RICE UNIVERSITY

Accelerated PDE Constrained Optimization using Direct Solvers

by

Peter Geldermans

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Art

Approved, Thesis Committee:

Adrianna Gillman, Co-Chair
Assistant Professor of Computational and Applied Mathematics

Matthias Heinkenschloss, Co-Chair
Noah G. Harding Chair and Professor of Computational and Applied Mathematics

Jesse Chan
Assistant Professor of Computational and Applied Mathematics

Houston, Texas

November, 2017

Abstract

Accelerated PDE Constrained Optimization Using Direct Solvers

by

Peter J. Geldermans

In this thesis, I propose a method to reduce the cost of computing solutions to optimization problems governed by partial differential equations (PDEs). Standard second order methods such as Newton apply an iterative method to solve the Newton system. Iteratively solving the Newton system requires the solution of two PDEs per iteration, which can be prohibitively expensive when applying iterative solvers to the PDEs. In contrast, this work takes advantage a recently developed high order discretization method that comes with an efficient direct solver. The new technique precomputes a solution operator that can be reused for any body load, which is applied whenever a PDE solve is required. Thus the precomputation cost is amortized over many PDE solves. This approach will make second order optimization algorithms computationally affordable for practical applications such as photoacoustic tomography and optimal design problems.

Acknowledgements

This work is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1450681.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The numerical solution of optimization problems governed by partial differential equations (PDEs) is important for optimal design, modeling of physical systems, and inverse problems in many fields of science and engineering. Computing solutions to practical PDE constrained optimization problems requires the solution of the governing PDE many times. Repeatedly solving the PDE at each optimization iteration can easily cause the cost of solving these problems to become very expensive. Additionally, in many real applications, it is desirable to solve the optimization problem over a range (or sampling) of some parameter rather than solving a single deterministic problem. If solving the PDE constrained optimization problem for a single parameter value is expensive, then solving the problem over a range of parameter values may be computationally intractible. Reducing the cost of computing solutions to PDE constrained optimization problems will extend the range of computationally affordable problems.

Solving the governing PDE for many different right hand sides dominates the cost of computing solutions to PDE constrained optimization problems. The Hierarchical Poincaré-Steklov (HPS) method, is a recently developed high order accurate discretization technique that comes with an efficient direct solver. The HPS method uses the pseudospectral (or spectral collocation) method on a collection of disjoint

leaf boxes whose union is the domain.

The boxes are then hierarchically merged to construct local solution operators at each step by computing discrete Poincaré-Steklov operators on the union of two boxes at a time. Merging each of the local solution operators results in an efficient direct solver such that solution to the PDE may be evaluated very rapidly for new boundary conditions and body loads. This thesis examines the HPS method in the setting of PDE constrained optimization. The objective of this work is to exploit the efficiency of the direct solver that comes with the HPS discretization to obtain second order optimization algorithms at low cost relative to first order methods.

1.1 Model Problem

To examine the HPS discretization technique in the optimization setting, consider the following linear quadratic optimal control problem.

$$\underset{u \in \mathcal{U}}{\text{Minimize}} \ \frac{1}{2}\int_{\Omega} \left(y(x;u) - z(x)\right)^2 dx + \frac{\alpha}{2}\int_{\Omega} u^2(x) dx \tag{1.1.1a}$$

where $y(x;u) \in \mathcal{Y}$ satisfies the differential equation

$$\begin{cases} -\Delta y(x) = u(x) + f(x), & x \in \Omega = (0,1)^2, \\ \quad y(x) = g(x), & x \in \partial\Omega, \end{cases} \tag{1.1.1b}$$

for a given function $u \in \mathcal{U}$.

This simple model problem will provide insight for how to extend the HPS method to the optimization setting. It should be noted that the extension is not limited to problems of the form (1.1.1). The optimization methods developed can also be applied to problems with boundary control or inverse problems in which the PDE coefficients are recovered, but for simplicity of presentation I focus on problems of the form (1.1.1).

Throughout the remainder of the thesis I refer to y as the *state* $\mathcal{Y}$ as the state space, u as the *control*, $\mathcal{U}$ as the control space, and (1.1.1b) as the *state equation*.

1.2 Organization

The organization of the thesis is as follows. Chapter 2 reviews relevant literature from PDE constrained optimization as well as for the HPS method to provide context for this work. Chapter 3 reviews basic material that will be used throughout the thesis including polynomials, quadrature, and differentiation matrices. Chapter 4 presents the weak formulation and strong formulation for discretizing a PDE by the HPS method. Then Chapter 5 presents the infinite dimensional optimal control problem and examines its discretization in detail by considering each formulation presented in Chapter 4. The direct solver is presented in Chapter 6 along with how to exploit its efficiency in the optimization setting. Additionally, a numerical experiment is provided to illustrate the benefit of using the direct solver in the context of optimization. Finally, Chapter 7 summarizes the contributions and identifies several areas for future work.

Chapter 2

Literature Review

Taking advantage of the performance of the HPS method in PDE constrained optimization intersects the optimization community with active research in discretization techniques for PDEs. It is important to understand the context of this work from the perspective of each community. This chapter first reviews literature regarding PDE constrained optimization. Then it provides context for the HPS method in the PDE discretization community. Finally a recent publication is highlighted that illustrates the potential of using the HPS discretization method for PDE constrained optimization problems. This provides the starting point for developing a more general framework for exploiting the performance of the HPS discretization method in PDE constrained optimization.

2.1 PDE Constrained Optimization

For optimization problems governed by PDEs there are two fundamentally different approaches to computing numerical solutions: optimize-then-discretize, and discretize-then-optimize. In the optimize-then-discretize approach, optimality conditions are derived in the appropriate function space (as in Lions [12, Ch. 2], or Tröltzsch [24, Ch. 2]) and then the infinite dimensional optimality conditions are discretized to ob-

tain the solution. On the other hand, in the discretize-then-optimize approach, the objective function and constraint equation are discretized directly to obtain a finite dimensional problem which may then be solved using standard quadratic programming techniques (as in Heinkenschloss [9]). In general, the two approaches to solving PDE constrained optimization problems do not lead to the same algorithm. Furthermore, in certain applications one approach may be preferable to the other (see discussion in Quarteroni [18, Ch. 16]). Therefore in developing a framework it is important that either approach may be used to efficiently and accurately compute solutions. Under either approach, discretization results in a finite dimensional problem that can be solved by standard techniques. However, numerically solving these problems is computationally expensive or may even be intractable due to the large number of PDE solves required in a typical optimization problem.

Optimization algorithms for solving PDE constrained optimization problems are typically based on either first or second order derivative information. First order (gradient-based) methods require the solution of two PDEs, the state and adjoint equations, at each optimization iteration. Second order methods require the gradient as well as the solution of the Newton system (or the Newton-like system as in the Gauss-Newton method) at each optimization iteration. Since the Hessian is both very large and may be dense, often it is unsuitable to store or compute with it explicitly. Instead, when solving the Newton system, well known iterative methods such as the conjugate gradient (CG) method or the generalized minimal residual (GMRES) method are employed, which do not require the Hessian itself, but rather the application of the Hessian to a vector. The Hessian-times-vector computation can be performed by solving two PDEs similar to the state and adjoint equations (see for example Heinkenschloss [9]). Thus second order optimization methods require the solution of $2k + 2$ PDEs per optimization iteration, where k is the number of inner iterations (or Hessian-times-vector computations) required to solve the Newton system. Typically an iterative solver is used to compute solutions to each PDE

when solving PDE constrained optimization problems. However, for problems where the Newton system is large, the number of PDE solves required by second order optimization algorithms can grow large enough that the cost of solving the PDEs by an iterative method becomes impractical.

In contrast to solving each PDE by an iterative method, the method I propose takes advantage of the efficient direct solver that comes with the HPS discretization. By choosing to work with a direct solver for the PDEs, the solution operator is precomputed once, and then it is applied to efficient evaluate each PDE solution required by the optimization algorithm. Applying the solution operator is more efficient than calling an iterative solver. However, making use of a direct solver also incurs the cost of constructing the solution operator. While constructing the solution operator and then applying it is relatively expensive for a single PDE compared to calling an iterative solver, the solution operator may be reused for many PDE solves. That is, the cost of the precomputation is amortized over many PDE solves. Thus efficient direct solvers for PDEs are ideally suited for applying iterative methods to optimality systems in PDE constrained optimization.

An alternative approach that is used to reduce the computational burden of solving PDE constrained optimization problems is to employ reduced order models (also known as surrogates). Reduced order modeling is an active research area in the context of PDE constrained optimization. Typically, a reduced order model or surrogate for the governing PDE is used in place of the PDE constraint (see Sachs [22], Benner [2]). Computing solutions to the surrogate is significantly less expensive than computing solutions to the full order model. The surrogate models can be obtained by mathematical techniques such as the proper orthogonal decomposition, or they can be given as an engineering or physics model. Additionally, the solution to the optimization problem governed by the surrogate model can be used as an initial guess for the problem governed by the full order model. This ensures accuracy of the solution to the original problem, while minimizing the number of full order model evaluations

required. This idea can be generalized by including a hierarchy of surrogate models and selecting which to use at each optimization step (see the survey provided by Peherstorfer et al. [16]). The reduction in cost achieved by employing reduced order models can lead to efficient methods for computing solutions to some PDE constrained optimization problems. The approach of reduced order modeling is not considered in this work.

2.2 The Hierarchical Poincaré-Steklov Method

The HPS method is a high order accurate discretization technique for PDEs that was first proposed by Martinsson [14] in 2013. After discretization, the direct solver consists of a build stage which precomputes the solution operator in a factored form, and a solve stage which applies the solution operator given a body load and boundary conditions consisting of a collection of small matrix vector multiplies.

To discretize, the domain is broken into a collection of rectangular patches. These patches are organized via a binary space partitioning tree with the whole domain at the root, and the collection of patches as the leaves, which are called leaf boxes. Figure 2.1 illustrates the collection of patches and Figure 2.2 illustrates the binary space partitioning tree. On each leaf box, the local boundary value problem is discretized by spectral collocation on a tensor product grid of either Chebyshev of Legendre nodes, and requires that the solution and its derivative are continuous across leaf edges.

For the build stage, a solution operator and a Dirichlet-to-Neumann (DtN) operator for the local boundary value problem is constructed directly via the spectral collocation method. Then the local DtN operators are used to hierarchically merge the operators on the leaf boxes, sweeping up the binary tree, to obtain a solution operator and a DtN operator for the whole domain. The build stage scales as $\mathcal{O}(N^{3/2})$ where N is the number of unknowns.

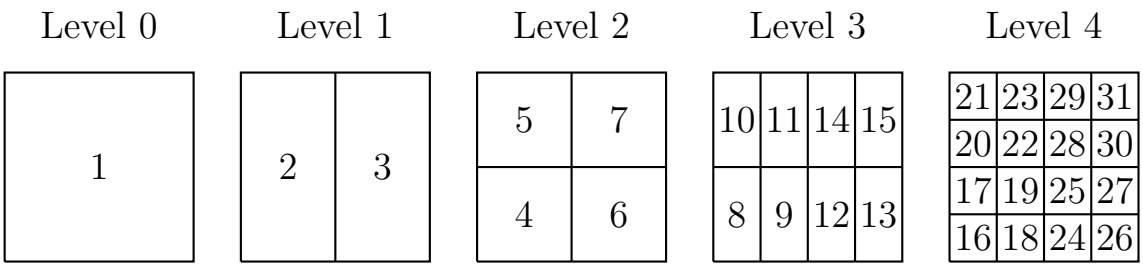

Figure 2.1: Illustration of domain partitioning for a four level binary space partitioning tree. The whole domain is the box numbered 1 (far left) and the leaf boxes are numbered 16-31 (far right). Each subdomain corresponds to the node in the binary space partitioning tree (see Figure2.2) with the same number.

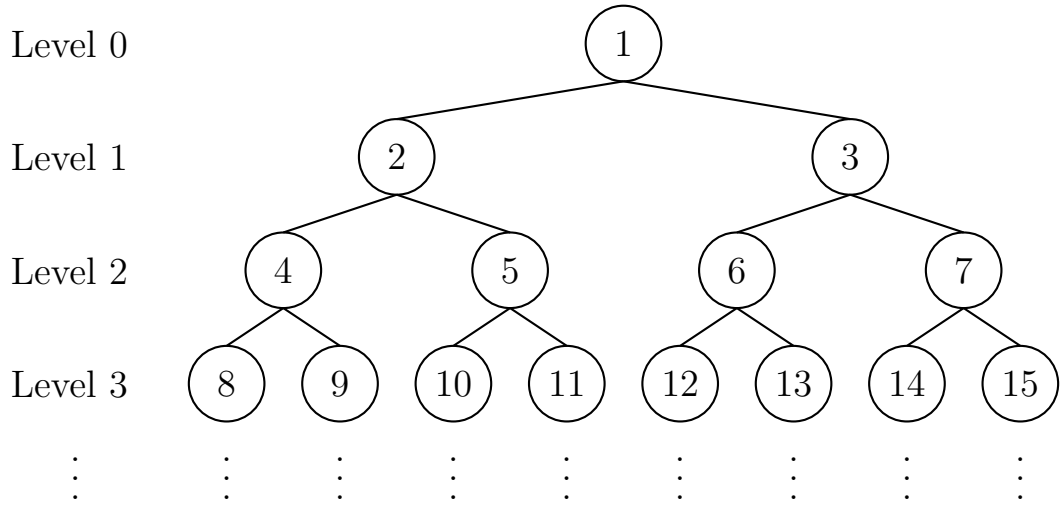

Figure 2.2: Illustration of the binary space partitioning tree. Node numbers correspond to the subdomains in Figure 2.1. The root is at the top of the tree and the leaf boxes are at the bottom.

During the solve stage the solution operator constructed during the build stage is applied to the given boundary condition for the PDE to obtain boundary data on each leaf box, sweeping down the binary tree from the root to the leaves. At the leaf level, the local solution operators are applied to obtain the solution on the interior of the leaf boxes. Since the boundary data maps and local solution operators were precomputed during the build stage, the solve is very efficient and scales as $\mathcal{O}(N \log N)$.

The HPS method is closely related to domain decomposition methods for applying spectral methods to solving PDEs. The discretization method employed in the HPS method is similar to the multidomain spectral discretization proposed by Orszag [15] in 1980 to apply spectral methods to problems with complex geometries. Both methods discretize local subdomains with spectral collocation (as in Boyd [5], Trefethen [23]) and require the solution and its derivative to match at the subdomain interfaces. The work by Orzsag was later generalized by Pfeiffer et al. [17] in 2003 to allow for a wider range of basis functions, to handle overlapping subdomains, and to provide the capability to match higher order derivatives across subdomain boundaries. A key distinction between the HPS method and earlier work on composite spectral collocation techniques is that the HPS method immediately eliminates interior unknowns. The earlier works produced large discretization matrices to be solved by an iterative solver such as GMRES or another Krylov subspace method. In contrast, the hierarchical elimination of interior unknowns and construction of local solution operators in the HPS method results in an efficient direct solver that is a series of small matrix vector multiplications the size of information on the boundary of the subdomains. In 2014, Gillman and Martinsson [8] demonstrated that for problems that are not highly oscillatory, Martinsson's original scheme can be further accelerated to achieve $\mathcal{O}(N)$ complexity for both the build and solve stages. They achieved the speedup by exploiting the special structure of hierarchical block separable (HBS) matrices to make use of fast linear algebra.

Then Babb et al. [1] in 2016 provided an efficient method for handling body loads in the build and solve stages to reuse the precomputation for multiple body loads. Unlike in the original method, the solution is represented as the superposition of a homogeneous solution and a particular solution. Then during the build stage, solution operators and DtN operators are constructed for both the homogeneous and particular solutions. The solve stage for the body load problem includes an additional sweep up the tree. For each new body load, the particular solution operator is applied to the body load on each leaf box. Then the contribution to the Neumann data from the particular solution is computed by applying the particular DtN operator. This information is swept up the tree to the root, where the boundary conditions are known. Then the solve stage sweeps down the tree mapping boundary data to the leaf boxes and applying the local solution operators as in the case without a body load. This algorithm will be reviewed in detail in Section 6.1.

The HPS method has been used in the optimization context by Borges et al. [4] for application to an inverse acoustic imaging problem from the medical imaging community. In the largest problem considered, a Matlab implementation of the HPS method was used to solve approximately one million PDEs, each with 19,600 unknowns, in approximately two days (on a multi-core workstation). The authors report that the reconstructions presented in their paper are among the largest ever computed in the medical imaging community.

This success demonstrates the potential of exploiting the HPS discretization and its direct solver in PDE constrained optimization. The goal of this work is to develop a general framework for taking advantage of the HPS method in the optimization setting. This work illustrates how the HPS method can be applied for both the optimize-then-discretize approach as well as the discretize-then-optimize approach. Additionally, optimization algorithms that take advantage of the efficiency of the direct solver to reduce the cost of solving PDE constrained optimization problems are provided.

Chapter 3

Background

This chapter reviews background material that is used throughout the thesis. First, Section 3.1 reviews a set of 1D polynomials, differentiation matrices, quadrature rules, and interpolation. Then Section 3.2 provides the extension of the 1D tools to 2D. Note that the extension can be performed for N dimensional polynomials in general, but the 2D case is emphasized as this work considers 2D problems. The results reviewed in this section can be found in many places, for example in the books by Canuto et al. [6, Ch. 2] or Quarteroni and Valli [20, Ch. 4] or in the paper by Bernardi and Maday [3, Thm. 13.4].

3.1 Polynomials, Differentiation Matrices, Quadrature Rules, and Interpolation

Consider the Legendre-Gauss-Lobatto (LGL) collocation points

$$-1 = \mathbf{x}_1 < \mathbf{x}_2 < \cdots < \mathbf{x}_{q-1} < \mathbf{x}_q = 1, \tag{3.1.1}$$

which are the roots of $(1-x^2)\frac{d}{dx}[\mathscr{L}_{q-1}(x)]$, where $\mathscr{L}_{q-1}$ is the Legendre polynomial of degree $q-1$.

The set of polynomials on $[a, b]$ of degree less than or equal to p is denoted by

$$\mathcal{P}_p[a, b].$$

Let the Lagrange interpolation polynomials be given by,

$$\psi_j(x) = \prod_{\substack{k=1 \\ k \neq j}}^{q} \frac{x - \mathbf{x}_k}{\mathbf{x}_j - \mathbf{x}_k} \in \mathcal{P}_{q-1}[-1, 1], \qquad j = 1, \ldots, q. \tag{3.1.2}$$

Furthermore, given the Legendre-Gauss-Lobatto points (3.1.1) and the corresponding Lagrange polynomials (3.1.2), let $I_{q-1} : C([-1, 1]) \to \mathcal{P}_{q-1}[-1, 1]$ be the interpolation operator

$$(I_{q-1}u)(x) = \sum_{j=1}^{q} u(\mathbf{x}_j)\psi_j(x). \tag{3.1.3}$$

The LGL quadrature rule is given by

$$\int_{-1}^{1} r(x)dx \approx \int_{-1}^{1} (I_{q-1}r)(x)dx = \sum_{j=1}^{q} w_j r(\mathbf{x}_j). \tag{3.1.4}$$

where the quadrature weights are

$$w_j = \int_{-1}^{1} \psi_j(x)dx = \frac{2}{q(q-1)} \frac{1}{[\mathscr{L}_{q-1}(\mathbf{x}_j)]^2}. \tag{3.1.5}$$

The exactness of the quadrature rule is given by Theorem 3.1.1 below. This is a standard result from Gauss-Lobatto quadrature. Details on the proof of the following theorem can be found in Chapter 10.4 of [19].

Theorem 3.1.1 (Legendre-Gauss-Lobatto Quadrature Exactness) *The Legendre-Gauss-Lobatto quadrature rule with q points given by (3.1.5) and (3.1.4) is exact for all all polynomials on $[-1, 1]$ of degree less than or equal to $2q - 3$, i.e.,*

$$\int_{-1}^{1} r(x)dx = \sum_{j=1}^{q} w_j r(\mathbf{x}_j) \qquad \forall r \in \mathcal{P}_{2q-3}[-1, 1]. \tag{3.1.6}$$

The following results on interpolation error can be found, e.g., Bernardi and Maday [3, Thm. 13.4] or Canuto et al. [6, Sec 5.4.3].

Theorem 3.1.2 (Legendre-Gauss-Lobatto Interpolation Error) *Let $k \in \{0, 1\}$ and $m > (1 + k)/2$. There exists a constant $C > 0$ such that for all functions $u \in H^m(-1, 1)$ the following interpolation error estimate holds*

$$\|u - I_{q-1}u\|_{H^k(-1,1)} \leq C(q-1)^{k-m}\|u\|_{H^m(-1,1)}. \tag{3.1.7}$$

The derivative matrices $\mathbf{D}, \mathbf{D}^{(2)} \in \mathbb{R}^{q\times q}$ have entries

$$\mathbf{D}_{k,j} = \frac{d}{dx}\psi_j(\mathbf{x}_k), \tag{3.1.8a}$$

$$\mathbf{D}^{(2)}_{k,j} = \frac{d^2}{dx^2}\psi_j(\mathbf{x}_k), \tag{3.1.8b}$$

so that if $\mathbf{r} = (r_q(\mathbf{x}_1), \ldots, r_q(\mathbf{x}_q))^T$, then the derivatives of the interpolating polynomial, r_q, at the collocation points are given by

$$\frac{d}{dx}r_q(\mathbf{x}_k) = \sum_{j=1}^{q} r(\mathbf{x}_j)\frac{d}{dx}\psi_j(\mathbf{x}_k) = \mathbf{e}_k^T\mathbf{D}\mathbf{r}, \tag{3.1.9a}$$

$$\frac{d^2}{dx^2}r_q(\mathbf{x}_k) = \sum_{j=1}^{q} r(\mathbf{x}_j)\frac{d^2}{dx^2}\psi_j(\mathbf{x}_k) = \mathbf{e}_k^T\mathbf{D}^{(2)}\mathbf{r}, \tag{3.1.9b}$$

where $\mathbf{e}_k \in \mathbb{R}^q$ is the k-th standard basis unit vector. Welfert [25, Thm. 6.1] shows that the second order derivative matrix is the square of the first order derivative matrix, i.e.

$$\mathbf{D}^{(2)} = \mathbf{D}^2. \tag{3.1.10}$$

Each of the results on the reference interval $[-1, 1]$ can be generalized to a target interval $[a, b]$ by mapping the collocation points from the reference interval to the target interval. Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_q)^T$ be the LGL collocation points on the reference interval. Then the LGL collocation points on the target interval are given by

$$\mathbf{x}^{(a,b)} = \frac{(b-a)}{2}\mathbf{x} + \frac{(b+a)}{2}. \tag{3.1.11}$$

Define the Lagrange polynomials on the target interval by

$$\psi_j^{(a,b)}(x) = \prod_{\substack{k=1\\k\neq j}}^{q} \frac{x - \mathbf{x}_k^{(a,b)}}{\mathbf{x}_j^{(a,b)} - \mathbf{x}_k^{(a,b)}}, \qquad j = 1, \ldots, q. \tag{3.1.12}$$

Let $I_{q-1} : C[a,b] \to \mathcal{P}_{q-1}[a,b]$ be the interpolation operator

$$(I_{q-1}u)(x) = \sum_{j=1}^{q} u(\mathbf{x}_j^{(a,b)})\psi_j^{(a,b)}(x). \tag{3.1.13}$$

Let the $\{w_j\}_{j=1}^q$ be the quadrature weights associated with the LGL collocation points on the reference interval. Then the quadrature weights for the generic interval $[a,b]$ are given by

$$w_j^{(a,b)} = \frac{(b-a)}{2} w_j, \text{ for } j = 1, \ldots, q, \tag{3.1.14}$$

so that the quadrature formula over the generic interval given by

$$\int_a^b r(x)dx \approx \int_a^b (I_{q-1}r)(x)dx = \sum_{j=1}^{q} w_j^{(a,b)} r(\mathbf{x}_j^{(a,b)}), \tag{3.1.15}$$

is exact for all $r \in \mathcal{P}_{2q-3}(a,b)$ similar to the result in Theorem 3.1.1. The interpolation error (4.3.9) generalizes to the following estimate, see Canuto et al. [6, Eqn. (5.4.42)].

$$\|u - I_{q-1}u\|_{H^k(a,b)} \leq C\,(b-a)^{k-\min\{m,q-1\}}\,(q-1)^{k-m}\,\|u\|_{H^{m,q-1}(-1,1)}. \tag{3.1.16}$$

The differentiation matrix on the generic interval is obtained by scaling the differentiation matrix from the reference interval

$$\mathbf{D}^{(a,b)} = \frac{2}{(b-a)}\mathbf{D}. \tag{3.1.17}$$

Now that the 1D polynomial approximation, the quadrature rule, and the differentiation matrices are known, it is necessary to extend these tools to higher dimensions to use them in solving partial differential equations via spectral collocation.

3.2 Extension to Higher Dimensions

To represent polynomials in multiple dimensions, a tensor product grid of LGL points is used. As noted before, the extension may be carried out to represent polynomials

in N dimensions, but as the work in the thesis focuses on 2D problems, I will discuss extending the results from Section 3.1 to two dimensions.

Consider the domain $\Omega = (a, b) \times (c, d) \subset \mathbb{R}^2$, and define the 1D LGL collocation points $\mathbf{x}^{(a,b)}$ on the interval (a, b) and $\mathbf{x}^{(c,d)}$ on (c, d) as in (3.1.11). Then the 2D LGL collocation points on $\overline{\Omega}$ are given by the tensor product grid

$$(\mathbf{x}_i^{(a,b)}, \mathbf{x}_j^{(c,d)}), \quad \forall\, i, j \in \{1, \ldots, q\}. \tag{3.2.1}$$

Figure 3.1 provides an illustration of the 2D LGL collocation points.

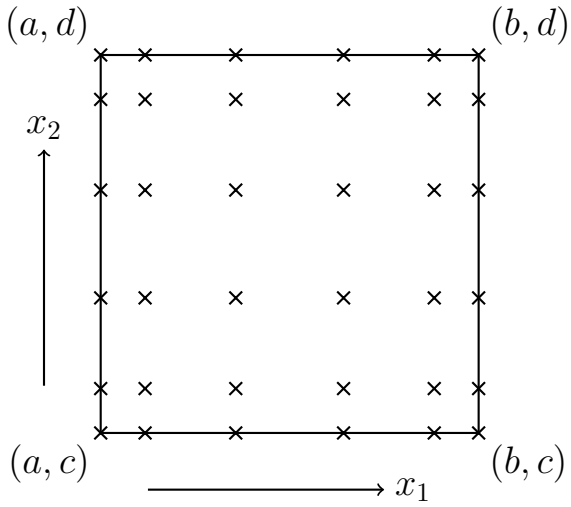

Figure 3.1: Illustration of the tensor product grid of LGL quadrature nodes on $\overline{\Omega}$ for $q = 6$. Observe that this is given by the Cartesian product of the set of 1D LGL quadrature nodes in each direction.

Order the collection of q^2 collocation points by the single index $\sigma(i, j, q)$ where

$$\sigma(i, j, q) = (i - 1)q + j, \tag{3.2.2}$$

and define

$$\mathbf{x}_{\sigma(i,j,q)} = (\mathbf{x}_i^{(a,b)}, \mathbf{x}_j^{(c,d)}). \tag{3.2.3}$$

Let $x = (x_1, x_2) \in \overline{\Omega}$. Then the 2D interpolation basis functions are given by

$$\psi_{\sigma(i,j,q)}(x) = \psi_i^{(a,b)}(x_1)\psi_j^{(c,d)}(x_2). \tag{3.2.4}$$

Note that upon inspection it is immediate that

$$\psi_{\sigma(i,j,q)}(\mathbf{x}_k^{(a,b)}, \mathbf{x}_\ell^{(c,d)}) = \begin{cases} 1, & (k,\ell) = (i,j), \\ 0, & (k,\ell) \neq (i,j), \end{cases} \tag{3.2.5}$$

since

$$\psi_i^{(a,b)}(\mathbf{x}_k^{(a,b)}) = \begin{cases} 1, & k = i, \\ 0, & k \neq i, \end{cases}$$

and

$$\psi_j^{(c,d)}(\mathbf{x}_\ell^{(c,d)}) = \begin{cases} 1, & \ell = j, \\ 0, & \ell \neq j. \end{cases}$$

The set of polynomials on $\bar{\Omega}$ of degree less than or equal to p in each variable is denoted by

$$\mathcal{P}_p(\Omega).$$

Furthermore, given the tensor product grid Legendre-Gauss-Lobatto points (3.2.1) and the corresponding Lagrange polynomials (3.2.4), let $I_{q-1} : C(\bar{\Omega}) \to \mathcal{P}_{q-1}(\Omega)$ be the interpolation operator

$$(I_{q-1}u)(x) = \sum_{\sigma=1}^{q^2} u(\mathbf{x}_\sigma)\psi_\sigma(x). \tag{3.2.6}$$

The polynomial interpolation error is bounded according to the following theorem (for details see Bernardi and Maday [3, Thm. 14.2] and Canuto et al. [6, Sec. 5.8.2]).

Theorem 3.2.1 *Let $k \in \{0, 1\}$ and $m > (2 + k)/2$. There exists a constant $C > 0$ such that for all functions $u \in H^m(\Omega)$ the following interpolation error estimate holds*

$$\|u - I_{q-1}u\|_{H^k(\Omega)} \leq C(q-1)^{k-m}\|u\|_{H^m(\Omega)}. \tag{3.2.7}$$

The 2D LGL quadrature formula is given by

$$\int_\Omega r(x)dx \approx \int_\Omega (I_{q-1}r)(x)dx = \sum_{\sigma=1}^{q^2} \mathbf{w}_\sigma r(\mathbf{x}_\sigma), \tag{3.2.8}$$

where the quadrature weights are

$$\begin{aligned} \mathbf{w}_{\sigma(i,j,q)} &= \int_\Omega \psi_\sigma(i,j,q)(x)dx \\ &= \int_c^d \int_a^b \psi_i^{(a,b)}(x_1)\psi_j^{(c,d)}(x_2)dx_1dx_2 \\ &= \int_c^d \psi_j^{(c,d)}(x_2)dx_2 \int_a^b \psi_i^{(a,b)}(x_1)dx_1 \\ &= w_j^{(c,d)} w_i^{(a,b)}. \end{aligned} \tag{3.2.9}$$

Thus the 2D quadrature weight $\mathbf{w}_{\sigma(i,j,q)}$ is simply the product of the 1D quadrature weights associated with the collocation point coordinate values in each direction. Theorem 3.2.2 defines the exactness of the Legendre-Gauss-Lobatto quadrature rule on Ω. For details, again refer to [19] and the references therein.

Theorem 3.2.2 (2D Legendre-Gauss-Lobatto Quadrature Exactness) *The Legendre-Gauss-Lobatto quadrature rule given by (3.2.9) and (3.2.8) is exact for all polynomials on Ω of degree less than or equal to $2q-3$ in each variable,*

$$\int_\Omega r(x)dx = \sum_{\sigma=1}^{q^2} \mathbf{w}_\sigma r(\mathbf{x}_\sigma) \qquad \forall r \in \mathcal{P}_{2q-3}(\Omega).$$

The structure of the tensor product grid allows the partial differentiation matrices to be defined in terms of a Kronecker product of the 1D differentiation matrices and the identity matrix. Let $\mathbf{I} \in \mathbb{R}^{q\times q}$, then the first partial derivative matrices are given by

$$\mathbf{D_1} = \mathbf{D}^{(a,b)} \otimes \mathbf{I}, \tag{3.2.10}$$

$$\mathbf{D_2} = \mathbf{I} \otimes \mathbf{D}^{(c,d)}. \tag{3.2.11}$$

Let $\mathbf{r} = (r_q(\mathbf{x}_1), \ldots, r_q(\mathbf{x}_{q^2}))^T$, and let $\mathbf{e}_\sigma$ the σ-th standard basis vector in $\mathbb{R}^{q^2}$. Then the partial derivatives of the interpolation polynomial r_q evaluated at the collocation

points are given by

$$\frac{d}{dx_1} r_q(\mathbf{x}_\sigma) = \mathbf{e}_\sigma^T \mathbf{D_1} \mathbf{r}, \tag{3.2.12}$$

$$\frac{d}{dx_2} r_q(\mathbf{x}_\sigma) = \mathbf{e}_\sigma^T \mathbf{D_2} \mathbf{r}. \tag{3.2.13}$$

More generally, define the partial differentiation matrix

$$\mathbf{D_1^k D_2^\ell} = (\mathbf{D}^{(a,b)})^k \otimes (\mathbf{D}^{(c,d)})^\ell, \tag{3.2.14}$$

so that

$$\frac{d^{k+\ell}}{dx_1^k dx_2^\ell} r_q(\mathbf{x}_\sigma) = \mathbf{e}_\sigma^T \mathbf{D_1^k D_2^\ell} \mathbf{r}. \tag{3.2.15}$$

Now that the 2D polynomial approximation, the quadrature rule, and the partial differentiation matrices are known, these tools can be used to approximate solutions to partial differential equations via spectral collocation.

Chapter 4

Discretization of the State Equation

This chapter describes two discretizations for the state equation (the differential equation that relates the control to the state in the model optimization problem). The first discretization approach is based on the Galerkin discretization of the weak form of the state equation as in [20]. The second approach is based on discretizing the strong form of the state equation via a composite spectral collocation scheme. In this approach, subdomains are discretized by spectral collocation as in [23], [5] and the solutions on subdomains are related by requiring that the solution and normal derivative match at the interface between subdomains. This is the discretization that the Hierarchical Poincaré-Steklov method, for details see [14]. The major difference between the two approaches is in the implementation of the Neumann condition. The Galerkin approach uses the weak form of the Neumann condition whereas the composite spectral collocation scheme uses the strong form of the Neumann condition.

Section 4.1 reviews the weak formulation and its Galerkin discretization Then Sections 4.2 and 4.3 describe the discretizations for a single domain and then for multiple subdomains respectively. Finally, the performance of the weak form and strong form discretizations are illustrated in Section 4.4 for a test problem.

4.1 Weak Formulation

Given a real Hilbert space $\mathcal{V}$ with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \| = \langle \cdot, \cdot \rangle^{1/2}$, a bilinear operator

$$a : \mathcal{V} \times \mathcal{V} \to \mathbb{R},$$

and a continuous linear functional

$$b : \mathcal{V} \to \mathbb{R},$$

consider the following problem. Find $y \in \mathcal{V}$ such that

$$a(y, \phi) = b(\phi), \quad \forall\, \phi \in \mathcal{V}. \tag{4.1.1}$$

Theorem 4.1.1 (stated without proof) is a standard result from functional analysis that provides conditions for the existence of a unique solution to (4.1.1).

Theorem 4.1.1 (Lax-Milgram Theorem) *Let $\mathcal{V}$ be a (real) Hilbert space, endowed with the norm $\| \cdot \|$, $a(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ a bilinear form and $b : \mathcal{V} \to \mathbb{R}$ a continuous linear functional, i.e. $b \in \mathcal{V}'$ where $\mathcal{V}'$ denotes the dual space of $\mathcal{V}$. If there exist constants $\beta_1, \beta_2 > 0$ such that*

$$|a(\psi, \phi)| \leq \beta_1 \|\psi\|_{\mathcal{V}} \, \|\phi\|_{\mathcal{V}}, \qquad \forall\, \psi, \phi \in \mathcal{V}, \tag{4.1.2}$$

$$\beta_2 \|\phi\|_{\mathcal{V}}^2 \leq a(\phi, \phi), \qquad \forall\, \phi \in \mathcal{V}, \tag{4.1.3}$$

i.e. $a(\cdot, \cdot)$ is continuous and coercive, then there exists a unique solution $y \in \mathcal{V}$ to (4.1.1) and

$$\|y\|_{\mathcal{V}} \leq \frac{1}{\beta_2} \|b\|_{\mathcal{V}'}. \tag{4.1.4}$$

The proof of Theorem 4.1.1 can be found in almost any text on the treatment of partial differential equations. In particular see [20] Thm 5.1.1.

To discretize the weak formulation, consider a finite dimensional subspace $\mathcal{V}_q \subset \mathcal{V}$. The discretized weak form is then given by, find $y_q \in \mathcal{V}_q$ such that

$$a(y_q, \phi_q) = b(\phi_q), \quad \forall\, \phi_q \in \mathcal{V}_q. \tag{4.1.5}$$

Lemma 4.1.2 establishes a general error bound for the discretization of the weak formulation.

Lemma 4.1.2 (Céa's Lemma) *Under the assumptions of Theorem 4.1.1 there exists a unique solution $y_q \in \mathcal{V}_q$ to (4.1.5). Moreover, if y is the solution to (4.1.1), then*

$$\|y - y_q\|_{\mathcal{V}} \leq \frac{\beta_1}{\beta_2} \inf_{\phi_q \in \mathcal{V}_q} \|y - \phi_q\|_{\mathcal{V}}. \tag{4.1.6}$$

In Section 4.2 and Section 4.3, an appropriate finite dimensional subspace $\mathcal{V}_q$ is identified and the general error estimate via Céa's Lemma is made specific by approximation results for the chosen subspace.

4.2 Single Domain Discretization

The state equation considered in this section is a slight variation of (1.1.1b). This variation will be useful when considering multi-domain discretizations. Let $\Omega = (0,1)^2$, $\Gamma_D = \{x \in \partial\Omega \mid x_1 = 0\} \cup \{x \in \partial\Omega \mid x_2 = 0\}$ and $\Gamma_N = \partial\Omega \setminus \Gamma_D$, and consider the boundary value problem

$$\begin{cases} -\Delta y(x) = u(x) + f(x), & x \in \Omega \\ y(x) = 0, & x \in \Gamma_D \\ \dfrac{\partial}{\partial n} y(x) = v(x), & x \in \Gamma_N. \end{cases} \tag{4.2.1}$$

The geometry for the boundary value problem is provided in Figure 4.1.

This section presents two spectral collocation approaches to discretize the state equation (4.2.1). First, I present discretization of the weak formulation in Section 4.2.1. Then I provide the discretization of the strong formulation in Section 4.2.3.

4.2.1 Discretization of the Weak Formulation

Let

$$\mathcal{V} = \{y \in H^1(\Omega) \mid y = 0 \text{ on } \Gamma_D\}$$

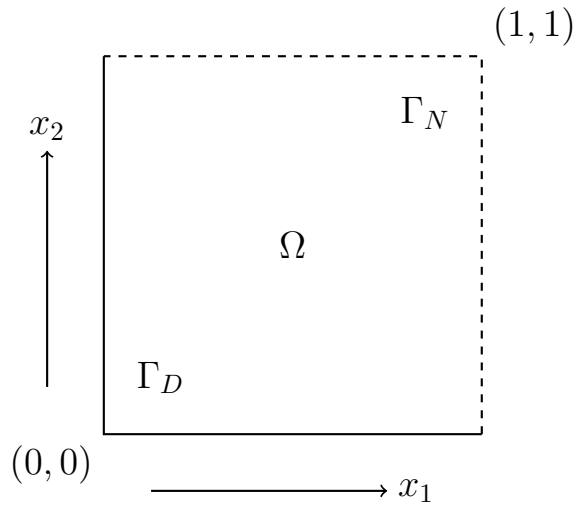

Figure 4.1: Illustration of geometry for the single domain problem. Γ_D is the solid line (bottom and left faces) and Γ_N is the dashed line (top and right faces).

be endowed with the $H^1(\Omega)$ norm. The weak formulation of (4.2.1) is given by

$$\begin{aligned}&\int_\Omega \nabla y(x)\cdot\nabla\phi(x)dx\\&=\int_\Omega \left(u(x)+f(x)\right)\phi(x)dx+\int_{\Gamma_N} v(x)\phi(x)dx,\quad \forall\,\phi\in\mathcal{V}.\end{aligned}\tag{4.2.2}$$

This is the identity (4.1.1) if the bilinear operator is defined as

$$a(y,\phi)=\int_\Omega \nabla y(x)\cdot\nabla\phi(x)dx,\tag{4.2.3}$$

and the continuous linear functional is defined as

$$b(\phi)=\int_\Omega (u(x)+f(x))\phi(x)dx+\int_{\Gamma_N} v(x)\phi(x)dx\tag{4.2.4}$$

Corollary 4.2.1 is then immediate by applying Theorem 4.1.1 to the weak formulation (4.1.1) with $\mathcal{V}=H^1(\Omega)$.

Corollary 4.2.1 *For any $f,u\in L^2(\Omega)$ and $v\in L^2(\Gamma_N)$ the state equation (4.2.1) has a unique weak solution $y\in\mathcal{V}$. Moreover, there exists a constant $C>0$ (independent of f and v) such that*

$$\|y\|_{H^1(\Omega)}\le C\left(\|f\|_{L^2(\Omega)}+\|u\|_{L^2(\Omega)}+\|v\|_{L^2(\Gamma_N)}\right).$$

Let $\mathcal{P}_{q-1}(\Omega)$ be the set of polynomials of degree less than or equal to $(q-1)$. The finite dimensional subspace is given by

$$\mathcal{V}_q = \mathcal{P}_{q-1}(\Omega) \cap \mathcal{V}, \tag{4.2.5}$$

and the Galerkin discretization of the (4.2.2) is

$$\begin{aligned} &\int_\Omega \nabla y_q(x) \cdot \nabla \phi(x) dx \\ &= \int_\Omega \left(u(x) + f(x)\right) \phi(x) dx + \int_{\Gamma_N} v(x)\phi(x) dx, \quad \forall\, \phi \in \mathcal{V}_q. \end{aligned} \tag{4.2.6}$$

A bound for the error between the exact solution and the solution to the finite dimensional subspace approximation is given by the following lemma.

Lemma 4.2.2 *Let the weak solution y to (4.2.1) satisfy $y \in H^m(\Omega)$ with $m > 3/2$. There exists a constant $C > 0$ such that the error between y and the solution y_q of (4.2.6) satisfies*

$$\|y - y_q\|_{H^1(\Omega)} \leq C(q-1)^{1-m} \|y\|_{H^m(\Omega)}. \tag{4.2.7}$$

Proof: Lemma 4.1.2 gives

$$\|y - y_q\|_{H^1(\Omega)} \leq C_1 \inf_{v_q \in \mathcal{V}_q} \|y - v_q\|_{H^1(\Omega)}.$$

Applying Theorem 3.2.1 with $k = 1$ to bound the right hand side

$$\inf_{v_q \in \mathcal{V}_q} \|y - v_q\|_{H^1(\Omega)} \leq \|y - I_{q-1} y\|_{H^1(\Omega)}$$

gives the desired result. □

Remark 4.2.3 *The error bound in Lemma 4.2.2 requires that integrals such as $\int_\Omega \left(u(x) + f(x)\right) \phi(x) dx$ for $\phi \in \mathcal{V}_q$ are evaluated exactly. If instead they are approximated by quadrature an additional term on the right hand side arises, which is proportional to the quadrature error.*

4.2.2 Linear System

Now I set up and solve a linear system corresponding to the discretization of the weak formulation (4.2.6). Let $\mathbf{x}$ be a tensor product grid of $q \times q$ Legendre-Gauss-Lobatto (LGL) quadrature points on $\overline{\Omega}$, where $\mathbf{x}_j$ denote the j-th point of the tensor product grid. Let $\mathbf{w}_j$ be the 2D quadrature weight associated with $\mathbf{x}_j$, and let $\psi_j(x)$ be the 2D Lagrange interpolation basis function associated with $\mathbf{x}_j$.

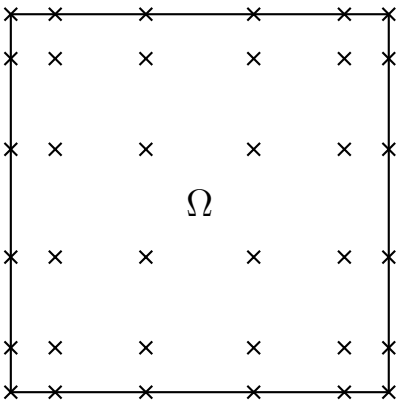

Figure 4.2: Illustration of the tensor product grid of LGL quadrature points $\{\mathbf{x}_j\}_{j=1}^{q^2}$ on $\overline{\Omega}$ for $q = 6$.

First write

$$\Gamma_N = \Gamma_{N(E)} \cup \Gamma_{N(N)},$$

where

$$\Gamma_{N(E)} = \{x \in \partial\Omega \mid x_1 = 1\} \quad \text{and} \quad \Gamma_{N(N)} = \{x \in \partial\Omega \mid x_2 = 1\}.$$

Because y_q is a polynomial and, hence, smooth, the left hand side in the discretized weak form (4.2.6) may be rewritten using the divergence theorem to obtain

$$\begin{aligned}
&\int_\Omega \nabla y_q(x) \cdot \nabla \phi(x) dx \\
&= \int_\Omega -\Delta y_q(x)\phi(x)dx + \int_{\Gamma_N} (\nabla y_q(x) \cdot \overline{n})\phi(x)dx \\
&= \int_\Omega -\Delta y_q(x)\phi(x)dx + \int_{\Gamma_{N(E)}} \frac{\partial}{\partial x_1} y_q(x)\phi(x)dx + \int_{\Gamma_{N(N)}} \frac{\partial}{\partial x_2} y_q(x)\phi(x)dx. \quad (4.2.8)
\end{aligned}$$

Substituting this expression for the left hand side in (4.2.6) yields the following equivalent expression for the weak form (4.2.6)

$$\int_\Omega -\Delta y_q(x)\phi(x)dx + \int_{\Gamma_{N(E)}} \frac{\partial}{\partial x_1} y_q(x)\phi(x)dx + \int_{\Gamma_{N(N)}} \frac{\partial}{\partial x_2} y_q(x)\phi(x)dx \tag{4.2.9}$$
$$= \int_\Omega (u(x) + f(x))\,\phi(x)dx + \int_{\Gamma_{N(E)}} v(x)\phi(x)dx + \int_{\Gamma_{N(N)}} v(x)\phi(x)dx \quad \forall\, \phi \in \mathcal{V}_q.$$

To obtain the linear system corresponding to (4.2.9), replace the integrals by quadrature and require the resulting equation to hold for all $\phi \in \text{span}\{\psi_1, \ldots, \psi_{q^2}\}$ such that $\phi = 0$ on Γ_D. Note that the quadrature rule is exact for the integrals in (4.2.9) involving y_q due to Theorem 3.2.2.

Let $\mathbf{y}$ be the vector given by

$$\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_{q^2})^T, \tag{4.2.10}$$

and let $\mathbf{y}_j$ denote the j-th element of $\mathbf{y}$ (i.e. $\mathbf{y}_j = y_q(\mathbf{x}_j)$). Furthermore, let

$$\mathbf{u} = (u(\mathbf{x}_1), \ldots, u(\mathbf{x}_{q^2}))^T, \tag{4.2.11}$$

$$\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{q^2}))^T, \tag{4.2.12}$$

$$\mathbf{v} = (v(\mathbf{x}_1), \ldots, v(\mathbf{x}_{q^2}))^T. \tag{4.2.13}$$

Given interpolation points $\mathbf{x}_j$ and the Lagrange basis functions $\psi_j(x)$, the solution of the discretized weak form (4.2.6) is

$$y_q(x) = \sum_{j=1}^{q^2} \mathbf{y}_j \psi_j(x). \tag{4.2.14}$$

Let $\mathbf{D_1^k}$ be the k-th order partial differentiation matrix in the x_1 direction, and let $\mathbf{D_2^k}$ be the k-th order partial differentiation matrix in the x_2 direction as defined in (3.2.14). Then the discretized differential operator $\mathbf{L} \in \mathbb{R}^{q^2 \times q^2}$ is given by

$$\mathbf{L} = -(\mathbf{D_1^2} + \mathbf{D_2^2}), \tag{4.2.15}$$

so that

$$-\Delta y_q(\mathbf{x}_j) = \mathbf{e}_j^T \mathbf{L} \mathbf{y}. \tag{4.2.16}$$

Next, partition

$$J := \{1, \ldots, q^2\} = J_I \cup J_D \cup J_{N(E)} \cup J_{N(N)} \cup J_C$$

where the index sets $J_I, J_D, J_{N(E)}, J_{N(N)}$ and J_C are defined next. See Figure 4.3 for an illustration of these index sets.

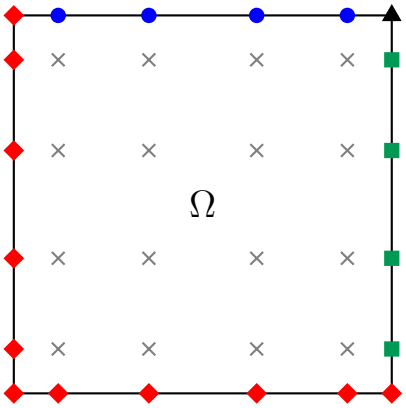

Figure 4.3: Illustration of the collocation points $\{\mathbf{x}_j\}_{j=1}^{q^2}$ by index sets. The gray crosses denote the interior points corresponding to J_I. The red diamonds denote the points corresponding to J_D, where the Dirichlet boundary condition is applied. The green squares, blue circles, and black triangle denote the points where the Neumann condition is applied, corresponding to $J_{N(E)}, J_{N(N)}$, and J_C respectively.

Let

$$J_I = \{j \mid \mathbf{x}_j \in \Omega\}$$

be the set of indices of interior collocation points and let

$$J_D = \{j \mid \mathbf{x}_j \in \Gamma_D\}$$

be the set of indices of collocation points where the Dirichlet boundary condition is enforced. To define the remaining index sets, first define sets to describe the northeast, southeast, and northwest corners of the domain

$$\mathcal{C}_{NE} = \{(1,1)\}, \quad \mathcal{C}_{SE} = \{(1,0)\}, \quad \mathcal{C}_{NW} = \{(0,1)\}.$$

Then let

$$J_{N(E)} = \{j \mid \mathbf{x}_j \in \Gamma_{N(E)} \setminus (\mathcal{C}_{NE} \cup \mathcal{C}_{SE})\},$$

be the set of indices of collocation points on the interior of the east edge of the domain and let

$$J_{N(N)} = \{j \mid \mathbf{x}_j \in \Gamma_{N(N)} \setminus (\mathcal{C}_{NE} \cup \mathcal{C}_{NW})\}.$$

be the set of indices of collocation points on the interior of the north edge of the domain. Finally, let

$$J_C = \{j \mid \mathbf{x}_j \in \mathcal{C}_{NE}\}$$

be the index of the collocation point at the northeast corner of the domain.

The Dirichlet boundary condition $y_q(x) = 0$ for all $x \in \Gamma_D$ implies that

$$\mathbf{e}_\mu^T \mathbf{y} = 0 \qquad \text{for } \mu \in J_D. \tag{4.2.17}$$

Since

$$\mathcal{V}_q = \mathcal{P}_{q-1}(\Omega) \cap \mathcal{V} = \{\psi_\mu \mid \mu \notin J_D\}$$

(4.2.9) is equivalent to

$$\begin{aligned}
&\int_\Omega -\Delta y_q(x)\psi_\mu(x)dx + \int_{\Gamma_{N(E)}} \frac{\partial}{\partial x_1} y_q(x)\psi_\mu(x)dx + \int_{\Gamma_{N(N)}} \frac{\partial}{\partial x_2} y_q(x)\psi_\mu(x)dx \\
&= \int_\Omega (u(x) + f(x))\,\psi_\mu(x)dx + \int_{\Gamma_{N(E)}} v(x)\psi_\mu(x)dx + \int_{\Gamma_{N(N)}} v(x)\psi_\mu(x)dx \quad \forall\, \mu \notin J_D.
\end{aligned} \tag{4.2.18}$$

Next, replace the integrals by quadrature. Note that the quadrature rule is exact for the integrals in (4.2.18) involving y_q due to Theorem 3.2.2.

$$\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell \Delta y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) + \sum_{\ell\in J_E} w_{\ell,2}\frac{\partial}{\partial x_1} y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) + \sum_{\ell\in J_N} w_{\ell,1}\frac{\partial}{\partial x_2} y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell)$$
$$= \sum_{\ell=1}^{q^2} \mathbf{w}_\ell \left(u(\mathbf{x}_\ell) + f(\mathbf{x}_\ell)\right)\psi_\mu(\mathbf{x}_\ell)$$
$$+ \sum_{\ell\in J_E} w_{\ell,2} v(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) + \sum_{\ell\in J_N} w_{\ell,1} v(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) \qquad \forall\, \mu \notin J_D. \tag{4.2.19}$$

For μ corresponding to the interior collocation points, $\mu \in J_I$, (4.2.19) simplifies as

$$\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell \Delta y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) = \sum_{\ell=1}^{q^2} \mathbf{w}_\ell \left(u(\mathbf{x}_\ell) + f(\mathbf{x}_\ell)\right)\psi_\mu(\mathbf{x}_\ell).$$

Since $\psi_\mu(\mathbf{x}_\ell) = 0$ for each $\ell \neq \mu$, the sums each reduce to the single term corresponding to the index μ

$$-\mathbf{w}_\mu \Delta y_q(\mathbf{x}_\mu) = \mathbf{w}_\mu \left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right).$$

Finally, dividing each side by the constant $\mathbf{w}_\mu$ yields

$$-\Delta y_q(\mathbf{x}_\mu) = u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu),$$

which can be interpreted as enforcing the strong form of the PDE at each interior collocation point. This can be written in terms of the discretized differential operators as

$$\mathbf{e}_\mu^T \mathbf{L}\mathbf{y} = \mathbf{e}_\mu^T \mathbf{u} + \mathbf{f}_\mu \qquad \text{for } \mu \in J_I. \tag{4.2.20a}$$

For μ corresponding to collocation points on the interior of the east edge of the domain, $\mu \in J_{N(E)}$, (4.2.19) simplifies as

$$\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell \Delta y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) + \sum_{\ell\in J_E} w_{\ell,2}\frac{\partial}{\partial x_1} y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell)$$
$$= \sum_{\ell=1}^{q^2} \mathbf{w}_\ell \left(u(\mathbf{x}_\ell) + f(\mathbf{x}_\ell)\right)\psi_\mu(\mathbf{x}_\ell) + \sum_{\ell\in J_E} w_{\ell,2} v(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell).$$

Again since $\psi_\mu(\mathbf{x}_\ell) = 0$ for each $\ell \neq \mu$, the sums each reduce to the single term corresponding to the index μ

$$\left(-\mathbf{w}_\mu \Delta + w_{\mu,2}\frac{\partial}{\partial x_1}\right) y_q(\mathbf{x}_\mu) = \mathbf{w}_\mu \left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right) + w_{\mu,2} v(\mathbf{x}_\mu).$$

Dividing both sides by the constant $w_{\mu,2}$ yields

$$\left(-w_{\mu,1}\Delta + \frac{\partial}{\partial x_1}\right) y_q(\mathbf{x}_\mu) = w_{\mu,1} \left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right) + v(\mathbf{x}_\mu).$$

This can be written in terms of the discretized differential operators as

$$\mathbf{e}_\mu^T \left(w_{\mu,1}\mathbf{L} + \mathbf{D_1}\right) \mathbf{y} = w_{\mu,1} \left(\mathbf{e}_\mu^T \mathbf{u} + \mathbf{f}_\mu\right) + \mathbf{v}_\mu \qquad \text{for } \mu \in J_{N(E)}. \tag{4.2.20b}$$

Similarly, for μ corresponding to collocation points on the interior of the north edge of the domain, $\mu \in J_{N(N)}$, (4.2.19) simplifies as

$$\begin{aligned}\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell \Delta y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) + \sum_{\ell \in J_N} w_{\ell,1}\frac{\partial}{\partial x_2} y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell) \\ = \sum_{\ell=1}^{q^2} \mathbf{w}_\ell \left(u(\mathbf{x}_\ell) + f(\mathbf{x}_\ell)\right) \psi_\mu(\mathbf{x}_\ell) + \sum_{\ell \in J_N} w_{\ell,1} v(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell).\end{aligned}$$

As $\psi_\mu(\mathbf{x}_\ell) = 0$ for each $\ell \neq \mu$, the sums each reduce to the single term corresponding to the index μ

$$\left(-\mathbf{w}_\mu \Delta + w_{\mu,1}\frac{\partial}{\partial x_2}\right) y_q(\mathbf{x}_\mu) = \mathbf{w}_\mu \left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right) + w_{\mu,1} v(\mathbf{x}_\mu).$$

Dividing by the constant $w_{\mu,1}$ yields

$$\left(-w_{\mu,2}\Delta + \frac{\partial}{\partial x_2}\right) y_q(\mathbf{x}_\mu) = w_{\mu,2} \left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right) + v(\mathbf{x}_\mu).$$

This can be written in terms of the discretized differential operators as

$$\mathbf{e}_\mu^T \left(w_{\mu,2}\mathbf{L} + \mathbf{D_2}\right) \mathbf{y} = w_{\mu,2} \left(\mathbf{e}_\mu^T \mathbf{u} + \mathbf{f}_\mu\right) + \mathbf{v}_\mu \qquad \text{for } \mu \in J_{N(N)}. \tag{4.2.20c}$$

Finally, for μ corresponding to the northeast corner of the domain, $\mu \in J_C$, (4.2.19) simplifies as

$$\begin{aligned}-\mathbf{w}_\mu \Delta y_q(\mathbf{x}_\mu) + w_{\mu,2}\frac{\partial}{\partial x_1} y_q(\mathbf{x}_\mu) + w_{\mu,1}\frac{\partial}{\partial x_2} y_q(\mathbf{x}_\mu) \\ = \mathbf{w}_\mu \left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right) + w_{\mu,2} v(\mathbf{x}_\mu) + w_{\mu,1} v(\mathbf{x}_\mu),\end{aligned}$$

which can be written in factored form as

$$\left(-\mathbf{w}_\mu \Delta + w_{\mu,2}\frac{\partial}{\partial x_1} + w_{\mu,1}\frac{\partial}{\partial x_2}\right) y_q(\mathbf{x}_\mu) = \mathbf{w}_\mu \left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right) + \left(w_{\mu,2} + w_{\mu,1}\right) v(\mathbf{x}_\mu).$$

This can be written in terms of the discretized differential operators as

$$\mathbf{e}_\mu^T \left(\mathbf{w}_\mu \mathbf{L} + w_{\mu,2}\mathbf{D_1} + w_{\mu,1}\mathbf{D_2}\right) \mathbf{y} = \mathbf{w}_\mu \left(\mathbf{e}_\mu^T \mathbf{u} + \mathbf{f}_\mu\right) + \left(w_{\mu,2} + w_{\mu,1}\right) \mathbf{v}_\mu \qquad \text{for } \mu \in J_C. \tag{4.2.20d}$$

In summary, after an approximation of the integrals by quadrature, the discretized weak form (4.2.9) leads to the linear system (4.2.17), (4.2.20) in $\mathbf{y}$ which is denoted by

$$\mathbf{A}\mathbf{y} = -\mathbf{B}\mathbf{u} + \mathbf{c}. \tag{4.2.21}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$, $\mathbf{y}, \mathbf{u}, \mathbf{c} \in \mathbb{R}^N$. Solving the linear system for $\mathbf{y}$ provides the coefficient values of the composite polynomial approximation of the weak solution y_q.

The matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ in (4.2.21) inherits important properties from the bilinear form (4.2.3).

Theorem 4.2.4 *The matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ in (4.2.21) is symmetric positive definite on $\{\mathbf{v} \mid \mathbf{v}_\mu = 0, \quad \mu \in J_D\}$.*

Proof: Let $\mathbf{y}, \mathbf{v}$ be vectors with $\mathbf{y}_\mu = \mathbf{v}_\mu = 0$, $\mu \in J_D$, and define

$$y_q(x) = \sum_{j=1}^{q^2} \mathbf{y}_j \psi_j(x), \qquad v_q(x) = \sum_{j=1}^{q^2} \mathbf{v}_j \psi_j(x).$$

Following the derivations (4.2.20), (4.2.19), and (4.2.18) yields

$$\begin{aligned}\mathbf{v}^T\mathbf{A}\mathbf{y} &= \sum_{\mu=1}^{q^2}\mathbf{v}_\mu\Big(\sum_{\ell=1}^{q^2}-\mathbf{w}_\ell\Delta y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell)+\sum_{\ell\in J_E}w_{\ell,2}\frac{\partial}{\partial x_1}y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell)\\ &\qquad+\sum_{\ell\in J_N}w_{\ell,1}\frac{\partial}{\partial x_2}y_q(\mathbf{x}_\ell)\psi_\mu(\mathbf{x}_\ell)\Big)\\ &=\sum_{\mu=1}^{q^2}\mathbf{v}_\mu\Big(\int_\Omega-\Delta y_q(x)\psi_\mu(x)dx+\int_{\Gamma_N}\frac{\partial}{\partial x_1}y_q(x)\psi_\mu(x)dx\Big)\\ &=\sum_{\mu=1}^{q^2}\mathbf{v}_\mu\Big(\int_\Omega\nabla y_q(x)^T\nabla\psi_\mu(x)dx\Big)\\ &=\int_\Omega\nabla y_q(x)^T\nabla v_q(x)dx.\end{aligned}$$

This shows the symmetry, $\mathbf{v}^T\mathbf{A}\mathbf{y}=\mathbf{y}^T\mathbf{A}\mathbf{v}$ for all vectors $\mathbf{y},\mathbf{v}$ with $\mathbf{y}_\mu=\mathbf{v}_\mu=0$, $\mu\in J_D$, and the positive definiteness,

$$\mathbf{v}^T\mathbf{A}\mathbf{v}=\int_\Omega\nabla v_q(x)^T\nabla v_q(x)dx>0$$

for all vectors $\mathbf{v}\neq\mathbf{0}$ with $\mathbf{v}_\mu=0$, $\mu\in J_D$. □

4.2.3 Discretization of the Strong Formulation

Consider the boundary value problem (4.2.1). Note that in the strong sense, the normal derivative is not well defined at the corners of the domain. Place a tensor product grid of LGL quadrature points *less corners* on Ω. The tensor product grid of quadrature points less corners is illustrated in Figure 4.4.

To define the 2D Lagrange basis functions on $\overline{\Omega}$, let $\mathbf{z}$ be a set of 1D LGL points on a general interval (a,b). Then consider the points on the interior of the interval (i.e. consider $\{\mathbf{z}_k\}_{k=2}^{q-1}$). Define the Lagrange basis polynomials for interpolating the interior points on the interval as follows

$$\varphi_k(z)=\prod_{\substack{j=2\\k\neq j}}^{q-1}\frac{z-\mathbf{z}_k}{\mathbf{z}_j-\mathbf{z}_k},\qquad j=2,\ldots,q-1. \tag{4.2.22}$$

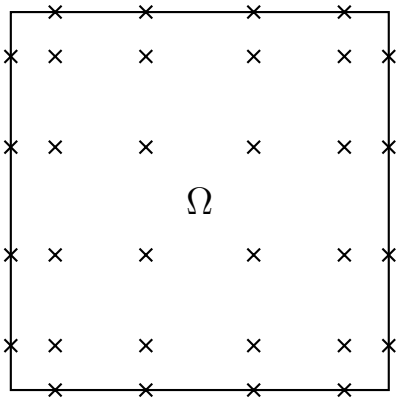

Figure 4.4: Illustration of the tensor product grid of LGL quadrature points less corners $\{\widetilde{\mathbf{x}}_\sigma\}_{\sigma=1}^{q^2-4}$ on $\overline{\Omega}$ for $q = 6$. Observe that the only difference between this set of points and the set considered in Figure 4.2 is the removal of the collocation points on the corners of the domain.

Analogous to (3.2.2), define the mapping $\sigma : (i, j, q) \mapsto \mathbb{Z}^+$ such that $\widetilde{\mathbf{x}}_{\sigma(i,j,q)} = (\mathbf{x}_i^{(a,b)}, \mathbf{x}_j^{(c,d)})$ for each (i, j) such that $(\mathbf{x}_i^{(a,b)}, \mathbf{x}_j^{(c,d)})$ is in the tensor product grid less corners.

The 2D Lagrange basis polynomials for the tensor product grid of LGL points less corners are defined as follows. For basis functions corresponding to points on the interior of the domain, $\widetilde{\mathbf{x}}_{\sigma(i,j,q)} \in \Omega$, the basis function is the same as in the weak formulation

$$\widetilde{\psi}_{\sigma(i,j,q)}(x) = \psi_i(x_1)\psi_j(x_2). \tag{4.2.23}$$

However, for basis functions corresponding to collocation nodes on the boundary, the basis functions are modified to account for the removal of the corner nodes. For basis functions corresponding to points on the north and south edges of the domain, $\widetilde{\mathbf{x}}_{\sigma(i,j,q)} \in \partial\Omega \cap (\{x \mid x_2 = 0\} \cup \{x \mid x_2 = 1\})$, the basis functions are given by

$$\widetilde{\psi}_{\sigma(i,j,q)}(x) = \varphi_i(x_1)\psi_j(x_2). \tag{4.2.24}$$

Similarly, for basis function corresponding to points on the east and west edges,

$\widetilde{\mathbf{x}}_{\sigma(i,j,q)} \in \partial\Omega \cap (\{x \mid x_1 = 0\} \cup \{x \mid x_1 = 1\})$, the basis functions are given by

$$\widetilde{\psi}_{\sigma(i,j,q)}(x) = \psi_i(x_1)\varphi_j(x_2). \tag{4.2.25}$$

Given this definition of the basis functions, the polynomial interpolation of a function $r : \Omega \to \mathbb{R}$ is given by

$$r(x) \approx \widetilde{r}_q(x) = \sum_{\sigma=1}^{q^2-4} r(\widetilde{\mathbf{x}}_\sigma)\widetilde{\psi}_\sigma(x). \tag{4.2.26}$$

The quadrature is obtained by integrating the polynomial approximation

$$\int_\Omega r(x)dx \approx \int_\Omega \widetilde{r}_q(x)dx = \sum_{\sigma=1}^{q^2-4} \widetilde{\mathbf{w}}_\sigma r(\widetilde{\mathbf{x}}_\sigma), \tag{4.2.27}$$

where the quadrature weights are given by

$$\widetilde{\mathbf{w}}_\sigma = \int_\Omega \widetilde{\psi}_\sigma(x)dx. \tag{4.2.28}$$

Again, note that due to the tensor product grid, the 2D quadrature weights are given by the product of the 1D quadrature weights of the polynomial approximation in each coordinate direction.

To discretize the boundary value problem (4.2.1), approximate the solution y by

$$\widetilde{y}_q(x) = \sum_{\sigma=1}^{q^2-4} \widetilde{\mathbf{y}}_\sigma \widetilde{\psi}_\sigma(x) \tag{4.2.29}$$

insert $\widetilde{y}_q$ into the boundary value problem (4.2.1), and require that (4.2.1) holds at the collocation points $\{\widetilde{\mathbf{x}}_\sigma\}_{\sigma=1}^{q^2-4}$.

To derive the corresponding linear equation, let $\widetilde{\mathbf{y}}$ be the vector given by

$$\widetilde{\mathbf{y}} = (\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_{q^2-4})^T, \tag{4.2.30}$$

and similarly, let

$$\widetilde{\mathbf{u}} = (u(\widetilde{\mathbf{x}}_1), \ldots, u(\widetilde{\mathbf{x}}_{q^2-4}))^T, \tag{4.2.31}$$

$$\widetilde{\mathbf{f}} = (f(\widetilde{\mathbf{x}}_1), \ldots, f(\widetilde{\mathbf{x}}_{q^2-4}))^T, \tag{4.2.32}$$

$$\widetilde{\mathbf{v}} = (v(\widetilde{\mathbf{x}}_1), \ldots, v(\widetilde{\mathbf{x}}_{q^2-4}))^T. \tag{4.2.33}$$

Define the entries of the k-th order partial differentiation matrices $\widetilde{\mathbf{D}_{\mathbf{1}}^k}$ and $\widetilde{\mathbf{D}_{\mathbf{2}}^k}$ by

$$\widetilde{\mathbf{D}_{\mathbf{1}}^k}_{\sigma,\ell} = \frac{\partial^k}{\partial x_1^k}\widetilde{\psi}_\sigma(\widetilde{\mathbf{x}}_\ell), \tag{4.2.34}$$

$$\widetilde{\mathbf{D}_{\mathbf{2}}^k}_{\sigma,\ell} = \frac{\partial^k}{\partial x_2^k}\widetilde{\psi}_\sigma(\widetilde{\mathbf{x}}_\ell), \tag{4.2.35}$$

so that the discretized differential operator is given by

$$\widetilde{\mathbf{L}} = -(\widetilde{\mathbf{D}_{\mathbf{1}}^2} + \widetilde{\mathbf{D}_{\mathbf{2}}^2}). \tag{4.2.36}$$

As in the weak formulation, it will be useful to partition

$$\widetilde{J} := \{1, \ldots, q^2 - 4\} = \widetilde{J}_I \cup \widetilde{J}_D \cup \widetilde{J}_{N(E)} \cup \widetilde{J}_{N(N)},$$

where the index sets $\widetilde{J}_I, \widetilde{J}_D, \widetilde{J}_{N(E)}$ and $\widetilde{J}_{N(N)}$ are defined next See also Figure 4.5 for an illustration of these index sets.

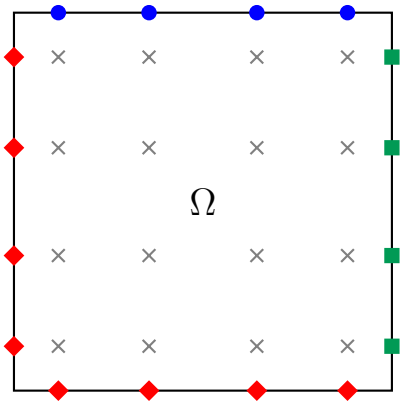

Figure 4.5: Illustration of the collocation points $\{\widetilde{\mathbf{x}}_\sigma\}_{\sigma=1}^{q^2-4}$ by index sets. The gray crosses denote the interior points corresponding to $\widetilde{J}_I$. The red diamonds denote the points corresponding to $\widetilde{J}_D$, where the Dirichlet boundary condition is applied. The green squares and blue circles denote the points where the Neumann condition is applied, corresponding to $\widetilde{J}_{N(E)}$ and $\widetilde{J}_{N(N)}$. Observe that the removal of corner points eliminates the need to apply a Neumann condition on a corner of the domain as was required for the weak form discretization (compare to Figure 4.3).

Let

$$\widetilde{J}_I = \{\sigma \mid \widetilde{\mathbf{x}}_\sigma \in \Omega\}$$

be the set of indices of collocation points in the interior of the domain and let

$$\widetilde{J}_D = \{\sigma \mid \widetilde{\mathbf{x}}_\sigma \in \Gamma_D\}$$

be the set of indices of collocation points where the Dirichlet boundary condition will be applied. The index sets corresponding to the Neumann boundary are defined as follows. Let

$$\widetilde{J}_{N(E)} = \{\sigma \mid \widetilde{\mathbf{x}}_\sigma \in \Gamma_{N(E)} \setminus (\mathcal{C}_{NE} \cup \mathcal{C}_{SE})\}$$

be the set of indices of collocation points along the east edge of the domain where the Neumann boundary condition will be applied and let

$$\widetilde{J}_{N(N)} = \{\sigma \mid \widetilde{\mathbf{x}}_\sigma \in \Gamma_{N(N)} \setminus (\mathcal{C}_{NE} \cup \mathcal{C}_{NW})\}$$

be the set of indices of collocation points along the north edge of the domain where the Neumann boundary condition will be applied. The indexed collocation points are illustrated in Figure 4.5.

As mentioned before, to discretize (4.2.1), approximate y by the interpolating polynomial $\widetilde{y}_q$, then require the resulting equation to hold at each collocation point. For the Dirichlet boundary condition, require that $\widetilde{y}^{(q)}(x) = 0$ for all $x \in \Gamma_D$. Explicitly, for $\mu \in \widetilde{J}_D$ enforce the Dirichlet boundary condition by

$$\mathbf{e}_\mu^T \widetilde{\mathbf{y}} = 0 \qquad \text{for } \mu \in \widetilde{J}_D. \tag{4.2.37}$$

For μ corresponding to collocation points on the interior of the domain, $\mu \in \widetilde{J}_I$ enforce the PDE by

$$-\Delta \widetilde{y}_q(\widetilde{\mathbf{x}}_\mu) = u(\widetilde{\mathbf{x}}_\mu) + f(\widetilde{\mathbf{x}}_\mu),$$

which can be written in terms of the discretized differential operators as

$$\mathbf{e}_\mu^T \widetilde{\mathbf{L}}\widetilde{\mathbf{y}} = \mathbf{e}_\mu^T \widetilde{\mathbf{u}} + \widetilde{\mathbf{f}}_\mu \qquad \text{for } \mu \in \widetilde{J}_I. \tag{4.2.38a}$$

For μ corresponding to collocation points on the east edge of the domain, $\mu \in \widetilde{J}_{N(E)}$ enforce the strong form of the Neumann condition by

$$\frac{\partial}{\partial x_1}\widetilde{y}_q(\widetilde{\mathbf{x}}_\mu) = v(\widetilde{\mathbf{x}}_\mu),$$

which can be written in terms of the discretized differential operators as

$$\mathbf{e}_\mu^T \widetilde{\mathbf{D_1}}\widetilde{\mathbf{y}} = \mathbf{v}_\mu \qquad \text{for } \mu \in \widetilde{J}_{N(E)}. \tag{4.2.38b}$$

Finally, for μ corresponding to collocation points on the north edge of the domain $\mu \in \widetilde{J}_{N(N)}$ enforce the strong form of the Neumann condition by

$$\frac{\partial}{\partial x_2}\widetilde{y}_q(\widetilde{\mathbf{x}}_\mu) = v(\widetilde{\mathbf{x}}_\mu),$$

which can be written in terms of the discretized differential operators as

$$\mathbf{e}_\mu^T \widetilde{\mathbf{D_2}}\widetilde{\mathbf{y}} = \mathbf{v}_\mu \qquad \text{for } \mu \in \widetilde{J}_{N(N)}. \tag{4.2.38c}$$

In summary, the collocation approximation of the strong form (4.2.1) leads to the linear system (4.2.37), (4.2.38) in $\mathbf{y}$.

Comparing the strong formulation with the weak formulation, observe that the discretized equations that enforce the PDE at the interior collocation points and the Dirichlet boundary condition are the same. However, the discretized equations that enforce the Neumann boundary condition are different. In particular, the weak formulation includes the body load (or source term) in the implementation of the Neumann boundary condition, but the strong formulation does not.

Now that the each discretization of the state equation for a single domain is understood, both can be extended to the multidomain case as will be necessary for the HPS method.

4.3 Multidomain Discretization

Let $\Omega = (0,1)^2$ and consider the boundary value problem

$$\begin{cases} -\Delta y(x) = u(x) + f(x), & x \in \Omega, \\ \quad y(x) = 0, & x \in \partial\Omega. \end{cases} \tag{4.3.1}$$

The HPS method discretization partitions the domain via a binary space partitioning tree, then formulates a local boundary value problem each subdomain. The local boundary value problems require that the solution to the a given local boundary value problem has consistent Dirichlet and Neumann boundary conditions with the solution to each neighoring local boundary value problem. That is, at each subdomain interface the solution and its derivative are continuous. The discussion of the discretization is restricted to a problem with four subdomains (corresponding to four leafs in the binary space partitioning tree). Considering the "four leaf problem" simplifies discussion, but extends naturally to the more general case.

4.3.1 The Four Leaf Discretization

Consider the uniform partition of the domain Ω into four subdomains

$$\Omega^1 = (0,0.5)^2, \quad \Omega^2 = (0,0.5) \times (0.5,1), \quad \Omega^3 = (0.5,1) \times (0,0.5), \quad \Omega^4 = (0.5,1)^2,$$

so that $\overline{\Omega} = \cup_{k=1}^4 \overline{\Omega^k}$. The geometry partition is illustrated in Figure 4.6.

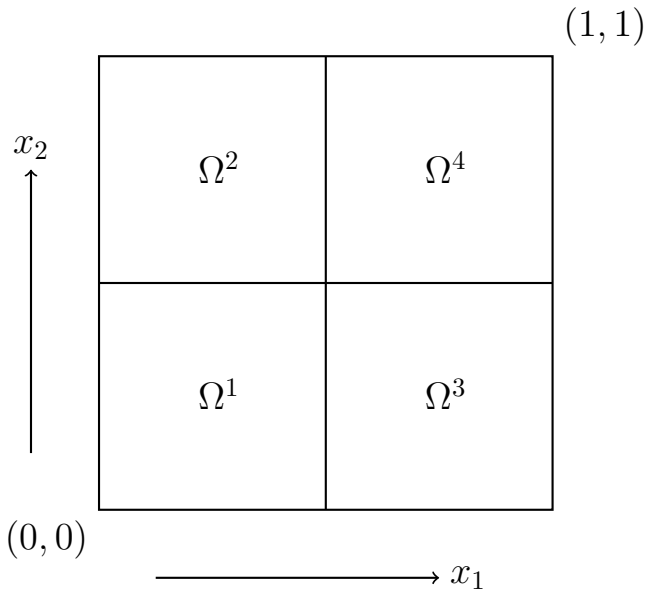

Figure 4.6: Illustration of the partition of $\overline{\Omega}$ for the four leaf problem. Note this corresponds to two levels of binary space partitioning.

From the geometry of the partition, introduce the following sets which will be useful for defining index sets as in the single domain case. Let $\mathcal{C}$ denote the intersection of each of the four subdomains

$$\mathcal{C} = \cap_{k=1}^{4} \partial\Omega^k = \{(0.5, 0.5)\}.$$

Let $\Gamma^{(k,\ell)}$ denote the interior of the shared edge between neighboring subdomains Ω^k and Ω^ℓ

$$\Gamma^{k,\ell} = (\partial\Omega^k \cap \partial\Omega^\ell) \setminus (\mathcal{C} \cup \partial\Omega).$$

The solution of (4.3.1) restricted to the subdomain Ω^k will be denoted by $y^{(k)}$,

$$y^{(k)} = y|_{\Omega^k}.$$

Then by requiring the solution to the differential equation (4.3.1) to have consistent Dirichlet and Neumann conditions at the subdomain interfaces $\Gamma^{k,\ell}$ and $\mathcal{C}$, the

differential equation on the partitioned geometry is formally given by

$$-\Delta y^{(k)}(x) = u(x) + f(x), \qquad x \in \Omega^k, k \in \{1, \ldots, 4\}, \tag{4.3.2a}$$

$$y^{(k)}(x) = 0, \qquad x \in \partial\Omega \cap \partial\Omega^k, \tag{4.3.2b}$$

$$y^{(k)}(x) - y^{(\ell)}(x) = 0, \qquad x \in \Gamma^{k,\ell}, k, \ell \in \{1, \ldots, 4\}, k \neq \ell, \tag{4.3.2c}$$

$$\frac{\partial}{\partial n^{(k)}} y^{(k)}(x) + \frac{\partial}{\partial n^{(\ell)}} y^{(\ell)}(x) = 0, \qquad x \in \Gamma^{k,\ell}, k, \ell \in \{1, \ldots, 4\}, k \neq \ell, \tag{4.3.2d}$$

$$\sum_{k=1}^{4} \frac{\partial}{\partial n^{(k)}} y^{(k)}(x) = 0, \qquad x \in \mathcal{C}, \tag{4.3.2e}$$

where $\frac{\partial}{\partial n^{(k)}}$ denotes the partial derivative with respect to the outward pointing normal vector of Ω^k. Note that along an edge shared by two neighboring subdomains, the outward pointing normal vectors have opposite direction. Additionally it is important to recognize that the normal derivative at the corner point $x \in \mathcal{C}$ is only well defined in the weak sense. As such the discretization of (4.3.2) corresponding to the weak form treats (4.3.2e) explicitly. In contrast, the discretization of (4.3.2) corresponding to the strong form does not have a collocation point at $\mathcal{C}$. Instead, the continuity of the solution and derivative at $\mathcal{C}$ is enforced implicitly by the approximation of the solution as a polynomial. These differences in interpretation and implementation ultimately result in different discretizations of (4.3.2).

4.3.2 Weak Formulation Error Estimates

In this section, a multidomain error estimate is developed for one dimensional problems using approximation results from [13] and the references therein.

Consider the one dimensional boundary value problem

$$\begin{cases} -\dfrac{d^2}{dx^2} y(x) = u(x) + f(x), & x \in (-1, 1), \\ \quad y(-1) = y(1) = 0. \end{cases} \tag{4.3.3}$$

Let $\mathcal{V} = H_0^1(-1,1)$. Then the weak form of (4.3.3) is given by, find $y \in \mathcal{V}$ such that

$$\int_{-1}^{1} \frac{d}{dx}y(x)\frac{d}{dx}\phi(x)dx = \int_{-1}^{1}(u(x)+f(x))\phi(x)dx, \qquad \forall \phi \in \mathcal{V}. \tag{4.3.4}$$

For the one dimensional multidomain problem, let

$$-1 = \mathtt{x}_0 < \mathtt{x}_1 < \ldots < \mathtt{x}_{K-1} < \mathtt{x}_K = 1$$

and define the subintervals $\mathcal{I}^\tau = (\mathtt{x}_{\tau-1}, \mathtt{x}_\tau)$ for $\tau = 1, \ldots, K$. Let $\mathcal{P}_{q-1}^K(-1,1)$ be the set of functions in $L^2(-1,1)$ such that for the restriction to a subdomain $\mathcal{I}^\tau$ the function is a polynomial of degree less than or equal to $(q-1)$, that is

$$\mathcal{P}_{q-1}^K(-1,1) = \{\phi \in L^2(-1,1) \mid \phi\big|_{\mathcal{I}^\tau} \in \mathcal{P}_{q-1}(\mathcal{I}^\tau), \tau = 1.\ldots, K\}. \tag{4.3.5}$$

Define the finite dimensional subspace

$$\mathcal{V}_q(-1,1) = \mathcal{P}_{q-1}^K(-1,1) \cap H_0^1(-1,1). \tag{4.3.6}$$

Then the Galerkin discretization of (4.3.4) is given by, find $y_q \in \mathcal{V}_q$ such that

$$\int_{-1}^{1} \frac{d}{dx}y_q(x)\frac{d}{dx}\phi(x)dx = \int_{-1}^{1}(u(x)+f(x))\phi(x)dx, \qquad \forall \phi \in \mathcal{V}_q. \tag{4.3.7}$$

Theorem 4.3.1 below provides an approximation error result for the piecewise polynomial subspace $\mathcal{V}_q$ (for details refer to [13]).

For $k = 1, \ldots, K$, let $\mathbf{x}_j^k \in [\mathtt{x}_{k-1}, \mathtt{x}_k]$, $j = 1, \ldots, q$, be the LGL collocation points in $[\mathtt{x}_{k-1}, \mathtt{x}_k]$, let ψ_j^k, $j = 1, \ldots, q$, be the corresponding Lagrange polynomials, and let w_j^k, $j = 1, \ldots, q$, be the LGL quadrature nodes. Define the interpolation operator Let $I_{q-1} : C[a,b] \to \mathcal{P}_{q-1}^K(-1,1)$ be the interpolation operator

$$(I_{q-1}u)(x) = \sum_{k=1}^{K}\sum_{j=1}^{q} u(\mathbf{x}_j^k)\psi_j^k(x). \tag{4.3.8}$$

Theorem 4.3.1 *Let $k \in \{0,1\}$ and $m > (1+k)/2$. There exists a constant $C > 0$ such that for all functions $r \in H^m(-1,1)$ the following interpolation error estimate holds*

$$\|r - I_{q-1}r\|_{H^k(-1,1)} \le C(q-1)^{k-m}\|r\|_{H^m(-1,1)}. \tag{4.3.9}$$

Remark 4.3.2 *The right hand side in (4.3.9) also depends on the subinterval lengths $\boldsymbol{x}_k - \boldsymbol{x}_{k-1}$. Since uniform partitions are used in this thesis, this term is dropped. See, e.g., [13] for details.*

A bound for the error between the exact solution and the solution to the finite dimensional subspace approximation is given by Lemma 4.3.3.

Lemma 4.3.3 *Let the weak solution y to (4.3.4) satisfy $y \in H^m(-1,1)$ with $m > 3/2$. There exists a constant $C > 0$ such that the error satisfies*

$$\|y - y_q\|_{H^1(-1,1)} \leq C(q-1)^{1-m}\|y\|_{H^m(-1,1)}. \tag{4.3.10}$$

Similar to the error bound for the single domain discretization in Lemma 4.2.2, the error bound in Lemma 4.3.3 is obtained by substituting the polynomial approximation error estimate from Theorem 4.3.1 into Céa's Lemma.

Remark 4.3.4 *The error bound in Lemma 4.3.3 requires that integrals such as $\int_\Omega (u(x) + f(x))\,\phi(x)dx$ for $\phi \in \mathcal{V}_q$ are evaluated exactly. If instead they are approximted by quadrature an additional term on the right hand side arises, which is proportional to the quadrature error.*

4.3.3 Discretization of the Weak Formulation

Let $\mathcal{V} = \{\phi \in H^1(\Omega) \mid \phi = 0 \text{ on } \partial\Omega\}$. Then the weak formulation for (4.3.1) is given by, find $y \in \mathcal{V}$ such that

$$a(y,\phi) = b(\phi), \quad \forall\, \phi \in \mathcal{V}, \tag{4.3.11}$$

where

$$a(y,\phi) = \int_\Omega \nabla y(x) \cdot \nabla\phi(x)dx \quad \text{and} \quad b(\phi) = \int_\Omega (u(x) + f(x))\phi(x)dx.$$

Let $\mathcal{P}^4_{q-1}(\Omega)$ be the set of functions ϕ on Ω such that ϕ restricted to Ω^k for $k \in \{1,2,3,4\}$ is a polynomial of degree no more than $(q-1)$ and define the subspace

$$\mathcal{V}_q = \mathcal{P}^4_{q-1}(\Omega) \cap \mathcal{V}.$$

The Galerkin discretization of (4.3.11) is given by, find $y_q \in \mathcal{V}_q$ such that

$$\int_\Omega \nabla y_q(x) \nabla \phi(x) dx = \int_\Omega \left(u(x) + f(x)\right) \phi(x) dx, \qquad \forall\, \phi \in \mathcal{V}_q. \tag{4.3.12}$$

The solution of (4.3.12) restricted to the subdomain Ω^k will be denoted by $y_q^{(k)}$,

$$y_q^{(k)} = y|_{\overline{\Omega^k}}.$$

In terms of the partitioned geometry, (4.3.12) can be written as

$$\sum_{k=1}^4 \int_{\Omega^k} \nabla y_q^{(k)}(x) \nabla \phi(x) dx = \sum_{k=1}^4 \int_{\Omega^k} \left(u(x) + f(x)\right) \phi(x) dx, \qquad \forall\, \phi \in \mathcal{V}_q. \tag{4.3.13}$$

Because y_q is a polynomial on each subdomain, and thus smooth on each subdomain, the left hand side can be rewritten by the diverence theorem, which yields

$$\begin{aligned} \sum_{k=1}^4 \int_{\Omega^k} -\Delta y_q^{(k)}(x) \phi(x) dx + \sum_{k=1}^4 \int_{\partial\Omega^k \setminus \partial\Omega} (\nabla y_q^{(k)}(x) \cdot \bar{n}) \phi(x) dx \\ = \sum_{k=1}^4 \int_{\Omega^k} \left(u(x) + f(x)\right) \phi(x) dx, \quad \forall\, \phi \in \mathcal{V}_q. \end{aligned} \tag{4.3.14}$$

Now I set up and solve a linear system corresponding to the discretization of the weak formulation of the state equation on four leaf boxes. Discretize each subdomain as in the single domain case by placing a tensor product grid of LGL quadrature points on each subdomain. Let $\mathbf{x}_j^{(k)}$ denote collocation point j of the tensor product grid on $\overline{\Omega^k}$, and let $\psi_j^{(k)}(x)$ be the 2D Lagrange interpolation basis function for the collocation points on domain $\overline{\Omega^k}$ associated with the point $\mathbf{x}_j^{(k)}$. Let $\mathbf{w}_j^{(k)}$ be the 2D LGL quadrature weight for the collocation points $\mathbf{x}_j^{(k)}$. Note that $\mathbf{w}_j^{(k)} = w_{j,1}^{(k)} w_{j,2}^{(k)}$, where $w_{j,\ell}^{(k)}$ is the 1D LGL quadrature weight in the x_ℓ direction associated with the point $\mathbf{x}_j^{(k)}$.

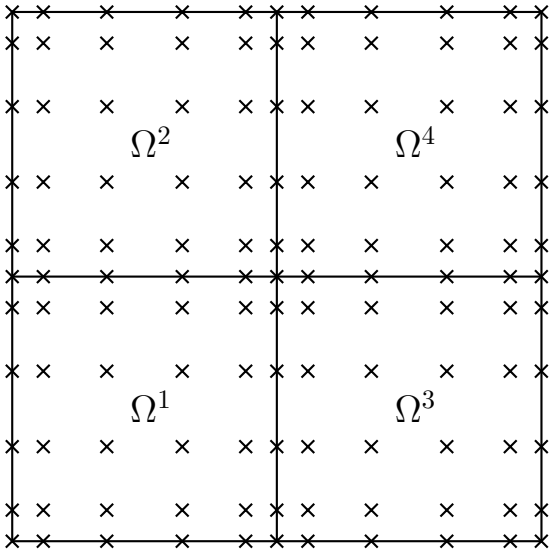

Figure 4.7: Illustration of the tensor product grid of LGL quadrature points on each subdomain $\overline{\Omega^k}$ for $q = 6$. Observe that there are not $4q^2$ unique collocation points due the fact collocation points that lie on the intersection of subdomain boundaries are described from each subdomain.

Let π be a mapping such that the index pair (j, k) for each collocation point $\mathbf{x}_j^{(k)}$ is mapped to a single index (i.e. $\pi(j, k) = \mu$ means that $\mathbf{x}_j^{(k)} = \mathbf{x}_\mu$). Note that this mapping is not one-to-one as for some μ, $\mathbf{x}_\mu \in \overline{\Omega^k} \cap \overline{\Omega^\ell}$.

It will prove useful to be able to "undo" the mapping π. To accomplish this, define the mapping ρ such that if $\pi(j, k) = \mu$ then $\rho(\mu, k) = j$ (i.e. $\rho(\mu, k) = j$ means that $\mathbf{x}_\mu = \mathbf{x}_j^{(k)}$).

As in the single domain case, it will be useful to partition

$$J := \{1, \ldots, q^2\} = J_{I(k)} \cup J_B \cup J_{E(k,\ell)} \cup J_{M(k,\ell)} \cup J_C$$

where the index sets $J_{I(k)}, J_B, J_{E(k,\ell)}, J_{M(k,\ell)}$ and J_C are defined next. See Figure 4.8 for an illustration of these index sets.

As in the single domain discretization, introduce the following convenient sets of indices.

Let $J_{I(k)}$ index the collocation points on the interior of the k-th subdomain

$$J_{I(k)} = \{\pi(j,k) \mid \mathbf{x}_j^{(k)} \in \Omega^k\}.$$

Let J_B index the collocation points that intersect the boundary of the whole domain

$$J_B = \{\pi(j,k) \mid \mathbf{x}_j^{(k)} \in \partial\Omega\}.$$

Let $J_{E(k,\ell)}$ index the collocation points on the shared edge of neighboring subdomains Ω^k and Ω^ℓ

$$J_{E(k,\ell)} = \{\pi(j,k) \mid \mathbf{x}_j^{(k)} \in \partial\Omega^k \cap \partial\Omega^\ell\}.$$

Let $J_{M(k,\ell)}$ index the collocation points on the interior of the shared edge of neighboring subdomains Ω^k and Ω^ℓ

$$J_{M(k,\ell)} = \{\pi(j,k) \mid \mathbf{x}_j^{(k)} \in \Gamma^{k,\ell}\},$$

so that $J_{M(k,\ell)} \subset J_{E(k,\ell)}$. Finally, let J_C index the collocation point at the intersection of each of the four subdomains

$$J_C = \{\pi(j,k) \mid \mathbf{x}_j^{(k)} \in \mathcal{C}\}.$$

To represent a function on Ω in terms of the basis functions $\psi_j^{(k)}(x)$ it is necessary to extend the basis functions to $\overline{\Omega}$ while maintaining orthogonality.

For the μ corresponding to collocation points on the interior of a subdomain, $\mu = \pi(j,k) \in J_{I(k)}$, extend the local basis functions to the whole domain by

$$\psi_\mu(x) = \begin{cases} \psi_j^{(k)}(x), & x \in \overline{\Omega^k}, \\ 0, & x \in \overline{\Omega} \setminus \overline{\Omega^k}. \end{cases}$$

For μ corresponding to collocation points on the boundary of the whole domain that

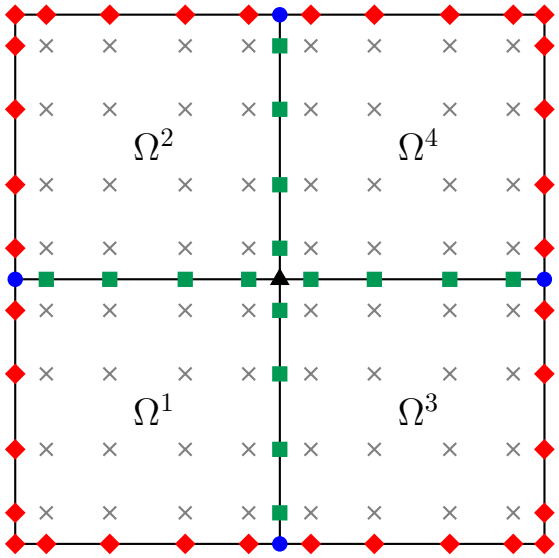

Figure 4.8: Illustration of the collocation points for the four leaf problem by index sets. The gray crosses denote interior points of $\overline{\Omega^k}$, corresponding to $J_{I(k)}$. The red diamonds denote points on the boundary of $\overline{\Omega}$ that lie on the boundary of a single subdomain, corresponding to J_B. The blue circles denotes points on the boundary of $\overline{\Omega}$ that lie in the intersection of two subdomain boundaries corresponding to $J_B \cap J_{E(k,\ell)}$. The green squares denote interior points of $\overline{\Omega}$ that lie in the intersection of exactly two subdomain boundaries $\partial\Omega^k$ and $\partial\Omega^\ell$, corresponding to $J_{M(k,\ell)}$. Finally, the black triangle denotes the point that lies in the intersection of all four subdomain boundaries.

only lie on the boundary of a single subdomain, $\mu = \pi(j,k) \in J_B \setminus J_{E(k,\ell)}$, extend the local basis functions by

$$\psi_\mu(x) = \begin{cases} \psi_j^{(k)}(x), & x \in \overline{\Omega^k}, \\ 0, & x \in \overline{\Omega} \setminus \overline{\Omega^k}. \end{cases}$$

For μ corresponding to collocation points on a subdomain interface, $\mu = \pi(j,k) = \pi(m,\ell) \in J_{E(k,\ell)} \setminus J_C$, extend the local basis functions by

$$\psi_\mu(x) = \begin{cases} \psi_j^{(k)}(x), & x \in \overline{\Omega^k}, \\ \psi_m^{(\ell)}(x), & x \in \overline{\Omega^\ell} \setminus \overline{\Omega^k}, \\ 0, & x \in \overline{\Omega} \setminus (\overline{\Omega^k} \cup \overline{\Omega^\ell}). \end{cases}$$

Finally, for μ corresponding the collocation point at the intersection of all four subdomains (i.e. at the point $(0.5, 0.5)$), $\mu = \pi(j,1) = \pi(k,2) = \pi(\ell,3) = \pi(m,4) \in J_C$, extend the basis function by

$$\psi_\mu(x) = \begin{cases} \psi_j^{(1)}(x), & x \in \overline{\Omega^1}, \\ \psi_k^{(2)}(x), & x \in \overline{\Omega^2} \setminus \overline{\Omega^1}, \\ \psi_\ell^{(3)}(x), & x \in \overline{\Omega^3} \setminus \overline{\Omega^1}, \\ \psi_m^{(4)}(x), & x \in \overline{\Omega^4} \setminus (\overline{\Omega^2} \cup \overline{\Omega^3}). \end{cases}$$

Now that the local basis functions have been extended to the whole domain, a function $r \in \mathcal{V}_q$ may be represented by the basis expansion

$$r(x) = \sum_{\mu \in J} r(\mathbf{x}_\mu)\psi_\mu(x). \tag{4.3.15}$$

Let $\mathbf{y}$ be given by

$$\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T \tag{4.3.16}$$

and let $\mathbf{y}_\mu$ denote the μ-th element of $\mathbf{y}$ (i.e. $\mathbf{y}_\mu = y_q(\mathbf{x}_\mu)$). Furthermore, let

$$\mathbf{u} = (u(\mathbf{x}_1), \ldots, u(\mathbf{x}_N))^T, \tag{4.3.17}$$

$$\mathbf{f} = (f(\mathbf{x}_1), \ldots, y(\mathbf{x}_N))^T. \tag{4.3.18}$$

Let $\mathbf{D_1}^{(k)}, \mathbf{D_2}^{(k)}$ and $\mathbf{L}^{(k)}$ be the local partial differentiation matrices and the local discretized differential operator on Ω^k, so that

$$\frac{\partial}{\partial x_1} y^{(k)}(\mathbf{x}_j^{(k)}) = \mathbf{e}_j^T \mathbf{D_1}^{(k)} \mathbf{y}^{(k)}, \tag{4.3.19a}$$

$$\frac{\partial}{\partial x_2} y^{(k)}(\mathbf{x}_j^{(k)}) = \mathbf{e}_j^T \mathbf{D_2}^{(k)} \mathbf{y}^{(k)}, \tag{4.3.19b}$$

$$-\Delta y^{(k)}(\mathbf{x}_j^{(k)}) = \mathbf{e}_j^T \mathbf{L}^{(k)} \mathbf{y}^{(k)}. \tag{4.3.19c}$$

To obtain the linear system corresponding to (4.3.14), replace the integrals by quadrature and require the resulting equation to hold for all $\phi \in \text{span}\{\psi_1, \ldots, \psi_N\}$ such that $\phi = 0$ on $\partial\Omega$. Note that the quadrature rule is exact for the integrals in (4.3.14) involving y_q.

The Dirichlet boundary condition $y_q(x) = 0$ for all $x \in \partial\Omega$ implies

$$\mathbf{e}_\mu^T \mathbf{y} = 0, \qquad \text{for } \mu \in J_B. \tag{4.3.20}$$

Thus for all μ such that the collocation point $\mathbf{x}_\mu$ is not in $\partial\Omega$, i.e. $\mu \in J \setminus J_B$ the

discretization of (4.3.14) is given by

$$\begin{aligned}\sum_{k=1}^{4}\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell^{(k)}\Delta y_q^{(k)}(\mathbf{x}_\ell^{(k)})\psi_\mu(\mathbf{x}_\ell^{(k)}) &+ \sum_{j\in J_{E(1,3)}} w_{j,2}\left(\frac{\partial y_q^{(1)}}{\partial x_1}(\mathbf{x}_j) - \frac{\partial y_q^{(3)}}{\partial x_1}(\mathbf{x}_j)\right)\psi_\mu(\mathbf{x}_j)\\ &+ \sum_{j\in J_{E(2,4)}} w_{j,2}\left(\frac{\partial y_q^{(2)}}{\partial x_1}(\mathbf{x}_j) - \frac{\partial y_q^{(4)}}{\partial x_1}(\mathbf{x}_j)\right)\psi_\mu(\mathbf{x}_j)\\ &+ \sum_{j\in J_{E(1,2)}} w_{j,1}\left(\frac{\partial y_q^{(1)}}{\partial x_2}(\mathbf{x}_j) - \frac{\partial y_q^{(2)}}{\partial x_2}(\mathbf{x}_j)\right)\psi_\mu(\mathbf{x}_j)\\ &+ \sum_{j\in J_{E(3,4)}} w_{j,1}\left(\frac{\partial y_q^{(3)}}{\partial x_2}(\mathbf{x}_j) - \frac{\partial y_q^{(4)}}{\partial x_2}(\mathbf{x}_j)\right)\psi_\mu(\mathbf{x}_j)\\ &= \sum_{k=1}^{4}\sum_{\ell=1}^{q^2}\mathbf{w}_\ell^{(k)}\left(u(\mathbf{x}_\ell^{(k)}) + f(\mathbf{x}_\ell^{(k)})\right)\psi_\mu(\mathbf{x}_\ell^{(k)}).\end{aligned} \tag{4.3.21}$$

For μ corresponding to collocation points on the interior of a subdomain, $\mu \in J_{I(k)}$, $k = 1, \ldots, 4$, (4.3.21) simplifies as follows

$$\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell^{(k)}\Delta y_q^{(k)}(\mathbf{x}_\ell^{(k)})\psi_\mu(\mathbf{x}_\ell^{(k)}) = \sum_{\ell=1}^{q^2}\mathbf{w}_\ell^{(k)}\left(u(\mathbf{x}_\ell^{(k)}) + f(\mathbf{x}_\ell^{(k)})\right)\psi_\mu(\mathbf{x}_\ell^{(k)}).$$

Due to the Lagrange basis functions, each sum reduces to a single term

$$-\mathbf{w}_\mu\Delta y_q^{(k)}(\mathbf{x}_\mu) = \mathbf{w}_\mu\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right).$$

Dividing by the constant $\mathbf{w}_\mu$ yields

$$-\Delta y_q^{(k)}(\mathbf{x}_\mu) = u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu).$$

This can be written in terms of the local discretized differential operators as

$$\mathbf{e}_{\rho(\mu,k)}^T\mathbf{L}^{(k)}\mathbf{y}_{\pi(1:q^2,k)} = \mathbf{e}_\mu^T\mathbf{u} + \mathbf{f}_\mu, \qquad \text{for } \mu \in J_{I(k)}. \tag{4.3.22a}$$

For μ corresponding to collocation points on the interior of the shared edge between

Ω^1 and Ω^3, that is $\mu \in J_{M(1,3)}$, (4.3.21) simplifies as

$$\sum_{k=1}^{4}\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell^k \Delta y_q^{(k)}(\mathbf{x}_\ell^{(k)})\psi_\mu(\mathbf{x}_\ell^{(k)}) + \sum_{j\in J_{E(1,3)}} w_{j,2}\left(\frac{\partial y_q^{(1)}}{\partial x_1}(\mathbf{x}_j) - \frac{\partial y_q^{(3)}}{\partial x_1}(\mathbf{x}_j)\right)\psi_\mu(\mathbf{x}_j)$$
$$= \sum_{k=1}^{4}\sum_{\ell=1}^{q^2} \mathbf{w}_\ell^{(k)}\left(u(\mathbf{x}_\ell^{(k)}) + f(\mathbf{x}_\ell^{(k)})\right)\psi_\mu(\mathbf{x}_\ell^{(k)}).$$

Again each sum has only one non-zero term, which yields

$$\left(-\mathbf{w}_\mu \Delta y_q^{(1)}(\mathbf{x}_\mu) - \mathbf{w}_\mu \Delta y_q^{(3)}(\mathbf{x}_\mu) + w_{\mu,2}\frac{\partial y_q^{(1)}}{\partial x_1}(\mathbf{x}_\mu) - w_{\mu,2}\frac{\partial y_q^{(3)}}{\partial x_1}(\mathbf{x}_\mu)\right)$$
$$= 2\mathbf{w}_\mu\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right).$$

Dividing by the constant $w_{\mu,2}$ gives

$$\left(-w_{\mu,1} \Delta y_q^{(1)}(\mathbf{x}_\mu) + \frac{\partial y_q^{(1)}}{\partial x_1}(\mathbf{x}_\mu) - w_{\mu,1}\Delta y_q^{(3)}(\mathbf{x}_\mu) - \frac{\partial y_q^{(3)}}{\partial x_1}(\mathbf{x}_\mu)\right)$$
$$= 2w_{\mu,1}\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right).$$

This can be written in terms of the local discretized differential operators as

$$\mathbf{e}_{\rho(\mu,1)}^T\left(w_{\mu,1}\mathbf{L}^{(1)} + \mathbf{D}_\mathbf{1}^{(1)}\right)\mathbf{y}_{\pi(1:q^2,1)} + \mathbf{e}_{\rho(\mu,3)}^T\left(w_{\mu,1}\mathbf{L}^{(3)} - \mathbf{D}_\mathbf{1}^{(3)}\right)\mathbf{y}_{\pi(1:q^2,3)}$$
$$= 2w_{\mu,1}\left(\mathbf{e}_\mu^T\mathbf{u} + \mathbf{f}_\mu\right), \qquad \text{for } \mu \in J_{M(1,3)}. \tag{4.3.22b}$$

Following a similar process for $\mu \in J_{M(2,4)}$ yields

$$\left(-w_{\mu,1} \Delta y_q^{(2)}(\mathbf{x}_\mu) + \frac{\partial y_q^{(2)}}{\partial x_1}(\mathbf{x}_\mu) - w_{\mu,1}\Delta y_q^{(4)}(\mathbf{x}_\mu) - \frac{\partial y_q^{(4)}}{\partial x_1}(\mathbf{x}_\mu)\right)$$
$$= 2w_{\mu,1}\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right),$$

which in terms of the local discretized differential operators is given by

$$\mathbf{e}_{\rho(\mu,2)}^T\left(w_{\mu,1}\mathbf{L}^{(2)} + \mathbf{D}_\mathbf{1}^{(2)}\right)\mathbf{y}_{\pi(1:q^2,2)} + \mathbf{e}_{\rho(\mu,4)}^T\left(w_{\mu,1}\mathbf{L}^{(4)} - \mathbf{D}_\mathbf{1}^{(4)}\right)\mathbf{y}_{\pi(1:q^2,4)}$$
$$= 2w_{\mu,1}\left(\mathbf{e}_\mu^T\mathbf{u} + \mathbf{f}_\mu\right), \qquad \text{for } \mu \in J_{M(2,4)}. \tag{4.3.22c}$$

For μ corresponding to collocation points on the interior of the shared edge between Ω^1 and Ω^2, that is $\mu \in J_{M(1,2)}$, (4.3.21) simplifies as follows

$$\sum_{k=1}^{4}\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell^k \Delta y_q^{(k)}(\mathbf{x}_\ell^{(k)})\psi_\mu(\mathbf{x}_\ell^{(k)}) + \sum_{j\in J_{E(1,2)}} w_{j,1}\left(\frac{\partial y_q^{(1)}}{\partial x_2}(\mathbf{x}_j) - \frac{\partial y_q^{(2)}}{\partial x_2}(\mathbf{x}_j)\right)\psi_\mu(\mathbf{x}_j)$$
$$= \sum_{k=1}^{4}\sum_{\ell=1}^{q^2} \mathbf{w}_\ell^{(k)}\left(u(\mathbf{x}_\ell^{(k)} + f(\mathbf{x}_\ell^{(k)})\right)\psi_\mu(\mathbf{x}_\ell^{(k)}).$$

As before, the Lagrange basis causes each sum to reduce to a single non-zero term

$$\left(-\mathbf{w}_\mu \Delta y_q^{(1)}(\mathbf{x}_\mu) - \mathbf{w}_\mu \Delta y_q^{(2)}(\mathbf{x}_\mu) + w_{\mu,1}\frac{\partial y_q^{(1)}}{\partial x_2}(\mathbf{x}_\mu) - w_{\mu,1}\frac{\partial y_q^{(2)}}{\partial x_2}(\mathbf{x}_\mu)\right)$$
$$= 2\mathbf{w}_\mu\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right).$$

Dividing by the constant $w_{\mu,1}$ yields

$$\left(-w_{\mu,2}\Delta y_q^{(1)}(\mathbf{x}_\mu) + \frac{\partial y_q^{(1)}}{\partial x_2}(\mathbf{x}_\mu) - w_{\mu,2}\Delta y_q^{(2)}(\mathbf{x}_\mu) - \frac{\partial y_q^{(2)}}{\partial x_2}(\mathbf{x}_\mu)\right)$$
$$= 2w_{\mu,2}\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right).$$

This can be written in terms of the local discretized differential operators as

$$\mathbf{e}_{\rho(\mu,1)}^T\left(w_{\mu,2}\mathbf{L}^{(1)} + \mathbf{D_2}^{(1)}\right)\mathbf{y}_{\pi(1:q^2,1)} + \mathbf{e}_{\rho(\mu,2)}^T\left(w_{\mu,2}\mathbf{L}^{(2)} - \mathbf{D_2}^{(2)}\right)\mathbf{y}_{\pi(1:q^2,2)}$$
$$= 2w_{\mu,2}\left(\mathbf{e}_\mu^T\mathbf{u} + \mathbf{f}_\mu\right), \qquad \text{for } \mu \in J_{M(1,2)}. \tag{4.3.22d}$$

Following a similar process for $\mu \in J_{M(3,4)}$ yields

$$\left(-w_{\mu,2}\Delta y_q^{(3)}(\mathbf{x}_\mu) + \frac{\partial y_q^{(3)}}{\partial x_2}(\mathbf{x}_\mu) - w_{\mu,2}\Delta y_q^{(4)}(\mathbf{x}_\mu) - \frac{\partial y_q^{(4)}}{\partial x_2}(\mathbf{x}_\mu)\right)$$
$$= 2w_{\mu,2}\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right),$$

which in terms of the local discretized differential operators is given by

$$\mathbf{e}_{\rho(\mu,3)}^T\left(w_{\mu,2}\mathbf{L}^{(3)} + \mathbf{D_2}^{(3)}\right)\mathbf{y}_{\pi(1:q^2,3)} + \mathbf{e}_{\rho(\mu,4)}^T\left(w_{\mu,2}\mathbf{L}^{(4)} - \mathbf{D_2}^{(4)}\right)\mathbf{y}_{\pi(1:q^2,4)}$$
$$= 2w_{\mu,2}\left(\mathbf{e}_\mu^T\mathbf{u} + \mathbf{f}_\mu\right), \qquad \text{for } \mu \in J_{M(3,4)}. \tag{4.3.22e}$$

For μ corresponding to the collocation point on the intersection of each of the four subdomains, that is $\mu \in J_C$, (4.3.21) simplifies as

$$\begin{aligned}\sum_{k=1}^{4} -\mathbf{w}_\mu \Delta y_q^{(k)}(\mathbf{x}_\mu) + w_{\mu,2}&\left(\frac{\partial y_q^{(1)}}{\partial x_1}(\mathbf{x}_\mu) - \frac{\partial y_q^{(3)}}{\partial x_1}(\mathbf{x}_\mu) + \frac{\partial y_q^{(2)}}{\partial x_1}(\mathbf{x}_\mu) - \frac{\partial y_q^{(4)}}{\partial x_1}(\mathbf{x}_\mu)\right)\\ &+w_{\mu,1}\left(\frac{\partial y_q^{(1)}}{\partial x_2}(\mathbf{x}_\mu) - \frac{\partial y_q^{(2)}}{\partial x_2}(\mathbf{x}_\mu) + \frac{\partial y_q^{(3)}}{\partial x_2}(\mathbf{x}_\mu) - \frac{\partial y_q^{(4)}}{\partial x_2}(\mathbf{x}_\mu)\right)\\ &= 4\mathbf{w}_\mu\left(u(\mathbf{x}_\mu) + f(\mathbf{x}_\mu)\right),\end{aligned}$$

This can be written in terms of the local discretized differential operators as

$$\begin{aligned}&\mathbf{e}_{\rho(\mu,1)}^T\left(-\mathbf{w}_\mu\mathbf{L}^{(1)} + w_{\mu,2}\mathbf{D}_\mathbf{1}^{(1)} + w_{\mu,1}\mathbf{D}_\mathbf{2}^{(1)}\right)\mathbf{y}_{\pi(1:q^2,1)}\\ &\quad+\mathbf{e}_{\rho(\mu,2}^T\left(-\mathbf{w}_\mu\mathbf{L}^{(2)} + w_{\mu,2}\mathbf{D}_\mathbf{1}^{(2)} - w_{\mu,1}\mathbf{D}_\mathbf{2}^{(2)}\right)\mathbf{y}_{\pi(1:q^2,2)}\\ &\quad+\mathbf{e}_{\rho(\mu,3}^T\left(-\mathbf{w}_\mu\mathbf{L}^{(3)} - w_{\mu,2}\mathbf{D}_\mathbf{1}^{(3)} + w_{\mu,1}\mathbf{D}_\mathbf{2}^{(3)}\right)\mathbf{y}_{\pi(1:q^2,3)}\\ &\quad+\mathbf{e}_{\rho(\mu,4}^T\left(-\mathbf{w}_\mu\mathbf{L}^{(4)} - w_{\mu,2}\mathbf{D}_\mathbf{1}^{(4)} - w_{\mu,1}\mathbf{D}_\mathbf{2}^{(4)}\right)\mathbf{y}_{\pi(1:q^2,4)}\\ &\quad=4\mathbf{w}_\mu\left(\mathbf{e}_\mu^T\mathbf{u} + \mathbf{f}_\mu\right), \qquad \text{for } \mu \in J_C.\end{aligned} \tag{4.3.22f}$$

In summary, approximating the integrals in (4.3.14) by quadrature results in the linear system (4.3.20),(4.3.22) in $\mathbf{y}$ which is denoted by

$$\mathbf{A}\mathbf{y} = -\mathbf{B}\mathbf{u} + \mathbf{c}. \tag{4.3.23}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$, $\mathbf{y}, \mathbf{u}, \mathbf{c} \in \mathbb{R}^N$. Solving the linear system for $\mathbf{y}$ provides the coefficient values of the composite polynomial approximation of the weak solution $y_q(x)$. Although the matrices in (4.2.21) and (4.3.23) are different, the same notation is used since ultimately only the multidomain discretization will be considered. Note that the matrix $\mathbf{B}$ is a diagonal matrix with entries

$$B_{\mu\mu} = \begin{cases} -1, & \mu \in \cup_{k=1}^4 J_{I(k)},\\ -2w_{\mu,1}, & \mu \in J_{M(1,3)} \cup J_{M(2,4)},\\ -2w_{\mu,2}, & \mu \in J_{M(1,2)} \cup J_{M(3,4)},\\ -4\mathbf{w}_\mu, & \mu \in J_C,\\ 0, & \mu \in J_B. \end{cases} \tag{4.3.24}$$

Analogous to Theorem 4.2.4 the following result can be proven.

Theorem 4.3.5 *The matrix* $\mathbf{A} \in \mathbb{R}^{N\times N}$ *in* (4.3.23) *is symmetric positive definite on* $\{\mathbf{v} \mid \mathbf{v}_\mu = 0, \quad \mu \in J_D\}$.

4.3.4 Discretization of the Strong Formulation

Discretize each subdomain as in the single domain case by placing a tensor product grid of LGL quadrature points less corners on each subdomain. Let $\widetilde{\mathbf{x}}_j^{(k)}$ denote collocation point j of the tensor product grid on $\overline{\Omega^k}$, and let $\widetilde{\psi}_j^{(k)}(x)$ be the 2D Lagrange interpolation basis function for the collocation points on domain $\overline{\Omega^k}$ associated with the point $\widetilde{\mathbf{x}}_j^{(k)}$. The quadrature points are illustrated in Figure 4.9.

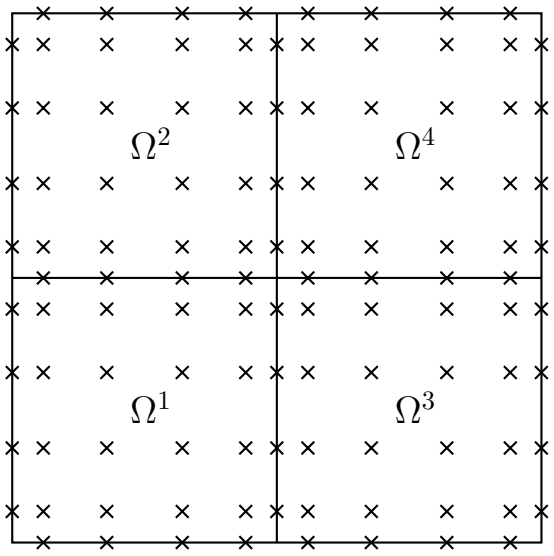

Figure 4.9: Illustration of the tensor product grid of LGL quadrature points less corners on each subdomain $\overline{\Omega^k}$ for $q = 6$. Observe that the only difference between this set of points and the set considered in Figure 4.7 is the removal of collocation points on corners of each subdomain.

Define the mapping $\widetilde{\pi}$ and $\widetilde{\rho}$ (analogous to π and ρ in Section 4.3.4) such that $\widetilde{\pi}(j,k) = \mu$ implies that $\widetilde{\mathbf{x}}_j^{(k)} = \widetilde{\mathbf{x}}_\mu$ and $\widetilde{\rho}(\mu,k) = j$.

It will again be useful to partition the index set

$$\widetilde{J} := \widetilde{J}_{I(k)} \cup \widetilde{J}_B \cup \widetilde{J}_{M(k,\ell)}$$

where the index sets $\widetilde{J}_{I(k)}, \widetilde{J}_B$, and $\widetilde{J}_{M(k,\ell)}$ will be defined next. The indexed collocation points are illustrated in Figure 4.10.

Define the index sets

$$\begin{aligned}
\widetilde{J}_{I(k)} &= \{\widetilde{\pi}(j,k) \mid \widetilde{\mathbf{x}}_j^{(k)} \in \Omega^k\},\\
\widetilde{J}_B &= \{\widetilde{\pi}(j,k) \mid \widetilde{\mathbf{x}}_j^{(k)} \in \partial\Omega\},\\
\widetilde{J}_{M(k,\ell)} &= \{\widetilde{\pi}(j,k) \mid \widetilde{\mathbf{x}}_j^{(k)} \in \Gamma^{k,\ell}\}.
\end{aligned}$$

Again it is necessary to extend the local basis functions $\widetilde{\psi}_j^{(k)}(x)$ to the entire domain $\overline{\Omega}$.

For $\mu = \widetilde{\pi}(j,k) \in \widetilde{J}_{I(k)} \cup \widetilde{J}_B$

$$\widetilde{\psi}_\mu(x) = \begin{cases} \widetilde{\psi}_j^{(k)}(x), & x \in \overline{\Omega^k},\\ 0, & x \in \overline{\Omega} \setminus \overline{\Omega^k}. \end{cases}$$

For $\mu = \widetilde{\pi}(j,k) = \pi(m,\ell) \in \widetilde{J}_{M(k,\ell)}$

$$\widetilde{\psi}_\mu(x) = \begin{cases} \widetilde{\psi}_j^{(k)}(x), & x \in \overline{\Omega^k},\\ \widetilde{\psi}_m^{(\ell)}(x), & x \in \overline{\Omega^\ell} \setminus \overline{\Omega^k},\\ 0, & x \in \overline{\Omega} \setminus (\overline{\Omega^k} \cup \overline{\Omega^\ell}). \end{cases}$$

Given the collocation points $\{\widetilde{\mathbf{x}}_\mu\}_{\mu\in\widetilde{J}}$ and the extended basis functions $\{\widetilde{\psi}_\mu(x)\}_{\mu\in\widetilde{J}}$, the composite polynomial approximation of the function y is given by

$$y(x) \approx \widetilde{y}_q(x) = \sum_{\mu\in\widetilde{J}} \widetilde{\mathbf{y}}_\mu \widetilde{\psi}_\mu(x). \tag{4.3.25}$$

The restriction of (4.3.25) to the subdomain Ω^k will be denoted by $\widetilde{y}_q^{(k)}$,

$$\widetilde{y}_q^{(k)} = \widetilde{y}|_{\overline{\Omega^k}}.$$

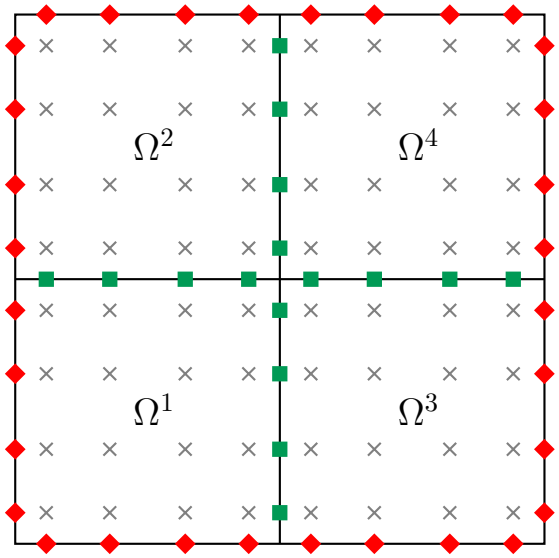

Figure 4.10: Illustration of the collocation points for the four leaf problem by index sets. The gray crosses denote interior points of $\overline{\Omega^k}$, corresponding to $\widetilde{J}_{I(k)}$. The red diamonds denote points on the boundary of $\overline{\Omega}$ that lie on the boundary of a single subdomain, corresponding to $\widetilde{J}_B$. Finally, the green squares denote interior points of $\overline{\Omega}$ that lie in the intersection of exactly two subdomain boundaries $\partial\Omega^k$ and $\partial\Omega^\ell$, corresponding to $\widetilde{J}_{M(k,\ell)}$. The important difference between this set of points and the set considered in Figure 4.8 is that there are no points that lie in the intersection of all four subdomain boundaries.

Let $\widetilde{\mathbf{D_1}}^{(k)}, \widetilde{\mathbf{D_2}}^{(k)}$ and $\widetilde{\mathbf{L}}^{(k)}$ be the partial differentiation matrices and the discretized differential operator on Ω^k as in the single domain case.

To discretize (4.3.2), approximate y by the polynomial $\widetilde{y}_q$ as in (4.3.25), then require the resulting equation to hold at each collocation point.

For the Dirichlet boundary condition, require that $\widetilde{y}_q(\widetilde{\mathbf{x}}_\mu) = 0$ for all $\widetilde{\mathbf{x}_\mu} \in \partial\Omega$

$$\widetilde{\mathbf{e}}_\mu^T \widetilde{\mathbf{y}} = 0, \qquad \text{for } \mu \in \widetilde{J}_B. \tag{4.3.26}$$

Note that the Dirichlet boundary condition is enforced explicitly as in the discretization of the weak formulation (4.3.20). For μ corresponding to collocation points on the interior of a subdomain, that is $\mu \in \widetilde{J}_{I(k)}$

$$-\Delta \widetilde{y}_q^{(k)}(\widetilde{\mathbf{x}}_\mu) = u(\widetilde{\mathbf{x}}_\mu) + f(\widetilde{\mathbf{x}}_\mu),$$

which can be written in terms of the local discretized differential operators as

$$\widetilde{\mathbf{e}}_{\widetilde{\rho}(\mu,k)}^T \widetilde{\mathbf{L}}^{(k)} \widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,k)} = \widetilde{\mathbf{e}}_\mu^T \widetilde{\mathbf{u}} + \widetilde{\mathbf{f}}_\mu, \qquad \text{for } \mu \in \widetilde{J}_{I(k)}. \tag{4.3.27a}$$

This is equivalent to the condition enforced in the discretization of the weak formulation (4.3.22a).

For μ corresponding to collocation points on the shared edge of Ω^1 and Ω^3, that is $\mu \in \widetilde{J}_{M(1,3)}$

$$\frac{\partial \widetilde{y}_q^{(1)}}{\partial x_1}(\widetilde{\mathbf{x}}_\mu) - \frac{\partial \widetilde{y}_q^{(3)}}{\partial x_1}(\widetilde{\mathbf{x}}_\mu) = 0,$$

which can be written in terms of the local discretized differential operators as

$$\widetilde{\mathbf{e}}_{\widetilde{\rho}(\mu,1)}^T \widetilde{\mathbf{D_1}}^{(1)} \widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,1)} - \widetilde{\mathbf{e}}_{\widetilde{\rho}(\mu,3)}^T \widetilde{\mathbf{D_1}}^{(3)} \widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,3)} = 0, \qquad \text{for } \mu \in \widetilde{J}_{M(1,3)}. \tag{4.3.27b}$$

In contrast to the weak formulation Neumann condition (4.3.22b), the strong formulation of the Neumann condition (4.3.27b) does not include a linear combination of the differential operator and right hand side. Similarly, for μ corresponding to collocation points on the shared edge of Ω^2 and Ω^4, that is $\mu \in \widetilde{J}_{M(2,4)}$

$$\frac{\partial \widetilde{y}_q^{(2)}}{\partial x_1}(\widetilde{\mathbf{x}}_\mu) - \frac{\partial \widetilde{y}_q^{(4)}}{\partial x_1}(\widetilde{\mathbf{x}}_\mu) = 0,$$

which can be written in terms of the local discretized differential operators as

$$\widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,2)}\widetilde{\mathbf{D_1}}^{(2)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,2)} - \widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,4)}\widetilde{\mathbf{D_1}}^{(4)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,4)} = 0, \qquad \text{for } \mu \in \widetilde{J}_{M(2,4)}. \tag{4.3.27c}$$

For μ corresponding to collocation points on the shared edge of Ω^1 and Ω^2, that is $\mu \in \widetilde{J}_{M(1,2)}$

$$\frac{\partial \widetilde{y}^{(1)}_q}{\partial x_2}(\widetilde{\mathbf{x}}_\mu) - \frac{\partial \widetilde{y}^{(2)}_q}{\partial x_2}(\widetilde{\mathbf{x}}_\mu) = 0,$$

which can be written in terms of the local discretized differential operators as

$$\widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,1)}\widetilde{\mathbf{D_2}}^{(1)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,1)} - \widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,2)}\widetilde{\mathbf{D_2}}^{(2)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,2)} = 0, \qquad \text{for } \mu \in \widetilde{J}_{M(1,2)}. \tag{4.3.27d}$$

Finally, for μ corresponding to collocation points on the shared edge of Ω^3 and Ω^4, that is $\mu \in \widetilde{J}_{M(3,4)}$

$$\frac{\partial \widetilde{y}^{(3)}_q}{\partial x_2}(\widetilde{\mathbf{x}}_\mu) - \frac{\partial \widetilde{y}^{(4)}_q}{\partial x_2}(\widetilde{\mathbf{x}}_\mu) = 0,$$

which can be written in terms of the local discretized differential operators as

$$\widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,3)}\widetilde{\mathbf{D_2}}^{(3)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,3)} - \widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,4)}\widetilde{\mathbf{D_2}}^{(4)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,4)} = 0, \qquad \text{for } \mu \in \widetilde{J}_{M(3,4)}. \tag{4.3.27e}$$

In summary, discretizing the strong formulation by composite collocation results in the linear system (4.3.26), (4.3.27) which is denoted by

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{y}} = -\widetilde{\mathbf{B}}\widetilde{\mathbf{u}} + \widetilde{\mathbf{c}}. \tag{4.3.28}$$

where $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}} \in \mathbb{R}^{(N-9)\times(N-9)}$, $\widetilde{\mathbf{y}}, \widetilde{\mathbf{u}}, \widetilde{\mathbf{c}} \in \mathbb{R}^{(N-9)}$. Solving the linear system for $\widetilde{\mathbf{y}}$ provides the coefficient values for the composite polynomial approximation of the strong solution $\widetilde{y}_q(x)$.

Note that the matrix $\widetilde{\mathbf{B}}$ is a diagonal matrix with entries

$$\widetilde{\mathbf{B}}_{\mu\mu} = \begin{cases} -1, & \mu \in \cup_{k=1}^4 \widetilde{J}_{I(k)}, \\ 0 & \mu \in \widetilde{J}_{M(1,2)} \cup \widetilde{J}_{M(1,3)} \cup \widetilde{J}_{M(2,4)} \cup \widetilde{J}_{M(3,4)}, \\ 0, & \mu \in \widetilde{J}_B. \end{cases} \tag{4.3.29}$$

There are two distinctions between the linear systems corresponding to the weak form discretization and the strong form discretization for the four leaf problem. First, the weak form discretization has collocation points at the corners of each subdomain whereas the strong form discretization does not (thus the linear system for the strong form discretization is slightly smaller than for the weak form). Second, the implementation of the Neumann condition for the weak form discretization includes a linear combination of the differential operator and right hand side whereas the implementation of the Neumann condition in the strong form discretization does not. This is made more precise in the following remark.

Remark 4.3.6 *In the strong form multidomain discretization, the control at the subdomain interfaces (i.e.* $\widetilde{\mathbf{u}}_\mu$ *for* $\mu \in \widetilde{J}_{M(k,\ell)}$*) does not enter the discretization. See* (4.3.29)*. This is the most important distinction between the strong form discretization* (4.3.28) *and the weak form discretization* (4.3.23) *of the state equation.*

In the next section, the performance of each discretization is examined for solving a simple boundary value problem.

4.4 State Equation Numerical Example

Let $\Omega = (0,1)^2$ and consider the boundary value problem

$$\begin{cases} -\Delta y(x) = u(x) + f(x), & x \in \Omega \\ \quad y(x) = 0, & x \in \partial\Omega, \end{cases}$$

where

$$\begin{aligned} u(x) &= 10\sin(3\pi x_1)\sin(\pi x_2), \\ f(x) &= 10\pi^2 \sin(\pi x_1)\sin(3\pi x_2) - 10\sin(3\pi x_1)\sin(\pi x_2), \end{aligned}$$

which yields the exact solution

$$y_{ex}(x) = \sin(\pi x_1)\sin(3\pi x_2).$$

To evaluate the performance of the weak and strong discretizations for the four leaf boundary value problem, the linear systems corresponding to the weak form and the strong form ((4.3.23) and (4.3.28) respectively) were constructed and solved over a range of polynomial orders. Figure 4.11 provides the convergence behavior by comparing the relative error in the approximate solutions y_q for the weak form and $\widetilde{y}_q$ for the strong form.

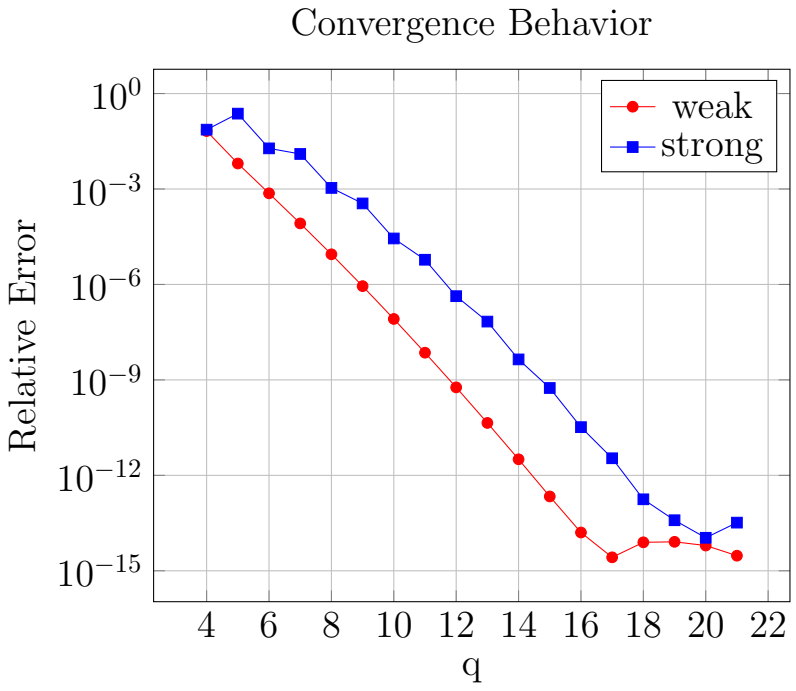

Figure 4.11: The relative L^2 errors vs. q for the weak and strong four leaf formulations applied to the test problem. Both formulations converge at similar rates. The error in the weak formulation is smaller than the error in the strong formulation for the same value of q as the presence of corner nodes allows the weak composite polynomial approximation (y_q) to represent more functions exactly compared to the strong composite polynomial approximation ($\widetilde{y}_q$).

Define the relative errors

$$E_{L^2}(y_q) = \frac{(\mathbf{y} - y_{ex}(\mathbf{x}))^T \mathbf{W} (\mathbf{y} - y_{ex}(\mathbf{x}))}{\max_j |y(\mathbf{x}_j)|}, \tag{4.4.1a}$$

for the weak form, and

$$E_{L^2}(\widetilde{y}_q) = \frac{(\widetilde{\mathbf{y}} - y_{ex}(\widetilde{\mathbf{x}}))^T \widetilde{\mathbf{W}} (\widetilde{\mathbf{y}} - y_{ex}(\widetilde{\mathbf{x}}))}{\max_j |y(\widetilde{\mathbf{x}}_j)|}, \tag{4.4.1b}$$

for the strong form.

Both discretizations perform well and achieve errors on the order of machine precision. As from the 1D error estimate for the multidomain weak form discretization, see Lemma 4.3.3, the numerical results indicate that the order of accuracy of the solution increases with q for smooth solutions. Additionally, the error from the solution corresponding to the weak form is less than the error for the solution corresponding to the strong form. This behavior is expected as the composite polynomial representation of the solution for the weak form $y_q(x)$ can represent higher order composite polynomials exactly than the composite polynomial representation of the solution for the strong form $\widetilde{y}_q(x)$ due to the presence of the corner nodes in the weak form.

Chapter 5

The Optimal Control Problem

The overall goal of this thesis is to accelerate the solution of PDE constrained optimization problems by exploiting the efficiency of the HPS method. The strong form discretization as presented in Section 4.3.4 underlies the standard HPS method. In this Chapter, the convergence behavior of the strong form discretization is examined in both the optimize-then-discretize and the discretize-then-optimize approaches. In particular for the discretize-then-optimize approach it is observed that strong form discretization does not provide high order accurate convergence. For this case several modified discretizations are examined and ultimately it is concluded that in the discretize-then-optimize approach the weak form discretization (as presented in Section 4.3.3) should be used.

First in Section 5.1 the infinite dimensional optimal control problem is presented. Then, the performance of the discretizations in the optimization context are investigated. In Section 5.2, I derive the optimality system for the model problem under the optimize-then-discretize approach. Then the corresponding optimality system is solved for a test problem to observe the behavior of the both the weak and strong form discretizations under the optimize-then-discretize approach. In Section 5.3, I derive the optimality system for the model problem under the discretize-then-optimize approach. Both the weak and strong form discretizations are considered as well as

several modifications to the strong form discretization. The optimality system corresponding to a test problem is solved for each discretization to examine the behavior under the discretize-then-optimize approach. Finally, I given an error estimate for the weak form discretization in the context of optimization in Section 5.4.

5.1 The Infinite Dimensional Problem

Let $\Omega = (0,1)^2$. For any $u \in L^2(\Omega)$ the state equation (1.1.1b) has a unique solution $y(\,\cdot\,, u) \in H^1(\Omega)$. Therefore the model problem can be written as

$$\underset{u}{\text{Minimize}}\ J(u) \stackrel{\text{def}}{=} \frac{1}{2}\int_\Omega \left(y(x;u) - z(x)\right)^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx \tag{5.1.1}$$

where $y(\,\cdot\,;u) \in H^1(\Omega)$ is the solution of (1.1.1b) for a given control u.

The analytical treatement of the optimal control problem (5.1.1) is based on the weak formulation of the state equation. See, e.g., Hinze et al. [11, Ch. 1], Lions [12], or Tröelsch [24, Ch. 2]. Consider the weak formulation of the state equation: find y such that $y = g$ on $\partial\Omega$ and

$$\int_\Omega \nabla y(x)\nabla\phi(x)dx = \int_\Omega (u(x) + f(x))\phi(x)dx, \quad \forall\, \phi \in \mathcal{V}. \tag{5.1.2}$$

Let $y_d \in H^1(\Omega)$ be a function that satisfies the inhomogeneous Dirichlet boundary conditions $y = g$ on $\partial\Omega$, and define

$$\mathcal{V} = \left\{v \in H^1(\Omega)\ :\ v(x) = 0 \text{ on } \partial\Omega\right\}.$$

Define $y_0 = y - y_d$ so that $y_0 \in \mathcal{V}$. The state space is $\mathcal{Y} = y_d + \mathcal{V}$ and the control space is $\mathcal{U} = L^2(\Omega)$.

The Lagrange functional

$$\mathcal{L}\ :\ \mathcal{V}\times\mathcal{U}\times\mathcal{V} \to \mathbb{R} \tag{5.1.3a}$$

associated with (1.1.1) is given by

$$\begin{aligned}\mathcal{L}(y_0,u,p) = \frac{1}{2}\int_\Omega (y_d(x)+y_0(x)-z(x))^2dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx \\ + \int_\Omega \nabla(y_d(x)+y_0(x))\nabla p(x)dx - \int_\Omega (u(x)+f(x))p(x)dx.\end{aligned} \tag{5.1.3b}$$

The partial Fréchet derivative $\partial_p\mathcal{L}(y_0,u,p)\phi = 0$ for all $\phi \in \mathcal{V}$ gives the weak form of the state equation

$$\int_\Omega \nabla y(x)\nabla\phi(x)dx = \int_\Omega (u(x)+f(x))\phi(x)dx \quad \text{for all } \phi\in\mathcal{V}. \tag{5.1.4a}$$

The partial Fréchet derivative $\partial_{y_0}\mathcal{L}(y_0,u,p)\phi = 0$ for all $\phi \in \mathcal{V}$ gives the weak form of the adjoint equation

$$\int_\Omega \nabla\phi(x)\nabla p(x)dx = -\int_\Omega (y(x)-z(x))\phi(x)dx \quad \text{for all } \phi\in\mathcal{V}. \tag{5.1.4b}$$

Finally, the partial Fréchet derivative $\partial_u\mathcal{L}(y,u,p)\omega = 0$ for all $\omega \in \mathcal{U}$ gives

$$\alpha u(x) - p(x) = 0 \quad \text{almost everywhere in } \Omega. \tag{5.1.4c}$$

The gradient of J defined in (5.1.1) can be computed using the adjoint equation approach. Specifically, the gradient of J is

$$\nabla J(u) = \alpha u - p, \tag{5.1.5}$$

where $p \in H^1(\Omega)$ is the solution of the adjoint equation (5.1.4b) with $y = y(\,\cdot\,,u)$ the solution of the state equation (5.1.4a).

Since (5.1.1) is a strictly convex quadratic problem, the condition $\nabla J(u) = \alpha u - p = 0$ almost everywhere on Ω is a necessary and sufficient condition for u to be the solution of (5.1.1). Using the definition (5.1.5) of the gradient, finding u that solves $\nabla J(u) = 0$ is equivalent to finding $y \in H^1(\Omega)$, $u \in L^2(\Omega)$, and $p \in H^1(\Omega)$, such that the coupled system (5.1.4) is satisfied. Under additional regularity assumptions, the

weak solution of (5.1.4) is also the strong solution of

$$-\Delta p(x) = -(y(x) - z(x)), \qquad x \in \Omega, \tag{5.1.6a}$$

$$p(x) = 0, \qquad x \in \partial\Omega, \tag{5.1.6b}$$

$$-p(x) + \alpha u(x) = 0, \tag{5.1.6c}$$

$$-\Delta y(x) = u(x) + f(x), \qquad x \in \Omega, \tag{5.1.6d}$$

$$y(x) = g(x), \qquad x \in \partial\Omega. \tag{5.1.6e}$$

Note that the optimality conditions (5.1.6c) and (5.1.6b) imply that the optimal control satisfies $u(x) = 0$ for $x \in \partial\Omega$.

Since the optimality conditions (5.1.6) are necessary and sufficient for the solution of the optimal control problem (1.1.1), there are two main directions to pursue for discretization of the problem, the optimize-then-discretize approach, and discretize-then-optimize approach. First the optimize-then-discretize approach is presented for the four leaf problem in Section 5.2. Under this approach, the continuous optimality system is derived and then each of the differential equations is discretized to obtain a finite dimensional system. Then the discretize-then-optimize approach for the four leaf problem is presented in Section 5.3. In this approach the optimal control problem is discretized directly, which leads to a finite dimensional quadratic optimization problem.

5.2 Optimize-then-Discretize Approach

Solving the optimal control problem (5.1.1) is equivalent to solving the necessary and sufficient optimality conditions (5.1.6). In the optimize-then-discretize approach, the optimality conditions (5.1.6) are discretized and the discretized optimality conditions are then solved. Both the discretization based on the weak form and the discretization based on the strong form of the state equation (5.1.6d)–(5.1.6e) and the adjoint equation (5.1.6a)–(5.1.6b) can be used. Again for simplicity, I consider the four leaf

problem as discussed in Section 4.3.1.

5.2.1 Weak Form Discretization of the Model Problem

The weak form discretization of the state equation (5.1.6d)–(5.1.6e) is given by

$$\mathbf{A}\mathbf{y} = -\mathbf{B}\mathbf{u} + \mathbf{c}, \tag{5.2.1}$$

for details refer to Section 4.3.3. Similarly, the discretization of the adjoint equation (5.1.6a)–(5.1.6b) is given by

$$\mathbf{A}\mathbf{p} = \mathbf{B}\mathbf{y} + \mathbf{d}, \tag{5.2.2}$$

and finally, the gradient condition (5.1.6c) is enforced at each collocation point to obtain the discretized condition

$$\alpha\mathbf{u} - \mathbf{p} = \mathbf{0}. \tag{5.2.3}$$

Collecting equations (5.2.1)–(5.2.3) yields the linear system

$$\begin{bmatrix} -\mathbf{B} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \alpha\mathbf{I} & -\mathbf{I} \\ \mathbf{A} & \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \\ \mathbf{c} \end{bmatrix}. \tag{5.2.4}$$

The linear system (5.2.4) is invertible (for details see Corollary 5.3.2, Remark 5.3.3). Computing the solution to the linear system (5.2.4) provides the composite polynomial approximation coefficients for the for the state, control, and adjoint variables corresponding to the weak form discretization.

5.2.2 Strong Form Discretization of the Model Problem

The strong form discretization of the state equation (5.1.6d)–(5.1.6e) is given by

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{y}} = -\widetilde{\mathbf{B}}\widetilde{\mathbf{u}} + \widetilde{\mathbf{c}}, \tag{5.2.5}$$

for details refer to Section 4.3.4. Similarly, the discretization of the adjoint equation (5.1.6a)–(5.1.6b) is given by

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{p}} = \widetilde{\mathbf{B}}\widetilde{\mathbf{y}} + \widetilde{\mathbf{d}}, \tag{5.2.6}$$

and finally the gradient condition (5.1.6c) is enforced at each collocation point by

$$\alpha\widetilde{\mathbf{u}} - \widetilde{\mathbf{p}} = \mathbf{0}. \tag{5.2.7}$$

Collecting equations (5.2.5)–(5.2.7) yields the linear system

$$\begin{bmatrix} -\widetilde{\mathbf{B}} & \mathbf{0} & \widetilde{\mathbf{A}} \\ \mathbf{0} & \alpha\widetilde{\mathbf{I}} & -\widetilde{\mathbf{I}} \\ \widetilde{\mathbf{A}} & \widetilde{\mathbf{B}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{y}} \\ \widetilde{\mathbf{u}} \\ \widetilde{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{d}} \\ \mathbf{0} \\ \widetilde{\mathbf{c}} \end{bmatrix}. \tag{5.2.8}$$

Computing the solution to the linear system (5.2.8) provides the composite polynomial approximation coefficients for the for the state, control, and adjoint variables corresponding to the strong form discretization.

5.2.3 Numerical Experiment

Consider the optimal control problem

$$\underset{u}{\text{Minimize}}\ J(u) \stackrel{\text{def}}{=} \frac{1}{2}\int_{\Omega} \left(y(x;u) - z(x)\right)^2 dx + \frac{\alpha}{2}\int_{\Omega} u^2(x)dx$$

where $\Omega = (0,1)^2$, $\alpha = 0.1$, and $y(\,\cdot\,;u) \in H_0^1(\Omega)$ satisfies

$$\begin{aligned} -\Delta y(x) &= u(x) + f(x), && x \in \Omega, \\ y(x) &= 0, && x \in \partial\Omega, \end{aligned}$$

for a given control u.

To construct an example with a known exact solution, choose exact state and adjoint functions y_{ex} and p_{ex} that satisfy the appropriate boundary conditions from the infinite dimensional optimality conditions (5.1.6).

Let

$$y_{ex}(x) = \sin(\pi x_1)\sin(3\pi x_2),$$
$$p_{ex}(x) = \sin(3\pi x_1)\sin(\pi x_2).$$

Then the optimality condition (5.1.6c) gives

$$u_{ex}(x) = \frac{1}{\alpha} p_{ex}(x).$$

Finally, the functions z and f are computed from the optimality conditions (5.1.6a) and (5.1.6d) respectively

$$z(x) = -\Delta p_{ex}(x) + y_{ex}(x),$$
$$f(x) = -\Delta y_{ex}(x) - u_{ex}(x).$$

To evaluate the performance of the weak and strong four leaf discretizations for the optimize-then-discretize approach to solving the optimal control problem, the optimality systems corresponding to the weak form of the state and adjoint equations (5.2.4) and the strong form of the state and adjoint equations (5.2.8) were constructed and solved over a range of q values. Figure 5.1 provides the convergence behavior by comparing the relative error in the L^2-norm of the state, control, and adjoint for the weak form and the strong form discretizations. Note the relative errors for the weak and strong formulations are defined as in (4.4.1a) and (4.4.1b) respectively.

Each discretization exhibits high order accurate convergence to the exact solution and achieves relative errors on the order of machine precision and they both appear to converge at a similar rate. As anticipated, the solution from the weak form discretization admits smaller relative errors than the solution from the strong form discretization for the same value of q since the weak form is able to represent higher order polynomials exactly than the strong form due to the presence of the corner points. This is consistent with the convergence results for solving the state equation by each discretization (see Section 4.4).

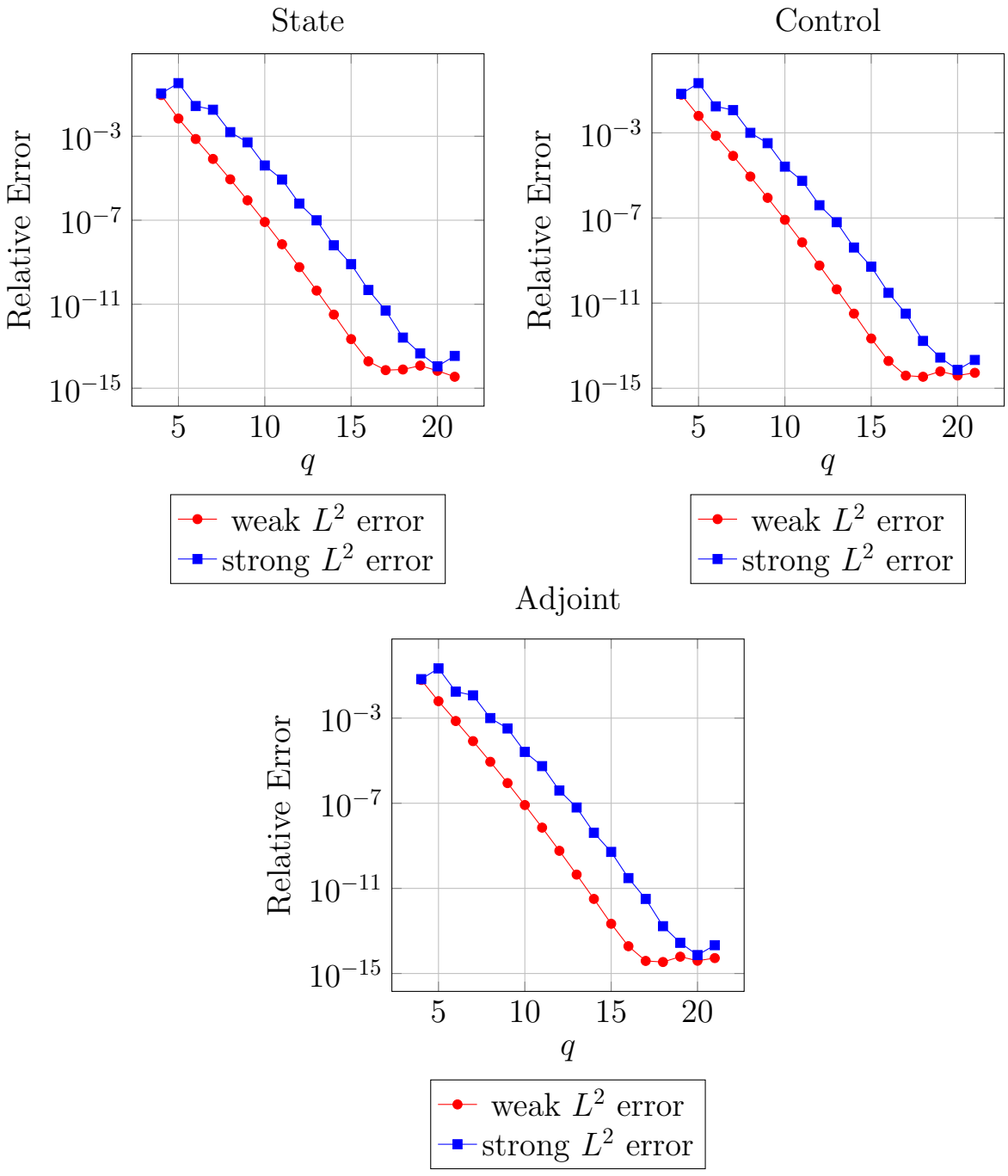

Figure 5.1: The relative L^2 errors vs. q for the state, control, and adjoint for the weak and strong four leaf formulations applied to the test problem for the optimize-then-discretize approach. The state, control, and adjoint errors converge at a similar rate for both formulations.

The strong form discretization underlies the standard HPS method. Since the strong form discretization exibits high order convergence behavior with the optimize-then-discretize approach, the standard HPS method can be used under this approach without modification. Note that the weak form discretization can also be used with the HPS method but this requires some modification (refer to Section 6.1).

5.3 Discretize-then-Optimize Approach

Under the discretize-then-optimize approach, the equality constraint (state equation) and the objective function in (5.1.1) are each discretized to obtain a finite dimensional optimal control problem which then can be solved numerically.

I start with the discretization based on the weak form in Section 5.3.1. In particular, will show that of the discretization based on the weak form is used, the optimize-then-discretize approach and the discretize-then-optimize approach leads to the same system. Then I examine the behavior of the strong form discretization presented. This is the underlying discretization in the standard HPS method and the goal is to determine what modifications (if any) need to be made to the this underlying discretization in order to accelerate the solution of optimal control problems.

5.3.1 Weak Form Discretization of the Model Problem

For the weak form discretization of the state equation (refer to Section 4.3.3), the objective function is discretized by applying the LGL quadrature rule on each subdomain. That is

$$\mathbf{J}(\mathbf{y},\mathbf{u}) = \frac{1}{2}\sum_{\mu\in J}\mathbf{w}_\mu(y(\mathbf{x}_\mu)-z(\mathbf{x}_\mu))^2 + \frac{\alpha}{2}\sum_{\mu\in J}\mathbf{w}_\mu u(\mathbf{x}_\mu)^2. \tag{5.3.1}$$

For convenience, define

$$\mathbf{W} = diag(\mathbf{w}_1,\ldots,\mathbf{w}_{|J|}),$$
$$\mathbf{z} = (z(\mathbf{x}_1),\ldots,z(\mathbf{x}_{|J|}))^T,$$

so that the (5.3.1) may be written as

$$\mathbf{J}(\mathbf{y},\mathbf{u}) = \frac{1}{2}(\mathbf{y}-\mathbf{z})^T\mathbf{W}(\mathbf{y}-\mathbf{z}) + \frac{\alpha}{2}\mathbf{u}^T\mathbf{W}\mathbf{u}.$$

Thus the finite dimensional optimal control problem corresponding to the weak form discretization is given by

$$\underset{\mathbf{u}}{\text{Minimize}}\ \frac{1}{2}(\mathbf{y}-\mathbf{z})^T\mathbf{W}(\mathbf{y}-\mathbf{z}) + \frac{\alpha}{2}\mathbf{u}^T\mathbf{W}\mathbf{u} \tag{5.3.2a}$$

where $\mathbf{y}$ is the solution to

$$\mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} - \mathbf{c} = \mathbf{0}. \tag{5.3.2b}$$

To obtain the optimality conditions for the finite dimensional problem (5.3.2), introduce the vector of Lagrange multipliers $\boldsymbol{\lambda}$ associated with the equality constraint (5.3.2b). Then define the Lagrangian function $\boldsymbol{L}$ for the discretized optimal control problem (5.3.2) by

$$\boldsymbol{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \mathbf{J}(\mathbf{y},\mathbf{u}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} - \mathbf{c}).$$

Then the KKT conditions require that at optimality

$$\nabla_{\mathbf{y}}\boldsymbol{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \mathbf{0}, \tag{5.3.3a}$$

$$\nabla_{\mathbf{u}}\boldsymbol{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \mathbf{0}, \tag{5.3.3b}$$

$$\nabla_{\boldsymbol{\lambda}}\boldsymbol{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \mathbf{0}. \tag{5.3.3c}$$

The KKT condition (5.3.3a) yields the optimality condition

$$\mathbf{W}\mathbf{y} - \mathbf{W}\mathbf{z} + \mathbf{A}^T\boldsymbol{\lambda} = \mathbf{0}, \tag{5.3.4a}$$

(5.3.3b) yields

$$\alpha\mathbf{W}\mathbf{u} + \mathbf{B}^T\boldsymbol{\lambda} = \mathbf{0}, \tag{5.3.4b}$$

and (5.3.3c) simply yields the discretized state equation

$$\mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} - \mathbf{c} = \mathbf{0}. \tag{5.3.4c}$$

Then the optimality system for (5.3.2) is given by collecting equations Equations (5.3.4a)–(5.3.4c) to form the linear system

$$\begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{A}^T \\ \mathbf{0} & \alpha\mathbf{W} & \mathbf{B}^T \\ \mathbf{A} & \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{W}\mathbf{z} \\ \mathbf{0} \\ \mathbf{c} \end{bmatrix}. \tag{5.3.5}$$

It is necessary to relate the Lagrange multipliers $\boldsymbol{\lambda}$ to the adjoint p for developing error estimates for the discretize-then-optimize approach. The following theorem provides the relationship between the Lagrange multipliers and the adjoint.

Theorem 5.3.1 *Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{|J|})^T$ be the vector of Lagrange multipliers corresponding to the optimal solution of (5.3.5). Then the polynomial*

$$p_q(x) = \sum_{j \in J} p_j \psi_j(x) \tag{5.3.6a}$$

where

$$p_j = \begin{cases} \mathbf{w}_j^{-1} \boldsymbol{\lambda}_j, & j \in \cup_{k=1}^4 J_{I(k)} \\ w_{j,2}^{-1} \boldsymbol{\lambda}_j, & j \in J_{M(1,3)} \cup J_{M(2,4)} \\ w_{j,1}^{-1} \boldsymbol{\lambda}_j, & j \in J_{M(1,2)} \cup J_{M(3,4)} \\ \boldsymbol{\lambda}_j, & j \in J_C \\ 0, & j \in J_B \end{cases} \tag{5.3.6b}$$

is the solution to the weak form discretization of the adjoint equation (5.1.4b).

Proof: The first row equation of (5.3.5) is equivalent to

$$\boldsymbol{\lambda}^T \mathbf{A}\mathbf{v} = \mathbf{v}^T \mathbf{A}^T \boldsymbol{\lambda} = -\mathbf{v}^T \mathbf{W}(\mathbf{y} - \mathbf{z}) \quad \forall \mathbf{v} \in \mathbb{R}^{|J|} \tag{5.3.7}$$

Given $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_{|J|})^T \in \mathbb{R}^{|J|}$ such that $\mathbf{v}_j = 0$ for $j \in J_B$, associate it with the polynomial

$$v_q(x) = \sum_{j \in J} \mathbf{v}_j \psi_j(x) \tag{5.3.8}$$

Let $J_M = J_{M(1,3)} \cup J_{M(2,4)} \cup J_{M(1,2)} \cup J_{M(3,4)}$. Then the right hand side of (5.3.7) yields

$$\begin{aligned}
-\mathbf{v}^T\mathbf{W}(\mathbf{y}-\mathbf{z}) &= \sum_{k=1}^{4}\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell^{(k)}\left(y_q(\mathbf{x}_\ell^{(k)}) - z_q(\mathbf{x}_\ell^{(k)})\right) v_q(\mathbf{x}_\ell^{(k)}) \\
&= \sum_{k=1}^{4}\sum_{j\in J_{I(k)}} -\mathbf{w}_j\left(\mathbf{y}_j-\mathbf{z}_j\right)\mathbf{v}_j - \sum_{j\in J_B}\mathbf{w}_j\left(\mathbf{y}_j-\mathbf{z}_j\right)\mathbf{v}_j \\
&\quad - \sum_{j\in J_M} 2\mathbf{w}_j\left(\mathbf{y}_j-\mathbf{z}_j\right)\mathbf{v}_j - \sum_{j\in J_C} 4\mathbf{w}_j\left(\mathbf{y}_j-\mathbf{z}_j\right)\mathbf{v}_j \\
&\approx \int_\Omega (y_q(x)-z_q(x))v_q(x)dx
\end{aligned} \tag{5.3.9}$$

Note that the quadrature is not exact for $(y_q - z_q)v_q$.

In the following let $v_q^{(k)}$, $p_q^{(k)}$ denote the restriction of v_q, p_q to the subdomain Ω^k,

$$v_q^{(k)} = v|_{\overline{\Omega^k}}, \quad p_q^{(k)} = p|_{\overline{\Omega^k}}.$$

Then from the definition of $\mathbf{A}$ as in (4.3.22) and (4.3.23)

$$\begin{aligned}
\boldsymbol{\lambda}^T\mathbf{A}\mathbf{v} = &\sum_{k=1}^{4}\sum_{j\in J_{I(k)}} -\boldsymbol{\lambda}_j\Delta v_q^{(k)}(\mathbf{x}_j)\\
&+\sum_{j\in J_{M(1,3)}}\boldsymbol{\lambda}_j\left(-w_{j,1}\Delta v_q^{(1)}(\mathbf{x}_j)+\frac{\partial}{\partial x_1}v_q^{(1)}(\mathbf{x}_j)-w_{j,1}\Delta v_q^{(3)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}v_q^{(3)}(\mathbf{x}_j)\right)\\
&+\sum_{j\in J_{M(2,4)}}\boldsymbol{\lambda}_j\left(-w_{j,1}\Delta v_q^{(2)}(\mathbf{x}_j)+\frac{\partial}{\partial x_1}v_q^{(2)}(\mathbf{x}_j)-w_{j,1}\Delta v_q^{(4)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}v_q^{(4)}(\mathbf{x}_j)\right)\\
&+\sum_{j\in J_{M(1,2)}}\boldsymbol{\lambda}_j\left(-w_{j,2}\Delta v_q^{(1)}(\mathbf{x}_j)+\frac{\partial}{\partial x_2}v_q^{(1)}(\mathbf{x}_j)-w_{j,2}\Delta v_q^{(2)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}v_q^{(2)}(\mathbf{x}_j)\right)\\
&+\sum_{j\in J_{M(3,4)}}\boldsymbol{\lambda}_j\left(-w_{j,2}\Delta v_q^{(3)}(\mathbf{x}_j)+\frac{\partial}{\partial x_2}v_q^{(3)}(\mathbf{x}_j)-w_{j,2}\Delta v_q^{(4)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}v_q^{(4)}(\mathbf{x}_j)\right)\\
&+\sum_{j\in J_C}\boldsymbol{\lambda}_j\Bigg[w_{j,2}\left(\frac{\partial}{\partial x_1}v_q^{(1)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}v_q^{(3)}(\mathbf{x}_j)+\frac{\partial}{\partial x_1}v_q^{(2)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}v_q^{(4)}(\mathbf{x}_j)\right)\\
&\quad+w_{j,1}\left(\frac{\partial}{\partial x_2}v_q^{(1)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}v_q^{(2)}(\mathbf{x}_j)+\frac{\partial}{\partial x_2}v_q^{(3)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}v_q^{(4)}(\mathbf{x}_j)\right)\\
&\quad-\mathbf{w}_j\left(\Delta v_q^{(1)}(\mathbf{x}_j)+\Delta v_q^{(2)}(\mathbf{x}_j)+\Delta v_q^{(3)}(\mathbf{x}_j)+\Delta v_q^{(4)}(\mathbf{x}_j)\right)\Bigg]
\end{aligned} \tag{5.3.10}$$

Writing $\boldsymbol{\lambda}$ in terms of $\mathbf{p}$ via (5.3.6b) and grouping like terms yields

$$\begin{aligned}
\boldsymbol{\lambda}^T\mathbf{A}\mathbf{v} = &\sum_{k=1}^{4}\sum_{\ell=1}^{q^2}-\mathbf{p}_\ell^{(k)}\mathbf{w}_\ell^{(k)}\Delta v_q^{(k)}(\mathbf{x}_\ell^{(k)})\\
&+\sum_{j\in J_{E(1,3)}}\mathbf{p}_j w_{j,2}\left(\frac{\partial}{\partial x_1}v_q^{(1)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}v_q^{(3)}(\mathbf{x}_j)\right)\\
&+\sum_{j\in J_{E(2,4)}}\mathbf{p}_j w_{j,2}\left(\frac{\partial}{\partial x_1}v_q^{(2)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}v_q^{(4)}(\mathbf{x}_j)\right)\\
&+\sum_{j\in J_{E(1,2)}}\mathbf{p}_j w_{j,1}\left(\frac{\partial}{\partial x_2}v_q^{(1)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}v_q^{(2)}(\mathbf{x}_j)\right)\\
&+\sum_{j\in J_{E(3,4)}}\mathbf{p}_j w_{j,1}\left(\frac{\partial}{\partial x_2}v_q^{(3)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}v_q^{(4)}(\mathbf{x}_j)\right)
\end{aligned} \tag{5.3.11}$$

Since p_q and v_q are polynomials of sufficiently small degree, the quadrature formulas

are exact. Writing (5.3.11) as integrals yields

$$\begin{aligned}
\boldsymbol{\lambda}^T \mathbf{A}\mathbf{v} = &\sum_{k=1}^{4} \int_{\Omega^k} -p_q(x)\Delta v_q^{(k)}(x)dx \\
&+ \int_{\Gamma^{1,3}} p_q(x)\left(\frac{\partial}{\partial x_1}v_q^{(1)}(x) - \frac{\partial}{\partial x_1}v_q^{(3)}(x)\right)ds(x) \\
&+ \int_{\Gamma^{2,4}} p_q(x)\left(\frac{\partial}{\partial x_1}v_q^{(2)}(x) - \frac{\partial}{\partial x_1}v_q^{(4)}(x)\right)ds(x) \\
&+ \int_{\Gamma^{1,2}} p_q(x)\left(\frac{\partial}{\partial x_2}v_q^{(1)}(x) - \frac{\partial}{\partial x_2}v_q^{(2)}(x)\right)ds(x) \\
&+ \int_{\Gamma^{3,4}} p_q(x)\left(\frac{\partial}{\partial x_2}v_q^{(3)}(x) - \frac{\partial}{\partial x_2}v_q^{(4)}(x)\right)ds(x) \\
= &\sum_{k=1}^{4} \int_{\Omega^k} -p_q(x)\Delta v_q^{(k)}(x)dx \\
&+ \sum_{k=1}^{4} \int_{\partial\Omega^k \setminus \partial\Omega} p_q(x)\left(\nabla v_q^{(k)}(x)\cdot \overline{n^k}\right)ds(x)
\end{aligned} \tag{5.3.12}$$

where $\overline{n^k}$ is the outward pointing normal direction with respect to Ω^k. Applying the divergence theorem to (5.3.12) once yields

$$\boldsymbol{\lambda}^T \mathbf{A}\mathbf{v} = \sum_{k=1}^{4} \int_{\Omega^k} \nabla p_q(x)\nabla v_q^{(k)}(x)dx,$$

and applying it a second time yields

$$\begin{aligned}\boldsymbol{\lambda}^T\mathbf{A}\mathbf{v} =& \sum_{k=1}^{4}\int_{\Omega^k} -\Delta p_q(x)v_q^{(k)}(x)dx \\ &+ \sum_{k=1}^{4}\int_{\partial\Omega^k\setminus\partial\Omega}\left(\nabla p_q^{(k)}(x)\cdot\overline{n^k}\right)v_q(x)ds(x) \\ =& \sum_{k=1}^{4}\int_{\Omega^k} -\Delta p_q(x)v_q^{(k)}(x)dx \\ &+ \int_{\Gamma^{1,3}}\left(\frac{\partial}{\partial x_1}p_q^{(1)}(x)-\frac{\partial}{\partial x_1}p_q^{(3)}(x)\right)v_q(x)ds(x) \\ &+ \int_{\Gamma^{2,4}}\left(\frac{\partial}{\partial x_1}p_q^{(2)}(x)-\frac{\partial}{\partial x_1}p_q^{(4)}(x)\right)v_q(x)ds(x) \\ &+ \int_{\Gamma^{1,2}}\left(\frac{\partial}{\partial x_2}p_q^{(1)}(x)-\frac{\partial}{\partial x_2}p_q^{(2)}(x)\right)v_q(x)ds(x) \\ &+ \int_{\Gamma^{3,4}}\left(\frac{\partial}{\partial x_2}p_q^{(3)}(x)-\frac{\partial}{\partial x_2}p_q^{(4)}(x)\right)v_q(x)ds(x)\end{aligned} \tag{5.3.13a}$$

(5.3.13b)

Writing each of the integrals in (5.3.13b) by the corresponding quadrature rule yields

$$\begin{aligned}\boldsymbol{\lambda}^T\mathbf{A}\mathbf{v} =& \sum_{k=1}^{4}\sum_{\ell=1}^{q^2} -\mathbf{w}_\ell^{(k)}\Delta p_q^{(k)}(\mathbf{x}_\ell^{(k)})\mathbf{v}_\ell^{(k)} \\ &+ \sum_{j\in J_{E(1,3)}}\mathbf{v}_j w_{j,2}\left(\frac{\partial}{\partial x_1}p_q^{(1)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}p_q^{(3)}(\mathbf{x}_j)\right) \\ &+ \sum_{j\in J_{E(2,4)}}\mathbf{v}_j w_{j,2}\left(\frac{\partial}{\partial x_1}p_q^{(2)}(\mathbf{x}_j)-\frac{\partial}{\partial x_1}p_q^{(4)}(\mathbf{x}_j)\right) \\ &+ \sum_{j\in J_{E(1,2)}}\mathbf{v}_j w_{j,1}\left(\frac{\partial}{\partial x_2}p_q^{(1)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}p_q^{(2)}(\mathbf{x}_j)\right) \\ &+ \sum_{j\in J_{E(3,4)}}\mathbf{v}_j w_{j,1}\left(\frac{\partial}{\partial x_2}p_q^{(3)}(\mathbf{x}_j)-\frac{\partial}{\partial x_2}p_q^{(4)}(\mathbf{x}_j)\right).\end{aligned} \tag{5.3.14}$$

Again because p_q and v_q are polynomials of sufficiently small degree, the quadrature is exact. Comparing (5.3.14) with (5.3.11) and (5.3.10), observe that (5.3.14) is equivalent to $\mathbf{v}^T\mathbf{A}\mathbf{p}$ up to row scaling. More precisely, let $\mathbf{T}\in\mathbb{R}^{|J|\times|J|}$ be a diagonal

matrix with entries

$$\mathbf{T}_{jj} = \begin{cases} \mathbf{w}_j, & j \in \cup_{k=1}^{4} J_I(k) \\ w_{j,2}, & j \in J_{M(1,3)} \cup J_{M(2,4)} \\ w_{j,1}, & j \in J_{M(1,2)} \cup J_{M(3,4)} \\ 1, & j \in J_C \cup J_B \end{cases} \tag{5.3.15}$$

Then

$$\mathbf{v}^T \mathbf{A}^T \boldsymbol{\lambda} = \mathbf{v}^T \mathbf{T} \mathbf{A} \mathbf{p}, \quad \forall \mathbf{v} \in \{\mathbb{R}^{|J|} \mid \mathbf{v}_j = 0 \text{ for } j \in J_B\}. \tag{5.3.16}$$

Similarly, comparing (5.3.9) and the definition of $\mathbf{B}$ in (4.3.24) yields

$$-\mathbf{v}^T \mathbf{W}(\mathbf{y} - \mathbf{z}) = \mathbf{v}^T \mathbf{T}(\mathbf{B}\mathbf{y} + \mathbf{d}), \quad \forall \mathbf{v} \in \{\mathbb{R}^{|J|} \mid \mathbf{v}_j = 0 \text{ for } j \in J_B\}. \tag{5.3.17}$$

The desired result is obtained by combining (5.3.16) and (5.3.17). □

Corollary 5.3.2 *For the weak form discretization applied to the model problem* (1.1.1)*, the optimality systems for the discretize-then-optimize approach* (5.3.5) *and the optimize-then-discretize approach* (5.2.4) *are equivalent.*

Proof: Theorem 5.3.1 shows that

$$\mathbf{W}\mathbf{y} + \mathbf{A}^T \boldsymbol{\lambda} = \mathbf{W}\mathbf{z} \quad \Longleftrightarrow \quad -\mathbf{B}\mathbf{y} + \mathbf{A}\mathbf{p} = \mathbf{d}$$

It remains to show that

$$\alpha \mathbf{W}\mathbf{u} + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{0} \quad \Longleftrightarrow \quad \alpha \mathbf{u} - \mathbf{p} = \mathbf{0} \tag{5.3.18}$$

Comparing the definitions of $\mathbf{p}$ in (5.3.6b), $\mathbf{T}$ in (5.3.15) and $\mathbf{B}$ in (4.3.24), observe that

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{T}\mathbf{B}\mathbf{p}$$

Thus

$$\mathbf{W}^{-1}(\alpha\mathbf{W}\mathbf{u}+\mathbf{B}\boldsymbol{\lambda}) = \mathbf{W}^{-1}(\alpha\mathbf{W}\mathbf{u}+\mathbf{T}\mathbf{B}\mathbf{p}) = \alpha\mathbf{u}+\mathbf{W}^{-1}\mathbf{T}\mathbf{B}\mathbf{p} = 0 \tag{5.3.19}$$

Furthermore, by comparison with (5.3.9) the diagonal matrix

$$(\mathbf{W}^{-1}\mathbf{T}\mathbf{B})_{jj} = \begin{cases} -1, & j \notin J_B \\ 0, & j \in J_B \end{cases} \tag{5.3.20}$$

That

$$\begin{aligned} \alpha\mathbf{u}_j - \mathbf{p}_j &= 0, && \text{for } j \notin J_B \\ \alpha\mathbf{u}_j &= 0, && \text{for } j \in J_B \end{aligned}$$

Finally, taking into account the boundary condition $\mathbf{p}_j = 0$ for $j \in J_B$ (see (5.1.6b)) shows that the two linear systems are equivalent. □

Remark 5.3.3 *Solvability of* (5.3.5) *follows from invertability of* $\mathbf{A}$ *and the positive definiteness of* $\mathbf{W}$. *By Corollary 5.3.2 the linear system* (5.2.4) *is equivalent to* (5.3.5) *and thus is invertible.*

5.3.2 Strong Form Discretization of the Model Problem

Now I investigate the performance of the strong form discretization to see what (if any) modifications need to be made to the standard HPS method for use under the discretize-then-optimize approach.

The state equation is discretized by (4.3.28) and the quadrature rule (4.2.27) corresponding to the strong discretization collocation points is applied on each subdomain to discretize the objective function Define

$$\begin{aligned} \widetilde{\mathbf{W}} &= diag(\widetilde{\mathbf{w}}_1, \ldots, \widetilde{\mathbf{w}}_{|\widetilde{J}|}), \\ \widetilde{\mathbf{z}} &= (z(\widetilde{\mathbf{x}}_1), \ldots, z(\widetilde{\mathbf{x}}_{|\widetilde{J}|}))^T, \end{aligned}$$

so that the finite dimensional optimal control problem corresponding to the strong form discretization is given by

$$\underset{\widetilde{\mathbf{u}}}{\text{Minimize}}\ \frac{1}{2}(\widetilde{\mathbf{y}}-\widetilde{\mathbf{z}})^T\widetilde{\mathbf{W}}(\widetilde{\mathbf{y}}-\widetilde{\mathbf{z}})+\frac{\alpha}{2}\widetilde{\mathbf{u}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{u}} \stackrel{\text{def}}{=} \widetilde{\mathbf{J}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}}) \tag{5.3.21a}$$

where $\widetilde{\mathbf{y}}$ is the solution to

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{y}}+\widetilde{\mathbf{B}}\widetilde{\mathbf{u}}-\widetilde{\mathbf{c}}=\mathbf{0}. \tag{5.3.21b}$$

Recall from the discretization of the state equation in Section 4.3.4 that in contrast to the weak form discretization in (5.3.2), the control along the shared subdomain boundaries does not enter the strong form discretization of the state equation as given by (5.3.21b) (see Remark 4.3.6).

Introduce the vector of Lagrange multipliers $\widetilde{\boldsymbol{\lambda}}$ associated with the equality constraint (5.3.21b), and define the Lagrangian function $\widetilde{\boldsymbol{L}}$ for the discretized optimal control problem (5.3.21) by

$$\widetilde{\boldsymbol{L}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}},\widetilde{\boldsymbol{\lambda}})=\widetilde{\mathbf{J}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}})+\widetilde{\boldsymbol{\lambda}}^T(\widetilde{\mathbf{A}}\widetilde{\mathbf{y}}+\widetilde{\mathbf{B}}\widetilde{\mathbf{u}}-\widetilde{\mathbf{c}}).$$

At optimality the KKT conditions require that

$$\nabla_{\mathbf{y}}\widetilde{\boldsymbol{L}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}},\widetilde{\boldsymbol{\lambda}})=\mathbf{0}, \tag{5.3.22a}$$

$$\nabla_{\mathbf{u}}\widetilde{\boldsymbol{L}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}},\widetilde{\boldsymbol{\lambda}})=\mathbf{0}, \tag{5.3.22b}$$

$$\nabla_{\boldsymbol{\lambda}}\widetilde{\boldsymbol{L}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}},\widetilde{\boldsymbol{\lambda}})=\mathbf{0}. \tag{5.3.22c}$$

The KKT condition (5.3.22a) yields the optimality condition

$$\widetilde{\mathbf{W}}\widetilde{\mathbf{y}}-\widetilde{\mathbf{W}}\widetilde{\mathbf{z}}+\widetilde{\mathbf{A}}^T\widetilde{\boldsymbol{\lambda}}=\mathbf{0}, \tag{5.3.23a}$$

(5.3.22b) yields

$$\alpha\widetilde{\mathbf{W}}\widetilde{\mathbf{u}}+\widetilde{\mathbf{B}}^T\widetilde{\boldsymbol{\lambda}}=\mathbf{0}. \tag{5.3.23b}$$

and (5.3.22c) simply yields the discretized state equation

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{y}}+\widetilde{\mathbf{B}}\widetilde{\mathbf{u}}-\widetilde{\mathbf{c}}=\mathbf{0}. \tag{5.3.23c}$$

Then the optimality system for (5.3.21) is given by collecting equations Equations (5.3.23a)–(5.3.23c) to form the linear system

$$\begin{bmatrix} \widetilde{\mathbf{W}} & \mathbf{0} & \widetilde{\mathbf{A}}^T \\ \mathbf{0} & \alpha\widetilde{\mathbf{W}} & \widetilde{\mathbf{B}}^T \\ \widetilde{\mathbf{A}} & \widetilde{\mathbf{B}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{y}} \\ \widetilde{\mathbf{u}} \\ \widetilde{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{W}}\widetilde{\mathbf{z}} \\ \mathbf{0} \\ \widetilde{\mathbf{c}} \end{bmatrix}. \tag{5.3.24}$$

Again consider the test problem from Section 5.2.3 with exact solution

$$\begin{aligned} y_{ex}(x) &= \sin(\pi x_1)\sin(3\pi x_2), \\ u_{ex}(x) &= 10\sin(3\pi x_1)\sin(\pi x_2), \\ p_{ex}(x) &= \sin(3\pi x_1)\sin(\pi x_2). \end{aligned}$$

To evaluate the performance of the strong four leaf discretizations for the discretize-then-optimize approach to solving the optimal control problem, the optimality system (5.3.24) was constructed and solved over a range of q values. Figure 5.2 provides the convergence behavior of the relative error in the L^2-norm of the state and control for the strong four leaf formulation applied to the test problem. Note the relative errors are defined as in (4.4.1b).

The strong form discretization does not converge to the exact solution for either the state or the control. To determine what is inhibiting the rapid convergence behavior expected of the strong form discretization, I examine the discretized optimal control problem.

Solving (5.3.21) is equivalent to choosing the vector $\widetilde{\mathbf{u}}$ that minimizes the objective function from the set vectors that satisfy the linear constraint ($\widetilde{\mathbf{A}}\widetilde{\mathbf{y}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{u}} - \widetilde{\mathbf{c}} = \mathbf{0}$). It has already been shown that the exact solution evaluated at the collocation points satisfies the linear constraint (see Section 4.4). This suggests that the objective function is penalizing the exact control $\widetilde{\mathbf{u}}_{ex}$ more than some other vector that also satisfies the linear constraint. Rewriting the objective function as a sum of the individual

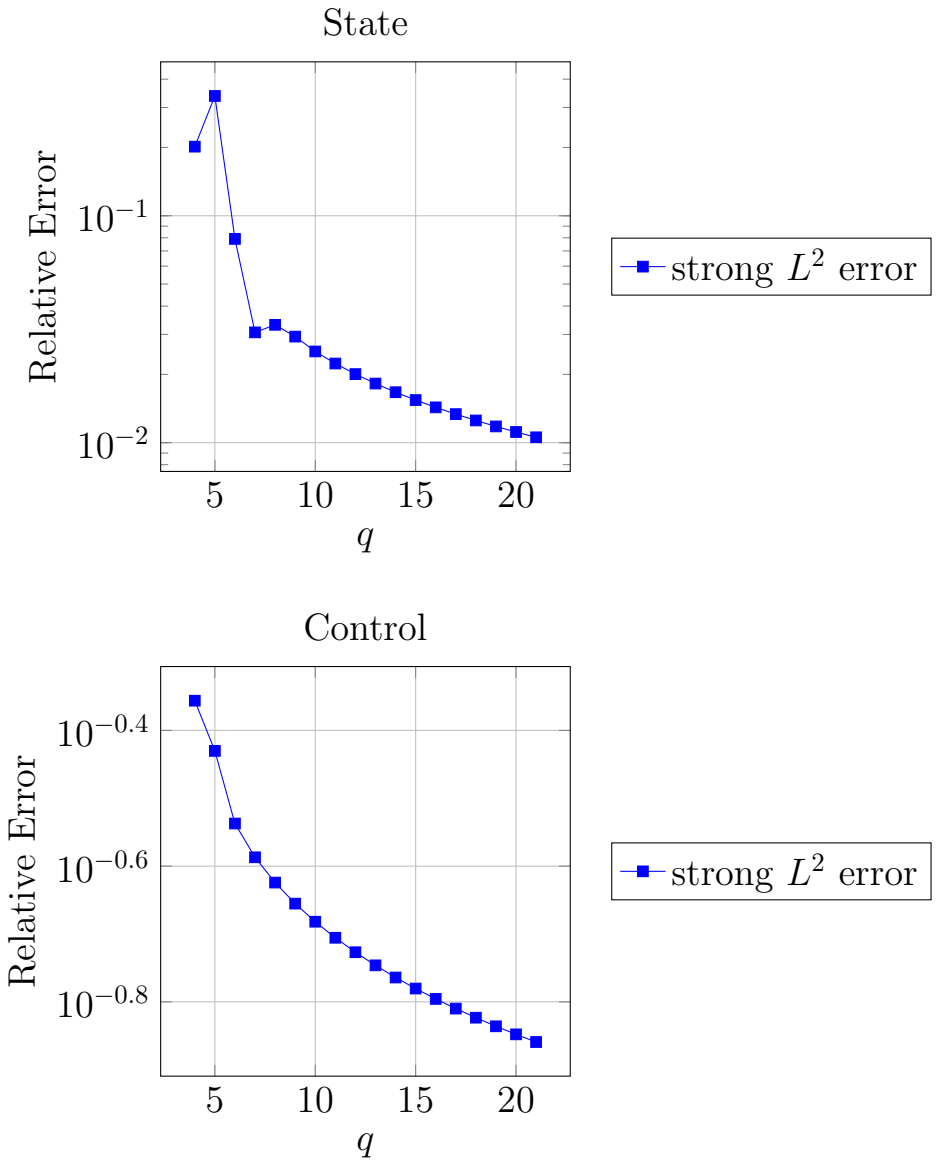

Figure 5.2: The relative L^2 error vs. q for the state and control for the strong form four leaf discretization applied to the test problem under the discretize-then-optimize approach. Both the state and control errors exhibit very poor convergence compared to the optimize-then-discretize approach (compare to Figure 5.1) and do not achieve errors on the order of machine precision.

components of the vectors $\widetilde{\mathbf{y}}, \widetilde{\mathbf{d}}$, and $\widetilde{\mathbf{u}}$ yields

$$\widetilde{\mathbf{J}}(\widetilde{\mathbf{y}}, \widetilde{\mathbf{u}}) = \sum_{\mu \in |\widetilde{J}|} \frac{1}{2}\mathbf{w}_\mu \widetilde{\mathbf{y}}_\mu^2 - \mathbf{w}_\mu \widetilde{\mathbf{y}}_\mu \widetilde{\mathbf{z}}_\mu + \frac{\alpha}{2}\mathbf{w}_\mu \widetilde{\mathbf{u}}_\mu^2. \tag{5.3.25}$$

Temporarily consider minimizing the objective function (5.3.25) without any constraints relating the state $\widetilde{\mathbf{y}}$ and the control $\widetilde{\mathbf{u}}$. Then the control $\widetilde{\mathbf{u}}$ must be equal to zero or the objective function has not been minimized. Now again consider the discretized optimal control problem with the equality constraint given by the strong form discretization of the state equation (5.3.21b). As observed in Remark 4.3.6, the control along the subdomain interfaces does not enter the discretization. In other words, the control values at collocation points on the subdomain interfaces are unconstrained and must be set equal to zero in order to minimize the objective function (5.3.25).

Specifically, let $\mu \in \widetilde{J}_{M(k,\ell)}$. Then in the strong form discretization of the state equation the value of the control at the μ-th collocation point is unconstrained (i.e. $\widetilde{\mathbf{B}}\widetilde{\mathbf{e}}_\mu = \mathbf{0}$).

Examining the μ-th row equation from the gradient condition in the optimality system yields

$$\begin{aligned} \alpha \mathbf{e}_\mu^T \widetilde{\mathbf{W}} \widetilde{\mathbf{u}} + \mathbf{e}_\mu^T \widetilde{\mathbf{B}}^T \widetilde{\boldsymbol{\lambda}} &= 0, \\ \alpha \widetilde{\mathbf{w}}_\mu \widetilde{\mathbf{u}}_\mu + \mathbf{0}^T \widetilde{\boldsymbol{\lambda}} &= 0, \\ \widetilde{\mathbf{u}}_\mu &= 0. \end{aligned}$$

Since $\widetilde{\mathbf{u}}_\mu$ is unconstrained by the strong form discretization of the state equation, regardless of the value of the exact solution, the value of $\widetilde{\mathbf{u}}_\mu$ must equal zero to minimize the discretized objective function.

Indeed examining the computed control from the test problem, the numerical experiment supports these findings as the computed control solution is equal to zero at each collocation node on a subdomain interface.

5.3.3 Modifications to the Discretization to Improve Convergence Behavior

Modification I. As observed in Section 5.3.2, under the discretize-then-optimize approach, the strong form discretization leading to the linear system (5.3.24) does not converge at a similar rate compared to solving a boundary value problem as in Section 4.4. A natural question is if it is possible to modify the strong form discretization in some way to regain the high order converge observed for solving boundary value problems.

Upon closer examination of (5.3.24) it was observed that the control on the subdomain interfaces is unconstrained by the strong form discretization of the state equation. To correct this, consider adding an addition equality constraint to the discretized optimization problem that requires the control values on the merge interfaces to satisfy the differential equation. This can be thought of as post-processing the state solve in the following sense. From the discretization of the state equation, given the control values at the interior collocation points on each subdomain uniquely determines the state at each collocation point. Once the state is known, the control values on the merge interface can be solved for by requiring that the differential equation is satisfied at the collocation points on the merge interface.

Enforcing the differential equation on the merge interface is given by the following equations.

For $\mu \in \widetilde{J}_{M(k,\ell)}$

$$\frac{1}{2}\widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,k)}\widetilde{\mathbf{L}}^{(k)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,k)} + \frac{1}{2}\widetilde{\mathbf{e}}^T_{\widetilde{\rho}(\mu,\ell)}\widetilde{\mathbf{L}}^{(\ell)}\widetilde{\mathbf{y}}_{\widetilde{\pi}(1:q^2-4,\ell)} = \widetilde{\mathbf{e}}^T_{\mu}\widetilde{\mathbf{u}} + \mathbf{f}_{\mu} \tag{5.3.26}$$

Collecting these equations for each merge interface in the four leaf problem leads to the linear system

$$\mathbf{E}\widetilde{\mathbf{y}} + \mathbf{F}\widetilde{\mathbf{u}} = \mathbf{a}. \tag{5.3.27}$$

Adding this equality constraint to the formulation of the discretized optimization

problem yields

$$\underset{\widetilde{\mathbf{u}}}{\text{Minimize}}\ \frac{1}{2}(\widetilde{\mathbf{y}}-\widetilde{\mathbf{z}})^T\widetilde{\mathbf{W}}(\widetilde{\mathbf{y}}-\widetilde{\mathbf{z}})+\frac{\alpha}{2}\widetilde{\mathbf{u}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{u}} \tag{5.3.28a}$$

$$\text{s.t. } \widetilde{\mathbf{A}}\widetilde{\mathbf{y}}+\widetilde{\mathbf{B}}\widetilde{\mathbf{u}}-\widetilde{\mathbf{c}}=\mathbf{0} \tag{5.3.28b}$$

$$\mathbf{E}\widetilde{\mathbf{y}}+\mathbf{F}\widetilde{\mathbf{u}}-\mathbf{a}=\mathbf{0}. \tag{5.3.28c}$$

Introducing the vectors of Lagrange multipliers $\widetilde{\boldsymbol{\lambda}}$ and $\widetilde{\boldsymbol{\nu}}$ with the equality constraints, form the Lagrangian function

$$\widetilde{\boldsymbol{L}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}},\widetilde{\boldsymbol{\lambda}},\widetilde{\boldsymbol{\nu}})=\widetilde{\mathbf{J}}(\widetilde{\mathbf{y}},\widetilde{\mathbf{u}})+\widetilde{\boldsymbol{\lambda}}^T(\widetilde{\mathbf{A}}\widetilde{\mathbf{y}}+\widetilde{\mathbf{B}}\widetilde{\mathbf{u}}-\widetilde{\mathbf{c}})+\widetilde{\boldsymbol{\nu}}^T(\mathbf{E}\widetilde{\mathbf{y}}+\mathbf{F}\widetilde{\mathbf{u}}-\mathbf{a}). \tag{5.3.29}$$

Then the system of optimality conditions is given by

$$\begin{bmatrix}\widetilde{\mathbf{W}} & \mathbf{0} & \widetilde{\mathbf{A}}^T & \mathbf{E}^T\\ \mathbf{0} & \alpha\widetilde{\mathbf{W}} & \widetilde{\mathbf{B}}^T & \mathbf{F}^T\\ \widetilde{\mathbf{A}} & \widetilde{\mathbf{B}} & \mathbf{0} & \mathbf{0}\\ \mathbf{E} & \mathbf{F} & \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{y}}\\ \widetilde{\mathbf{u}}\\ \widetilde{\boldsymbol{\lambda}}\\ \widetilde{\boldsymbol{\nu}}\end{bmatrix}=\begin{bmatrix}\widetilde{\mathbf{W}}\widetilde{\mathbf{z}}\\ \mathbf{0}\\ \widetilde{\mathbf{c}}\\ \mathbf{a}\end{bmatrix}. \tag{5.3.30}$$

Modification II. A second modification to the strong form discretization under the discretize-then-optimize approach that has potential to improve the convergence behavior is to discretize the control such that it does not have collocation points located on the merge interfaces. Intuitively, this removes the issue of control variables being set equal to zero along the merge interface.

To accomplish this, place a tensor product grid of $(q-2)^2$ Legendre-Gauss (LG) quadrature points on each subdomain. Figure 5.3 compares the Legendre-Gauss quadrature points with the LGL points less corners for the four leaf problem.

Let $\widehat{\psi}_\mu$ and $\widehat{\mathbf{w}}_\mu$ be the 2D Lagrange basis polynomial and 2D quadrature weight corresponding to the collocation point $\widehat{\mathbf{x}}_\mu$. Then the 2D composite polynomial approximation of the function u is given by

$$u(x)\approx\sum_{\mu=1}^{|\widehat{J}|}u(\widehat{\mathbf{x}}_\mu)\widehat{\psi}_\mu(x). \tag{5.3.31}$$

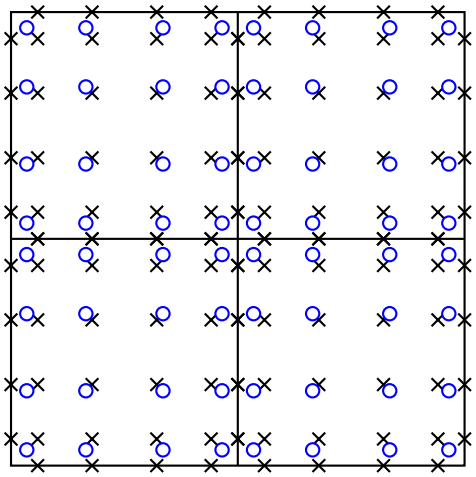

Figure 5.3: Comparison of the LG points (blue circles) and the LGL points less corners (black crosses) for the four leaf problem. Observe that the LG points do not lie on the boundary of any of the subdomains.

As before, define the vector

$$\widehat{\mathbf{u}} = (u(\widehat{\mathbf{x}}_1), \ldots, \widehat{\mathbf{x}}_{(q-2)^2})^T, \tag{5.3.32}$$

and let

$$\widehat{\mathbf{W}} = diag(\widehat{\mathbf{w}}_1, \ldots, \widehat{\mathbf{w}}_{(q-2)^2}) \tag{5.3.33}$$

be the diagonal matrix of Legendre-Gauss quadrature weights. Next define the 2D interpolation matrix $\mathbf{Q}$ such that

$$\widetilde{\mathbf{x}} = \mathbf{Q}\widehat{\mathbf{x}}. \tag{5.3.34}$$

Finally, let

$$\widehat{\mathbf{B}} = \widetilde{\mathbf{B}}\mathbf{Q} \tag{5.3.35}$$

so that the discretization of the state equation is given by

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{y}} + \widehat{\mathbf{B}}\widehat{\mathbf{u}} = \widetilde{\mathbf{c}}. \tag{5.3.36}$$

Then the discretized optimization problem is given by

$$\underset{\widehat{\mathbf{u}}}{\text{Minimize}} \ \frac{1}{2}(\widetilde{\mathbf{y}}-\widetilde{\mathbf{z}})^T\widetilde{\mathbf{W}}(\widetilde{\mathbf{y}}-\widetilde{\mathbf{z}})+\frac{\alpha}{2}\widehat{\mathbf{u}}^T\widehat{\mathbf{W}}\widehat{\mathbf{u}} \tag{5.3.37a}$$

$$\text{s.t. } \widetilde{\mathbf{A}}\widetilde{\mathbf{y}}+\widehat{\mathbf{B}}\widehat{\mathbf{u}}-\widetilde{\mathbf{c}}=\mathbf{0}, \tag{5.3.37b}$$

which corresponds to the optimality system

$$\begin{bmatrix} \widetilde{\mathbf{W}} & \mathbf{0} & \widetilde{\mathbf{A}}^T \\ \mathbf{0} & \alpha\widehat{\mathbf{W}} & \widehat{\mathbf{B}}^T \\ \widetilde{\mathbf{A}} & \widehat{\mathbf{B}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{y}} \\ \widehat{\mathbf{u}} \\ \widetilde{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{W}}\widetilde{\mathbf{z}} \\ \mathbf{0} \\ \widetilde{\mathbf{c}} \end{bmatrix}. \tag{5.3.38}$$

5.3.4 Numerical Experiment

Again consider the test problem from Section 5.2.3. To examine the performance of the weak form discretization, the linear system (5.3.5) was solved for the test problem.

Define the relative errors for the weak form discretization to be

$$E_{L^2}(y_q)=\frac{\mathbf{y}^T\mathbf{W}\mathbf{y}}{\max_j|y(\mathbf{x}_j)|}, \qquad E_{L^2}(u_q)=\frac{\mathbf{u}^T\mathbf{W}\mathbf{u}}{\max_j|u(\mathbf{x}_j)|}$$

where $\mathbf{y}$ and $\mathbf{u}$ are the state and control components of the solution to (5.3.5).

The relative errors for the (unmodified) strong form discretization are given by

$$E_{L^2}(\widetilde{y}_q)=\frac{\widetilde{\mathbf{y}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{y}}}{\max_j|y(\mathbf{x}_j)|}, \qquad E_{L^2}(\widetilde{u}_q)=\frac{\widetilde{\mathbf{u}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{u}}}{\max_j|u(\mathbf{x}_j)|}$$

where $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{u}}$ are the state and control components of the solution to (5.3.24).

The relative errors for the strong form discretization with the additional constraint are given by

$$E_{L^2,\nu}(\widetilde{y}_q)=\frac{\widetilde{\mathbf{y}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{y}}}{\max_j|y(\mathbf{x}_j)|}, \qquad E_{L^2,\nu}(\widetilde{u}_q)=\frac{\widetilde{\mathbf{u}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{u}}}{\max_j|u(\mathbf{x}_j)|}$$

where $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{u}}$ are the state and control components of the solution to (5.3.30).

Finally, the relative errors for the strong form discretization with the control discretized on Legendre-Gauss points are given by

$$E_{L^2,\widehat{u}}(\widetilde{y}_q)=\frac{\widetilde{\mathbf{y}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{y}}}{\max_j|y(\widetilde{\mathbf{x}}_j)|}, \qquad E_{L^2,\widehat{u}}(\widehat{u}_q)=\frac{\widehat{\mathbf{u}}^T\widehat{\mathbf{W}}\widehat{\mathbf{u}}}{\max_j|u(\widehat{\mathbf{x}}_j)|}$$

where $\widetilde{\mathbf{y}}$ and $\widehat{\mathbf{u}}$ are the state and control components of the solution to (5.3.38).

Figure 5.4 compares the relative errors for the various discretizations under the discretize-then-optimize approach applied to the test problem. Each of the proposed modifications to the strong form discretization improves the convergence behavior with the discretize-then-optimize approach relative to the strong form discretization. However, both the state and control errors do not exhibit the high order convergence behavior expected.

In contrast, the weak form discretization under the discretize-then-optimize approach exhibits the expected desirable convergence behavior. This indicates that only the weak form discretization as presented in Section 4.3.3 should be used with the discretize-then-optimize approach. The standard HPS method must be modified to use the weak form discretization in order to accelerate the solution of PDE constrained optimization problems under the discretize-then-optimize approach.

5.4 Error Estimate for the Weak Discretization

In this section I present error analysis for the discretize-then-optimize approach based on the weak form multidomain discretization presented in Section 4.3.3. As noted in Section 5.3.1 the discretize-then-optimize approach is equivalent to the optimize-then-discretize approach when the weak form discretization used. So the error estimate holds for either approach. For simplicity I present results for the 1D case,

$$\Omega = (-1, 1),$$

but the results can be generalized to higher dimensions.

Let $\mathcal{V} = H_0^1(\Omega)$. Let the state space $\mathcal{Y} = \mathcal{V}$ and the control space $\mathcal{U} = L^2(\Omega)$ so that the model optimization problem is given by

$$\underset{u \in L^2(\Omega)}{\text{Minimize}}\ J(u) \stackrel{\text{def}}{=} \frac{1}{2}\int_\Omega \left(y(u;x) - z(x)\right)^2 dx + \frac{\alpha}{2}\int_\Omega u(x)^2 dx \tag{5.4.1a}$$

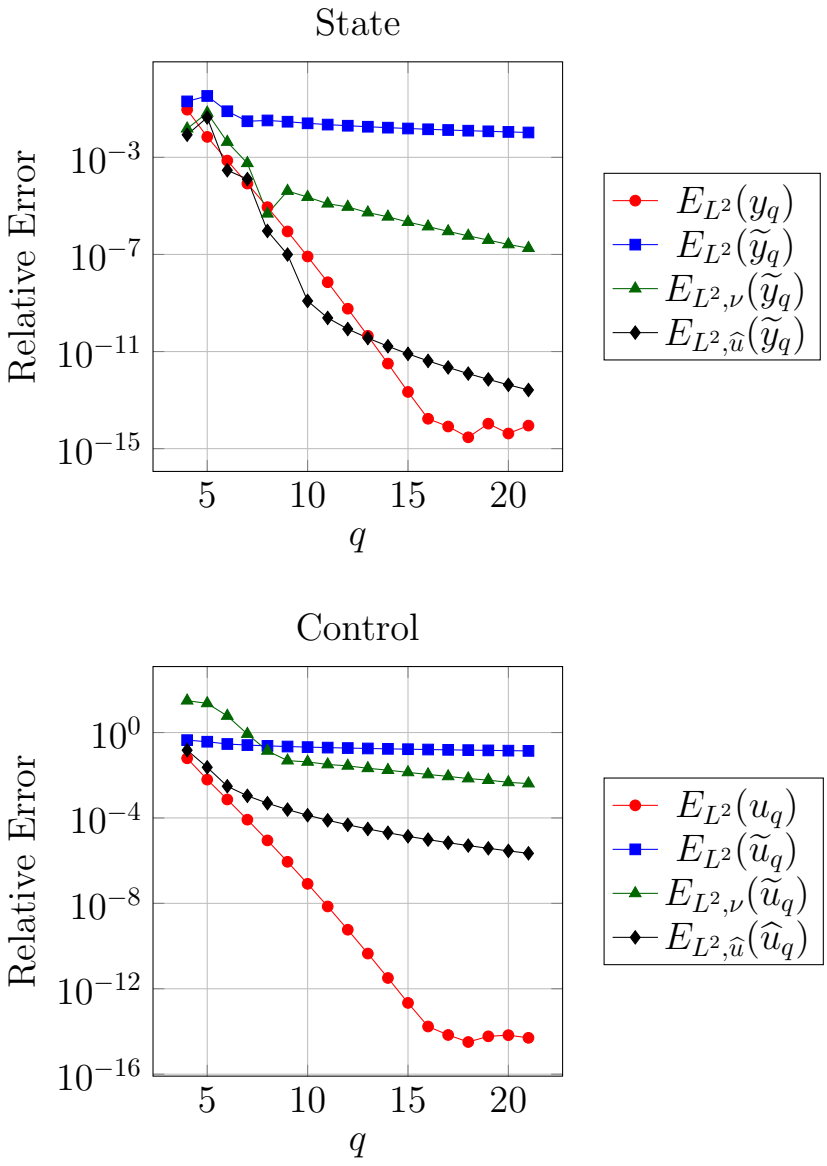

Figure 5.4: The L^2 errors for the state and control for the discretize-then-optimize approach to solving the test problem. Each attempt to restore the convergence for the strong form improves the error, but only the weak form discretization obtains the desired convergence behavior (as seen in the optimize-then-discretize approach). The weak formulation should be used for the discretize-then-optimize approach.

where $y(u;\cdot) \in \mathcal{V}$ solves

$$a(y,v) = b(u+f,v), \qquad \forall v \in \mathcal{V} \tag{5.4.1b}$$

where

$$a(y,u) \stackrel{\text{def}}{=} \int_\Omega \frac{d}{dx}y(x)\frac{d}{dx}v(x)dx, \qquad b(u+f,v) \stackrel{\text{def}}{=} \int_\Omega (u(x)+f(x))v(x)dx.$$

Define the finite dimensional subspaces $\mathcal{V}_q \subset \mathcal{V}$, $\mathcal{U}_q \subset \mathcal{U}$ by

$$\mathcal{V}_q = \mathcal{P}_q^K(-1,1) \cap H_0^1(-1,1), \quad \mathcal{U}_q = \mathcal{P}_q^K(-1,1)$$

as in (4.3.6). The discretized optimization problem is then given by replacing the state space and the control space by the finite dimensional subspace $\mathcal{V}_q$.

$$\underset{u_q \in \mathcal{U}_q}{\text{Minimize}}\ J_q(u_q) \stackrel{\text{def}}{=} \frac{1}{2}\int_\Omega \left(y_q(u_q;x) - z(x)\right)^2 dx + \frac{\alpha}{2}\int_\Omega u_q(x)^2 dx \tag{5.4.2a}$$

where $y_q(u_q;\cdot) \in \mathcal{V}_q$ solves

$$a(y_q,v) = b(u_q+f,v), \qquad \forall v \in \mathcal{V}_q. \tag{5.4.2b}$$

Theorem 5.4.1 *Let u be the solution to the continuous optimization problem (5.4.1) and u_q be the solution to the discretized optimization problem (5.4.2). Furthermore, let $\overline{p}(u_q) \in \mathcal{V}$ be the solution of*

$$a(\overline{p}(u_q),v) = -\int_\Omega (y_q(u_q;x) - z(x))v(x)dx \qquad \forall\, v \in \mathcal{V} \tag{5.4.3}$$

and let $y(u_q) \in \mathcal{V}$ be the solution of the state equation (5.4.1b) with u replaced by u_q. If $y(u_q), \overline{p}(u_q) \in H^m(\Omega)$ with $m > 3/2$. Then

$$\|u - u_q\|_{L^2(\Omega)} \le C(q-1)^{1-m}\big(\|\overline{p}(u_q)\|_{H^m(\Omega)} + \|y(u_q)\|_{H^m(\Omega)}\big).$$

Proof: The gradients of the infinite dimensional and the discretized problem are given by

$$\nabla J(u) = \alpha u + p, \qquad \nabla J_q(u_q) = \alpha u_q + p_q,$$

where p, p_q solve the weak form of the adjoint equations

$$a(v,p) = -\int_\Omega (y(u;x) - z(x))v(x)dx, \qquad \forall\, v \in \mathcal{V},$$
$$a(v_q,p_q) = -\int_\Omega (y_q(u_q;x) - z(x))v_q(x)dx, \qquad \forall\, v_q \in \mathcal{V}_q.$$

At the solutions u and u_q of the infinite dimensional and the discretized problem,

$$\nabla J(u) = \alpha u + p = 0, \qquad \nabla J_q(u_q) = \alpha u_q + p_q = 0.$$

Since the map $u \to J(u)$ is strongly convex with parameter $\alpha > 0$

$$\alpha\|u-w\|^2_{L^2(\Omega)} \le \langle \nabla J(u) - \nabla J(w), u-w\rangle_{L^2(\Omega)} \tag{5.4.4}$$

for all $u, w \in L^2(\Omega)$. Choosing $w = u_q$ and using the optimality conditions $\nabla J(u) = 0$ and $\nabla J_q(u_q) = 0$ gives

$$\begin{aligned}\alpha\|u-u_q\|^2_{L^2(\Omega)} &\le \langle \nabla J(u) - \nabla J(u_q), u-u_q\rangle_{L^2(\Omega)}\\ &\le \langle \nabla J_q(u_q) - \nabla J(u_q), u-u_q\rangle_{L^2(\Omega)}\\ &\le \|\nabla J_q(u_q) - \nabla J(u_q)\|_{L^2(\Omega)}\|u-u_q\|_{L^2(\Omega)}.\end{aligned} \tag{5.4.5}$$

Using the definition of the gradients gives

$$\begin{aligned}&\langle \nabla J_q(u_q) - \nabla J(u_q), w\rangle_{L^2(\Omega)}\\ &= \int_\Omega \big(\alpha u_q(x) + p_q(u_q;x)\big)w(x) - \big(\alpha u_q(x) + p(u_q;x)\big)w(x)dx\\ &\le \|w\|_{L^2(\Omega)}\|p_q(u_q) - p(u_q)\|_{L^2(\Omega)},\end{aligned} \tag{5.4.6}$$

where $p_q(u_q)$ solves

$$a(p_q(u_q),v_q) = -\int_\Omega (y_q(u_q;x) - z(x))v_q(x)dx \qquad \forall\, v_q \in \mathcal{V}_q, \tag{5.4.7}$$

and $p(u_q)$ solves

$$a(p(u_q),v) = -\int_\Omega (y(u_q;x) - z(x))v(x)dx \qquad \forall\, v \in \mathcal{V}.$$

Combining equations (5.4.5) and (5.4.6) gives

$$\|u - u_q\|_{L^2(\Omega)} \leq \alpha^{-1} \|p_q(u_q) - p(u_q)\|_{L^2(\Omega)}.$$

Introduce $\overline{p}(u_q)$ as the solution to (5.4.3). By the triangle inequality,

$$\begin{aligned} \|u - u_q\|_{L^2} &\leq \alpha^{-1} \|p_q(u_q) - p(u_q)\|_{L^2(\Omega)} \\ &\leq \alpha^{-1} \big(\|p_q(u_q) - \overline{p}(u_q)\|_{L^2(\Omega)} + \|\overline{p}(u_q) - p(u_q)\|_{L^2(\Omega)} \big) \\ &\leq \alpha^{-1} \big(\|p_q(u_q) - \overline{p}(u_q)\|_{H^1(\Omega)} + \|\overline{p}(u_q) - p(u_q)\|_{H^1(\Omega)} \big). \end{aligned} \tag{5.4.8}$$

The first term on the right hand side in (5.4.8) is bounded by the discretization error for the adjoint equation (5.4.3) and its discretization (5.4.7). The second term on the right hand side in (5.4.8) is bounded as follows. By definition of $\overline{p}(u_q)$ and $p(u_q)$,

$$\begin{aligned} a(\overline{p}(u_q) - p(u_q), v) &= \int_\Omega (y_q(u_q) - y(u_q)) v dx \\ &\leq \|y_q(u_q) - y(u_q)\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\ &\leq \|y_q(u_q) - y(u_q)\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \qquad \forall v \in \mathcal{V}. \end{aligned}$$

The choice of $v = \overline{p}(u_q) - p(u_q)$ and the ellipticity f the bilinear form a give

$$\begin{aligned} \beta_2 \|\overline{p}(u_q) - p(u_q)\|^2_{H^1(\Omega)} &\leq a(\overline{p}(u_q) - p(u_q), \overline{p}(u_q) - p(u_q)) \\ &\leq \|y_q(u_q) - y(u_q)\|_{H^1(\Omega)} \|\overline{p}(u_q) - p(u_q)\|_{H^1(\Omega)}, \end{aligned}$$

i.e.,

$$\|\overline{p}(u_q) - p(u_q)\|_{H^1(\Omega)} \leq \beta_2^{-1} \|y_q(u_q) - y(u_q)\|_{H^1(\Omega)}.$$

Inserting this bound into (5.4.8) gives

$$\|u - u_q\|_{L^2(\Omega)} \leq C \big(\|p_q(u_q) - \overline{p}(u_q)\|_{H^1(\Omega)} + \|y_q(u_q) - y(u_q)\|_{H^1(\Omega)} \big)$$

for some $C > 0/$ Applying the discretization error estimate from Lemma 4.3.3 for both the adjoint equation discretization error $\|p_q(u_q) - \overline{p}(u_q)\|_{H^1(\Omega)}$ and the state equation discretization error $\|y_q(u_q) - y(u_q)\|_{H^1(\Omega)}$ yields the desired result. □

Now that the performance of the multidomain discretizations is understood in the context of the optimization problem, the remaining chapters focus on using the efficient direct solver that comes with the HPS method to accelerate the solution of optimization problems of the form (1.1.1). For simplicity, attention is restricted to the optimize-then-discretize approach and the strong form discretization is used. Chapter 6 presents the direct solver that comes with the HPS discretization method and provides algorithms for computing solutions to the optimization problem. Then in Section 6.3 a simple numerical example illustrates the performance benefit of using an efficient direct solver in the optimization setting.

Chapter 6

The Hierarchical Poincaré-Steklov Method

This chapter presents the efficient direct solver that comes with the HPS discretization method that will be used to accelerate the solutino of PDE constrained optimization problems. Section 6.1 presents the direct solver for solving a boundary value problem (i.e. the state equation (1.1.1b)). Then Section 6.2 provides algorithms for using the direct solver to solve the optimize-then-discretize formulation of the optimization model problem (1.1.1). Section 6.3 uses a simple numerical example to illustrates the performance benefit of using an efficient direct solver in the optimization setting.

6.1 Solving a Differential Equation

6.1.1 Overview of the Direct Solver

Consider the boundary value problem

$$-\Delta y(x) = u(x) + f(x), \qquad x \in \Omega = (0,1)^2, \tag{6.1.1a}$$

$$y(x) = g(x), \qquad x \in \partial\Omega. \tag{6.1.1b}$$

The direct solver that comes with the Hierarchical Poincaré-Steklov method constructs an approximation to the solution operator of (6.1.1). The domain is partitioned hierarchically via a binary space partitioning tree (refer to Figures 2.1 and 2.2). Once the hierarchical tree has been constructed, the solver consists of a build stage, which constructs the approximation to the solution operator for the differential equation (6.1.1), and a solve stage, which applies the approximation solution operator for a given boundary condition and body load to obtain the solution to the differential equation (6.1.1). The build stage consists of a single upward sweep through the tree from the leaves to the root (from the smallest boxes to the largest box, see Figures 2.1 and 2.2) as described in Algorithm 6.1.1. On each leaf box, a $q \times q$ tensor product grid of collocation points is placed, and the restriction of (6.1.1) to the leaf box is discretized by the standard spectral collocation approach based on the strong form as in Section 4.3.4. See also Boyd [5], or Trefethen [23]. By performing dense linear algebra on matrices of at most size $q^2 \times q^2$ a local solution operator and local DtN operator is formed for each leaf as described in Section 6.1.2. Then beginning the upward sweep, a local solution operator and DtN operator is constructed for each parent in the tree by "merging" the DtN operators from each child as described in Section 6.1.3. This results in a hierarchical representation for approximate solution operator of (6.1.1) and ends the build stage.

Once the solution operator is available the solve stage takes in the bodyload $u + f$ and the boundary data g and returns the approximate solution y. The solve stage consists of an upward sweep and then a downward sweep through the hierarchical tree as described in Algorithm 6.1.2. Given the bodyload $u+f$, precomputed operators are applied on each leaf to evaluate the contribution to the local Neumann data due to the particular solution. Then during the upward sweep through the tree, precomputed operators are applied to evaluate the contribution to the local Neumann data on the parent due to the local Neumann data on each child. Then starting at the root of the tree and sweeping down to the leaves, precomputed operators map the boundary

data g to the boundary data for each child. Finally, at the leaf level, the precomputed local solution operators are applied to obtain the solution everywhere in the domain.

6.1.2 Leaf Computations

On a leaf box Ω^τ, consider the local boundary value problem

$$-\Delta y^\tau(x) = u(x) + f(x), \qquad x \in \Omega^\tau, \tag{6.1.2a}$$

$$y^\tau(x) = g(x), \qquad x \in \partial\Omega^\tau. \tag{6.1.2b}$$

Observe that for this local problem, the body load $(u + f)$ is known, but the local Dirichlet boundary condition g is an unknown that must also be solved for. This is done by finding g such that the normal derivatives $\partial y^\tau/\partial n^\tau$, where n^τ is the unit outward normal of leaf box Ω^τ match the normal derivatives of solution on neighboring boxes on the interface between the boxes.

Let r^τ be the homogeneous solution which satisfies

$$-\Delta r^\tau(x) = 0, \qquad x \in \Omega^\tau, \tag{6.1.3a}$$

$$r^\tau(x) = g(x), \qquad x \in \partial\Omega^\tau, \tag{6.1.3b}$$

and t^τ be the particular solution which satisfies

$$-\Delta t^\tau(x) = u(x) + f(x), \qquad x \in \Omega^\tau, \tag{6.1.4a}$$

$$t^\tau(x) = 0, \qquad x \in \partial\Omega^\tau. \tag{6.1.4b}$$

Then it is clear that $y^\tau = r^\tau + t^\tau$ solves (6.1.2a).

As indicated before, it is necessary to find the normal derivative $\partial r^\tau/\partial n^\tau$ or more precisely the Dirichlet-to-Neumann (DtN) map

$$g \mapsto \partial r^\tau/\partial n^\tau$$

that maps the Dirichlet boundary data g into the normal derivative of the solution r^τ of (6.1.3), as well as the normal derivative $\partial t^\tau/\partial n^\tau$ of the solution of (6.1.4) for given body loads $u + f$.

As in Section 4.2.3, to discretize place a tensor product grid of LGL quadrature points less corners on $\overline{\Omega^\tau}$ (refer to Figure 4.4. As before, let $\widetilde{\mathbf{x}}_j^\tau$ denote the j-th quadrature point, and define the vectors

$$\begin{aligned}
\widetilde{\mathbf{y}}^\tau &= (y(\widetilde{\mathbf{x}}_1^\tau), \ldots, y(\widetilde{\mathbf{x}}_{q^2-4}^\tau))^T, \\
\widetilde{\mathbf{u}}^\tau &= (u(\widetilde{\mathbf{x}}_1^\tau), \ldots, u(\widetilde{\mathbf{x}}_{q^2-4}^\tau))^T, \\
\widetilde{\mathbf{f}}^\tau &= (f(\widetilde{\mathbf{x}}_1^\tau), \ldots, f(\widetilde{\mathbf{x}}_{q^2-4}^\tau))^T, \\
\widetilde{\mathbf{g}}^\tau &= (g(\widetilde{\mathbf{x}}_1^\tau), \ldots, g(\widetilde{\mathbf{x}}_{q^2-4}^\tau))^T.
\end{aligned}$$

Approximate the homogeneous and particular solutions by the poloynomial representations

$$\begin{aligned}
r^\tau(x) \approx r_q^\tau(x) &= \sum_{j=1}^{q^2-4} r(\widetilde{\mathbf{x}}_j^\tau)\widetilde{\psi}_j^\tau(x), \\
t^\tau(x) \approx t_q^\tau(x) &= \sum_{j=1}^{q^2-4} t(\widetilde{\mathbf{x}}_j^\tau)\widetilde{\psi}_j^\tau(x),
\end{aligned}$$

and define the coefficient vectors

$$\begin{aligned}
\widetilde{\mathbf{r}}^\tau &= (r(\widetilde{\mathbf{x}}_1^\tau), \ldots, r(\widetilde{\mathbf{x}}_{q^2-4}^\tau))^T, \\
\widetilde{\mathbf{t}}^\tau &= (t(\widetilde{\mathbf{x}}_1^\tau), \ldots, t(\widetilde{\mathbf{x}}_{q^2-4}^\tau))^T.
\end{aligned}$$

Next, define the partial differentiation matrices $\widetilde{\mathbf{D1}}^\tau$, $\widetilde{\mathbf{D1}}^\tau$, and $\widetilde{\mathbf{L}}^\tau$ as in Section 4.2.3.

Finally, introduce the index sets

$$\begin{aligned}
I &= \{j \mid \widetilde{\mathbf{x}}_j^\tau \in \Omega^\tau\}, \\
B &= \{j \mid \widetilde{\mathbf{x}}_j^\tau \in \partial\Omega^\tau\}.
\end{aligned}$$

Then the discretization of the local homogeneous boundary value problem (6.1.3) is given by

$$\begin{bmatrix} \mathbf{I}_{BB} & \mathbf{0} \\ \widetilde{\mathbf{L}}_{IB}^\tau & \widetilde{\mathbf{L}}_{II}^\tau \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{r}}_B^\tau \\ \widetilde{\mathbf{r}}_I^\tau \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{g}}_B^\tau \\ \mathbf{0} \end{bmatrix}$$

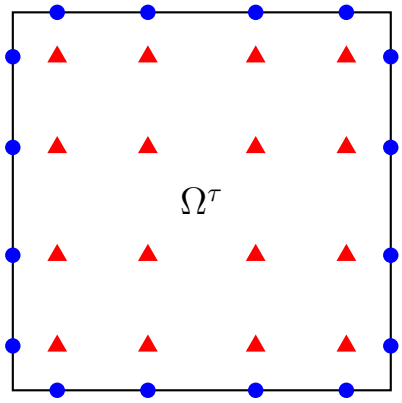

Figure 6.1: Illustration of indexed collocation points on Ω^τ. The blue circles denote collocation points where the Dirichlet boundary condition will be applied. The red triangles denote the interior collocation points where the PDE will be enforced.

Then immediately,

$$\widetilde{\mathbf{r}}_B^\tau = \widetilde{\mathbf{g}}_B^\tau \tag{6.1.5}$$

and solving for $\widetilde{\mathbf{r}}_I^\tau$ yields

$$\widetilde{\mathbf{r}}_I^\tau = -\left(\widetilde{\mathbf{L}}_{II}^\tau\right)^{-1} \widetilde{\mathbf{L}}_{IB}^\tau \, \widetilde{\mathbf{g}}_B^\tau. \tag{6.1.6}$$

Define the local homogeneous solution operator as the operator that acts on the boundary data $\widetilde{\mathbf{g}}_B^\tau$ and returns the local homogeneous solution $\mathbf{r}^\tau$. From (6.1.5) and (6.1.6) the local homgeneous solution operator is given by

$$\mathbf{S}^\tau = \begin{bmatrix} \mathbf{I}_{BB} \\ -\left(\widetilde{\mathbf{L}}_{II}^\tau\right)^{-1} \widetilde{\mathbf{L}}_{IB}^\tau \end{bmatrix} \tag{6.1.7}$$

so that

$$\mathbf{r}^\tau = \mathbf{S}^\tau \, \widetilde{\mathbf{g}}_B^\tau.$$

Similarly, the discretization of the local particular boundary value problem (6.1.4) is given by

$$\begin{bmatrix} \mathbf{I}_{BB} & \mathbf{0} \\ \widetilde{\mathbf{L}}_{IB}^\tau & \widetilde{\mathbf{L}}_{II}^\tau \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{t}}_B^\tau \\ \widetilde{\mathbf{t}}_I^\tau \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \widetilde{\mathbf{u}}_B^\tau + \widetilde{\mathbf{f}}_B^\tau \end{bmatrix}$$

Immediately, $\widetilde{\mathbf{t}}_B^\tau = \mathbf{0}$ and solving for $\widetilde{\mathbf{t}}_I^\tau$ yields

$$\widetilde{\mathbf{t}}_I^\tau = \left(\widetilde{\mathbf{L}}_{II}^\tau\right)^{-1} (\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau). \tag{6.1.8}$$

Define the particular solution operator as the operator that acts on the source data $\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau$ and returns the local particular solution $\widetilde{\mathbf{t}}^\tau$. Then from (6.1.8) the particular solution operator is given by

$$\mathbf{F}^\tau = \begin{bmatrix} \mathbf{0} \\ \left(\widetilde{\mathbf{L}}_{II}^\tau\right)^{-1} \end{bmatrix}. \tag{6.1.9}$$

Then the total local solution $\mathbf{y}^\tau$ is given by

$$\mathbf{y}^\tau = \mathbf{S}^\tau \, \widetilde{\mathbf{g}}_B^\tau + \mathbf{F}^\tau(\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau). \tag{6.1.10}$$

Note that after the construction of the local particular and homogeneous solution operators, all of the unknowns live on the boundary of Ω^τ. That is, given the local boundary data on Ω^τ then simply applying the solution operators yields the solution on the interior of Ω^τ.

Next construct the local DtN operator for Ω^τ. Begin by partitioning the index set B into the subsets S, E, N, and W that correspond to the collocation points on the north, east, south, and west boundaries respectively. Then define the operator $\mathbf{D}^\tau$

$$\mathbf{D}^\tau = \begin{bmatrix} -\widetilde{\mathbf{D}_{\mathbf{2}}}^\tau_{S,(B,I)} \\ \widetilde{\mathbf{D}_{\mathbf{1}}}^\tau_{E,(B,I)} \\ \widetilde{\mathbf{D}_{\mathbf{2}}}^\tau_{N,(B,I)} \\ -\widetilde{\mathbf{D}_{\mathbf{1}}}^\tau_{W,(B,I)} \end{bmatrix}.$$

The local homogeneous DtN operator is given by

$$\mathbf{T}^\tau = \mathbf{D}^\tau \mathbf{S}^\tau, \tag{6.1.11}$$

and similarly the local particular DtN operator is given by

$$\mathbf{H}^\tau = \mathbf{D}^\tau \mathbf{F}^\tau. \tag{6.1.12}$$

Finally, denote the contribution of the particular solution to the Neumann data by

$$\mathbf{h}_B^\tau = \mathbf{H}^\tau(\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau)$$

(this is the discretization of $\partial t^\tau / \partial n^\tau$), so that

$$\widetilde{\mathbf{v}}_B^\tau = \mathbf{T}^\tau \widetilde{\mathbf{g}}_B^\tau + \mathbf{h}_B^\tau. \tag{6.1.13}$$

($\mathbf{T}^\tau \widetilde{\mathbf{g}}_B^\tau$ is the discretization of $g \mapsto \partial r^\tau / \partial n^\tau$.)

Note that going forward I will drop the subscript B as each vector corresponds to points on the boundary.

6.1.3 Merge Operations

Consider a parent box Ω^τ with child boxes Ω^β and Ω^γ such that $\overline{\Omega^\tau} = \overline{\Omega^\beta} \cup \overline{\Omega^\gamma}$. Suppose that the interior unknowns on the child boxes have already been eliminated and the DtN operators on the child boxes are given by

$$\widetilde{\mathbf{v}}^\beta = \mathbf{T}^\beta \widetilde{\mathbf{g}}^\beta + \mathbf{h}^\beta, \qquad \widetilde{\mathbf{v}}^\gamma = \mathbf{T}^\gamma \widetilde{\mathbf{g}}^\gamma + \mathbf{h}^\gamma.$$

Let $\mathbf{x}_j$ be the j-th collocation point on the interface $\partial\Omega^\beta \cup \partial\Omega^\gamma$ and introduce the index sets

$$\begin{aligned} J_1 &= \{j \mid \mathbf{x}_j \in \partial\Omega^\beta \cup \partial\Omega^\tau\}, \\ J_2 &= \{j \mid \mathbf{x}_j \in \partial\Omega^\gamma \cup \partial\Omega^\tau\}, \\ J_3 &= \{j \mid \mathbf{x}_j \in \partial\Omega^\beta \cup \partial\Omega^\gamma\}, \end{aligned}$$

which are illustrated in Figure 6.2.

Then the DtN operators on the child boxes can be written as

$$\begin{bmatrix} \widetilde{\mathbf{v}}_1^\beta \\ \widetilde{\mathbf{v}}_3^\beta \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11}^\beta & \mathbf{T}_{13}^\beta \\ \mathbf{T}_{31}^\beta & \mathbf{T}_{33}^\beta \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{g}}_1^\beta \\ \widetilde{\mathbf{g}}_3^\beta \end{bmatrix} + \begin{bmatrix} \mathbf{h}_1^\beta \\ \mathbf{h}_3^\beta \end{bmatrix}, \qquad \begin{bmatrix} \widetilde{\mathbf{v}}_2^\gamma \\ \widetilde{\mathbf{v}}_3^\gamma \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{22}^\gamma & \mathbf{T}_{23}^\gamma \\ \mathbf{T}_{32}^\gamma & \mathbf{T}_{33}^\gamma \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{g}}_2^\gamma \\ \widetilde{\mathbf{g}}_3^\gamma \end{bmatrix} + \begin{bmatrix} \mathbf{h}_2^\gamma \\ \mathbf{h}_3^\gamma \end{bmatrix}.$$

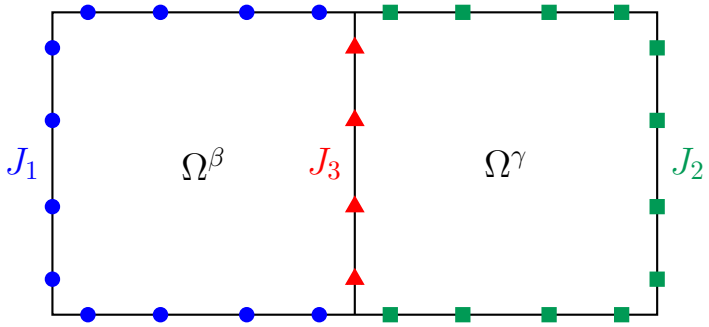

Figure 6.2: Illustration of the indexed points for the merge. The goal of the merge is to eliminate unknowns on the interior edge indexed by J_3.

To enforce that the solution on each subdomain has consistent Dirichlet and Neumann boundary data on the shared edge (i.e. at collocation points in the index set J_3), it is required that $\widetilde{\mathbf{g}}_3^\beta = \widetilde{\mathbf{g}}_3^\gamma = \widetilde{\mathbf{g}}_3$ and $\widetilde{\mathbf{v}}_3^\beta + \widetilde{\mathbf{v}}_3^\gamma = \mathbf{0}$. Then write the combined equation

$$\begin{bmatrix} \widetilde{\mathbf{v}}_1^\beta \\ \widetilde{\mathbf{v}}_2^\gamma \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11}^\beta & \mathbf{0} & \mathbf{T}_{13}^\beta \\ \mathbf{0} & \mathbf{T}_{22}^\gamma & \mathbf{T}_{23}^\gamma \\ \mathbf{T}_{13}^\beta & \mathbf{T}_{23}^\gamma & (\mathbf{T}_{33}^\beta + \mathbf{T}_{33}^\gamma) \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{g}}_1^\beta \\ \widetilde{\mathbf{g}}_2^\gamma \\ \widetilde{\mathbf{g}}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{h}_1^\beta \\ \mathbf{h}_2^\gamma \\ \mathbf{h}_3^\beta + \mathbf{h}_3^\gamma \end{bmatrix}. \tag{6.1.14}$$

To eliminate the unknowns on the interior edge, solve for $\widetilde{\mathbf{g}}_3$ from the bottom row equation in (6.1.14). Define $\mathbf{K}^\tau = \left(\mathbf{T}_{33}^\beta + \mathbf{T}_{33}^\gamma\right)^{-1}$. Then the unknowns on the interior edge are given by

$$\widetilde{\mathbf{g}}_3 = -\mathbf{K}^\tau \begin{bmatrix} \mathbf{T}_{31}^\beta & \mathbf{T}_{32}^\gamma \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{g}}_1^\beta \\ \widetilde{\mathbf{g}}_2^\gamma \end{bmatrix} - \mathbf{K}^\tau \left(\mathbf{h}_3^\beta + \mathbf{h}_3^\gamma\right). \tag{6.1.15}$$

Observe that the first term yields the homogeneous solution on the interior edge. That is, define the local homogeneous solution operator for Ω^τ by

$$\mathbf{S}^\tau = -\mathbf{K}^\tau \begin{bmatrix} \mathbf{T}_{31}^\beta & \mathbf{T}_{32}^\gamma \end{bmatrix}, \tag{6.1.16}$$

so that

$$\mathbf{r}_3^\beta = \mathbf{r}_3^\gamma = \mathbf{S}^\tau \begin{bmatrix} \widetilde{\mathbf{g}}_1^\beta \\ \widetilde{\mathbf{g}}_2^\gamma \end{bmatrix}. \tag{6.1.17}$$

Similarly, the particular solution on the interior edge is given by the second term. That is

$$\mathbf{t}_3^\beta = \mathbf{t}_3^\gamma = -\mathbf{K}^\tau\big(\mathbf{h}_3^\beta + \mathbf{h}_3^\gamma\big). \tag{6.1.18}$$

Since the total solution on the interior edge of the parent box is known, the unknowns on the interior edge are now eliminated.

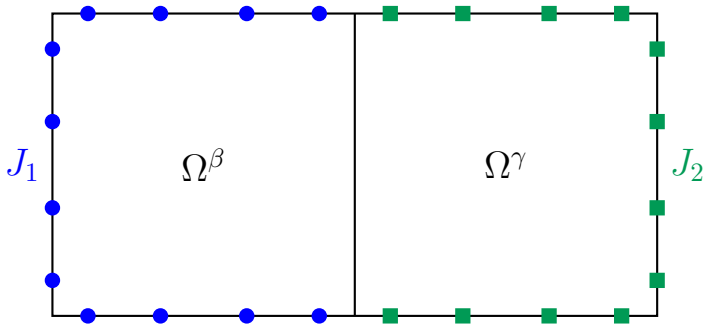

Figure 6.3: Illustration of the indexed points for the merge after the elimination of the unknowns on the interior edge. Observe that the unknowns now lie on the boundary of the parent box Ω^τ.

Then the DtN operators for the parent box Ω^τ is obtained by eliminting $\widetilde{\mathbf{g}}_3$ from the first two row equations in (6.1.14) via (6.1.15)

$$\underbrace{\begin{bmatrix}\widetilde{\mathbf{v}}_1^\beta\\ \widetilde{\mathbf{v}}_2^\gamma\end{bmatrix}}_{\widetilde{\mathbf{v}}^\tau} = \underbrace{\left(\begin{bmatrix}\mathbf{T}_{11}^\beta & \mathbf{0}\\ \mathbf{0} & \mathbf{T}_{22}^\gamma\end{bmatrix} + \begin{bmatrix}\mathbf{T}_{13}^\beta\\ \mathbf{T}_{23}^\gamma\end{bmatrix}\mathbf{S}^\tau\right)}_{\mathbf{T}^\tau}\underbrace{\begin{bmatrix}\widetilde{\mathbf{g}}_1^\beta\\ \widetilde{\mathbf{g}}_2^\gamma\end{bmatrix}}_{\widetilde{\mathbf{g}}^\tau} + \underbrace{\begin{bmatrix}\mathbf{h}_1^\beta\\ \mathbf{h}_2^\gamma\end{bmatrix} - \begin{bmatrix}\mathbf{T}_{13}^\beta\\ \mathbf{T}_{23}^\gamma\end{bmatrix}\mathbf{K}^\tau\big(\mathbf{h}_3^\beta + \mathbf{h}_3^\gamma\big)}_{\mathbf{h}^\tau}. \tag{6.1.19}$$

so that the DtN map for the parent box Ω^τ is given by

$$\widetilde{\mathbf{v}}^\tau = \mathbf{T}^\tau\widetilde{\mathbf{g}}^\tau + \mathbf{h}^\tau. \tag{6.1.20}$$

6.1.4 The Full Solver on a Uniform Grid

Once the domain has been partitioned via the binary space paritioning tree, a pre-computation or *build stage* is performed. The build stage consists of a single upward

sweep through the tree (from the smallest boxes to the largest boxes). For the build stage, any ordering may be used as long as the child nodes in the tree are processed before their parent node.

On each leaf box, solution operators and DtN operators are approximated for the homogeneous and particular solutions by the method described in Section 6.1.2. For a leaf box Ω^τ, the following operators are computed:

- $\mathbf{F}^\tau$ The local particular solution operator maps the evaluation of the source term at the interior collocation points to the values of the local particular solution at each the collocation point. In other words $\widetilde{\mathbf{t}}^\tau = \mathbf{F}^\tau(\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau)$.
- $\mathbf{S}^\tau$ The local homogeneous solution operator maps the values of the local Dirichlet boundary data to the local homogeneous solution at each collocation point. In other words $\widetilde{\mathbf{r}}^\tau = \mathbf{S}^\tau \widetilde{\mathbf{g}}_B^\tau$.
- $\mathbf{H}^\tau$ The local particular DtN operator maps the evaluation of the source term at the interior collocation points to the boundary flux due to the local particular solution at each boundary collocation point. In other words $\mathbf{h}_B^\tau = \mathbf{H}^\tau(\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau)$.
- $\mathbf{T}^\tau$ The local homogeneous DtN operator maps the values of the local Dirichlet boundary data to the boundary flux due to the local homogeneous solution. In other words $\widetilde{\mathbf{v}}_B^\tau = \mathbf{T}^\tau \widetilde{\mathbf{g}}_B^\tau + \mathbf{h}_B^\tau$.

Then on each parent box, the boundary data maps and DtN operators are constructed as described in Section 6.1.3. For each parent node τ with child nodes β and γ, the following operators are computed:

$\mathbf{K}^\tau$ The particular contribution to the boundary data map acts on the flux in the particular solutions on each child at the interior nodes of the parent and returns the particular solution on the interior of the parent. In other words $\widetilde{\mathbf{t}}_3^\beta = \widetilde{\mathbf{t}}_3^\gamma = -\mathbf{K}^\tau(\mathbf{h}_3^\beta + \mathbf{h}_3^\gamma)$.

$\mathbf{S}^\tau$ The homogeneous contribution to the boundary data map acts on the Dirichlet boundary data on the parent and returns the homogeneous solution on the interior of the parent. In other words $\widetilde{\mathbf{r}}_3^\beta = \widetilde{\mathbf{r}}_3^\gamma = \mathbf{S}^\tau \widetilde{\mathbf{g}}^\tau$.

$\mathbf{T}^\tau$ The homogeneous DtN operator for the parent maps the local Dirichlet boundary data on the parent to the boundary flux on the parent due to the local homogeneous soloution. In other words $\widetilde{\mathbf{v}}^\tau = \mathbf{T}^\tau \widetilde{\mathbf{g}}^\tau + \mathbf{h}^\tau$.

An outline for the build stage is provided in Algorithm 6.1.1. Since the algorithm is potentially applied to different PDEs and associated discretizations, the discretized differential operator $\widetilde{\mathbf{A}}$ is used as an input. However, $\widetilde{\mathbf{A}}$ is not needed explicitly.

Algorithm 6.1.1

Input: *discretized differential operator* $\widetilde{\mathbf{A}}$

for $\tau = N_{boxes}, N_{boxes} - 1, \ldots, 1$

if *(τ is a leaf)*

$$\mathbf{F}^\tau = \begin{bmatrix} \mathbf{0} \\ \left(\widetilde{\mathbf{L}}_{II}^\tau\right)^{-1} \end{bmatrix}$$

$$\mathbf{S}^\tau = \begin{bmatrix} \mathbf{I} \\ -\left(\widetilde{\mathbf{L}}_{II}^\tau\right)^{-1} \widetilde{\mathbf{L}}_{IB}^\tau \end{bmatrix}$$

$$\mathbf{H}^\tau = \mathbf{N}^\tau \mathbf{F}^\tau$$

$$\mathbf{T}^\tau = \mathbf{N}^\tau \mathbf{S}^\tau$$

else

Let β and γ be children of τ.

Partition into vectors J_1, J_2, and J_3 as shown in Figure 6.2.

$$\mathbf{K}^\tau = (\mathbf{T}_{33}^\beta + \mathbf{T}_{33}^\gamma)^{-1}$$

$$\mathbf{S}^\tau = -\mathbf{K}^\tau \begin{bmatrix} \mathbf{T}_{31}^\beta & \mathbf{T}_{32}^\gamma \end{bmatrix}$$

$$\mathbf{T}^\tau = \begin{bmatrix} \mathbf{T}_{11}^\beta & 0 \\ 0 & \mathbf{T}_{22}^\gamma \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{13}^\beta \\ \mathbf{T}_{23}^\gamma \end{bmatrix} \mathbf{S}^\tau$$

end if

end for

Output: *HPS solution operator needed to apply* $\widetilde{\mathbf{A}}^{-1}$ *(input for Algorithm 6.1.2)*

After the executing Algorithm 6.1.1, given a specific Dirichlet boundary condition and source term the solution to the boundary value problem may be evaluated rapidly by the *solve stage* of the HPS method. The solve stage consists of a single upward sweep followed by a single downward sweep of the tree. In the upward sweep (from smallest to largest boxes) the particular solutions and boundary fluxes due to the particular solutions are computed. At the root of the tree (the largest box) the solution on the boundary of the domain is then given by the provided Dirichlet boundary condition. During the downward sweep (from largest to smallest boxes) the boundary data maps computed during the build phase are applied to map the boundary data from the root box to the leaf boxes, and finally the local solution operators on each leaf box are applied to obtain the approximate solution. The outline for the solve stage is given by Algorithm 6.1.2.

Algorithm 6.1.2 : $\widetilde{\mathbf{y}} = hps\_solve(\widetilde{\mathbf{A}}^{-1}, \widetilde{\mathbf{u}}, \widetilde{\mathbf{f}}, \widetilde{\mathbf{g}})$

Input: *HPS solution operator needed to apply* $\widetilde{\mathbf{A}}^{-1}$ *(output of Algorithm 6.1.1),*

$\widetilde{\mathbf{u}}$ *- control evaluated at collocation points,*

$\widetilde{\mathbf{f}}$ *- body load component evaluated at collocation points,*

$\widetilde{\mathbf{g}}$ *- Dirichlet data evaluated at collocation points.*

Upward sweep – construct all particular solutions:

for $\tau = N_{boxes}, N_{boxes} - 1, \ldots, 1$

if *(τ is a leaf)*

 # Compute boundary flux due to local particular solution

$\mathbf{h}^\tau = \mathbf{H}^\tau(\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau)$

else

Let β and γ be children of τ.

 # Compute the local particular solution

$\widetilde{\mathbf{t}}_I^\tau = -\mathbf{K}^\tau(\mathbf{h}_3^\beta + \mathbf{h}_3^\gamma)$

 # Compute the boundary flux due to the particular solution

$$\mathbf{h}^\tau = \begin{bmatrix} \mathbf{h}_1^\beta \\ \mathbf{h}_2^\gamma \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{13}^\beta \\ \mathbf{T}_{23}^\gamma \end{bmatrix} \widetilde{\mathbf{t}}_I^\tau$$

end if

end for

Downward sweep – construct all potentials:

 # Use the provided Dirichlet data to set solution on the boundary of the root

$\widetilde{\mathbf{y}}_B^1 = \widetilde{\mathbf{g}}_B^1$

for $\tau = 1, 2, \ldots, N_{boxes}$

if *(τ is a parent)*

 # Add the homogeneous term and the particular term

$\widetilde{\mathbf{y}}_I^\tau = \mathbf{S}^\tau \widetilde{\mathbf{y}}_B^\tau + \widetilde{\mathbf{t}}_I^\tau$.

else

 # Add the homogeneous term and the particular term

$\widetilde{\mathbf{y}}^\tau = \mathbf{S}^\tau \widetilde{\mathbf{y}}_B^\tau + \mathbf{F}^\tau(\widetilde{\mathbf{u}}_I^\tau + \widetilde{\mathbf{f}}_I^\tau)$.

end if

end for

Output: $\widetilde{\mathbf{y}}$ *- computed solution evaluated at collocation points*

6.1.5 Direct Solver Complexity

Let N denote the total number of discretization points in $\Omega \subset \mathbb{R}^d$. Let q be the number of collocation points per linear dimension across a leaf box so that the tensor product grid has q^d collocation points per leaf box. Finally let L be the total number of levels in the binary space partitioning tree. Then there are 2^L leaf boxes and

$N \sim 2^L q^d$ discretization points.

The build stage of the algorithm can be broken into two primary steps, constructing operators on the leaf boxes, and constructing operators for the merge operations. The computational complexity for constructing operators on the leaf boxes is dominated by inverting the dense matrices $\widetilde{\mathbf{L}}_{II}^{\tau}$ of size $\mathcal{O}(q^d) \times \mathcal{O}(q^d)$ on each leaf. Since there 2^L leaf boxes, the total cost for the leaf computations is approximately

$$2^L \times q^{3d} \sim N q^{2d}. \tag{6.1.21}$$

During the merge operations, on each level ℓ in the tree, operators are constructed by inverting the dense matrices $(\mathbf{T}_{33}^{\beta} + \mathbf{T}_{33}^{\gamma})$ of the size $\mathcal{O}(2^{-(d-1)\ell/d} N^{(d-1)/d}) \times \mathcal{O}(2^{-(d-1)\ell/d} N^{(d-1)/d})$. To understand this formula, $N^{(d-1)/d}$ is the number of collocation points along one face of the domain. In 2D this corresponds to $N^{1/2}$ which is the number of nodes along a single edge of the domain. The $2^{-(d-1)\ell/d}$ factor represents the ratio of the size of a merge interface on level ℓ of the binary tree to the size of a face of the domain. For example, for a 2D problem on level $\ell = 2$ of the tree, the size of a merge interface is one half of the size of a face of the total domain. Similarly on level $\ell = 4$ the merge interface is one fourth of the size of a face of the total domain. Since on level ℓ there are 2^ℓ boxes, the total cost for the merge operations is approximately

$$\sum_{\ell=1}^{L} 2^\ell \times 2^{-3(d-1)\ell/d} N^{3(d-1)/d} \sim N^{3(d-1)/d} \sum_{\ell=1}^{L} 2^{(-2d\ell+3\ell)/d}. \tag{6.1.22}$$

For the case $d = 2$ the approximate cost is given by

$$N^{3/2} \sum_{\ell=1}^{L} 2^{-\ell/2} \sim N^{3/2}. \tag{6.1.23}$$

Summing the contributions of the leaf and merge operations gives the computational complexity for the build stage of the algorithm as $\mathcal{O}(N^{3/2})$ for 2D problems.

For the upward sweep of the solve stage, on each of the leaf boxes the operator $\mathbf{H}^{\tau}$ of size $\mathcal{O}(2dq^{d-1}) \times \mathcal{O}(q^d)$ is applied to compute the boundary flux due to the local

particular solution. Thus the complexity of applying the operator on each of the 2^L leaf boxes is given by

$$2^L \times 2dq^{(d-1)d} \sim Nq^{d-1}. \tag{6.1.24}$$

Then on level ℓ in the tree, the matrix $\mathbf{K}^\tau$ of size $\mathcal{O}(2^{-(d-1)\ell/d}N^{(d-1)/d}) \times \mathcal{O}(2^{-(d-1)\ell/d}N^{(d-1)/d})$ is applied on each of the 2^ℓ boxes in the level. This gives the approximate cost

$$\sum_{\ell=1}^{L} 2^\ell \times 2^{-2(d-1)\ell/d}N^{2(d-1)/d} \sim N^{2(d-1)/d}\sum_{\ell=1}^{L} 2^{(-d\ell+2\ell)/d} \tag{6.1.25}$$

For the case $d = 2$ the approximate cost is given by

$$N\sum_{\ell=1}^{L} 1 \sim NL \sim N\log(N). \tag{6.1.26}$$

For the downward sweep, on level ℓ in the tree, the matrices $\mathbf{S}^\tau$ of size $\mathcal{O}(2^{-(d-1)\ell/d}N^{(d-1)/d}) \times \mathcal{O}(2^{-(d-1)\ell/d}N^{(d-1)/d})$ are applied on the 2^ℓ boxes, leading to the same computational complexity as for the upward sweep of the tree. Finally, on the leaf boxes, the homogeneous and particular solution operators $\mathbf{S}^\tau$ and $\mathbf{F}^\tau$ are applied to obtain the total solution. This cost is dominated by applying the particular solution operators, which are matrices of size $\mathcal{O}(q^d) \times \mathcal{O}(q^d)$ on the 2^L leaf boxes. This yields the cost

$$2^L \times q^{2d} \sim Nq^d. \tag{6.1.27}$$

Summing the contributions from each step in the solve algorithm yields the asymptotic complexity of $\mathcal{O}(N\log(N))$ for 2D problems.

Finally, note that for non-highly oscillatory problems, the asymptotic complexity of the build stage and the solve stage may each be improved by exploiting accelerated linear algebra as by Babb et al. [1]. However, this acceleration is not considered in the present work.

6.2 Solving the Optimal Control Problem

Recall the optimality system

$$\begin{bmatrix} -\widetilde{\mathbf{B}} & \mathbf{0} & \widetilde{\mathbf{A}} \\ \mathbf{0} & \alpha\widetilde{\mathbf{I}} & -\widetilde{\mathbf{I}} \\ \widetilde{\mathbf{A}} & \widetilde{\mathbf{B}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{y}} \\ \widetilde{\mathbf{u}} \\ \widetilde{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{c}} \\ \mathbf{0} \\ \widetilde{\mathbf{d}} \end{bmatrix} \tag{6.2.1}$$

for the optimize then discretize approach to the model problem, cf. (5.2.8). Note that the discretized differential operator $\widetilde{\mathbf{A}}$ appears twice in (6.2.1). This is due to the fact to the fact that the differential operator in the state equation of the model problem (1.1.1b) is self-adjoint. In general, $\widetilde{\mathbf{A}}$ in the first block row equation will be the discretization of the adjoint of the differential operator in the state equation. The third and first block can be used to express $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{p}}$ in terms of $\widetilde{\mathbf{u}}$. This gives

$$\widetilde{\mathbf{y}} = \widetilde{\mathbf{A}}^{-1}\left(-\widetilde{\mathbf{B}}\widetilde{\mathbf{u}} + \widetilde{\mathbf{c}}\right)$$

and

$$\begin{aligned} \widetilde{\mathbf{p}} &= \widetilde{\mathbf{A}}^{-1}\left(\widetilde{\mathbf{B}}\widetilde{\mathbf{y}} + \widetilde{\mathbf{d}}\right), \\ &= \widetilde{\mathbf{A}}^{-1}\left(\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\left(-\widetilde{\mathbf{B}}\widetilde{\mathbf{u}} + \widetilde{\mathbf{c}}\right) + \widetilde{\mathbf{d}}\right). \end{aligned}$$

Substituting $\widetilde{\mathbf{p}}$ into the remaining second block of the optimality system yields

$$\begin{aligned} 0 &= \alpha\widetilde{\mathbf{u}} - \widetilde{\mathbf{p}}, \\ &= \alpha\widetilde{\mathbf{u}} - \widetilde{\mathbf{A}}^{-1}\left(\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\left(-\widetilde{\mathbf{B}}\widetilde{\mathbf{u}} + \widetilde{\mathbf{c}}\right) + \widetilde{\mathbf{d}}\right), \\ &= \alpha\widetilde{\mathbf{u}} - \widetilde{\mathbf{A}}^{-1}\left(-\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{B}}\widetilde{\mathbf{u}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{c}} + \widetilde{\mathbf{d}}\right), \\ &= \left(\alpha\widetilde{\mathbf{I}} + \widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{B}}\right)\widetilde{\mathbf{u}} - \widetilde{\mathbf{A}}^{-1}\left(\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{c}} + \widetilde{\mathbf{d}}\right). \end{aligned}$$

This is a linear sytem

$$\underbrace{\left(\alpha\widetilde{\mathbf{I}} + \widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{B}}\right)}_{\stackrel{\text{def}}{=}\mathbf{M}}\widetilde{\mathbf{u}} = \underbrace{\widetilde{\mathbf{A}}^{-1}\left(\widetilde{\mathbf{B}}\widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{c}} + \widetilde{\mathbf{d}}\right)}_{\stackrel{\text{def}}{=}\mathbf{b}}. \tag{6.2.2}$$

in discretized controls $\widetilde{\mathbf{u}}$.

For problems of interest, the matrix $\mathbf{M}$ may be too large to store and compute with directly. Instead, ones applies an iterative scheme that only requires the matrix vector product $\mathbf{M}\,\mathbf{s}$ for a given vector $\mathbf{s}$ instead of an explicit representation of $\mathbf{M}$. If $\mathbf{M}$ is symmetric the conjugate gradient (CG) method [10] can be used. For nonsymmetric $\mathbf{M}$, which is the case in (6.2.2), the generalized minimal residual (GMRES) method [21] can be used.

To take advantage of the efficient direct solver presented in Section 6.1.4 when solving the reduced ptimality system (6.2.2), the direct solver can be used to compute the residual $\mathbf{M}\mathbf{s} - \mathbf{b}$ or the matrix vector product $\mathbf{M}\,\mathbf{s}$. The residual is computed by Algorithm 6.2.1.

Algorithm 6.2.1 : $\mathbf{r} = optsys\_res(\widetilde{\mathbf{A}}^{-1}, \mathbf{s}, \widetilde{\mathbf{f}}, \widetilde{\mathbf{z}}, \widetilde{\mathbf{g}})$

Input: *HPS state (and adjoint) solution operator (output of Algorithm 6.1.1),*

$\mathbf{s}$ - *trial control vector,*

$\widetilde{\mathbf{f}}$ - *state equation body load component,*

$\widetilde{\mathbf{z}}$ - *desired state,*

$\widetilde{\mathbf{g}}$ - *state equation Dirichlet boundary condition.*

$\widetilde{\mathbf{y}} = hps\_solve(\widetilde{\mathbf{A}}^{-1}, \mathbf{s}, \widetilde{\mathbf{f}}, \widetilde{\mathbf{g}})$.

$\widetilde{\mathbf{p}} = hps\_solve(\widetilde{\mathbf{A}}^{-1}, -\widetilde{\mathbf{y}}, \widetilde{\mathbf{z}}, \mathbf{0})$.

Output: $\mathbf{r} = \alpha\mathbf{s} - \widetilde{\mathbf{p}}$ - *residual of reduced optimality system* ($\mathbf{r} = \mathbf{M}\mathbf{s} - \mathbf{b}$).

Note that the matrix vector product $\mathbf{M}\,\mathbf{s}$ can be evaluated by (6.2.2) with $\widetilde{\mathbf{c}} = \widetilde{\mathbf{d}} = \mathbf{0}$ (which is equivalent to setting $\widetilde{\mathbf{g}} = \widetilde{\mathbf{f}} = \widetilde{\mathbf{z}} = \mathbf{0}$). Thus the matrix vector product is evaluated by Algorithm 6.2.2.

Algorithm 6.2.2 : $\mathbf{v} = optsys\_matvec(\widetilde{\mathbf{A}}^{-1}, \mathbf{s})$

Input: *HPS state (and adjoint) solution operator (output of Algorithm 6.1.1),*

$\mathbf{s}$ - *trial control vector.*

$\widetilde{\mathbf{y}} = hps\_solve(\widetilde{\mathbf{A}}^{-1}, \mathbf{s}, \mathbf{0}, \mathbf{0})$.

$\widetilde{\mathbf{p}} = hps\_solve(\widetilde{\mathbf{A}}^{-1}, -\widetilde{\mathbf{y}}, \mathbf{0}, \mathbf{0})$.

Output: $\mathbf{v} = \alpha\mathbf{s} - \widetilde{\mathbf{p}}$ - *reduced optimality system matrix-vector product* ($\mathbf{v} = \mathbf{Ms}$).

6.3 The Benefit of Using a Direct Solver in Optimization

So far, this work has illustrated how to use the HPS discretization and the direct solver in the optimization setting. However, the goal of this work is to reduce cost and thus extend the range of practical optimization problems that can be solved. It remains to be shown that the direct solver does in fact reduce the cost of solving the PDE constrained optimization problem. This section illustrates the benefit of using the efficient direct solver that comes with the HPS method for solving an optimization problem of the form of the model problem (1.1.1).

Let $\Omega = (0,1)^2$, $\alpha = 0.1$, and define the functions

$$z(x) = 10\pi^2 \sin(3\pi x_1)\sin(\pi x_2),$$
$$f(x) = 10\pi^2 \sin(\pi x_1)\sin(3\pi x_2) - \frac{1}{\alpha}\sin(3\pi x_1)\sin(\pi x_2).$$

Then consider the optimal control problem

$$\underset{u}{\text{Minimize}}\ J(u) \stackrel{\text{def}}{=} \frac{1}{2}\int_\Omega \left(y(x;u) - z(x)\right)^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx$$

where $y(\,\cdot\,;u) \in H_0^1(\Omega)$ is the solution of

$$\begin{cases} -\Delta y(x) = u(x) + f(x), & x \in \Omega, \\ \quad y(x) = 0, & x \in \partial\Omega, \end{cases}$$

for a given control u. The optimal control u_{ex} and the corresponding state y_{ex} and adjoint p_{ex} are given by

$$u_{ex}(x) = 10\sin(3\pi x_1)\sin(\pi x_2),$$
$$y_{ex}(x) = \sin(\pi x_1)\sin(3\pi x_2),$$
$$p_{ex}(x) = \sin(3\pi x_1)\sin(\pi x_2).$$

Let the strong form four leaf discretization of the state equation be given by

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{y}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{u}} = \widetilde{\mathbf{c}}.$$

Then under the optimize-then-discretize approach, the solution to the test problem is the solution to

$$\begin{bmatrix} -\widetilde{\mathbf{B}} & \mathbf{0} & \widetilde{\mathbf{A}} \\ \mathbf{0} & \alpha\widetilde{\mathbf{I}} & -\widetilde{\mathbf{I}} \\ \widetilde{\mathbf{A}} & \widetilde{\mathbf{B}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{y}} \\ \widetilde{\mathbf{u}} \\ \widetilde{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{W}}\widetilde{\mathbf{z}} \\ \mathbf{0} \\ \widetilde{\mathbf{c}} \end{bmatrix}$$

Finally, consider applying GMRES to the reduced optimality system

$$\mathbf{M}\widetilde{\mathbf{u}} = \mathbf{b}$$

as defined in (6.2.2).

In each GMRES iteration, the matrix vector product $\mathbf{M}$ applied to a vector is computed by solving the state equation and then the adjoint equation as described in Algorithm 6.2.2.

To determine if using the direct solver reduces the cost for this simple test problem, compare run times for two vwersions. In one version solving the state and adjoint solves needed to apply $\mathbf{M}$ to a vector are performed by precomputing the HPS solution operators. In the other version solving the state and adjoint solves needed to apply $\mathbf{M}$ to a vector are performed by applying GMRES. The Matlab `gmres` function was used as the iterative solver to obtain solutions to the state and adjoint equations. Table 6.1 reports the timings results for the experiment applied to the test problem with the strong form four leaf discretization.

The subscript "gm" refers to the results for which `gmres` was used to compute the solution to the state and adjoint equations and the subscript "hps" refers tot he results for which the HPS solution operator was applied to solve the state and adjoint equations.

Table 6.1: Timing Results for Four Leaf Test Problem

q	$E_{gm}(u)$	$E_{hps}(u)$	k_{gm}	k_{hps}	t_{gm}	t_{hps}	t_{ratio}
6	4.93e-02	4.93e-02	4	4	0.133	0.026	5.20
8	2.86e-03	2.86e-03	3	3	0.375	0.024	15.4
10	7.41e-05	7.41e-05	3	3	1.12	0.015	74.6
12	1.13e-06	1.14e-06	3	3	2.91	0.029	99.2
14	7.64e-09	1.17e-08	3	2	6.40	0.026	245
16	2.05e-08	8.72e-11	3	2	12.3	0.030	414

Define the relative error

$$E(u) = \frac{\max_j |\widetilde{\mathbf{u}}_j - u_{ex}(\widetilde{\mathbf{x}}_j)|}{\max_j |u_{ex}(\widetilde{\mathbf{x}}_j)|}.$$

Define k as the number of GMRES iterations required to solve the reduced optimality system. Finally, t_{gm} gives the total time to solve the reduced optimality system, t_{hps} gives the total time to precompute the state and adjoint HPS solution operators plus the time to solve the reduced optimality system, and

$$t_{ratio} = \frac{t_{gm}}{t_{hps}}.$$

As the polynomial order increases, using the direct solver is much more efficient than applying an iterative solver each time a PDE solution is required. While this is an encouraging first result, it is necessary to compare the performance of using the direct solver vs. leading methods in PDE constrained optimization on a more realistic problem to quantify the reduction in cost.

Chapter 7

Conclusion

This thesis developed a framework for using the HPS method in the context of PDE constrained optimization. In Chapter 5 the HPS discretization was examined in the optimization setting under both the optimize-then-discretize and discretize-then-optimize approaches.

In the optimize-then-discretize approach, optimization theory was used to derive the continuous optimality conditions which consisted of the state equation, the adjoint equation, and a relationship between the control and the adjoint variables. Then the continuous optimality system was discretized to provide a finite dimensional linear system to be solved. Discretizing the state and adjoint equations by the strong form of the HPS method provided the expected convergence behavior for the optimal control problem.

In the discretize-then-optimize approach, the objective function and constraint (state equation) were immediately discretized, resulting in a finite dimensional optimization problem. The finite dimensional optimality conditions were derived consisting of the discretized state equation, an equation involving the transpose of the discretized differential operator from the state equation, and a relationship between the control and the adjoint.

Discretization of the state equation by the strong form of the HPS method set all

of the control variables along merge interfaces equal to zero since the strong form of the Neumann condition does not touch the body load on a leaf box boundary. This prevents convergence to the exact solution for the optimization problem.

Several methods were examined to attempt to restore the convergence behavior under the discretize-then-optimize approach. First, an additional constraint was added to the finite dimensional optimization problem to explicitly require that the control along the merge interfaces satisfy the differential equation. Second, a modified discretization of the state equation was considered that discretized the control via a tensor product grid of Legendre-Gauss points on each leaf box (which only live on the interior of the domain). Each of these methods improved the errors, but did not restore the convergence to the desired behavior. Finally, the state equation was discretized by the weak form of the HPS method discretization, which restored the convergence behavior to the same rate observed in the optimize-then-discretize approach.

After establishing the performance of the HPS discretization in the optimization setting, a simple numerical test was used to examine the cost reduction by using the direct solver from the HPS method. Under the optimize-then-discretize approach, the reduced optimality system was solved iteratively by GMRES where the application of the matrix-vector product was computed on one hand by calling an inner loop of GMRES to solve the state and adjoint equations, and on the other hand using the direct solver from the HPS method to solve the state and adjoint equations. Timing results for the numerical experiment indicate that applying the direct solver significantly reduces the cost of solving the reduced optimality system.

However, to understand the cost reduction in a realistic scenario, a practical optimization problem should be considered, and computing solutions to the state and adjoint equations via the HPS direct solver should be compared to solving the state and adjoint equations by a problem-specific preconditioned iterative scheme. Comparing the performance of the HPS method in the optimization setting against a state

of the art method for PDE constrained optimization will illuminate cost reduction achieved by using the efficient direct solver.

Other areas for future work include implemented the accelerated $\mathcal{O}(N)$ version of the HPS method for non-oscillatory problems to obtain further efficiency from the direct solver. Additionally, when localized phenomena are expected, using the adaptive refinement version of the HPS method will lead to further cost reduction. Finally, as many real applications have additional constraints, for example bounds on the state or control variables, the model problem should be generalized to consider additional constraints as well as variable coefficient PDEs.

Bibliography

[1] T. Babb, A. Gillman, S. Hao, and P.-G. Martinsson. An accelerated Poisson solver based on multidomain spectral discretization. *arXiv:1612.02736v1*, 2016.

[2] P. Benner, E. Sachs, and S. Volkwein. Model order reduction for PDE constrained optimization. In G. Leugering, P. Benner, S. Engell, A. Griewank, H. Harbrecht, M. Hinze, R. Rannacher, and S. Ulbrich, editors, *Trends in PDE constrained optimization*, volume 165 of *Internat. Ser. Numer. Math.*, pages 303–326. Birkhäuser/Springer, Cham, 2014.

[3] C. Bernardi and Y. Maday. Spectral methods. In P. G. Ciarlet and J. L. Lions, editors, *Handbook of numerical analysis, Vol. V. Techniques of scientific computing. Part 2*, Handb. Numer. Anal., V, pages 209–486. North-Holland, Amsterdam, 1997.

[4] C. Borges, A. Gillman, and L. Greengard. High resolution inverse scattering in two dimensions using recursive linearization. *SIAM J. Imaging Sci.*, 10(2):641–664, 2017.

[5] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover, New York, 2000.

[6] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zhang. *Spectral methods. Fundamentals in single domains*. Scientific Computation. Springer-Verlag, Berlin, 2006.

[7] C. Canuto and A. Quarteroni. Approximation results for orthogonal polynomials in Sobolev spaces. *Math. Comp.*, 38(157):67–86, 1982.

[8] A. Gillman and P.-G. Martinsson. A direct solver with $O(N)$ complexity for variable coefficient elliptic PDEs discretized via a high-order composite spectral collocation method. *SIAM J. Sci. Comput.*, 36(4):A2023–A2046, 2014.

[9] M. Heinkenschloss. Numerical solution of implicitly constrained optimization problems. Technical Report TR08–05, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005–1892, 2008.

[10] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. of Research National Bureau of Standards*, 49:409–436, 1952.

[11] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling, Theory and Applications*. Springer Verlag, Heidelberg, New York, Berlin, 2009.

[12] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer Verlag, Berlin, Heidelberg, New York, 1971.

[13] Y. Maday and E. M. Rønquist. Optimal error analysis of spectral methods with emphasis on nonconstant coefficients and deformed geometries. *Comput. Methods Appl. Mech. Engrg.*, 80(1-3):91–115, 1990. Spectral and high order methods for partial differential equations (Como, 1989).

[14] P.-G. Martinsson. A direct solver for variable coefficient elliptic PDEs discretized via a composite spectral collocation method. *J. Comput. Phys.*, 242:460–479, 2013.

[15] S. A. Orszag. Spectral methods for problems in complex geometries. *J. Comput. Phys.*, 37(1):70–92, 1980.

[16] B. Peherstorfer, K. Willcox, and M. D. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Technical Report ACDL TR16-1, Department of Aeronautics & Astronautics, MIT, Cambridge, MA 02139, 2016.

[17] H. P. Pfeiffer, L. E. Kidder, M. A. Scheel, and S. A. Teukolsky. A multidomain spectral method for solving elliptic equations. *Comput. Phys. Comm.*, 152(3):253–273, 2003.

[18] A. Quarteroni. *Numerical models for differential problems*, volume 2 of *MS&A. Modeling, Simulation and Applications*. Springer-Verlag Italia, Milan, 2009. Translated from the 4th (2008) Italian edition by Silvia Quarteroni.

[19] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in Applied Mathematics, Vol. 37. Springer, Berlin, Heidelberg, New York, 2000.

[20] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, Berlin, Heidelberg, New York, 1994. First softcover printing 2008.

[21] Y. Saad and M. H. Schultz. GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7:856–869, 1986.

[22] E. W. Sachs and S. Volkwein. POD-Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):194–208, 2010.

[23] L. N. Trefethen. *Spectral Methods in Matlab*. SIAM, Philadelphia, 2000.

[24] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2010.

[25] B. D. Welfert. Generation of pseudospectral differentiation matrices I. *SIAM J. Numer. Anal.*, 34:1640–1657, 1997.