UNIVERSITY OF CALIFORNIA SAN DIEGO

Unstructured Space-Time Finite Element Methods in Four Dimensions

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Mathematics with Specialization in Computational Science

by

David Charles Lenz

Committee in Charge:

Professor Randolph Bank, Chair
Professor Jiun-Shyan Chen
Professor Michael Holst
Professor Petr Krysl
Professor Melvin Leok

2020

The dissertation of David Charles Lenz is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California San Diego

2020

DEDICATION

For Ashley, who makes time stand still.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I am incredibly grateful to my advisor, Randy Bank, who has been an invaluable guide to me in the course of this research. The study of finite element methods is deeply intertwined with ideas from approximation theory, numerical linear algebra, functional analysis, and high-performance computing, and it is very easy to get lost in the details, losing the forest for the trees. Throughout our many conversations, I routinely entered Randy's office with a slate of questions to ask and left his office with answers to the questions I should have been asking. As I leave the doctoral program, I attribute my ability to guide my own research and ask "the important questions" to Randy's example.

In this final year at UCSD, I have been extremely fortunate to work alongside Eric Lybrand, whose consistent encouragement and support was a relief in the midst of stress. Thank you Eric for your affirmation, patient listening when I wanted to rant, and for your gentle prodding to take a break every once in a while.

My research at UCSD would not be the same without the support of Yifeng Cui, who welcomed me into his research group at the San Diego Supercomputer Center and introduced me to the field of high-performance computing for the first time. I am also indebted to Alex Breuer, who guided my research at SDSC and patiently answered my hundreds of questions as I took my first steps into computational science research. And perhaps I would not have been so bold to apply to join Yifeng's group if it were not for Alan Hylton at NASA Glenn Research Center, who inspired me to apply my mathematical knowledge to real-world problems. Thank you Yifeng, Alex, and Alan.

My growth as a mathematician and as a student in general has been supported by more teachers than I can list here. To all the teachers who have built me up, thank you. From the bottom of my heart, I have the most profound gratitude for all of your efforts and sacrifices that have gone unnoticed, even by myself. You have all changed my life.

There are some forms of gratitude which cannot be captured by words, but in the case of my first teachers I cannot help but try. To my parents, I could not have entered this world with a greater blessing. I found my zeal for learning by watching you, and my courage to try new things comes from your unceasing support. Thank you for every sacrifice that you made to support me in my education. I realize now how unspeakably lucky I am to have been raised by such kind, smart, and joyful parents.

I am thankful for many people for many reasons, but none so much as my brilliant wife Ashley,

who stood by me and loved me every step of the way. Her tenacity and dedication is inspiring, and I could not have pushed through the challenges of the last five years without her example and encouragement. Doctoral study is a grueling, disorienting, and humbling experience for many people, and I am no exception. But - even on the most crushing and exhausting days - I was never without the support and assurance of my greatest advocate. I cannot overstate how much of a consolation this was.

To Ashley, thank you for the innumerable ways that you have sustained me through the last five years. I was able to write this dissertation because you made a choice every day to love and build me up. Thank you for all the times you helped me "really get out there" and step out of my comfort zone - without those steps, I wouldn't be where I am today. And of course, thank you for your patience, understanding, and encouragement when things didn't go to plan. Your faith in me is everything. I love you, Ashley.

In compliance with the university instructions on material in preparation, I also make the following acknowledgments:

Chapter 2, in part, is currently being prepared for submission for publication of the material. The dissertation author was the sole investigator and author of this material.

Chapter 3, in part, is currently being prepared for submission for publication of the material. The dissertation author was the sole investigator and author of this material.

VITA

2015 Bachelor of Science, University of Notre Dame

2017 Master of Arts, University of California San Diego

2020 Doctor of Philosophy, University of California San Diego

ABSTRACT OF THE DISSERTATION

Unstructured Space-Time Finite Element Methods in Four Dimensions

by

David Charles Lenz

Doctor of Philosophy in Mathematics with Specialization in Computational Science

University of California San Diego, 2020

Professor Randolph Bank, Chair

Large-scale simulations of time-dependent partial differential equations are, at present, largely reliant on massively parallel computers. As a result, the parallel scalability of numerical methods for partial differential equations is of crucial importance. In recent years, continuous space-time finite element methods have emerged as a promising technique for approximating these equations in a scalable, flexible way. In a space-time finite element method, the space and time variables of a time-dependent equation are treated as a single unified variable in higher-dimensional space. The higher-dimensional space-time domain is discretized into a collection of simplices and finite element methods may then be defined over this discretization. Parallelization is then achieved through domain decomposition techniques.

In this dissertation, we extend the theory of space-time finite element methods to a more general class of problems. We prove new theoretical results describing the stability of space-time methods applied to parabolic partial differential equations with nontrivial convection and reaction terms. In

particular, we define a streamline-upwind scheme which upwinds in the direction of the space-time convection. The stabilized method is proved to be coercive with respect to an energy norm and asymptotic error bounds are derived.

This dissertation also proposes several operations for the construction and refinement of unstructured, conforming four-dimensional simplex meshes. We define a simple algorithm which takes as input any tetrahedral mesh and produces a corresponding four-dimensional, simplicial space-time mesh. Our algorithm always produces conforming triangulations and may be run concurrently for each spatial element. In addition, we describe how four-dimensional simplex elements can be bisected in order to achieve local spatiotemporal refinement.

Chapter 1

Introduction

Over the last several decades, high-fidelity numerical simulations have been instrumental to the development of fields ranging from seismology to fluid dynamics to biotechnology. This explosion in simulation accuracy has been driven largely through improvements in CPU speeds, but these speeds have leveled off in recent years. Increases in computational power now rely on the simultaneous use of multiple processors, and this progression is even more pronounced in state-of-the-art supercomputers, where large-scale applications routinely use hundreds of thousands of cores. Unfortunately, creating mathematical methods that can be executed in parallel is a nontrivial task. In order to maintain the decades-long expansion of computational capability, new, scalable numerical methods are needed.

Finite element methods (FEMs) are a broad class of numerical methods which are used to solve partial differential equations (PDEs) in a wide variety of fields. These methods are integral to the inner workings of CAD (computer aided design) systems, which are ubiquitous in the practice of modern engineering. FEMs can also be exceptionally efficient - simulations conducted on the world's fastest supercomputers routinely utilize finite element methods for their numerical computations. In addition, the mathematical theory of FEMs has a rich structure that allows for detailed error estimates to be rigorously proven. Over the last fifty years, the breadth and depth of study of these methods has made FEMs one of the most widespread methods for approximating the solution to PDEs.

Over the years, a number of finite element methods have been studied which can run efficiently on parallel computers. An early approach to parallel finite element methods leveraged domain decomposition techniques for elliptic PDEs. In these methods, the domain of the PDE is subdivided into a number of smaller subdomains, which are assigned to different processors of a parallel computer.

Each processor then solves the PDE on its own subdomain, communicating and combining their local solutions with the other processors as necessary. Domain decomposition methods scale very efficiently on a wide variety of problems and are now a cornerstone of finite element solvers for elliptic PDEs.

It should be noted that the time-evolution equations which govern physical processes do not have an elliptic structure, which means that domain decomposition methods cannot be applied to them directly. However, in the context of time-stepping finite element methods, it is often necessary to solve a sequence of elliptic PDEs, where each step in the sequence corresponds to a further time in the simulation. By applying domain decomposition to each sub-problem in a time-stepping method, a good deal of the work of solving time-dependent PDEs may be carried out in parallel. Essentially, time-stepping methods with domain decomposition compute a sequence of sub-problems, where each sub-problem is carried out in parallel.

However, this broad-brush description of parallel finite element methods via domain decomposition hides a number of important details. In practice, the inter-processor communication that is required by domain decomposition methods can bog down the entire solver if one is not careful. As the number of subdomains increases, more time must be spent communicating between different processors. At the same time, each subdomain gets smaller, and the amount of work each processor applies to actually solving the PDE on its own subdomain decreases. Taken to an extreme, increasing the number of processors in a domain decomposition scheme can lead to a situation where the solver spends more time passing messages between processors than it does actually solving the PDE.

Given the huge number of parallel ranks on a modern supercomputer, it is now more challenging than ever to leverage the full power of these machines without becoming mired in communication overhead. When faced with a method that is dominated by message passing, the general solution is to modify the algorithm so that the ratio of communication to computation goes down. When solving time-dependent PDEs, one way to achieve this is to approximate the solution at various simulation times all at once. In other words, instead of solving a long sequence of elliptic problems one after another, one may structure the computation so that the serial time-stepping structure is avoided.

In the finite element literature, what we have just described is generally referred to as a "parallel-in-time" method. A number of approaches have been proposed over the years[21] which aim to reduce or eliminate the need for sequential time-stepping in time-dependent problems.

In this dissertation, we consider space-time finite element methods, which are a class of parallel-

in-time method. In particular, we develop the theory of continuous Galerkin methods on unstructured space-time meshes, which have received renewed attention over the last ten years. This dissertation presents a novel analysis of continuous space-time Galerkin methods for general linear parabolic PDEs, which to our knowledge is the first of its kind. In addition, we study the construction and manipulation of unstructured four-dimensional meshes for use with space-time methods. We give a new algorithm for four-dimensional mesh construction and exhibit a mesh refinement technique for four-dimensional simplices.

The dissertation is structured as follows. In the remainder of Chapter 1, we define the basic terminology and theory that will serve as building blocks for our analysis. We also introduce a model parabolic PDE, the transient convection-diffusion equation, and describe its structure and applications. In Chapter 2, we present an analysis of space-time finite element methods applied to general parabolic PDEs. The results in this chapter are the first rigorous analysis of space-time finite element methods applied to parabolic PDEs with nontrivial convection and reaction terms. We prove the stability of the numerical scheme and establish *a priori* error estimates on the finite element solution, even in the presence of low regularity. In Chapter 3, we address a major challenge in the implementation of space-time methods, which is the use of four-dimensional unstructured meshes. At present, research considering four-dimensional space-time finite element methods treat problems defined on very simple domains. We introduce a new algorithm which creates four-dimensional space-time meshes from a given three-dimensional spatial mesh. We also described methods for refining four-dimensional meshes via simplex bisection. Finally, in Chapter 4 we make some concluding remarks on the continued development of space-time methods.

Comparison to Time-Stepping Methods

In contrast to time-stepping methods, space-time methods are characterized by a simultaneous treatment of the space and time variables as a unified space-time variable $y = (x, t)$. In this paradigm, the fundamental domain is the space-time domain $Q = \Omega \times [0, T]$, which is one dimension higher than the spatial domain. This relatively minor change in perspective at a high level has profound consequences for the ways in which numerical methods are constructed.

For instance, consider an abstract PDE where a solution $u(x,t)$ satisfies

$$\mathcal{A}(u(x,t)) = g(x,t) \qquad \text{for almost every } x \in \Omega \quad \text{and} \quad t \in [0,T],$$

where $\mathcal{A}$ is some differential operator. This equation is a (almost-everywhere) pointwise condition on $\mathcal{A}(u(x,t))$. To make the problem more tractable, most numerical methods solve the related variational problem:

$$\int_\Omega \mathcal{A}(u(x,t))v(x)\,dx = \int_\Omega g(x,t)v(x)\,dx \qquad \text{for all } v \in V \quad \text{and almost every } t \in [0,T].$$

This condition relaxes the pointwise condition on x, but maintains a pointwise condition on t. This problem can be solved in a multitude of ways by further adjusting and discretizing the equation. At a high level, however, numerical schemes eventually define some discretization of the spatial domain and time domain; the structure of these two discretizations and how they interact with each other determine the overall form of the method. For our purposes, the key characteristic of these methods is the *separate* discretization of the space and time domains. While these two discretization may be coupled in various ways, the treatment of time and space variables is fundamentally distinct.

We emphasize that the separation of space and time discretizations is not an inherently negative thing. In fact, many numerical methods are exceptionally efficient as a direct result of this separation. However, it is possible to derive more general and flexible methods by considering a simultaneous approximation of space and time.

For instance, the integral equation listed above implies the related condition

$$\int_0^T \int_\Omega \mathcal{A}(u(x,t))v(x)\,dx\,dt = \int_0^T \int_\Omega g(x,t)v(x)\,dx\,dt \qquad \text{for all } v \in V,$$

and the two equations are equivalent if the functions are sufficiently smooth. If we define $Q = \Omega \times [0,T]$ and $y = (x,t)$, then this equation of double integrals can be recast a single integral over higher-dimensional space:

$$\int_Q \mathcal{A}(u(y))v(y)\,dy = \int_Q g(y)v(y)\,dy \qquad \text{for all } v \in V'$$

where V' is some extension of V to the higher-dimensional space. This final integral equation is just

another variational problem, which can be discretized as if there were no time-dependence at all. By imposing a particular structure on the discretization, one can recover particular time-stepping methods as special cases. However, it is equally valid to define an unstructured discretization on Q, which was not possible before. For this reason we say that space-time methods possess greater "flexibility" than time-stepping methods: they allow for the problem to be discretized in a more general setting.

1.1 Preliminaries

1.1.1 Convex Sets and Simplices

Definition 1.1. A set $\{v_i\}_{i=0}^m$ of $m+1$ vectors in Euclidean space is said to be *affinely independent* if $\{v_1 - v_0, v_2 - v_0, \dots, v_m - v_0\}$ is linearly independent. A set of vectors which is not affinely independent is said to be affinely dependent.

The set

$$\operatorname{affSpan}(V) = v_0 + \langle v_1 - v_0, \dots, v_m - v_0 \rangle$$

is called the *affine span* of $\{v_i\}_{i=0}^m$.

Intuitively, the affine span of $m+1$ affinely independent vectors form an "offset" m-dimensional subspace. For instance, three affinely independent vectors $\{v_0, v_1, v_2\}$ in $\mathbb{R}^3$ lie in the plane containing v_0 and spanned by $v_1 - v_0$ and $v_2 - v_0$.

There are a number of equivalent ways to define affine spans, each with their own strengths and weaknesses. It is often much more convenient to prove properties of affine spans by applying linear algebra techniques in these various settings. The following lemmas establish some of these correspondences.

Lemma 1.2. *Let $V = \{v_i\}_{i=0}^m$. Then*

$$u \in \operatorname{AffSpan}(V) \iff u = \sum_{i=0}^{m} \alpha_i v_i \qquad \text{where } \sum_{i=0}^{n} \alpha_i = 1. \tag{1.1}$$

Proof. If $u \in \operatorname{AffSpan}(V)$, then there exists some coefficients β_i such that

$$u = v_0 + \sum_{i=1}^{m} \beta_i (v_i - v_0).$$

Redistributing the sum, we have

$$u = \sum_{i=1}^{m} \beta_i v_i + \left(1 - \sum_{i=1}^{m} \beta_i\right) v_0,$$

and so under the definitions

$$\alpha_0 = 1 - \sum_{i=1}^{m} \beta_i, \quad \alpha_i = \beta_i \quad \text{for } 1 \leq i \leq m$$

the lemma is proved in one direction. The same argument can be carried out in reverse to prove the converse. □

Remark 1.3. When a vector u is written as a combination of vectors of the form Equation 1.1 (in particular, when the coefficients sum to 1), we say that u is an affine combination of $v_0, \ldots, v_m$.

Lemma 1.4. *Let $V = \{v_i\}_{i=0}^m$ and $V' = \{(v_i, 1)^T\}_{i=0}^m$. Then*

$$u \in \text{AffSpan}(V) \iff (u, 1)^T \in \langle V' \rangle.$$

Proof. By Lemma 1.2,

$$\begin{aligned} u \in \text{AffSpan}(V) &\iff \sum_{i=0}^{m} \alpha_i v_i = u \quad \text{and} \quad \sum_{i=0}^{m} \alpha_i = 1 \\ &\iff \begin{pmatrix} u \\ 1 \end{pmatrix} = \sum_{i=0}^{m} \alpha_i \begin{pmatrix} v_i \\ 1 \end{pmatrix} \\ &\iff \begin{pmatrix} u \\ 1 \end{pmatrix} \in \langle V' \rangle. \end{aligned}$$

□

Lemma 1.5. *The set of vectors $V = \{v_0, v_1, \ldots, v_m\} \subset \mathbb{R}^d$ is affinely independent if and only if the set $V' = \{(v_0, 1)^T, (v_1, 1)^T, \ldots, (v_m, 1)^T\} \subset \mathbb{R}^{d+1}$ is linearly independent.*

Proof. We will prove that V is affinely dependent if and only if V' is linearly dependent. Suppose V is affinely dependent. Then there exists a set of coefficients α_i, $1 \leq i \leq m$, not all zero such that $\sum_1^m \alpha_i (v_i - v_0) = 0$, and thus $\sum_1^m \alpha_i v_i = \left(\sum_1^m \alpha_i\right) v_0$.

If $\sum_1^m \alpha_i = 0$, then $0 \cdot (v_0, 1)^T + \sum_1^m \alpha_i \cdot (v_i, 1)^T = 0$ and V' is linearly dependent. If $\sum_1^m \alpha_i \neq 0$, define $\alpha_i' = \alpha_i / (\sum_1^m \alpha_i)$. Then $\sum_1^m \alpha_i' v_i = v_0$ and $\sum_1^m \alpha_i' = 1$, which means that $\sum_1^m \alpha_i' (v_i, 1)^T = (v_0, 1)^T$. Thus V' is linearly dependent in this case as well.

Now suppose that V' is linearly dependent. Then there is some set of coefficients β_i, $0 \leq i \leq m$, not all zero such that $\sum_0^m \beta_i v_i = 0$ and $\sum_0^m \beta_i = 0$. Hence

$$\begin{aligned}
\sum_1^m \beta_i (v_i - v_0) &= \sum_1^m \beta_i v_i - \left(\sum_1^m \beta_i \right) v_0 \\
&= \sum_1^m \beta_i v_i - \left(\left(\sum_0^m \beta_i \right) - \beta_0 \right) v_0 \\
&= \sum_1^m \beta_i v_i - (0 - \beta_0) v_0 \\
&= \sum_1^m \beta_i v_i + \beta_0 v_0 = 0,
\end{aligned}$$

and therefore V is affinely dependent. □

Corollary 1.6. *The ordering of the vectors $\{v_i\}_{i=0}^m$ in Definition 1.1 does not affect the affine independence and span of the set.*

In particular, if σ is a permutation on $\{0, 1, \ldots, m\}$ and we define $V = \{v_i\}_{i=0}^m$ and $V_\sigma = \{v_{\sigma(i)}\}_{i=0}^m$, then:

i) V is affinely independent if and only if V_σ is affinely independent,

ii) $\mathrm{AffSpan}(V) = \mathrm{AffSpan}(V_\sigma)$.

Proof. Let $V' = \{(v_i, 1)^T\}_{i=0}^m$ and $V_\sigma' = \{(v_{\sigma(i)}, 1)^T\}_{i=0}^m$. By Lemma 1.5, the set V is affinely independent if and only if V' is linearly independent. But V' is linearly independent whenever V_σ' is linearly independent, which is the case whenever V_σ is affinely independent. This proves (i).

Furthermore, by Lemma 1.4, $u \in \mathrm{AffSpan}(V)$ if and only if $(u, 1)^T \in \langle V' \rangle = \langle V_\sigma' \rangle$, and $u \in \langle V_\sigma' \rangle$ if and only if $u \in \mathrm{AffSpan}(V_\sigma)$. Thus (ii) holds as well. □

As described above, the affine span of $m+1$ affinely independent vectors in $\mathbb{R}^d$ can be considered to be an m-dimensional subspace of $\mathbb{R}^d$ shifted by translation. However, we will often consider such translated linear subspaces without referring to a generating set of vectors. In essence, we are concerned

with *affine subspaces*; we present a narrow definition of these spaces here, but remark that affine spaces may be defined more generally. For a more general discussion of affine spaces, see [44].

Definition 1.7. An *affine subspace* A of $\mathbb{R}^d$ is a subset of the form

$$A = p + W = \{p + w : w \in W\}, \tag{1.2}$$

where W is a linear subspace of $\mathbb{R}^d$. The *dimension* of A is equal to the dimension of W.

Clearly, the affine span of any collection of vectors forms an affine subspace, and in particular, the affine span of a collection of affinely dependent vectors is still an affine subspace. If $V = \{v_i\}_{i=0}^m$ is a set of vectors, then $\mathrm{AffSpan}(V)$ will be a shifted l-dimensional subspace of $\mathbb{R}^d$, where $l \leq m$. Often, we will be concerned more with the dimensionality of an affine span, and less with the number of vectors that generated it. This motivates the next definition.

Definition 1.8. We say that the *affine dimension* of a collection of vectors V is the dimension of the corresponding affine subspace $\mathrm{AffSpan}(V)$.

Definition 1.9 (Convexity)**.** Let $R = \{p_1, \ldots, p_n\}$ be a finite collection of points in $\mathbb{R}^d$. A *convex combination* p' of the points in R is a point of the form

$$p' = \sum_{i=1}^{n} \alpha_i p_i, \quad \text{where } \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{n} \alpha_i = 1.$$

Furthermore, the *convex hull* of a set $S \subset \mathbb{R}^d$ (possibly infinite) is the set of all convex combinations of points in S. We shall use the operator notation Conv to denote convex hulls. Therefore, we write

$$\mathrm{Conv}(S) = \{p \in \mathbb{R}^d : p \text{ is a convex combination of points in } S\}.$$

At times it will also be helpful to consider the convex hull of a union of sets S_j. For convenience of notation, in this case we shall use the convention

$$\mathrm{Conv}(S_1, S_2, \ldots, S_m) := \mathrm{Conv}\left(\bigcup_{j=1}^{m} S_j\right).$$

Finally, when $S = \mathrm{Conv}(S)$, we say that S is a *convex set*.

We remark that a convex combination of vectors is a special case of an affine combination where all coefficients α_i are non-negative. One effect of this is that the convex hull of finitely many points is bounded in $\mathbb{R}^d$, whereas affine hulls are in general unbounded.

The following lemma is a useful tool for describing points in the union of two convex sets, and will be used later in the dissertation.

Lemma 1.10. *If $S_1, S_2 \subset \mathbb{R}^d$ are two convex sets and $p \in \mathrm{Conv}(S_1, S_2)$, then $p = \alpha p_1 + (1-\alpha)p_2$, where $0 \leq \alpha \leq 1$ and $p_1 \in S_1$ while $p_2 \in S_2$.*

Proof. Since $p \in \mathrm{Conv}(S_1, S_2)$, p is a convex combination of some collection of points in $S_1 \cup S_2$. Without loss of generality we may write

$$p = \sum_{i=1}^{n_1} \beta_i q_i + \sum_{j=1}^{n_2} \gamma_j q_j', \tag{1.3}$$

where each $q_i \in S_1$, each $q_j' \notin S_1$, the β_i and γ_i are non-negative, and $\sum \beta_i + \sum \gamma_j = 1$. Next, let $B = \sum \beta_i$ and $C = \sum \gamma_j$. If either B or C is equal to 0, then immediately the lemma holds with $\alpha = 0$ or $\alpha = 1$, respectively. Thus, for the remainder of this proof we shall assume B and C are nonzero.

Factoring out these terms from Equation 1.3, we obtain

$$p = B \sum_{i=1}^{n_1} \frac{\beta_i}{B} q_i + C \sum_{j=1}^{n_2} \frac{\gamma_j}{C} q_j'. \tag{1.4}$$

By the definition of B and C, the terms

$$\sum_{i=1}^{n_1} \frac{\beta_i}{B} q_i \quad \text{and} \quad \sum_{j=1}^{n_2} \frac{\gamma_j}{C} q_j'$$

are convex combinations of points in S_1 and S_2, respectively. Since S_1 and S_2 are convex sets, these two terms must belong to S_1 and S_2. The lemma follows upon observing that $B + C = 1$ and $0 \leq B \leq 1$. □

Next, we move on to a discussion of the geometry of convex sets. Of particular importance to this dissertation will be convex sets which are the convex hull of finitely many points. This is a broad class of objects, which include line segments, convex polygons, and convex solids. In order to carry out a discussion of four-dimensional geometry, it is necessary to establish a dimension-independent language which unifies geometric objects of this type. The fundamental building block of this analysis is the polytope.

Definition 1.11. A *convex polytope* is the convex hull of finitely many points. If a polytope P is the convex hull of points $\{p_1, \ldots, p_m\}$ and the affine span of these points has dimension k, then we say that P is a *polytope of dimension k*. Throughout this dissertation we shall always take "polytope" to mean "convex polytope."

Remark 1.12. There are several widespread definitions for convex polytopes, which are more or less equivalent. However, some defnitions allow polytopes to be unbounded while others do not. By our definition, all polytopes are necessarily compact (and in particular, bounded). In other sources, such an object may be referred to as a "compact convex polytope."

Many common shapes are examples of polytopes. For instance, all convex polygons are polytopes of dimension 2. Cubes, pyramids, and triangular prisms are all polytopes of dimension 3. A singular point is a polytope of dimension 0. In addition, the empty set is often considered to be a polytope of dimension -1.

Since all polytopes are defined as the convex hull of some finite set of points, these points are of central importance to any analysis involving polytopes. However, multiple sets of points can generate the same polytope. For instance, the same rectangle in $\mathbb{R}^2$ is generated by $\{(0,0),(1,0),(1,1),(0,1)\}$ and $\{(0,0),(1,0),(1,1),(0,1),(0.5,0.5)\}$. In order to associate a polytope with a unique generating set of points, we introduce the notion of extremal points.

Definition 1.13. Let P be a convex set. The point p is an *extremal point* of P if $p \in P$ and there is no open line segment contained in P which contains p. Equivalently, whenever p is an extremal point of P, if $p = \alpha p_1 + (1-\alpha)p_2$ where $p_1, p_2 \in P$ and $0 \leq \alpha \leq 1$, then $p = p_1 = p_2$.

Remark 1.14. When P is a polytope, we shall refer to its extremal points as *vertices*. The intuitive understanding of vertices as "corners" of a shape coincides with this definition. We shall sometimes use the phrase *vertex set* to refer to the set of all extremal points of a polytope.

In higher dimensions, it will be especially handy to study and manipulate polytopes in terms of their vertex sets. In order to easily identify a polytope with its vertices, we establish the following notation.

Definition 1.15 (Vertex Representation of Polytopes)**.** Suppose S is a convex polytope with vertices

$p_1, \ldots, p_k$. We denote the convex hull of $\{p_1, \ldots, p_k\}$, which is equal to S, by

$$S = \{\{p_1, \ldots, p_k\}\} := \mathrm{Conv}\left(\{p_1, \ldots, p_k\}\right).$$

The final geometric object that we shall introduce in this section is arguably the most important for the purposes of space-time finite element methods.

Definition 1.16. A *k-simplex* (or a *simplex of dimension k*) is defined to be the convex hull of $k+1$ affinely independent points in Euclidean space. The convex hull of $k+1$ points which are *not* affinely independent will be described as a *degenerate k-simplex*. Throughout this dissertation, the term "simplex" will be taken to mean "non-degenerate simplex" unless otherwise stated.

Remark 1.17. A k-simplex may alternatively be characterized as a polytope of dimension k with $k+1$ vertices.

For example, a 0-simplex is a point and a 1-simplex is a line segment. In $\mathbb{R}^2$, a 2-simplex is a triangle, and in $\mathbb{R}^3$ a 3-simplex is a tetrahedron. As a further example, any collection of $k+2$ points in $\mathbb{R}^k$ will always form a degenerate simplex, since $k+2$ points in $\mathbb{R}^k$ are always affinely dependent.

One useful property of simplices is that the boundary of any k-simplex can be decomposed into sets which are each l-simplices, where $l < k$. For instance, the boundary of a triangle (a 2-simplex) can be decomposed into three line segments (each a 1-simplex). This property also holds on a more general level for polytopes: the boundary of any polytope of dimension k can be decomposed into a collection of polytopes of dimension l, where $l < k$. However, in the case of polytopes, the structure of these lower-dimensional polytopes is not always easy to deduce. In contrast, every l-simplex has the same essential topology, which makes the boundary structure of simplices easy to analyze.

Describing the boundaries of simplices can be made more precise with the general definition of simplex faces.

Definition 1.18. Let $K = \{s_1, \ldots, s_{k+1}\}$ be a k-simplex. A *j-face* of K is the convex hull of $j+1$ of the vertices of K. As a convention, we define the empty set $\varnothing$ to be a (-1)-face of any simplex.

Note that by definition, every j-face of a simplex is itself a simplex. In addition, the number of j-faces of a k-simplex is simply the number of unique combinations of $j+1$ out of $k+1$ points; therefore, the number of j-faces of a k-simplex is $\binom{k+1}{j+1}$ (for $-1 \leq j \leq k+1$).

1.1.2 Triangulations

A central component of space-time finite element methods is the discretization of a space-time domain into elements. As previously discussed, this dissertation considers space-time methods over conforming, simplicial meshes. The use of conforming triangular or tetrahedral meshes in two- or three-dimensional finite element methods is widespread, but their generalization to four dimensions is neither widely studied nor implemented.

At a very basic level, triangular and tetrahedral meshes are collections of simplices with extra conditions imposed. We may generalize these conditions to set an appropriate definition for a mesh of k-simplices. Then, in four-dimensional space-time, the appropriate space-time discretization will be a 4-simplex mesh.

Definition 1.19. A *triangulation of* $Q \subset \mathbb{R}^d$ is a collection of d-simplices $\mathcal{T} = \{\tau_i\}_{i=1}^n$ covering Q such that the intersection of any two d-simplices is a common j-face for both simplices, with $-1 \leq j \leq d$. In other words, the set $\mathcal{T}$ is a collection of d-simplices satisfying:

i) $\bigcup_{i=1}^{n} \tau_i = Q$,

ii) If $\lambda = \tau_i \cap \tau_j$, then λ is a face of both τ_i and τ_j.

Note that since $\varnothing$ is a face of every simplex, property (ii) holds even for disjoint simplices.

Many standard properties of triangular and tetrahedral meshes can be naturally extended to four dimensions. Let $\mathcal{T} = \{\tau_i\}_{i=1}^n$ be a triangulation of d-simplices, and let τ be an arbitrary simplex element in $\mathcal{T}$.

The *diameter* of τ is defined to be

$$h_\tau = \max_{x,y \in \tau} |x - y|; \tag{1.5}$$

that is, the largest distance between two points in τ. In general, we say that the "size" of the element τ is its diameter, unless another measure of size is specifically mentioned. Furthermore, the size of the largest element in $\mathcal{T}$ is denoted

$$h = \max_{\tau \in \mathcal{T}} h_\tau. \tag{1.6}$$

Another important measure of size, which better captures the d-dimensional volume of τ, is the *largest contained radius* parameter. This parameter is defined as

$$\rho_\tau = \sup\{r \,:\, \tau \text{ contains a } d\text{-ball of radius } r\}. \tag{1.7}$$

Following the nomenclature of [9], the *chunkiness parameter* of τ is defined to be

$$\sigma_\tau = \frac{h_\tau}{\rho_\tau}. \tag{1.8}$$

In line with the typical definition for two- and three-dimensional problems, the d-simplex mesh $\mathcal{T}$ is said to be *shape regular* if there exists some $\sigma > 0$ such that $\sigma_\tau \leq \sigma$ for all $\tau \in \mathcal{T}$.

Now, suppose we have a countable collection of triangulations $\mathcal{T}_s$, parameterized by a real number $s \to 0$. We say that the family $\{\mathcal{T}_s\}$ is *shape regular* if there exists a $\sigma > 0$ such that

$$\sup_s \sup_{\tau \in \mathcal{T}_s} \sigma_\tau \leq \sigma. \tag{1.9}$$

The family $\{\mathcal{T}_s\}$ is said to be *quasi-uniform* if there exists some $\nu > 0$ such that for all s and all $\tau \in \mathcal{T}_s$,

$$h_\tau \geq \nu \cdot \sup_s \sup_{\tau \in \mathcal{T}_s} h_\tau. \tag{1.10}$$

1.1.3 Reference Elements

Many operations on simplices are unaffected by the precise location of the simplex in $\mathbb{R}^d$. For instance, a procedure to subdivide a simplex is often performed in the same way no matter where that simplex lies in the domain. To simplify our analysis, we will define operations on a "reference simplex" with simple geometry, and then extend that definition to arbitrary simplices through the use of affine transformations. An *affine transformation* on $\mathbb{R}^d$ is map of the form $Fx = Bx + b$, where B is a linear map on $\mathbb{R}^d$ and $b \in \mathbb{R}^d$. In this section, we will explore some of the properties of affine transformations that transform an arbitrary simplex into the reference simplex, and vice-versa.

Let $K = \{\{v_0, v_1, \ldots, v_d\}\}$ be a d-simplex in $\mathbb{R}^d$ (c.f. Definition 1.15 for a description of $\{\{\cdot\}\}$ notation). Denote by $\widehat{K}$ the canonical d-simplex; i.e., the convex hull of $\{0, e_1, e_2, \ldots, e_d\}$. In addition, we

define the *canonical simplex vertices*:

$$\hat{v}_j = \begin{cases} 0 & \text{if } j = 0 \\ e_j & \text{if } j > 0 \end{cases} \tag{1.11}$$

Next, let F_K be an affine map such that $F_K(\widehat{K}) = K$; note that for any K there are multiple distinct choices for F_K. The *Barycentric Transformation Matrix* for K is the $(d+1)\times(d+1)$ matrix with columns partially defined by its vertices $v_0, \dots, v_d$; specifically,

$$A_K = \begin{pmatrix} | & | & & | \\ v_0 & v_1 & \dots & v_d \\ | & | & & | \\ 1 & 1 & & 1 \end{pmatrix}.$$

By the definition of the d-simplex, every point $p \in K$ is a convex combination of the vertices of K. Since $K = \{\{v_0, v_1, \dots, v_d\}\}$, there must be a set of scalars λ_i, $0 \le i \le d$, such that

$$p = \sum_{i=0}^{d} \lambda_i v_i, \qquad \text{where } \sum_{i=0}^{d} \lambda_i = 1 \text{ and } \lambda_i \ge 0 \text{ for all } 0 \le i \le d.$$

The tuple $\lambda = (\lambda_0, \lambda_1, \dots, \lambda_d)^T$ is called the *barycentric coordinates* of p in the simplex K. Essentially, λ describes the relative location of p within K: the larger the value of λ_i, the closer p is to v_i. In particular, if $\lambda_i = 1$ and $\lambda_j = 0$ for all $j \ne i$, then $p = v_i$. If all $v_i = 1/(d+1)$, then p is the barycenter of K.

The aptly-named barycentric transformation matrix directly relates the Cartesian coordinates of a point to its barycentric coordinates. By construction, one may verify that if λ is the barycentric coordinates of p in the simplex K, then

$$A_K\lambda = \begin{pmatrix} | & | & & | \\ v_0 & v_1 & \dots & v_d \\ | & | & & | \\ 1 & 1 & & 1 \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_d \end{pmatrix} = \begin{pmatrix} p \\ 1 \end{pmatrix}.$$

Furthermore, the representation of a point in barycentric coordinates within a simplex K is

unique (provided K is non-degenerate, which we assume throughout). Thus we may consider barycentric coordinates to be an equivalent local coordinate system within each simplex. Since affine transformations between finite elements and the reference element are defined in terms of element vertices, properties of these transformations may be proved more naturally in the setting of barycentric coordinates.

We now establish some fundamental properties of affine transformations and maps between simplex elements. Our goal is to show that every affine bijection from a simplex element K to the reference element $\widehat{K}$ has a known structure that is defined solely in terms of the vertices of K. Having established this structure, we may prove several important properties about affine maps to the reference element. To begin, we prove that the barycentric representation of a point is, indeed, unique.

Lemma 1.20. *The simplex K is non-degenerate if and only if A_K is non-singular.*

Proof. By definition, the simplex K is non-degenerate if and only if the vectors in $\{v_0, \dots, v_d\}$ are affinely independent. By Lemma 1.5, the set $\{v_0, \dots, v_d\}$ is affinely independent if and only if the associated set $\{(v_0, 1)^T, \dots, (v_d, 1)^T\}$ is linearly independent, which is the case precisely when A_K is non-singular. □

The following series of lemmas provides an explicit characterization of the affine bijections from the reference simplex $\widehat{K}$ to another simplex K.

Lemma 1.21. *If the map $F_K : \widehat{K} \to K$ is an affine bijection, then $F_K(\hat{v}_j) = v_{\sigma(j)}$ for $0 \le j \le d$, where σ is a permutation on $\{0, 1, \dots, d\}$.*

Proof. Let $x \in K$ be arbitrary and $\hat{x} = F_K^{-1}(x)$. Then for some barycentric coordinates λ_i, $\hat{x} = \sum_0^d \lambda_i \hat{v}_i$ and $\sum_0^d \lambda_i = 1$. Now since F_K is affine, there is some linear transformation B and vector b such that $F_K(\hat{x}) = B\hat{x} + b$. Thus

$$x = F_K(\hat{x}) = B\hat{x} + b = \sum_{i=0}^{d} (\lambda_i B\hat{v}_i) + b = \sum_{i=0}^{d} (\lambda_i B\hat{v}_i) + \left(\sum_{i=0}^{d} \lambda_i\right) b = \sum_{i=0}^{d} \lambda_i (B\hat{v}_i + b) = \sum_{i=0}^{d} \lambda_i F_K(\hat{v}_i)$$

and therefore x is a convex combination of $\{F_K(\hat{v}_i)\}_{i=0}^d$. Since x was arbitrary, this implies that every point in K is a convex combination of $\{F_K(\hat{v}_i)\}_{i=0}^d$. Now, since K was the simplex defined to be the convex hull of $\{v_i\}_{i=0}^d$, these two sets must be equal up reordering. Thus for all $0 \le i \le d$, $F_K(\hat{v}_j) = v_{\sigma(j)}$, where σ is a permutation on $\{0, 1, \dots, d\}$. □

Lemma 1.22. *Let K be a d-simplex with vertices given by $\{v_0, v_1, \dots, v_d\}$. For any permutation σ on the set $\{0, 1, \dots, d\}$, the map*

$$F_{K,\sigma}(\hat{x}) = \begin{pmatrix} | & | & & | \\ | & | & & | \\ v_{\sigma(1)} - v_{\sigma(0)} & v_{\sigma(2)} - v_{\sigma(0)} & \cdots & v_{\sigma(d)} - v_{\sigma(0)} \\ | & | & & | \\ | & | & & | \end{pmatrix} \hat{x} + v_{\sigma(0)}. \tag{1.12}$$

is a bijection from $\widehat{K}$ to K and $F_{K,\sigma}(\hat{v}_j) = v_{\sigma(j)}$ for $0 \leq j \leq d$.

Proof. The linear transformation in the definition of $F_{K,\sigma}$ is non-singular by Lemma 1.5, so $F_{K,\sigma}$ is bijective. To see that $F_{K,\sigma}(\widehat{K}) = K$, consider an arbitrary element $x = \sum_0^d \lambda_i v_i \in K$, where $\sum_0^d \lambda_i = 1$. Then

$$\begin{aligned} F_{K,\sigma}\left((\lambda_{\sigma(1)}, \dots, \lambda_{\sigma(d)})^T\right) &= v_{\sigma(0)} + \sum_{i=1}^{d} \lambda_{\sigma(i)}(v_{\sigma(i)} - v_{\sigma(0)}) \\ &= v_{\sigma(0)} - \sum_{i=1}^{d} \lambda_{\sigma(i)} v_{\sigma(0)} + \sum_{i=1}^{d} \lambda_{\sigma(i)} v_{\sigma(i)} \\ &= \left(1 - \sum_{i=1}^{d} \lambda_{\sigma(i)}\right) v_{\sigma(0)} + \sum_{i=1}^{d} \lambda_{\sigma(i)} v_{\sigma(i)} \\ &= \sum_{i=0}^{d} \lambda_{\sigma(i)} v_{\sigma(i)} \\ &= x. \end{aligned}$$

Finally, we note that since $\hat{v}_i = e_i$ and $\hat{v}_0 = 0$, $(\lambda_{\sigma(1)}, \dots, \lambda_{\sigma(d)})^T = \sum_{i=0}^d \lambda_{\sigma(i)} \hat{v}_i \in \widehat{K}$. Thus every element in K is the image under $F_{K,\sigma}$ of an element in $\widehat{K}$, and therefore $F_{K,\sigma}(\widehat{K}) = K$. The statement that $F_{K,\sigma}(\hat{v}_j) = v_{\sigma(j)}$ follows directly from the definition of $F_{K,\sigma}$ and $\hat{v}_j$. □

Lemma 1.23. *Every bijection from $\widehat{K}$ to K is of the form $F_{K,\sigma}$ for some permutation σ on the set $\{0, 1, \dots, d\}$.*

Proof. To begin, we show that every affine bijection of simplices is completely determined by the image of each of the vertices. Let $F : \widehat{K} \to K$ be an affine bijection and let $\hat{x} \in \widehat{K}$ be arbitrary. Since F is affine, $F(\hat{x}) = B\hat{x} + b$ for some matrix B and vector b. Since $\hat{x} \in \widehat{K}$, $\hat{x} = \sum_{j=0}^d \lambda_j \hat{v}_j$ with $\sum_{j=0}^d \lambda_j = 1$ and $\lambda_j \geq 0$

for $0 \leq j \leq d$. Then

$$F(\hat{x}) = F\left(\sum_{j=0}^{d} \lambda_j \hat{v}_j\right) = B\left(\sum_{j=0}^{d} \lambda_j \hat{v}_j\right) + b = \sum_{j=0}^{d} \lambda_j B\hat{v}_j + \sum_{j=0}^{d} \lambda_j b = \sum_{j=0}^{d} \lambda_j \left(B\hat{v}_j + b\right) = \sum_{j=0}^{d} \lambda_j F(\hat{v}_j).$$

Thus $F(\hat{x})$ is determined completely by the images $F(\hat{v}_j)$. By Lemma 1.21, the fact that an affine bijection of simplices is determined by its image on the vertices implies that there are at most $(d+1)!$ such bijections. Lemma 1.22 establishes the existence of $(d+1)!$ of these maps (all distinct). The distinctness of each $F_{K,\sigma}$ is obvious from the fact that the ordered set $\{F_{K,\sigma}(\hat{v}_0), \dots, F_{K,\sigma}(\hat{v}_d)\}$ is unique for each σ. Therefore, every affine bijection from $\widehat{K}$ to K must be of the form $F_{K,\sigma}$ for some σ a permutation on $\{0, \dots, d\}$. □

Lemma 1.23 describes explicitly the structure of any affine bijection between a simplex and the reference simplex. Since all $(d+1)!$ bijections are essentially equivalent up to a reordering of the vertices, the magnitude of the determinant of these maps must be invariant.

Proposition 1.24. *Let $F_K : \widehat{K} \to K$ be an affine bijection of d-simplices. Then* $|\det(\nabla F_K)| = d! \cdot \mathrm{vol}(K)$, *where* $\mathrm{vol}(\cdot)$ *denotes d-dimensional volume.*

Proof. By Lemma 1.23, $F_K = F_{K,\sigma}$ for some permutation σ. In addition,

$$\nabla F_{K,\sigma} = (v_{\sigma(1)} - v_{\sigma(0)}, \dots, v_{\sigma(d)} - v_{\sigma(0)}),$$

and hence

$$|\det(\nabla F_K)| = \left|\det(\nabla F_{K,\sigma})\right| = \left|\det(v_{\sigma(1)} - v_{\sigma(0)}, \dots, v_{\sigma(d)} - v_{\sigma(0)})\right|.$$

The volume of the d-simplex $K = \{v_0, \dots, v_d\}$ is given by

$$\mathrm{vol}(K) = \left|\frac{1}{d!} \det\left(v_1 - v_0, \dots, v_d - v_0\right)\right|,$$

and since the volume of K is unrelated to the ordering of the vertices $\{v_0, \dots, v_d\}$, it must be the case that

$$\mathrm{vol}(K) = \left|\frac{1}{d!} \det\left(v_{\sigma(1)} - v_{\sigma(0)}, \dots, v_{\sigma(d)} - v_{\sigma(0)}\right)\right|$$

as well. Thus $|\det(\nabla F_K)| = d! \cdot \mathrm{vol}(K)$. □

In the implementation of finite element methods, the quantity $|\det(\nabla F_K)|$ is frequently used for the efficient integration of various functions over finite elements. Proposition 1.24 shows that this quantity can be computed using any ordering on the vertices of K. In particular, this mean that even though each bijection is defined in terms of an ordering on the vertices, the quantity $|\det(\nabla F_K)|$ is invariant with respect to this ordering.

The maps $F_{K,\sigma}$ may also be used to estimate the shape regularity of individual elements, which has a direct impact on the accuracy of a finite element method. We recall from the previous section on Triangulations that the shape regularity of an element τ may be described in terms of its chunkiness parameter $\sigma_\tau = h_\tau/\rho_\tau$, where h_τ is the diameter of τ and ρ_τ is the radius of the largest ball contained in τ.

Let us consider an affine bijection $F_{K,\sigma} : \widehat{K} \to K$, where $F_{K,\sigma}\hat{x} = B\hat{x} + b$, and let $\widehat{h}$ and $\hat{\rho}$ be the diameter and largest contained radius of the reference simplex, respectively. It may be shown directly from the definitions of the parameters h and ρ that

$$\|B\| \leq \frac{h_\tau}{\hat{\rho}} \qquad \text{and} \qquad \left\|B^{-1}\right\| \leq \frac{\widehat{h}}{\rho_\tau},$$

where here $\|\cdot\|$ denotes the operator norm (see [12], Theorem 15.2, for a proof). By multiplying these two inequalities together, we have

$$\|B\|\left\|B^{-1}\right\| \leq \frac{\widehat{h}}{\hat{\rho}} \cdot \frac{h_\tau}{\rho_\tau} = C \cdot \sigma_\tau.$$

Thus the chunkiness parameter of τ increases along with the condition number $\|B\|\|B^{-1}\|$; as such, one may use the condition number of the reference mapping $F_{K,\sigma}$ as an estimate for the shape regularity of τ. In fact, one may even *define* the shape regularity of an element to be the condition number of the reference mapping.

1.1.4 Useful Inequalities

There are a number of inequalities that find applicability in the analysis of finite element methods. Due to their widespread use, however, many authors formulate slight variants for ease of use in their area of interest. We shall state a few such inequalities in this section for use later in the disserta-

tion.

Lemma 1.25. *Let $\{\mathcal{T}_s\}$ be a shape-regular family of triangulations, where σ is the uniform bound on shape regularity defined in Equation 1.9. Then for any $\mathcal{T} \in \{\mathcal{T}_s\}$, $w \in H^1(\mathcal{T})$, $\tau \in \mathcal{T}$, and any $(d-1)$-face F of τ,*

$$\|w\|^2_{L^2(F)} \leq \frac{1}{\sigma}\left(2\|\nabla w\|_{L^2(\tau)}\|w\|_{L^2(\tau)} + \frac{d}{h_\tau}\|w\|^2_{L^2(\tau)}\right) \tag{1.13}$$

and in particular,

$$\|w\|^2_{L^2(F)} \leq \frac{1}{\sigma}h_\tau\|\nabla w\|^2_{L^2(\tau)} + \frac{1+d}{\sigma}h_\tau^{-1}\|w\|^2_{L^2(\tau)}. \tag{1.14}$$

Proof. A proof of Equation 1.13 may be found in [15], Lemma 1.49. Equation 1.14 follows from Equation 1.13 since

$$2\|\nabla w\|_{L^2(\tau)}\|w\|_{L^2(\tau)} \leq h\|\nabla w\|^2_{L^2(\tau)} + h^{-1}\|w\|^2_{L^2(\tau)}$$

by Young's inequality with epsilon. □

1.2 A Model Problem

Throughout this dissertation, we will often return to the time-evolving convection-diffusion equation as a model problem for studying the behavior of space-time finite element methods. This equation is highly studied in the finite element literature and can exhibit behavior which ranges from straightforward to sophisticated, depending on how the problem is set up. For the purposes of this dissertation, the convection-diffusion equation is complex enough to illustrate the main components of space-time finite element methods, but simple enough to be analyzed without studying the PDE too closely. Our focus here is the behavior of space-time methods; as such, we would like to avoid or de-emphasize the particularities of any specific PDE as much as possible.

We define the time-dependent convection-diffusion equation to be:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \nabla_x \cdot \left(\tilde{D}\nabla_x u - \tilde{b}u\right) + cu = f & \text{for } x \in \Omega, t \in (0,T) \\ u = 0 & \text{for } x \in \partial\Omega \\ u = u_0 & \text{for } t = 0 \end{cases} \tag{1.15}$$

where $\Omega \subset \mathbb{R}^d$ is a domain with boundary $\partial\Omega$, and $[0,T]$ is the time interval of interest. The function

$\widetilde{D} = \widetilde{D}(x,t)$ is a smoothly varying matrix-valued function, which is assumed to be uniformly positive definite over all x and t; that is, there exists a constant $\varkappa > 0$ such that

$$\xi^T wtD(x,y)\xi > \varkappa|\xi|^2 \qquad \text{for all } \xi \in \mathbb{R}^d \text{ and all } x \in \Omega,\ t \in [0,T].$$

The function $\tilde{b} = \tilde{b}(x,t)$ is a smooth vector field, $f = f(x,t)$ and $c = c(x,t)$ are scalar fields, and $u_0 = u_0(x)$ is the initial condition of the PDE. We also assume that there is some constant $\beta > 0$ such that for all x and t, $c(x,t) + \frac{1}{2}\nabla \cdot b(x,t) \geq \beta$.

Equation 1.15 may be viewed as a generalization of the heat equation

$$\frac{\partial u}{\partial t} - \Delta u = f,$$

where the Laplacian Δ is replaced by a general second order linear elliptic operator $\mathcal{A}$:

$$\frac{\partial u}{\partial t} - \mathcal{A}u = f. \tag{1.16}$$

It can be shown that every second order linear elliptic operator can be written in the form $\mathcal{A}u = \nabla_x \cdot (\widetilde{D}\nabla_x u - bu) + cu$ for appropriately chosen $\widetilde{D}$, $\tilde{b}$, and c (see, e.g., [18]). Furthermore, Equation 1.16 is the general form for any second-order linear parabolic PDE. Therefore, by studying arbitrary convection-diffusion equations, we are actually considering second-order linear parabolic PDEs in the general sense.

The convection-diffusion equation also models a number of physical phenomena, with the functions $\widetilde{D}$, $\tilde{b}$, c, and f all holding physical significance. By choosing these functions in certain ways, one may model processes ranging from semiconductor current density, to heat transfer through a medium, to population migration of a biological species. Due to their connection to physical processes, it is often helpful to think about these functions in relation to the phenomena they can model.

Generally speaking, the convection-diffusion equation describes the evolution of some physical quantity in space. The value of $u(x,t)$ represents the value of this quantity (i.e. temperature, pressure, density) at location x and time t. The three coefficient functions $\widetilde{D}$, $\tilde{b}$, and c dictate the ways in which the quantity u changes with x and t.

The function $\widetilde{D}$ is called the *diffusion coefficient* and describes the movement of a substance

from regions of high concentration to regions of low concentration. For instance, pouring a spoonful of salt into a glass of water will cause the new salt ions to move through the liquid in a diffusive process: ions will disperse rapidly where the change in salt concentration is high, but will disperse slowly as the concentration of salt ions approaches uniformity throughout the cup.

The function $\tilde{b}$ is called the *convection coefficient* (or *advection coefficient*)[1]. This term describes the underlying motion of the medium carrying the quantity u. In semiconductor physics, for example, the function u may represent charge density, which is mediated by the movement of charged particles (electrons and "holes"). In this model, the convection term models an ambient electric field, which imparts directed forces on the charge carriers throughout the device. Alternatively, if u describes the concentration of silt in a river, the flow of the river would be encoded by the convection coefficient $\tilde{b}$.

The function c is the *reaction coefficient*, which is used for modeling processes where the change in the quantity u is affected by the value of that quantity at a particular point in space and time. This behavior is common when chemical reactions are a component of the process being modeled. For instance, in applications where u measures the temperature of a solution during a chemical reaction, higher temperatures may spur additional reactions which generate or consume heat.

Finally, the function f is called the *forcing function* and can take on a number of meanings. In many of the situations we have just described, f can represent sources or sinks of the quantity u. When u models heat, f might describe a heat source. When u measures particle concentration in a liquid, a forcing function might model the effects of a filter in portions of the domain.

In the following chapters, we will not concern ourselves with any particular application of the convection-diffusion equation. Instead, our focus will be on the ways in which the different terms of Equation 1.15 affect the behavior of space-time finite element methods applied to this problem.

[1]Formally speaking, convective transport and advective transport refer to two different phenomena, depending on the context. Advective transport refers to the directional, bulk movements of a medium (like river currents). Convection is a more general term, encompassing any type of bulk movement (for instance, the undirected movement of fluid due to thermal gradients). This distinction between modes of transport is outside the scope of this dissertation, so we will use the more general term (convection).

Chapter 2

Space-Time Finite Element Methods for Linear Parabolic PDEs

We present a continuous space-time finite element method for the d-dimensional transient convection-diffusion equation, with weak assumptions on the regularity of the problem data and solution. The numerical stability of the method is proven and asymptotic convergence rates are derived which are in agreement with existing related literature. In particular, we prove convergence so long as the solution has at least $1+\epsilon$ weak derivatives, where $\epsilon > 0$. Numerical experiments are also conducted which verify the theoretical convergence rates in three- and four-dimensional space-time domains. To the best of our knowledge, this is the first analysis of unstructured space-time finite element methods applied to general linear parabolic equations.

Our approach is directly influenced by Bank, Vassilevski, and Zikatanov[4], who proved basic properties of upwinded space-time methods for constant-coefficient parabolic equations. Another notable influence is the work of Langer, Neumüller, and Schafelner, who apply an element-wise upwinding scheme to study low-regularity solutions of the heat equation in [27–29].

In contrast to these earlier results, here we shall treat parabolic equations where the second-order term is a general elliptic operator. That is, we consider second-order terms of the form

$$\sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial}{\partial x_j}u\right) \qquad \text{where} \quad \kappa|\xi|^2 \le \sum_{i,j} D_{ij}(x,t)\xi_i\xi_j \le \delta|\xi|^2 \quad \text{for any } \xi \in \mathbb{R}^d, \text{ where } \kappa,\delta > 0.$$

In all of the previously cited literature on Galerkin space-time methods, the second-order term is always taken to be a scalar field, often a piecewise-constant function. Existing results for these special cases corroborate the more general results which we present here. For instance, when $D_{ij} = \nu\delta_{ij}$ is some piecewise-constant diagonal operator, our theory aligns with that of [29].

Furthermore, we extend the work of Langer et. al. to problems with non-autonomous spatial convection and reaction terms. The proposed method utilizes a space-time upwinding term, which provides a natural mechanism to handle cases where the spatial convection dominates the spatial diffusion.

In addition, we present convergence results for problems in which the problem data is somewhat nonsmooth. For example, when modeling the dispersion of a species through a heterogeneous material, the material properties which define the PDE may be discontinuous across (potentially moving) interfaces, but otherwise smooth. Problems of this type are included in the present analysis. Inspired by [27], we will assume that a coefficient function is regular within each element of the mesh but place limited restrictions on the smoothness across element boundaries. Given a function $c : Q \to \mathbb{R}$ and a triangulation $\mathcal{T}$, we say that

$$c \in H^k(\mathcal{T}) \quad \text{if and only if} \quad c \in H^k(\tau) \quad \text{for all } \tau \in \mathcal{T}. \tag{2.1}$$

Then, if $\mathcal{T}$ is chosen such that no mesh element crosses a material interface, c will have smoothness $H^k(\mathcal{T})$. Of course, when the problem data is nonsmooth across some interface, we generally expect a solution u to have reduced smoothness across these interfaces as well. Therefore, we treat solutions of the class $H^m(Q) \cap H^k(\mathcal{T})$, where $m < k$. In this scenario, u has some low level of global regularity, but potentially much higher regularity within mesh elements.

2.1 The Basic Setting of Space-Time Methods

At the core of the space-time formulation of a PDE is the space-time variable $y = (x, t)$. Once this variable is defined, the associated language of norms, derivatives, boundary conditions, and so on must be adjusted as well.

Written in terms of separate space and time variables, the model convection-diffusion equation

is

$$\begin{cases} \mathcal{L}u := \dfrac{\partial u}{\partial t} - \nabla_x \cdot \left(\widetilde{D}\nabla_x u - \tilde{b}u\right) + cu = f & \text{for } x \in \Omega, t \in (0,T) \\ u = 0 & \text{for } x \in \partial\Omega \\ u = u_0 & \text{for } t = 0 \end{cases} \tag{2.2}$$

Now, we define $y = (x,t) \in \mathbb{R}^{d+1}$ and may re-write Equation 2.2 in terms of derivatives with respect to y. Let ∇ and $\nabla\cdot$ denote the gradient and divergence operators with respect to the space-time variable. Then the model problem may be written as

$$\begin{cases} \mathcal{L}u = -\nabla \cdot (D\nabla u - bu) + cu = f & \text{for } y \in \Omega \times (0,T) \\ u = 0 & \text{for } y \in \partial\Omega \times (0,T) \\ u = u_0 & \text{for } y \in \Omega \times \{0\} \end{cases} \tag{2.3}$$

where

$$D = \begin{pmatrix} \widetilde{D} & 0^{d\times 1} \\ 0^{1\times d} & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \qquad \text{and} \qquad b = \begin{pmatrix} \tilde{b} \\ 1 \end{pmatrix} \in \mathbb{R}^{d+1}. \tag{2.4}$$

Following the nomenclature for the (spatial) coefficient functions, we will refer to D as the *space-time diffusion coefficient* and b as the *space-time convection coefficient*. We also reiterate, following the description in Section 1.2, that $\widetilde{D} = \widetilde{D}(y)$ is assumed to be a $d \times d$ matrix-valued function such that $\widetilde{D}(y)$ form a uniformly positive-definite family of matrices over all y. That is, for any vector $x \in \mathbb{R}^d$,

$$x^T \widetilde{D}(y) x \geq \kappa |x|^2 \quad \text{for all } y \in Q, \tag{2.5}$$

where $\kappa > 0$ is a constant independent of y.

Because we make no formal distinction between the variables x and t, it is important to also redefine the sets in Equation 2.2 describing the computational domain (e.g. $\Omega \times (0,T)$, $\partial\Omega \times (0,T)$, and $\Omega \times \{0\}$). From a purely notational perspective it makes sense to ignore the Cartesian product structure between the space and time domains, since we are not setting apart time as a special variable. In addition, the space-time domain of a PDE does not have a Cartesian product structure in all cases. For instance, for problems with moving domains, the space-time domain can "bend" in the time direction; such a case is considered in [37].

We denote by Q the full space-time domain for the problem; specifically, Q is the open set in $\mathbb{R}^{d+1}$ containing all points for which the differential operator $\mathcal{L}$ is defined. In the case of Equation 2.2, the fact that Q has a Cartesian product structure means that we can separate ∂Q into a few distinct pieces. We define:

$$Q = \Omega \times (0, T) \subset \mathbb{R}^{d+1}$$

$$\Sigma_0 = \Omega \times \{0\}$$

$$\Sigma_T = \Omega \times \{T\}$$

$$\Sigma = \partial\Omega \times [0, T]$$

Thus Σ_0 is the portion of ∂Q that contains the initial state of u, Σ_T contains the final state, and Σ describes the spatial boundary of Ω at all times between 0 and T. Furthermore, for any $t \in [0, T]$ we will denote

$$\Omega_t = \Omega \times \{t\} \subset \mathbb{R}^{d+1} \qquad \text{and} \qquad Q_t = \Omega \times (0, t). \tag{2.6}$$

The set Ω_t is called the *spatial domain at time t* or a *time slice at time t*, while Q_t is the *space-time domain through t* or the *space-time domain on* $(0, t)$.

When the space-time domain Q has the Cartesian product structure of $\Omega \times (0, T)$, it is often called the *space-time cylinder* for the problem. When it is helpful to consider the individual domains in this product, we will refer to $\Omega \subset \mathbb{R}^d$ as the *spatial domain* and $(0, T)$ as the *time interval*.

Furthermore, we define the following spaces:

$$H^{k,l}(Q) = \{\, u \in L^2(Q) \mid \partial_x^\alpha u \in L^2(Q), \partial_t^i u \in L^2(Q), \text{ for } 0 \le |\alpha| \le k \text{ and } 0 \le i \le l \,\}$$

$$H_0^{k,l}(Q) = \{\, u \in H^{k,l}(Q) \mid u = 0 \text{ on } \Sigma \,\}$$

$$H_{0,\underline{0}}^{k,l}(Q) = \{\, u \in H^{k,l}(Q) \mid u = 0 \text{ on } \Sigma \text{ and } u = 0 \text{ on } \Sigma_0 \,\}$$

$$H_{0,\overline{0}}^{k,l}(Q) = \{\, u \in H^{k,l}(Q) \mid u = 0 \text{ on } \Sigma \text{ and } u = 0 \text{ on } \Sigma_T \,\}$$

Each of the above spaces is a Sobolev space with particular smoothness and boundary conditions. The superscripts denote the smoothness of the class in the spatial and temporal variables, with the first superscript describing spatial smoothness and the second superscript describing temporal smoothness. The subscripts define the boundary conditions. The first 0 subscript signifies homogeneous boundary conditions on the spatial boundary, while the second 0 (if it appears) describes homogeneous boundary

conditions on the time boundaries Σ_0 or Σ_T.

2.1.1 Abstract Error Analysis

A significant portion of this chapter will consist of various lemmas and propositions that establish the stability and convergence of our space-time finite element method. The details of these statements are important, but technical. As such, it is helpful to have a "road map," or big-picture view, that contextualizes the various claims and characteristics that we prove in this chapter. In this section, an abstract framework will be described that highlights the most important properties of our proposed finite element method and illustrates how these properties fit together.

Our point of departure will be to consider the transient convection-diffusion equation as a differential equation in the traditional sense; that is, a problem of the form:

Problem 1 (Strong Form, Abstract Setting)
Find $\tilde{u} \in \widetilde{U}$ such that

$$\begin{cases} \mathcal{A}\tilde{u} = f & \text{for } y \in Q \\ \tilde{u} = g & \text{for } y \in \Sigma_0 \cup \Sigma \end{cases} \tag{2.7}$$

where $\mathcal{A}$ is a differential operator and the equalities are considered in the sense of L^2 functions.

This form of the PDE will be called the *strong form* and its corresponding solution will be called the *strong solution*.

Next, we will define a variational problem which is consistent with the strong form in the sense that any strong solution will also be a solution to the variational problem. For suitable function spaces U and V, this problem takes the form:

Problem 2 (Variational Form, Abstract Setting)
Find $u \in U$ such that for all $v \in V$,

$$B(u, v) = L(v) \tag{2.8}$$

where B is a continuous bilinear form on $U \times V$ and L is a continuous linear functional on V.

In contrast to another variational equation which we will define shortly, this problem is sometimes called the *continuous variational problem* (as opposed to the *discrete* problem).

Of course, Problem 1 and Problem 2 are not directly computable, so it is necessary to define a finite element scheme that produces an approximation u_h to the variational solution u. The finite element solution is defined to satisfy a variational equality, but the function spaces from which solutions and test functions are drawn will be finite dimensional. As a result, the finite element problem can be characterized as a linear algebra problem (and is therefore computationally tractable).

Problem 3 (Finite Element Form, Abstract Setting)
Find $u_h \in V_h$ such that for all $v_h \in V_h$,

$$B_h(u_h, v_h) = L_h(v_h) \tag{2.9}$$

where the space $V_h \subset U$ is finite dimensional, B_h is a bilinear form on $U \times V_h$, and L_h is a continuous linear functional on V_h.

As before, a key property of the finite element problem is consistency with the continuous variational problem. That is, if u is a solution to Problem 2, then it must also satisfy Equation 2.9. An immediate consequence of consistency between Problem 2 and Problem 3 is the notion of *Galerkin orthogonality*. For solutions u and u_h to the continuous and discrete variational problems, respectively, Galerkin orthogonality is the property that

$$B_h(u - u_h, v_h) = 0 \qquad \text{for all } v_h \in V_h. \tag{2.10}$$

That is, the difference between the continuous and discrete solutions is orthogonal to the entire space V_h (in the sense of B_h). Note that Equation 2.10 may be derived immediately by observing that for any $v_h \in V_h$, the definition of consistency means that

$$B_h(u, v_h) = L_h(v_h) = B_h(u_h, v_h).$$

In addition to consistency, it will be necessary to establish two further properties of the bilinear form B_h: coercivity and boundedness. Let $|||\cdot|||_h$ be a norm on V_h. We say that B_h is *coercive* with respect to the norm $|||\cdot|||_h$ if there is some positive constant C_c such that

$$B_h(v_h, v_h) \geq C_c |||v_h|||_h^2 \qquad \text{for all } v_h \in V_h. \tag{2.11}$$

Next, let $|||\cdot|||_{h,*}$ be some norm on U. The bilinear form B_h is said to be *bounded* with respect to $|||\cdot|||_{h,*}$ and $|||\cdot|||_h$ if there is another positive constant C_b such that

$$|B_h(w, v_h)| \leq C_b |||w|||_{h,*} |||v_h|||_h \qquad \text{for all } w \in U \quad \text{and} \quad v_h \in V_h. \tag{2.12}$$

The properties of consistency, coercivity, and boundedness may be combined to establish a best-approximation property for the finite element solution u_h:

$$|||u - u_h|||_h^2 \leq \left(1 + \frac{C_b^2}{C_c^2}\right) \inf_{v_h \in V_h} |||u - v_h|||_{h,*}^2. \tag{2.13}$$

That is, the distance from u_h to u is minimal among all functions in the solution space V_h. Finally, the size of the distance $|||u - u_h|||_h$ can be characterized asymptotically as a function of the discretization parameter h. Given a polynomial interpolation operator $\mathcal{I}_h$ satisfying

$$|||w - \mathcal{I}_h w|||_{h,*} \leq C_2 h^m |w|_{H^m},$$

for any $w \in U \subset H^m$, an *a priori* error estimate of the form

$$|||u - u_h|||_h \leq C h^{k-1} |u|_{H^m} \tag{2.14}$$

will be proven, for a specific k.

The present chapter will establish each of the preceding definitions and properties, taking care to define the appropriate function spaces, norms, and linear forms. We begin with formal statements of the strong, variational, and finite element problems, and demonstrate consistency among them. The first main result will be the proof of coercivity of B_h, which also establishes the numerical stability of the method. Following the coercivity result, the boundedness of B_h will be established and a best-approximation property will follow immediately. The last component of the discussion will center on polynomial interpolants, which involves some nuance for schemes in four-dimensional space-time. After establishing the necessary properties of polynomial interpolation operators, we will arrive at an *a priori* error estimate for the finite element scheme.

2.2 Space-Time Formulations of Parabolic Problems

When written in terms of the space-time variable y, the transient convection-diffusion equation takes on the form of a stationary convection-diffusion problem in $d+1$ variables, with the condition that the space-time diffusion operator D has a specific (positive semidefinite, singular) form. In the standard analysis of stationary convection-diffusion equations, the diffusion operator is symmetric positive definite; therefore, we cannot apply the general theory of convection-diffusion equations directly to this problem. Our goal is to recover as much of the general theory as possible and handle the singular diffusion term appropriately when necessary.

Let us define precisely the strong form of the transient convection-diffusion equation, which will serve as the starting point for all of the following analysis.

Problem 4 (Transient Convection-Diffusion, Strong Form)
Let $\Omega \subset \mathbb{R}^d$ be a domain and $Q \subset \mathbb{R}^{d+1}$ its associated space-time domain. The solution $u \in H^{2,1}(Q)$ to the transient convection-diffusion equation satisfies

$$\begin{cases} -\nabla \cdot (D\nabla u - bu) + cu = f & \text{for } y \in Q \\ u = 0 & \text{for } y \in \Sigma \\ u = u_0 & \text{for } y \in \Sigma_0 \end{cases} \tag{2.15}$$

where D and b are defined as in Equation 2.4. Furthermore, we assume that $u_0 \in L^2(\Omega)$, $f \in L^2(Q)$, $c \in L^\infty(Q)$, $b \in H(div; Q)$, and $c + \frac{1}{2}\nabla \cdot b \geq \beta > 0$ on all of Q.

A variational form of Problem 4 may be derived in the usual manner for second-order equations - we multiply both sides of Equation 2.3 by a test function and then integrate by parts. However, due to the structure of the space-time boundary conditions, we must be somewhat careful. The following derivation is based on the classical work of Ladyzhenska[25], where notational changes have been introduced as needed. These basic results will then motivate the construction of a consistent finite element method.

Suppose that $u \in H^{2,1}(Q)$ is a solution to Problem 4 and let $v \in H^1_{0,\overline{0}}(Q)$ be arbitrary. Then

$$\int_Q -\nabla \cdot (D\nabla u - bu)v + cuv \, dy = \int_Q fv \, dy$$

and by the divergence theorem,

$$-\int_{\partial Q}(D\nabla u - bu)v \cdot n\, dy + \int_Q \nabla v^T (D\nabla u - bu) + cuv\, dy = \int_Q fv\, dy. \tag{2.16}$$

We can show that the boundary integral simplifies immensely by decomposing ∂Q and considering the boundary conditions. Recall that the space-time boundary can be split up as

$$\partial Q = \Sigma \cup \Sigma_0 \cup \Sigma_T, \tag{2.17}$$

where Σ is the set of spatial boundary points, Σ_0 is the initial state, and Σ_T is the final state. Since $v \in H^1_{0,\overline{0}}(Q)$, it follows that $v = 0$ on $\Sigma_T \cup \Sigma$. Thus

$$-\int_{\partial Q}(D\nabla u - bu)\, v \cdot n\, dy = -\int_{\Sigma_0}(D\nabla u - bu)\, v \cdot n\, dy.$$

Now, due to the special structure of the space-time diffusion D, the $(d+1)^{th}$ component of $D\nabla u$ vanishes. Since $n = (0, \dots, 0, -1)$ on Σ_0, this implies that

$$-\int_{\Sigma_0}(D\nabla u - bu)\, v \cdot n\, dy = \int_{\Sigma_0}(buv) \cdot n\, dy = -\int_{\Sigma_0} uv\, dy = -\int_{\Sigma_0} u_0 v\, dy.$$

Inserting this back into the larger expression of Equation 2.16, we conclude that

$$\int_Q \nabla v^T (D\nabla u - bu) + cuv\, dy = \int_Q fv\, dy + \int_{\Sigma_0} u_0 v\, dy. \tag{2.18}$$

We remark that the deduced Equation 2.18 is well-defined so long as $u, \nabla_x u \in L^2(Q)$. This motivates the definition of a variational solution to the model equation in Problem 4.

Problem 5 (Transient Convection-Diffusion, Variational Form)
The function $u \in H_0^{1,0}(Q)$ is a variational solution of the transient convection-diffusion equation if

$$B(u, v) = L(v) \qquad \text{for all } v \in H^1_{0,\overline{0}}(Q) \tag{2.19}$$

where

$$\begin{aligned} B(u,v) &= \int_Q \nabla v^T \left(D\nabla u - bu\right) + cuv \, dy \quad \text{and} \\ L(v) &= \int_Q fv \, dy + \int_{\Sigma_0} u_0 v \, dy. \end{aligned} \tag{2.20}$$

In the study of elliptic equations with Dirichlet boundary conditions, variational equations are often posed with test functions that vanish at the points where a Dirichlet boundary condition is imposed. For the space-time formulation of the transient convection-diffusion problem in Equation 2.15, the initial-boundary conditions on $\Sigma_0 \cup \Sigma$ may be considered as Dirichlet boundary conditions on the space-time domain. However, one may note that in the definition of the test space $H^1_{0,\overline{0}}(Q)$ for the variational equation in Problem 5, functions are prescribed to vanish on $\Sigma_T \cup \Sigma$, not $\Sigma_0 \cup \Sigma$. We choose this particular test space in order to leverage the theory proven by Ladyzhenska in [25], who also shows that the solution to Equation 2.19 satisfies a more general variational equality as well.

Proposition 2.1. *Let $u \in H^{1,0}_0(Q)$ be the solution to Problem 5. Then for any $v \in H^1_0(Q)$ and $t \in [0,T]$,*

$$\int_{Q_t} \nabla^T v(D\nabla u - bu) + cuv \, dy + \int_{\Omega_t} uv \, dy = \int_{Q_t} fv \, dy + \int_{\Sigma_0} u_0 v \, dy \tag{2.21}$$

where Q_t and Ω_t are defined as in Equation 2.6; that is, $Q_t = \Omega \times (0,t)$ and $\Omega_t = \Omega \times \{t\}$. In particular, taking $t = T$, we have

$$\int_{Q} \nabla^T v(D\nabla u - bu) + cuv \, dy + \int_{\Sigma_T} uv \, dy = \int_{Q} fv \, dy + \int_{\Sigma_0} u_0 v \, dy. \tag{2.22}$$

Proof. See [25], Chapter III, Equation 3.20, and the surrounding discussion. □

Proposition 2.1 states that any solution to Problem 5 satisfies a variational equality with test functions drawn from the larger space $H^1_0(Q) \supset H^1_{0,\overline{0}}(Q)$, so long as we include an extra term containing contributions from the time-outflow boundary Σ_T. We will use this fact when defining a consistent finite-element scheme for Problem 5.

A natural way to discretize the variational problem is to consider finite-dimensional subspaces of $H^{1,0}_0(Q)$ which are made up of piecewise-polynomial functions over a triangulation $\mathcal{T}$. For a triangulation $\mathcal{T}$ which covers Q, we will choose the finite element space V_h to be the space of all piecewise-polynomial functions which are of degree r on each simplex $\tau \in \mathcal{T}$, and which vanish on Σ. That

is,

$$V_h = V_{h,0}^r(\mathcal{T}) = \left\{ p \in H_0^{1,0}(Q) \,:\, p\big|_\tau \in \mathbb{P}_r(\tau) \;\; \forall \tau \in \mathcal{T} \right\}. \tag{2.23}$$

Remark 2.2. Going forward we will also assume that the coefficient functions in the definition of Problem 4 possess a given smoothness on the interior of individual mesh elements. For example, instead of assuming that $\nabla \cdot b \in L^2(Q)$, we will only assume that $\nabla \cdot b \in L^2(\tau)$ for each $\tau \in \mathcal{T}$.

In order to maintain consistency between the continuous variational problem and the finite element problem, we will use Equation 2.22 to define the discrete bilinear form.

Problem 6 (Transient Convection-Diffusion, Galerkin FEM)
The finite element solution $u_h \in V_h$ satisfies

$$B_h(u_h, v_h) = L_h(v_h) \quad \text{for all } v_h \in V_h. \tag{2.24}$$

where

$$\begin{aligned} B_h(u_h, v_h) &= \int_Q \nabla v_h^T \left(D\nabla u_h - b u_h\right) + c u_h v_h \, dy + \int_{\Sigma_T} u_h v_h \, dy \quad \text{and} \\ L_h(v_h) &= \int_Q f v \, dy + \int_{\Sigma_0} u_0 v_h \, dy. \end{aligned} \tag{2.25}$$

Since $V_h \subset H_0^1(Q)$, Proposition 2.1 implies that any solution to Problem 5 satisfies Equation 2.25. Therefore Problem 6 is consistent with Problem 5. Unfortunately, the above finite element problem is inherently numerically unstable due to the presence of the singular diffusion term D. This scenario can be viewed as a limiting case of convection-dominated flow problems, where the size ratio of convection and diffusion terms approaches infinity. In the following section, we apply this methodology to the transient convection-diffusion equation and show that the stability and convergence behavior matches that of the steady-state problem.

2.3 Stability of a Space-Time Galerkin Method

Galerkin finite element methods applied to the convection-diffusion equation can exhibit poor behavior under certain conditions on the coefficients D, b, and c. In particular, continuous Galerkin methods produce oscillatory solutions when the strength of the convective field b is much stronger than the diffusive term D; see [23], [40], and [10] for a more detailed treatment of this phenomenon.

These erroneous oscillations are present in treatments of both the steady-state and transient convection-diffusion equations.

Consider first the steady-state problem, where the magnitude of convection dominates the diffusive strength. Mathematically, this is the case where $|\tilde{b}| \gg \kappa$ (recall that κ is the coercivity constant of $\widetilde{D}$). Now, the abstract error estimate in Equation 2.13 describes the error of Galerkin's method in terms of polynomial approximation, but it also depends on a (semi-)computable constant. The constant $C = 1 + C_b C_c^{-1}$ depends directly on the boundedness constant C_b and the coercivity constant C_c, which depend in turn on D, b, and c. In particular, $C_c = C'\kappa$, where C' depends only on the spatial domain, and $C_b \geq \|\tilde{b}\|_\infty$ (see [40], Chapters 6 and 8 for more detail). Therefore, as the ratio $\|\tilde{b}\|_\infty/\kappa$ increases, the method's control on $\|\nabla_x(u - u_u)\|_{L^2}$ diminishes, and spurious oscillations in the solution can develop.

In the case of the transient convection-diffusion equation, the situation is even worse. While the stability issues for steady-state convection-diffusion equations arose from the potentially large constant $(1 + C_b/C_c)$, in the transient setting such a (finite) constant does not even exist! If the above argument is applied to the transient equation, we find that C_c, the smallest eigenvalue of $\widetilde{D}$, will be 0. If we consider this to be the limiting case of $C_c \to 0$, it is immediately clear that the constant $(1 + C_b/C_c)$ appearing in the error bound will tend to infinity.

Informally, we can think of the size of $\partial_{x_i} u$ as being controlled by the ratio of the diffusive strength in the i^{th} dimension to the convective strength in the i^{th} dimension. Since the transient convection-diffusion equation contains "time convection" terms but no "time diffusion" terms, the space-time Galerkin method loses any form of control over the size of the time derivative.

Clearly, the simple space-time Galerkin method proposed above needs to be modified in order to obtain a measure of numerical stability. Several stabilization techniques have been considered for this purpose. In [4], the authors analyzed a streamline-upwind Petrov-Galerkin (SUPG) scheme and an Edge Average Finite Element (EAFE) scheme for the case of constant coefficients (an EAFE scheme is higher-dimensional generalization of a Scharfetter-Gummel scheme). The series [27–29] considers an SUPG scheme for the heat equation with only weak regularity assumptions on the coefficient functions. In addition, stabilization of the heat equation via bubble functions is studied in [26].

The defining characteristic of an SUPG finite element scheme is the augmentation of the test space $V_{h,0}$ with terms that are "biased" in the direction of a convective flow. We define the *upwinded*

test space

$$\overline{V}_h = \left\{ \sum_{\tau \in \mathcal{T}} v_\tau + \theta_\tau h_\tau^p b \cdot \nabla v_\tau \,:\, v \in V_h,\ v_\tau = v\big|_\tau \right\} = V_h + W_h. \tag{2.26}$$

We leave the integer p and the real parameters θ_τ unspecified for now.

The guiding principle when defining an SUPG method is that the solution of the original strong form of the PDE (c.f Equation 2.3) is consistent with the solution to the SUPG problem. That is, if u a solution to Equation 2.3, then it must be a solution to the SUPG problem.

To derive an appropriate SUPG form of the PDE, we introduce a test function in the same way as for Galerkin's method. However, we now choose $\overline{v} \in \overline{V}_h$. Furthermore, since the functions in $\overline{V}_h$ are defined element-wise, our analysis will also proceed on an element-by-element basis. Since the triangulations considered always cover Q, global estimates are obtained by summing together every element-wise contribution.

Remark 2.3. In the previous section, notational conventions implied that functions u and v belonged to spaces for the continuous variational problem, while functions u_h and v_h belonged to spaces for the discrete (i.e. finite element) variational problem. For the remainder of this section, we will discard this convention and use the symbols v for functions in V_h and $\overline{v}$ for functions in $\overline{V}_h$. This is done primarily to reduce notational clutter.

Suppose $u \in H_0^{2,1}(Q)$ is a solution to the strong form of the transient convection-diffusion equation (c.f. Problem 4) and let

$$\overline{v} = \sum_{\tau \in \mathcal{T}} v_\tau + \theta_\tau h_\tau^p b \cdot \nabla v_\tau \in \overline{V}_h \tag{2.27}$$

be arbitrary, where again we set v_τ to be the function which coincides with v on $\tau \in \mathcal{T}$ and vanishes elsewhere. Then

$$\int_Q -\nabla \cdot (D\nabla u - bu)\overline{v} + cu\overline{v}\, dy = \sum_{\tau \in \mathcal{T}} \int_\tau -\nabla \cdot (D\nabla u - bu)\,\overline{v} + cu\overline{v}\, dy = \sum_{\tau \in \mathcal{T}} \int_\tau f\overline{v}\, dy \tag{2.28}$$

and we deduce:

$$\begin{aligned}\sum_{\tau\in\mathcal{T}}\int_\tau f\overline{v}\,dy &= \sum_{\tau\in\mathcal{T}}\int_\tau -\nabla\cdot(D\nabla u - bu)\,v - \nabla\cdot(D\nabla u - bu)\cdot\theta_\tau h_\tau^p\,(b\cdot\nabla v) + cu\overline{v}\,dy \\ &= \sum_{\tau\in\mathcal{T}}\int_\tau (D\nabla u - bu)\cdot\nabla v - \theta_\tau h_\tau^p\nabla\cdot(D\nabla u - bu)\,(b\cdot\nabla v) + cuv + \theta_\tau h_\tau^p\,(b\cdot\nabla v)\,cu\,dy \\ &\quad - \sum_{\tau\in\mathcal{T}}\int_{\partial\tau}(D\nabla u - bu)\,v\cdot n\,dy\end{aligned} \tag{2.29}$$

Due to the smoothness of solution u, we can show that the boundary flux terms (the last integral in the previous expression) cancel along internal boundaries. In particular:

Proposition 2.4. *Let $\mathcal{T}$ be a triangulation covering Q, $u \in H_0^{2,1}(Q)$ a solution to Problem 4, and $v \in H_0^1(Q)$. Then*

$$-\sum_{\tau\in\mathcal{T}}\int_{\partial\tau}(D\nabla u - bu)v\cdot n\,dy = \int_{\Sigma_T} uv\,dy - \int_{\Sigma_0} uv\,dy. \tag{2.30}$$

Proof. Recalling the derivation of Problem 4, we know that

$$\int_Q \nabla^T v(D\nabla u - bu) + cuv\,dy + \int_{\Sigma_T} uv\,dy - \int_{\Sigma_0} u_0 v\,dy = \int_Q -\nabla\cdot(D\nabla u - bu)v + cuv\,dy. \tag{2.31}$$

Then since $\mathcal{T}$ covers Q,

$$\begin{aligned}\int_Q -\nabla\cdot(D\nabla u - bu)v + cuv\,dy &= \sum_{\tau\in\mathcal{T}}\int_\tau -\nabla\cdot(D\nabla u - bu)v + cuv\,dy \\ &= \sum_{\tau\in\mathcal{T}}\int_\tau \nabla^T v(D\nabla u - bu) + cuv\,dy - \int_{\partial\tau}(D\nabla u - bu)v\cdot n\,dy \\ &= \int_Q \nabla^T v(D\nabla u - bu) + cuv\,dy - \sum_{\tau\in\mathcal{T}}\int_{\partial\tau}(D\nabla u - bu)v\cdot n\,dy.\end{aligned} \tag{2.32}$$

Subtracting the integral over Q from Equation 2.31 and Equation 2.32 establishes the proposition. □

Applying Proposition 2.4 to the derivation in Equation 2.29 simplifies the boundary integrals

so that

$$\begin{aligned}\sum_{\tau\in\mathcal{T}}\int_\tau f\overline{v}\,dy &= \sum_{\tau\in\mathcal{T}}\int_\tau (D\nabla u)\cdot\nabla v - \theta_\tau h_\tau^p (b\cdot\nabla v)\nabla\cdot(D\nabla u) + \theta_\tau h_\tau^p (b\cdot\nabla v)\nabla\cdot(bu)\;dy \\ &\quad + \sum_{\tau\in\mathcal{T}}\int_\tau (\theta_\tau h_\tau^p c - 1)\,(b\cdot\nabla v)\,u + cuv\,dy + \int_{\Sigma_T} uv\,dy - \int_{\Sigma_0} uv\,dy \\ &= \sum_{\tau\in\mathcal{T}}\int_\tau \nabla^T v\left(D + \theta_\tau h_\tau^p bb^T\right)\nabla u - \theta_\tau h_\tau^p (b\cdot\nabla v)\nabla\cdot(D\nabla u)\,dy \\ &\quad + \sum_{\tau\in\mathcal{T}}\int_\tau (\theta_\tau h_\tau^p (c+\nabla\cdot b) - 1)(b\cdot\nabla v)u + cuv\,dy + \int_{\Sigma_T} uv\,dy - \int_{\Sigma_0} uv\,dy.\end{aligned}$$

Remark 2.5. For the sake of notational convenience, we will denote

$$D_{h,\theta,\tau} = D + \theta_\tau h_\tau^p bb^T. \tag{2.33}$$

When τ is clear from the context, or the element τ is arbitrary, this will often be abbreviated to $D_{h,\theta}$. Furthermore, when τ is arbitrary we will drop the subscripts on h and θ. We can consider $D_{h,\theta}$ to be an "augmented" space-time diffusion, where an artificial (or "numerical") diffusion $\theta h^p bb^T$ has been added to the natural diffusion D. The matrix bb^T is a rank-one projection in the direction of b; therefore, the artificial diffusion introduced by the SUPG problem exists only in the direction of the space-time convection. For this reason, the SUPG discretization described above is sometimes referred to as a "streamline diffusion" method.

Incorporating the notation $D_{h,\theta,\tau}$ into the above derivation, we conclude that the solution $u \in H_0^{2,1}(Q)$ satisfies

$$\overline{B}_h(u,v) = \overline{L}_h(v) \qquad \text{for any } v \in V_h \tag{2.34}$$

where

$$\begin{aligned}\overline{B}_h(u,v) &= \sum_{\tau\in\mathcal{T}}\int_\tau (\nabla v)^T D_{h,\theta,\tau}\nabla u - \theta_\tau h_\tau^p (b\cdot\nabla v)\nabla\cdot(D\nabla u)\,dy \\ &\quad + \int_\tau (\theta_\tau h_\tau^p (c+\nabla\cdot b) - 1)(b\cdot\nabla v)u + cuv\,dy + \int_{\Sigma_T} uv\,dy\end{aligned} \tag{2.35}$$

$$\overline{L}_h(v) = \sum_{\tau\in\mathcal{T}}\int_\tau fv + \theta_\tau h_\tau^p f(b\cdot\nabla v)\,dy + \int_{\Sigma_0} u_0 v\,dy \tag{2.36}$$

The above derivation shows that any strong solution $u \in H_0^{2,1}(Q)$ to the transient convection-diffusion

equation must also satisfy Equation 2.34 for any $v \in V_h$.

We are now ready to define the SUPG finite element problem in the space-time setting. To ensure consistency, we will utilize the linear forms defined in Equations 2.35 to 2.36. The only modification necessary is to adjust the trial space to be finite dimensional.

Problem 7 (Transient Convection-Diffusion, SUPG FEM)
Find $u \in V_h$ such that for all $v \in V_h$,

$$\overline{B}_h(u, v) = \overline{L}_h(v) \tag{2.37}$$

where $\overline{B}_h$ and $\overline{L}_h$ are defined as in Equation 2.35 and Equation 2.36, respectively.

We will prove in short order that Problem 7 is numerically stable and converges at a near-optimal rate. In fact, this space-time SUPG method for the transient convection-diffusion equation converges at the same rate as the classical SUPG method for steady-state convection-diffusion equations. This shows that SUPG stabilization of convection-dominated flow problems converges even in the presence of vanishing diffusion.

Proposition 2.6. *$D_{h,\theta}$ is positive definite for any $h > 0$, $\theta > 0$.*

Proof. Let $w \in \mathbb{R}^{d+1}$ be nonzero and arbitrary, and let $w = (w_1, w_2, \dots, w_{d+1})^T$. We denote the first d coordinates of w as $w_x = (w_1, \dots, w_d)$, so we may write (in a slight abuse of notation) that $w = (w_x, w_{d+1})^T$. Then

$$w^T D_{h,\theta} w = w^T (D + \theta h^p b b^T) w = w^T D w^T + \theta h^p w^T b b^T w = w^T D w + \theta h^p (w^T b)^2.$$

From the block diagonal structure of $\widetilde{D}$ and the uniform positive definiteness of D, we deduce

$$\begin{aligned} w^T D_{h,\theta} w &= w^T D w + \theta h^p (w^T b)^2 \\ &= w_x^T \widetilde{D} w_x + \theta h^p (w_x^T \tilde{b} + w_{d+1})^2 \\ &\geq \kappa |w_x|^2 + \theta h^p (w_x^T \tilde{b} + w_{d+1})^2. \end{aligned}$$

If $|w_x| > 0$, then clearly $w^T D_{h,\theta} w > 0$. If instead $|w_x| = 0$, then $w^T D w = \theta h^p (w_{d+1})^2$; since w is nonzero and $w_x = 0$, we conclude that $w_{d+1} \neq 0$ and thus $w^T D w > 0$. Therefore, $w^T D w > 0$ in all cases. □

With a little additional effort, we can estimate the uniform bound for the positive-definiteness of $D_{h,\theta}$ from below. The following lemma is necessary to establish this bound; however, the proof of Lemma 2.7 is quite messy and does not provide additional insight into the present discussion. As such, the details of the proof are deferred to Chapter A.

Lemma 2.7. *Given constants $A, C > 0$ and $B \geq 0$,*

$$\min_{0\leq z\leq 1} Az^2 + C\left(Bz - \sqrt{1-z^2}\right)^2 \geq \min\left(A + B^2C, \frac{AC}{A + C\,(B^2+B)}\right).$$

Proof. See Chapter A. □

Proposition 2.8. *For any $w \in \mathbb{R}^{d+1}$,*

$$w^T D_{h,\theta} w \geq \gamma |w|^2, \tag{2.38}$$

where

$$\gamma = \min\left(\kappa + \theta h^p |\tilde{b}|^2, \frac{\kappa \cdot \theta h^p}{\kappa + \theta h^p\left(|\tilde{b}|^2 + |\tilde{b}|\right)}\right). \tag{2.39}$$

Proof. Let $w \in \mathbb{R}^{d+1}$ have unit norm. We will show that $w^T D_{h,\theta} w \geq \gamma$. As in the proof of Proposition 2.6, let w be written as $(w_x, w_{d+1})^T$, where $w_x = (w_1, \dots, w_d)^T$. By the same argument as of Proposition 2.6, we know

$$w^T D_{h,\theta} w \geq \kappa |w_x|^2 + \theta h^p (w^T b)^2.$$

Since $|w| = 1$, we know that $w_{d+1} = \sqrt{1 - |w_x|^2}$. By definition, $b_{d+1} = 1$, and so we deduce that $w^T b = w_x^T \tilde{b} + \sqrt{1 - |w_x|^2}$. Plugging this into the inequality above, we see

$$\begin{aligned} w^T D_{h,\theta} w &\geq \kappa |w_x|^2 + \theta h^p \left(w_x^T \tilde{b} + \sqrt{1 - |w_x|^2}\right)^2 \\ &\geq \kappa |w_x|^2 + \theta h^p \left(|w_x||\tilde{b}| - \sqrt{1 - |w_x|^2}\right)^2. \end{aligned}$$

This expression is a continuous function of of $|w_x|$, for $0 \leq |w_x| \leq 1$. Now, let $A = \kappa$, $B = |\tilde{b}|$, and $C = \theta_\tau h_\tau^p$. By Lemma 2.7, we know that

$$\min_{0\leq z\leq 1} Az^2 + C\left(Bz - \sqrt{1-z^2}\right)^2 \geq \min\left(A + B^2C, \frac{AC}{A + C(B^2+B)}\right),$$

and therefore we conclude that for $0 \leq |w_x| \leq 1$,

$$\begin{aligned} w^T D_{h,\theta} w &\geq \kappa |w_x|^2 + \theta h^p \left(|w_x||\tilde{b}| - \sqrt{1 - |w_x|^2} \right)^2 \\ &\geq \min\left(\kappa + \theta h^p |\tilde{b}|^2, \frac{\kappa \theta h^p \rho}{\kappa + \theta h^p \left(|\tilde{b}|^2 + |\tilde{b}| \right)} \right) \\ &= \gamma. \end{aligned}$$

Thus for any unit w, we have $w^T \widetilde{D}_{h,\theta} w \geq \gamma$. The proposition is then proved by noting that for any nonzero $w \in \mathbb{R}^{d+1}$,

$$w^T D_{h,\theta} w = \frac{w^T}{|w|} D_{h,\theta} \frac{w}{|w|} \cdot |w|^2 \geq \gamma |w|^2.$$

□

Next, we define an associated energy norm for this problem:

For $v \in H_0^1(Q)$, let

$$\begin{aligned} |||v|||_h^2 &:= \int_{\Sigma_T} v^2 \, dy + \int_{\Sigma_0} v^2 \, dy + \sum_{\tau \in \mathcal{T}} \int_\tau \nabla^T v D_{h,\theta} \nabla v + v^2 \, dy \\ &= \|v\|_{L^2(\Sigma_T)}^2 + \|v\|_{L^2(\Sigma_0)}^2 + \sum_{\tau \in \mathcal{T}} \left\| D_{h,\theta}^{1/2} \nabla v \right\|_{L^2(\tau)}^2 + \|v\|_{L^2(\tau)}^2 \end{aligned}$$

We will show that the SUPG bilinear form $\overline{B}_h$ is coercive with respect to this norm, which is crucial in order for the method to be numerically stable. In particular, the presence of the term $\left\| D_{h,\theta}^{1/2} \right\|$ in the definition of $|||\cdot|||$ ensures that the gradient of the error will diminish with mesh size. This has the effect of removing the spurious oscillations present in the standard Galerkin finite element solution.

Theorem 2.9. *Let $p \geq 2$. Then there is a positive constant C_c such that for any $v \in V_h$,*

$$\overline{B}_h(v, v) \geq C_c |||v|||_h^2.$$

Proof. By definition,

$$\overline{B}_h(v, v) = \sum_{\tau \in \mathcal{T}} \int_\tau \nabla^T v D_{h,\theta} \nabla v - \theta_\tau h_\tau^p (b \cdot \nabla v) \nabla \cdot (D \nabla v) + (\theta_\tau h_\tau^p (c + \nabla \cdot b) - 1)(b \cdot \nabla v) v + c v^2 \, dy + \int_{\Sigma_T} v^2 \, dy. \tag{2.40}$$

We will approximate each term of the integrand separately.

Let $\tau \in \mathcal{T}$ be an arbitrary simplex and define:

$$
\begin{aligned}
I(\tau) &= \int_\tau (\nabla v)^T D_{h,\theta} \nabla v \, dy \\
II(\tau) &= -\int_\tau \theta_\tau h_\tau^p (b \cdot \nabla v) \nabla \cdot (D \nabla v) \, dy \\
III(\tau) &= \int_\tau (\theta_\tau h_\tau^p (c + \nabla \cdot b) - 1)(b \cdot \nabla v) v \, dy \\
IV(\tau) &= \int_\tau c v^2 \, dy \\
V &= \int_{\Sigma_T} v^2 \, dy.
\end{aligned}
$$

Then

$$
\overline{B}(v, v) = V + \sum_{\tau \in \mathcal{T}} I(\tau) + II(\tau) + III(\tau) + IV(\tau).
$$

First, we approximate $II(\tau)$ using an inverse inequality together with Young's inequality with epsilon and the Cauchy-Schwarz inequality.

$$
\begin{aligned}
II(\tau) &= -\theta_\tau h_\tau^p \left(\nabla \cdot (D \nabla v), b \cdot \nabla v\right)_{L^2(\tau)} \\
&\geq -\theta_\tau h_\tau^p \|\nabla \cdot (D \nabla v)\|_{L^2(\tau)} \|b \cdot \nabla v\|_{L^2(\tau)} \\
&\geq -\theta_\tau h_\tau^p \cdot C_\tau h_\tau^{-1} \|D \nabla v\|_{L^2(\tau)} \|b \cdot \nabla v\|_{L^2(\tau)} \\
&\geq -\theta_\tau h_\tau^p \left(C_\tau^2 h_\tau^{-2} \|D \nabla v\|_{L^2(\tau)}^2 + \frac{1}{4} \|b \cdot \nabla v\|_{L^2(\tau)}^2 \right) \\
&\geq -\theta_\tau h_\tau^p \left(C_\tau^2 h_\tau^{-2} \left\| D^{1/2} \right\|^2 \left\| D^{1/2} \nabla v \right\|_{L^2(\tau)}^2 + \frac{1}{4} \|b \cdot \nabla v\|_{L^2(\tau)}^2 \right) \\
&= -\theta_\tau C_\tau^2 h_\tau^{p-2} \left\| D^{1/2} \right\|^2 \int_\tau \nabla^T v D \nabla v \, dy - \frac{1}{4} \theta_\tau h_\tau^p \int_\tau \nabla^T v b b^T \nabla v \, dy \\
&= \int_\tau \nabla^T v \left(-\theta_\tau C_\tau^2 h_\tau^{p-2} \left\| D^{1/2} \right\|^2 D - \frac{1}{4} \theta_\tau h_\tau^p b b^T \right) \nabla v \, dy,
\end{aligned}
$$

Therefore, we have for $p \geq 2$ and sufficiently small θ_τ,

$$\begin{aligned} I(\tau) + II(\tau) &\geq \int_\tau \nabla^T v \left(\left(1 - \theta_\tau C_\tau^2 \left\| D^{1/2} \right\|^2 h_\tau^{p-2} \right) D + \frac{3}{4} \theta_\tau h_\tau^p b b^T \right) \nabla v \, dy \\ &\geq \widehat{C}_1 \int_\tau \nabla^T v \left(D + \theta_\tau h_\tau^p b b^T \right) \nabla v \, dy \\ &= \widehat{C}_1 \left\| D_{h,\theta}^{1/2} \nabla v \right\|_{L^2(\tau)}^2. \end{aligned}$$

We now proceed to estimate $III(\tau) + IV(\tau)$. Applying the product rule in reverse on the term $v \nabla v$ in $III(\tau)$,

$$\begin{aligned} III(\tau) &= \int_\tau (\theta_\tau h_\tau^p (c + \nabla \cdot b) - 1)(b \cdot \nabla v) v \, dy \\ &= \int_\tau (\theta_\tau h_\tau^p (c + \nabla \cdot b) b - b) \cdot \nabla \left(v^2 \right) dy - \int_\tau (\theta_\tau h_\tau^p (c + \nabla \cdot b) b - b) \cdot (\nabla v) v \, dy, \end{aligned}$$

and therefore

$$\begin{aligned} III(\tau) &= \frac{1}{2} \int_\tau (\theta_\tau h_\tau^p (c + \nabla \cdot b) b - b) \cdot \nabla \left(v^2 \right) dy \\ &= -\frac{1}{2} \int_\tau \nabla \cdot (\theta_\tau h_\tau^p (c + \nabla \cdot b) b - b) v^2 \, dy + \frac{1}{2} \int_{\partial\tau} (\theta_\tau h_\tau^p (c + \nabla \cdot b) - 1) v^2 b \cdot n \, dy \\ &= \frac{1}{2} \int_\tau (\nabla \cdot b) v^2 \, dy - \frac{1}{2} \theta_\tau h_\tau^p \int_\tau \nabla \cdot (cb + (\nabla \cdot b) b) v^2 \, dy + \frac{1}{2} \int_{\partial\tau} (\theta_\tau h_\tau^p (c + \nabla \cdot b) - 1) v^2 b \cdot n \, dy. \end{aligned}$$

Adding together integrals $III(\tau)$ and $IV(\tau)$, and applying the assumptions that $b \in H^2(Q), c \in H^1(Q)$, and $c(y) + \frac{1}{2} \nabla \cdot b(y) \geq \beta > 0$ for all $y \in Q$,

$$\begin{aligned} III(\tau) + IV(\tau) &= \int_\tau \left(c + \frac{1}{2} \nabla \cdot b \right) v^2 \, dy - \frac{\theta_\tau h_\tau^p}{2} \int_\tau \nabla \cdot (cb + (\nabla \cdot b) b) v^2 \, dy \\ &\quad + \frac{1}{2} \int_{\partial\tau} (\theta_\tau h_\tau^p (c + \nabla \cdot b) - 1) v^2 b \cdot n \, dy \\ &\geq \beta \|v\|_{L^2(\tau)}^2 - \frac{\theta_\tau h_\tau^p}{2} \|\nabla \cdot (cb + (\nabla \cdot b) b)\|_{L^2(Q)} \|v\|_{L^2(\tau)}^2 \\ &\quad + \frac{1}{2} \int_{\partial\tau} (\theta_\tau h_\tau^p (c + \nabla \cdot b) - 1) v^2 b \cdot n \, dy. \end{aligned}$$

Denote the constant $\|\nabla\cdot(cb+(\nabla\cdot b)b)\|_{L^2(Q)} =: C_d$. Then we have shown that

$$III(\tau)+IV(\tau)\geq\left(\beta-\frac{\theta_\tau h_\tau^p}{2}C_d\right)\|v\|^2_{L^2(\tau)}+\frac{1}{2}\int_{\partial\tau}(\theta_\tau h_\tau^p(c+\nabla\cdot b)-1)v^2b\cdot n\,dy,$$

and so, taking θ_τ sufficiently small (for example, $\theta_\tau<\beta\cdot(h_\tau^pC_d)^{-1}$) we conclude that

$$III(\tau)+IV(\tau)\geq\widehat{C}_2\|v\|^2_{L^2(\tau)}+\frac{1}{2}\int_{\partial\tau}(\theta_\tau h_\tau^p(c+\nabla\cdot b)-1)v^2b\cdot n\,dy.$$

Next, we remark that if θ_τ is chosen such that $\theta_\tau\leq(2h_\tau^p\|c+\nabla\cdot b\|_{L^\infty(Q)})^{-1}$, then

$$\frac{1}{2}\int_{\partial\tau}(\theta_\tau h_\tau^p(c+\nabla\cdot b)-1)v^2b\cdot n\,dy\geq-\frac{3}{4}\int_{\partial\tau}v^2b\cdot n\,dy.$$

Additionally, since $v^2\in H_0^1(Q)$ and $\nabla\cdot b\in L^2(Q)$, the internal boundary integrals above cancel:

$$\sum_{\tau\in\mathcal{T}}\int_{\partial\tau}v^2b\cdot n\,dy=\int_{\partial Q}v^2b\cdot n\,dy=\int_{\Sigma_T}v^2\,dy-\int_{\Sigma_0}v^2\,dy.$$

We remark that above equality holds because for any vector field $F\in H(\mathrm{div};Q)$,

$$\sum_{\tau\in\mathcal{T}}\int_{\partial\tau}F\cdot n\,dy=\sum_{\tau\in\mathcal{T}}\int_{\tau}-\nabla\cdot F\,dy=\int_Q-\nabla\cdot F\,dy=\int_{\partial Q}F\cdot n\,dy.$$

Finally, the above element-wise estimates are combined into the domain-wide estimate:

$$\begin{aligned}
\overline{B}(v,v)&=V+\sum_{\tau\in\mathcal{T}}I(\tau)+II(\tau)+III(\tau)+IV(\tau)\\
&\geq\int_{\Sigma_T}v^2\,dy+\sum_{\tau\in\mathcal{T}}\widehat{C}_1\left\|D^{1/2}_{h,\theta}\nabla v\right\|^2_{L^2(\tau)}+\widehat{C}_2\|v\|^2_{L^2(\tau)}-\frac{3}{4}\int_{\partial\tau}v^2b\cdot n\,dy\\
&=\int_{\Sigma_T}v^2\,dy+\sum_{\tau\in\mathcal{T}}\left(\widehat{C}_1\left\|D^{1/2}_{h,\theta}\nabla v\right\|^2_{L^2(\tau)}+\widehat{C}_2\|v\|^2_{L^2(Q)}\right)-\frac{3}{4}\int_{\Sigma_T}v^2\,dy+\frac{3}{4}\int_{\Sigma_0}v^2\,dy\\
&=\frac{1}{4}\|v\|^2_{L^2(\Sigma_T)}+\frac{3}{4}\|v\|^2_{L^2(\Sigma_0)}+\sum_{\tau\in\mathcal{T}}\widehat{C}_1\left\|D^{1/2}_{h,\theta}\nabla v\right\|^2_{L^2(\tau)}+\widehat{C}_2\|v\|^2_{L^2(\tau)}\\
&\geq C_c|||v|||^2_h.
\end{aligned}$$

□

Remark 2.10. In the proof of Theorem 2.9, the upwinding parameter θ had several conditions imposed

upon it. The conditions are:

$$\begin{aligned} \theta_\tau h_\tau^{p-2} &\leq \frac{3}{4} C_\tau^{-2} \left\| D^{1/2} \right\|^{-2} \\ \theta_\tau h_\tau^p &\leq \beta \| \nabla \cdot (cb + (\nabla \cdot b) b) \|_{L^2(Q)}^{-1} \\ \theta_\tau h_\tau^p &\leq \frac{1}{2} \| c + \nabla \cdot b \|_{L^\infty(Q)}^{-1} \end{aligned} \tag{2.41}$$

where we may safely ignore conditions in which infinity appears on the right-hand side. In all further discussion, θ_τ is chosen to be the largest value that satisfies all three inequalities, or 1, whichever is smaller. In particular, this means that θ_τ is bounded uniformly from below and $\theta_\tau h_\tau^p$ is bounded uniformly from above for all τ.

Remark 2.11. If V_h is the space of piece-wise linear polynomials over the triangulation $\mathcal{T}$ (i.e. the case $r = 1$), then Theorem 2.9 holds even with $p \geq 1$.

Proof. The only change lies in estimating the term $I(\tau) + II(\tau)$. If v is piece-wise linear, then $\nabla \cdot (D \nabla v)$ is identically 0 and $II(\tau)$ vanishes. Thus $I(\tau) + II(\tau) = \left\| D_{h,\theta}^{1/2} \nabla v \right\|_{L^2(\tau)}^2$, and no assumption that $p \geq 2$ is needed to make this estimation. The rest of the proof may be carried through unchanged. □

The remainder of this chapter is devoted to analyzing the error of the SUPG finite element method. Going forward, we will always assume that p and θ_τ satisfy the requirements of Theorem 2.9; that is, $p = 2$ and each θ_τ is chosen in the manner described by Remark 2.10. A parallel analysis could be carried out for the special case when $r = p = 1$; see the remarks in [27] for a discussion of this scenario.

2.4 Convergence of a Stabilized Space-Time Galerkin Method

One of the key properties of the proposed SUPG scheme is good convergence with respect to $|||\cdot|||_h$. To establish this property, it is necessary to show that $\overline{B}_h$ is uniformly bounded with respect to $|||\cdot|||_h$. To aid in this proof, we will introduce a second norm:

$$|||w|||_{h,*}^2 = |||w|||_h^2 + \sum_{\tau \in \mathcal{T}} \theta_\tau h_\tau^2 \| \nabla \cdot (D \nabla w) \|_{L^2(\tau)}^2 + \theta_\tau^{-1} h_\tau^{-2} \| w \|_{L^2(\tau)}^2, \tag{2.42}$$

defined for all $w \in H^{2,1}(\mathcal{T}) \cap H_0^{1,0}(Q)$. The norm $|||\cdot|||_{h,*}$ is a natural cousin to the energy norm $|||\cdot|||_h$. In fact, it is the uniform bound on $\overline{B}_h(u, v)$ for fixed u:

Proposition 2.12. *Let $\overline{B}_h$ be the SUPG bilinear form with $p = 2$ and θ_τ chosen elementwise to satisfy the inequalities in Equation 2.41. Then for all $u \in H^{2,1}(\mathcal{T}) \cap H_0^{1,0}(Q)$ and $v \in V_h$,*

$$\left|\overline{B}_h(u,v)\right| \leq C_b |||u|||_{h,*} |||v|||_h \tag{2.43}$$

for some constant $C_b > 0$.

Proof. The full expression for $\left|\overline{B}(u,v)\right|$ may be broken into several parts and estimated individually. Indeed,

$$\begin{aligned}\left|\overline{B}_h(u,v)\right| \leq \int_{\Sigma_T} |uv|\,dy + \sum_{\tau\in\mathcal{T}} \int_\tau \left|\nabla^T v D_{h,\theta} \nabla u\right| dy + \int_\tau \left|\theta_\tau h_\tau^2 (b\cdot\nabla v)\nabla\cdot(D\nabla u)\right| dy \\ + \int_\tau \left|(\theta_\tau h_\tau^2 (c + \nabla\cdot b) - 1)(b\cdot\nabla v) v\right| dy + \int_\tau |cuv|\,dy.\end{aligned} \tag{2.44}$$

We shall bound each term of Equation 2.44 by the right-hand side of Equation 2.43, moving in order from the first to last term.

By the Cauchy-Schwarz inequality, the first term satisfies

$$\int_{\Sigma_T} |uv|\,dy \leq ||u||_{L^2(\Sigma_T)} ||v||_{L^2(\Sigma_T)} \leq |||u|||_h |||v|||_h \leq |||u|||_{h,*} |||v|||_h. \tag{2.45}$$

Similarly, for each $\tau \in \mathcal{T}$, the Cauchy-Schwarz inequality yields

$$\int_\tau \left|\nabla^T v D_{h,\theta} \nabla u\right| dy \leq \left\|D_{h,\theta}^{1/2} \nabla u\right\|_{L^2(\tau)} \left\|D_{h,\theta}^{1/2} \nabla v\right\|_{L^2(\tau)}$$

and therefore after applying the Cauchy-Schwarz inequality for sums,

$$\begin{aligned}\sum_{\tau\in\mathcal{T}} \int_\tau \left|\nabla^T v D_{h,\theta} \nabla u\right| dy \leq \sum_{\tau\in\mathcal{T}} \left\|D_{h,\theta}^{1/2} \nabla u\right\|_{L^2(\tau)} \left\|D_{h,\theta}^{1/2} \nabla v\right\|_{L^2(\tau)} \leq \left(\sum_{\tau\in\mathcal{T}} \left\|D_{h,\theta}^{1/2} \nabla u\right\|_{L^2(\tau)}^2\right)^{\frac{1}{2}} \left(\sum_{\tau\in\mathcal{T}} \left\|D_{h,\theta}^{1/2} \nabla v\right\|_{L^2(\tau)}^2\right)^{\frac{1}{2}} \\ \leq |||u|||_{h,*} |||v|||_h.\end{aligned} \tag{2.46}$$

For the next sub-expression, the contribution of the $\theta_\tau h_\tau^2$ term can be split such that

$$\int_\tau |\theta_\tau h_\tau^2 (b \cdot \nabla v) \nabla \cdot (D \nabla u)| \, dy \leq \left\| \theta_\tau^{1/2} h_\tau \nabla \cdot (D \nabla u) \right\|_{L^2(\tau)} \left\| \theta_\tau^{1/2} h_\tau b \cdot \nabla v \right\|_{L^2(\tau)}.$$

Furthermore, by the following bound:

$$\left\| \theta_\tau^{1/2} h_\tau b \cdot \nabla v \right\|_{L^2(\tau)}^2 = \int_\tau \theta_\tau h_\tau^2 \nabla^T v b b^T \nabla v \, dy \leq \int_\tau \nabla^T v D_{h,\theta} \nabla v \, dy \leq \left\| D_{h,\theta}^{1/2} \nabla v \right\|_{L^2(\tau)}^2$$

and another application of the Cauchy Schwarz inequality, the third term is bounded as

$$\sum_{\tau \in \mathcal{T}} \int_\tau |\theta_\tau h_\tau^2 (b \cdot \nabla v) \nabla \cdot (D \nabla u)| \, dy \leq \left(\sum_{\tau \in \mathcal{T}} \theta_\tau h_\tau^2 \| \nabla \cdot (D \nabla u) \|_{L^2(\tau)}^2 \right)^{\frac{1}{2}} \left(\sum_{\tau \in \mathcal{T}} \left\| D_{h,\theta}^{1/2} \nabla v \right\|_{L^2(\tau)}^2 \right)^{\frac{1}{2}} \leq |||u|||_{h,*} |||v|||_h. \tag{2.47}$$

For the fourth term of Equation 2.44, we first note that since θ_τ satisfies Equation 2.41, the estimate $|\theta_\tau h_\tau^2 (c + \nabla \cdot b) - 1| \leq \frac{3}{2}$ holds. Therefore

$$\int_\tau |(\theta_\tau h_\tau^2 (c + \nabla \cdot b) - 1)(b \cdot \nabla v) u| \, dy \leq \frac{3}{2} \int_\tau |b \cdot \nabla v| |u| \, dy \leq \frac{3}{2} \left\| \theta_\tau^{-1/2} h_\tau^{-1} u \right\|_{L^2(\tau)} \left\| \theta_\tau^{1/2} h_\tau b \cdot \nabla v \right\|_{L^2(\tau)}$$

and thus

$$\begin{aligned} \sum_{\tau \in \mathcal{T}} \int_\tau |(\theta_\tau h_\tau^2 (c + \nabla \cdot b) - 1)(b \cdot \nabla v) v| \, dy &\leq \frac{3}{2} \left(\sum_{\tau \in \mathcal{T}} \theta_\tau^{-1} h_\tau^{-2} \|u\|_{L^2(\tau)}^2 \right)^{\frac{1}{2}} \left(\sum_{\tau \in \mathcal{T}} \theta_\tau h_\tau^2 \|b \cdot \nabla v\|_{L^2(\tau)}^2 \right)^{\frac{1}{2}} \\ &\leq \frac{3}{2} |||u|||_{h,*} |||v|||_h. \end{aligned} \tag{2.48}$$

Finally, the last term may be estimated in terms of the upper bound $\overline{c} = \max_Q |c|$ via the Cauchy-Schwarz inequality once more:

$$\sum_{\tau \in \mathcal{T}} \int_\tau |cuv| \, dy \leq \overline{c} \sum_{\tau \in \mathcal{T}} \|u\|_{L^(\tau)} \|v\|_{L^2(\tau)} \leq \overline{c} \left(\sum_{\tau \in \mathcal{T}} \|u\|_{L^2(\tau)}^2 \right)^{\frac{1}{2}} \left(\sum_{\tau \in \mathcal{T}} \|v\|_{L^2(\tau)}^2 \right)^{\frac{1}{2}} \leq \overline{c} |||u|||_{h,*} |||v|||_h. \tag{2.49}$$

Having estimated each term of $\left| \overline{B}_h(u, v) \right|$ separately, by combining Equations 2.45 to 2.49, the proof is complete. □

We recall from the derivation of Problem 7 that $\overline{B}_h$ and $\overline{L}_h$ were defined so that the finite element

scheme is consistent with the strong form of the transient convection-diffusion equation. Thus if $u \in H_0^{2,1}(Q)$ is a strong solution, then

$$\overline{B}_h(u, v) = \overline{L}_h(v) \qquad \text{and} \qquad \overline{B}_h(u_h, v) = \overline{L}_h(v) \qquad \text{for all } v \in V_h.$$

From this we conclude that for any $v \in V_h$, $\overline{B}_h(u - u_h, v) = 0$; that is, the finite element error $u - u_h$ possesses Galerkin orthogonality.

Proposition 2.13. *Let $u \in H^{2,1}(\mathcal{T}) \cap H_0^{1,0}(Q)$ be the solution to Problem 5, and let $u_h \in V_h$ be the finite element solution. Then u_h satisfies the best-approximation property:*

$$\begin{aligned} |||u - u_h|||_h^2 &\leq \inf_{v \in V_h} \left(1 + \frac{C_b^2}{C_c^2}\right) |||u - v|||_h^2 + \frac{C_b^2}{C_c^2} \sum_{\tau \in \mathcal{T}} \theta_\tau h_\tau^2 ||\nabla \cdot (D\nabla(u - v))||_{L^2(\tau)}^2 + \theta_\tau^{-1} h_\tau^{-2} ||u - v||_{L^2(\tau)}^2 \\ &\leq \left(1 + \frac{C_b^2}{C_c^2}\right) \inf_{v \in V_h} |||u - v|||_{h,*}^2. \end{aligned} \tag{2.50}$$

Proof. Let $v \in V_h$ be arbitrary. Since $u_h - v \in V_h$, the coercivity of $\overline{B}_h$ with respect to $|||\cdot|||_h$ means that

$$|||u_h - v|||_h^2 \leq \frac{1}{C_c} \overline{B}_h(u_h - v, u_h - v).$$

Since $\overline{B}_h(u - u_h, v) = 0$ for all $v \in V_h$ and the form $\overline{B}_h$ is bounded with respect to $|||\cdot|||_{h,*}$, we have

$$\overline{B}_h(u_h - v, u_h - v) = \overline{B}_h(u - v, u_h - v) \leq C_b |||u - v|||_{h,*} |||u_h - v|||_h, \tag{2.51}$$

and therefore

$$|||u_h - v|||_h \leq \frac{C_b}{C_c} |||u - v|||_{h,*}. \tag{2.52}$$

Now by definition, $|||w|||_h \leq |||w|||_{h,*}$ for any $w \in H_0^{2,1}(Q)$, so the triangle inequality applied to the left-

hand side of Equation 2.50 yields

$$\begin{aligned}
|||u - u_h|||_h^2 &\leq |||u - v|||_h^2 + |||u_h - v|||_h^2 \\
&\leq |||u - v|||_h^2 + \frac{C_b^2}{C_c^2} |||u - v|||_{h,*}^2 \\
&\leq \left(1 + \frac{C_b^2}{C_c^2}\right) |||u - v|||_h^2 + \frac{C_b^2}{C_c^2} \sum_{\tau \in \mathcal{T}} \theta_\tau h_\tau^2 \|\nabla \cdot (D\nabla(u - v))\|_{L^2(\tau)}^2 + \theta_\tau^{-1} h_\tau^{-2} \|u - v\|_{L^2(\tau)}^2 \\
&= \left(1 + \frac{C_b^2}{C_c^2}\right) |||u - v|||_{h,*}^2.
\end{aligned} \tag{2.53}$$

Since $v \in V_h$ was arbitrary, Equation 2.53 holds for the infimum over V_h as well. □

Approximation by Polynomials

In order to prove an *a priori* error estimate for the finite element solution, we will quantify the polynomial interpolation error in terms of the norms $|||\cdot|||_h$ and $|||\cdot|||_{h,*}$. In many cases, this is done with the standard Lagrange interpolator (see, e.g., [9], Theorem 4.4.4 or [12], Theorems III.15.3 and III.16.1).

Definition 2.14. Let $\mathcal{T}$ be some triangulation of Q and $\mathcal{V}(\mathcal{T}) = \{y_i\}_{i=1}^N$ the vertex set of $\mathcal{T}$. In addition, let p_i be the standard r-degree Lagrange basis polynomial which is 1 on y_i and 0 on all other $y_j \in \mathcal{V}(\mathcal{T})$. Finally, let v be a continuous function on Q. The *Lagrange interpolation operator*

$$\begin{aligned}
\mathcal{I}_h &: C(Q) \to V_h \\
w &\mapsto \mathcal{I}_h w
\end{aligned} \tag{2.54}$$

is defined such that

$$\mathcal{I}_h w = \sum_{i=1}^N w(y_i) p_i. \tag{2.55}$$

The Lagrange interpolator maps any continuous function into the finite element space V_h by evaluating the function w at each Lagrange node. This simple construction satisfies a number of nice properties, making it a useful tool in the error analysis of finite element methods. In particular, the difference between a continuous function w and its Lagrange interpolant can be quantified in terms of the weak derivatives of w.

Proposition 2.15. *Let $\tau \subset \mathbb{R}^{d+1}$ be a $(d+1)$-simplex and $w \in H^{r+1}(Q) \cap C(Q)$. Then for $0 \leq l \leq r+1$,*

$$|w - \mathcal{I}_h w|_{H^l(\tau)} \leq C h_\tau^{r+1-l} |w|_{H^{r+1}(\tau)} \tag{2.56}$$

where h_τ is the diameter of τ and C is independent of h_τ.

The requirement that w be continuous is critical to the construction of $\mathcal{I}_h$, and is often satisfied naturally by weak solutions to PDEs. The Sobolev embdedding theorem states that if $\Omega \subset \mathbb{R}^n$, then there is a continuous inclusion map from $H^k(\Omega)$ into $C(\Omega)$ when $k > n/2$. In other words, when $k > n/2$, every function in $H^k(\Omega)$ may be identified with a continuous function (and the $C(\Omega)$ norm varies continuously with the $H^k(\Omega)$ norm). When PDEs are posed on domains of dimension three or less, this means that point values are well-defined for finite element solutions in $H^2(\Omega)$.

However, in four-dimensional space-time domains, even relatively smooth functions $u \in H^2(Q)$ are not necessarily continuous, and Lagrange interpolation is not well-defined. To remedy this, we may apply the theory of *quasi-interpolation operators*, which are a generalization of Lagrange interpolators that do not rely on pointwise evaluations. A comprehensive treatment of polynomial interpolation is outside the scope of this dissertation and has been well-developed elsewhere. A comparison of quasi-interpolation operators may be found in [50]; additional background can be found in the survey [1].

In the present study, we shall apply the quasi-interpolant of Scott and Zhang [43], which is a nodal interpolator based on local integrals, not point evaluations. Let $\tilde{\mathcal{I}}_h$ denote the r-degree Scott-Zhang interpolation operator, which is well-defined for all functions $w \in H^{\frac{1}{2}}(Q)$. For each $\tau \in \mathcal{T}$, define

$$S_\tau = \bigcup \{\tau' \in \mathcal{T} : \tau' \cap \tau \neq \emptyset\}, \tag{2.57}$$

which is a neighborhood of τ containing every simplex adjacent to τ. Due to the structure of the Scott-Zhang interpolant, many element-wise estimates will be stated in terms of S_τ. We may now state the fundamental approximability result for this interpolant; for further details see [43].

Proposition 2.16. *Let $\tau \subset \mathbb{R}^{d+1}$ be a $(d+1)$-simplex, l be a non-negative integer, $\tau \in \mathbb{R}^{d+1}$, and $w \in H^m(Q)$, where $0 \leq l \leq m \leq r+1$. Then*

$$|w - \tilde{\mathcal{I}}_h w|_{H^l(\tau)} \leq C h_\tau^{m-l} |w|_{H^m(S_\tau)}. \tag{2.58}$$

where h_τ is the diameter of τ and C is independent of h_t.

By combining the best approximation result in Proposition 2.13 with the approximability result of Proposition 2.16, we can establish an asymptotic bound on the finite element error. However, first we must extend Proposition 2.16 to the norms $|||\cdot|||_h$ and $|||\cdot|||_{h,*}$.

Lemma 2.17. *Let $Q \subset \mathbb{R}^{d+1}$ be a space-time domain, $\mathcal{T}$ a shape-regular, quasi-uniform triangulation over Q, and V_h the previously-defined finite element space of piecewise degree-r polynomials. Furthermore, let $w \in H^m(Q) \cap H_0^{2,1}(\mathcal{T})$ and assume the polynomial degree r satisfies $1 \leq r \leq \lfloor m \rfloor$. Then the r-degree Scott-Zhang quasi-interpolation operator $\tilde{\mathcal{I}}_h$ satisfies*

$$|||w - \tilde{\mathcal{I}}_h w|||_h^2 \leq C \sum_{\tau \in \mathcal{T}} h_\tau^{2m-2} |w|_{H^m(S_\tau)}^2. \tag{2.59}$$

Proof. Let $E_h = w - \tilde{\mathcal{I}}_h w$. Each term in

$$|||E_h|||_h^2 = \|E_h\|_{L^2(\Sigma_T)}^2 + \|E_h\|_{L^2(\Sigma_0)}^2 + \sum_{\tau \in \mathcal{T}} \left\| D_{h,\theta}^{1/2} \nabla E_h \right\|_{L^2(\tau)}^2 + \|E_h\|_{L^2(\tau)}^2$$

may be bounded individually.

For the first term, we have

$$\|E_h\|_{L^2(\Sigma_T)}^2 = \sum_{\tau \in \mathcal{T}} \int_{\Sigma_T \cap \tau} E_h^2 \, dy = \sum_{\tau \in \mathcal{T}} \int_{F_\tau} E_h^2 \, dy = \sum_{\tau \in \mathcal{T}} \|E_h\|_{L^2(F_\tau)}^2$$

where each F_τ is the d-face F which is contained in Σ_T (if such a face exists). Now since $w \in H^m(Q)$ with $m > 1$ and $\overline{\mathcal{I}}_h w \in V_h \subset H^1(Q)$, $E_h \in H^1(Q)$ as well. Therefore we can apply the trace inequality Lemma 1.25 and Proposition 2.16 to obtain

$$\begin{aligned} \|E_h\|_{L^2(\Sigma_T)}^2 = \sum_{\tau \in \mathcal{T}} \|E_h\|_{L^2(F_\tau)}^2 &\leq \sum_{\tau \in \mathcal{T}} \frac{1}{\sigma} h_\tau \|\nabla E_h\|_{L^2(\tau)}^2 + \frac{1+d}{\sigma} h_\tau^{-1} \|E_h\|_{L^2(\tau)}^2 \\ &\leq C_1 \sum_{\tau \in \mathcal{T}} h_\tau^{2m-1} |w|_{H^m(S_\tau)}^2. \end{aligned}$$

The same argument can be applied to the second term, changing Σ_T to Σ_0, to arrive at the same bound for the second term.

The third and fourth terms of $|||E_h|||_h^2$ may be estimated with Proposition 2.16, using the fact that

$D_{h,\theta}$ is a bounded operator:

$$\sum_{\tau\in\mathcal{T}}\left\|D_{h,\theta}^{1/2}\nabla E_h\right\|_{L^2(\tau)}^2+\|E_h\|_{L^2(\tau)}^2\leq\sum_{\tau\in\mathcal{T}}\left\|D_{h,\theta}^{1/2}\right\|\|\nabla E_h\|_{L^2(\tau)}^2+\|E_h\|_{L^2(\tau)}^2$$
$$\leq C_2\sum_{\tau\in\mathcal{T}}h_\tau^{2m-2}|w|_{H^m(S_\tau)}^2.$$

Combining the estimates for each of the four terms, we have

$$|||E_h|||_h^2\leq(2+hC_1+C_2)\sum_{\tau\in\mathcal{T}}h_\tau^{2m-2}|w|_{H^m(S_\tau)}^2. \qquad \square$$

Lemma 2.18. *Let the assumptions of Lemma 2.17 hold. Then*

$$\sum_{\tau\in\mathcal{T}}\theta_\tau h_\tau^2\left\|\nabla\cdot(D\nabla(w-\tilde{\jmath}_h w))\right\|_{L^2(\tau)}^2+\theta_\tau^{-1}h_\tau^{-2}\left\|w-\tilde{\jmath}_h w\right\|_{L^2(\tau)}^2\leq C\sum_{\tau\in\mathcal{T}}h_\tau^{2m-2}\left(|w|_{H^m(S_\tau)}^2+\|w\|_{H^2(\tau)}^2\right). \tag{2.60}$$

Proof. The second term on the left-hand side of Equation 2.60 may be bounded immediately by Proposition 2.16:

$$\sum_{\tau\in T}\theta_\tau^{-1}h_\tau^{-2}\left\|w-\tilde{\jmath}_h w\right\|_{L^2(\tau)}^2\leq C\sum_{\tau\in\mathcal{T}}h_\tau^{2m-2}\|w\|_{H^m(S_\tau)}^2, \tag{2.61}$$

where $C=\max_\tau\theta_\tau^{-1}$. Note that this constant is uniformly bounded since θ_τ is bounded from below even as $h_\tau\to 0$ (c.f. Remark 2.10).

To bound the first term, we recall that by assumption, the entries of the diffusion coefficient matrix are in $H^{1,0}(\mathcal{T})$. For $1\leq i,j\leq d$, let

$$\left\|D_{ij}\right\|_{H^{1,0}(\mathcal{T})}^2=\sum_{\tau\in\mathcal{T}}\left\|D_{ij}\right\|_{L^2(\tau)}^2+\left\|\nabla_x D_{ij}\right\|_{L^2(\tau)}^2.$$

We shall also use the shorthand

$$C_{\tau,ij}^0:=\left\|D_{ij}\right\|_{L^2(\tau)}^2\qquad\text{and}\qquad C_{\tau,ij}^1:=\left\|\nabla_x D_{ij}\right\|_{L^2(\tau)}^2.$$

Under this notation, the first term satisfies

$$\begin{aligned}
\left\|\nabla\cdot(D\nabla(w-\tilde{\mathcal{J}}_h w))\right\|^2_{L^2(\tau)} &= \sum_{i=1}^{d}\left\|\frac{\partial}{\partial x_i}\sum_{j=1}^{d} D_{ij}\frac{\partial}{\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)} \\
&\leq \sum_{i=1}^{d}\sum_{j=1}^{d}\left\|\left(\frac{\partial D_{ij}}{\partial x_i}\right)\frac{\partial}{\partial x_j}(w-\tilde{\mathcal{J}}_h w) + D_{ij}\frac{\partial^2}{\partial x_i\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)} \\
&\leq \sum_{i=1}^{d}\sum_{j=1}^{d}\left\|\frac{\partial D_{ij}}{\partial x_i}\right\|^2_{L^2(\tau)}\left\|\frac{\partial}{\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)} + \left\|D_{ij}\right\|^2_{L^2(\tau)}\left\|\frac{\partial^2}{\partial x_i\partial x_j}(w-\tilde{\mathcal{J}}_h(w))\right\|^2_{L^2(\tau)} \\
&\leq \sum_{i=1}^{d}\sum_{j=1}^{d} C^1_{\tau,ij}\left\|\frac{\partial}{\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)} + C^0_{\tau,ij}\left\|\frac{\partial^2}{\partial x_i\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)}.
\end{aligned}$$

Proposition 2.16 may be applied to the first term of this double sum, and thus

$$\left\|\nabla\cdot(D\nabla(w-\tilde{\mathcal{J}}_h w))\right\|^2_{L^2(\tau)} \leq C^1_\tau h^{2m-2}_\tau |w|^2_{H^m(S_\tau)} + \sum_{i=1}^{d}\sum_{j=1}^{d} C^0_{\tau,ij}\left\|\frac{\partial^2}{\partial x_i\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)} \tag{2.62}$$

where $C^1_\tau = \sum_i \sum_j C^1_{\tau,ij}$.

Finally, the last term of the above equation can be treated in two cases. If $m \geq 2$, then Proposition 2.16 implies that

$$\sum_{i=1}^{d}\sum_{j=1}^{d} C^0_{\tau,ij}\left\|\frac{\partial^2}{\partial x_i\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)} \leq C^0_\tau h^{2m-4}_\tau |w|^2_{H^m(S_\tau)}. \tag{2.63}$$

where $C^0_\tau = \sum_i \sum_j C^0_{\tau,ij}$. If $m < 2$, then $r = 1$ and $\tilde{\mathcal{J}}_h w$ is piecewise-linear. In this case, all second derivatives of $\tilde{\mathcal{J}}_h w$ vanish and

$$\begin{aligned}
\sum_{i=1}^{d}\sum_{j=1}^{d} C^0_{\tau,ij}\left\|\frac{\partial^2}{\partial x_i\partial x_j}(w-\tilde{\mathcal{J}}_h w)\right\|^2_{L^2(\tau)} &\leq \sum_{i=1}^{d}\sum_{j=1}^{d} C^0_{\tau,ij}\left\|\frac{\partial^2 w}{\partial x_i\partial x_j}\right\|^2_{L^2(\tau)} \\
&\leq \sum_{i=1}^{d}\sum_{j=1}^{d} C^0_{\tau,ij}|w|^2_{H^{2,0}(\tau)} \\
&= C^0_\tau |w|^2_{H^{2,0}(\tau)} = C^0_\tau h^{4-2m}_\tau h^{2m-4}_\tau |w|^2_{H^{2,0}(\tau)}.
\end{aligned} \tag{2.64}$$

Note that since $m < 2$ in this case, h^{4-2m}_τ is uniformly bounded above by some constant C'.

Combining Equation 2.63 and Equation 2.64 and setting $C'' = \max(1, C')$, we conclude that for

any $m \geq 1$,

$$\theta_\tau h_\tau^2 \sum_{i=1}^{d}\sum_{j=1}^{d} C^0_{\tau,ij}\left\|\frac{\partial^2}{\partial x_i \partial x_j}(w - \tilde{\jmath}_h w)\right\|^2_{L^2(\tau)} \leq C^0_\tau \theta_\tau C'' h_\tau^{2m-2}\left(|w|^2_{H^m(S_\tau)} + |w|^2_{H^{2,0}(\tau)}\right)$$

Returning now to Equation 2.62, and summing over all $\tau \in \mathcal{T}$, we conclude

$$\begin{aligned}\sum_{\tau\in\mathcal{T}} \theta_\tau h_\tau^2 \left\|\nabla\cdot(D\nabla(w - \tilde{\jmath}_h w))\right\|^2_{L^2(\tau)} &\leq \sum_{\tau\in\mathcal{T}} C^1_\tau \theta_\tau h_\tau^{2m}|w|^2_{H^m(S_\tau)} + C^0_\tau\theta_\tau C'' h_\tau^{2m-2}\left(|w|^2_{H^m(S_\tau)} + |w|^2_{H^{2,0}(\tau)}\right)\\ &\leq C\sum_{\tau\in\mathcal{T}} h_\tau^{2m-2}\left(|w|^2_{H^m(S_\tau)} + |w|^2_{H^{2,0}(\tau)}\right),\end{aligned}$$

which completes the proof. □

Having established the approximability of Scott-Zhang interpolation in terms of the norms $|||\cdot|||_h$ and $|||\cdot|||_{h,*}$, we may now combine these results with the best-approximation property in Proposition 2.13 to quantify the finite element error.

Theorem 2.19. *Let $Q \subset \mathbb{R}^{d+1}$ be a space-time domain and $\mathcal{T}$ a shape-regular, quasi-uniform triangulation over Q. Suppose $u \in H^m(Q) \cap H^{2,1}_0(\mathcal{T})$ is the weak solution to the transient convection-diffusion equation (Problem 5) and u_h is the solution to the SUPG finite element method (Problem 7) with degree-r shape functions. Then*

$$|||u - u_h|||^2_h \leq C\sum_{\tau\in\mathcal{T}} h_\tau^{2k-2}\left(|u|^2_{H^k(S_\tau)} + |u|^2_{H^2(\tau)}\right)$$

where $k = \min(m, r+1)$ and S_τ is the neighborhood of τ defined in Equation 2.57 for Scott-Zhang quasi-interpolation.

Proof. If $m < r+1$, let $\tilde{\jmath}_h$ be the $\lfloor m\rfloor$-degree Scott-Zhang interpolator, otherwise let $\tilde{\jmath}_h$ be the r-degree Scott-Zhang interpolator. In both cases, Lemma 2.17 and Lemma 2.18 apply (with "m" in the statement of the lemmas changed to "k" as defined in the theorem). Since $\tilde{\jmath}_h u \in V_h$, Proposition 2.13 implies

$$\begin{aligned}|||u - u_h|||^2_h &\leq \inf_{v\in V_h}\left(1 + \frac{C_b^2}{C_c^2}\right)|||u - v|||^2_h + \frac{C_b^2}{C_c^2}\sum_{\tau\in\mathcal{T}} \theta_\tau h_\tau^2 \|\nabla\cdot(D\nabla(u-v))\|^2_{L^2(\tau)} + \theta_\tau^{-1}h_\tau^{-2}\|u - v\|^2_{L^2(\tau)}\\ &\leq \left(1 + \frac{C_b^2}{C_c^2}\right)|||u - \tilde{\jmath}_h u|||^2_h + \frac{C_b^2}{C_c^2}\sum_{\tau\in\mathcal{T}} \theta_\tau h_\tau^2\left\|\nabla\cdot(D\nabla(u - \tilde{\jmath}_h u))\right\|^2_{L^2(\tau)} + \theta_\tau^{-1}h_\tau^{-2}\left\|u - \tilde{\jmath}_h u\right\|^2_{L^2(\tau)}.\end{aligned}$$

Applying to Lemma 2.17 and Lemma 2.18 to the above expression proves the theorem. □

By separating out the constitutive terms of $|||\cdot|||_h$, Theorem 2.19 provides estimates on several different kinds of error. Expanding the term $|||u-u_h|||_h^2$ on the left-hand side of Theorem 2.19,

$$\left.\begin{aligned} \|u-u_h\|^2_{L^2(\Sigma_T)} \\ \sum_{\tau\in\mathcal{T}} \|u-u_h\|^2_{L^2(\tau)} \\ \kappa \sum_{\tau\in\mathcal{T}} \|\nabla_x(u-u_h)\|^2_{L^2(\tau)} \\ \sum_{\tau\in\mathcal{T}} \theta_\tau h_\tau^2 \|b\cdot\nabla(u-u_h)\|^2_{L^2(\tau)} \end{aligned}\right\} \leq |||u-u_h|||_h^2 \leq C \sum_{\tau\in\mathcal{T}} h_\tau^{2k-2}\left(|u|^2_{H^k(S_\tau)} + |u|^2_{H^2(\tau)}\right). \tag{2.65}$$

Thus the finite element error in $L^2(\Sigma_T)$ and $L^2(\mathcal{T})$ is $O(h^{k-1})$ which is slightly less than the optimal decay rate of $O(h^k)$. However, error in the spatial gradient is $O(h^{k-1})$, which is optimal. Finally, error of the gradient in the space-time streamline direction is $O(h^{k-2})$. In particular, this means that the solution u must be in $H^2(Q)$ in order to achieve asymptotic decay of the derivative in the streamline direction.

As previously mentioned, in the special case of piecewise linear elements, it is possible to construct a similar SUPG scheme where the coefficient on the upwinded term is $\theta_\tau h_\tau$, not $\theta_\tau h_\tau^2$. Under this scheme, the convergence of the streamline derivatives is $O(k-\frac{3}{2})$, which is better than the estimate in Equation 2.65 for $u\in H^k(Q)$ with $k<2$. In particular, this means that when $\frac{3}{2}<k\leq 2$, the streamline error $\|b\cdot\nabla(u-u_h)\|_{L^2(\tau)}$ will converge when using linear elements but may not converge for higher order elements. For a description of this alternative scheme, see [27].

2.5 Numerical Experiments

As a means of confirming the theory laid out in the prior sections, numerical experiments were conducted with an implementation of the space-time SUPG method. When the research for this dissertation began, no publicly-shared finite element codes treated the case of four-dimensional unstructured meshes. To remedy this, we developed a research code tailored to this dissertation which solves space-time parabolic equations in domains of dimension ≤ 4. We remark that in the recent series [27–29] the authors describe an implementation of space-time methods with the `MFEM`[33] library; however, this support does not appear to be included in the library documentation at present.

Our solver relies on the `Eigen` [17] library for basic linear algebra operations and uses the `Multigraph` [2] solver for sparse matrix solves. We create our own implementation of four-dimensional unstructured meshes by building a mesh data structure on top of the Combinatorial Map package of the `CGAL` [47] library.

In each of the following examples, we used piecewise-linear elements on uniformly refined meshes to study the asymptotic convergence of the method. All space-time meshes were constructed using the procedures of Chapter 3. The linear system is solved with `Multigraph`, which uses a composite step biconjugate gradient method with a 2-level preconditioner based on ILU factorization. In each test we choose the drop tolerance for the ILU factorizations to be 10^{-8}; since we were not concerned with time-to-solution, no effort was made to optimize the drop tolerance. All numerical experiments were run in serial on a single compute node of the Comet supercomputer at the San Diego Supercomputer Center.

As a first example, we study the heat equation in three-dimensional space-time on the box domain $[-1,1]\times[-1,1]\times[0,2]$. We consider two functions:

$$u_1(x,y,t) = \cos(x)e^{-t}$$
$$u_2(x,y,t) = \cos(x) + y^2 + t$$

and solve the model problem where $\widetilde{D} = I$, $\tilde{b} = 0$, $c = 0$, and f is chosen such that u_1 or u_2 is the true solution. Boundary conditions on Σ_0 and Σ are imposed which are in agreement with the true solution.

Figure 2.1 shows the convergence of the error in $L^2(Q)$ as the number of degrees of freedom increases. In the case of u_1, we observe an $O(h)$ error decay rate as the mesh size decreases, particularly for $h \leq 0.1$. The decay rate of the $L^2(Q)$ error for u_2 is steeper than $O(h)$, although not quite $O(h^2)$. This may indicate that the mesh size is too coarse for the error to be governed by the asymptotic decay rates. However, due to the serial implementation of our solver, we were not able to test problem sizes with greater than 2.5M degrees of freedom.

It is also possible that the $|||\cdot|||_h$ error is dominated by the error of the derivatives, which decay like $O(h)$, but the L^2 error (shown above) actually decays at a faster rate. An immediate extension of the present research would be to further disect the various errors that comprise the $|||\cdot|||_h$ error; for instance, measuring the streamline and crosswind derivatives directly.

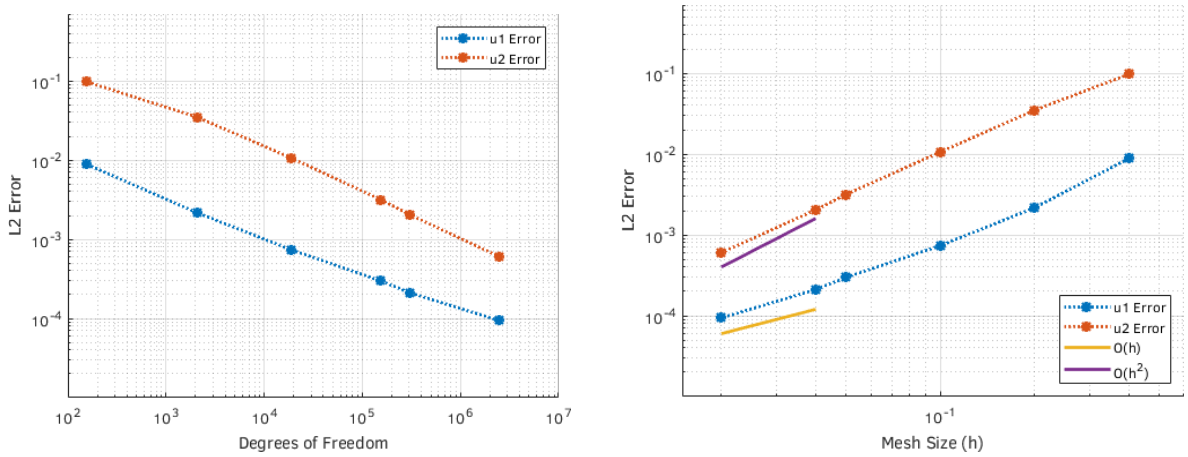

Figure 2.1: Convergence of finite element error in $L^2(Q)$ for the solution of the heat equation on a three-dimensional space-time domain.

We obtain similar results for tests in four-dimensional space-time. Taking again the computational domain to be a box, $Q = [1,3]\times[1,3]\times[1,3]\times[0,2]$, we define two more manufactured solutions. Let

$$u_3(x,y,z,t) = x^{2.5}y + t\cos(z)$$

$$u_4(x,y,z,t) = yt^{1.25} + xz^{1.75}$$

As before, we set $\widetilde{D} = I$, $\tilde{b} = 0$, $c = 0$, and manufacture the problem data to ensure that u_3 and u_4 are the true solutions. We remark that functions u_3 and u_4 are both smooth on the domain Q, since the spatial origin is excluded. At present, our implementation of the space-time finite element method cannot handle functions with singularities. This limits our ability to study the convergence of u_4 near the origin.

In this test, both problems display $O(h^2)$ convergence with respect to the mesh size. Due to the huge memory footprint necessary to maintain a four-dimensional mesh (both degrees of freedom and adjaceny information), these tests are further constrained in mesh size. A pentatope mesh with average edge length of just 0.1 requires 100,000 degrees of freedom in order to cover Q! These numerical tests underscore the double-edged nature of space-time methods, as discussed in [19]: these schemes require an increased overhead to set up, which is only compensated for when running in parallel.

It is clear that larger-scale numerical experiments will require a parallel implementation of our space-time solver, especially for four-dimensional problems. However, these tests confirm the theoretical error estimates that were established earlier in this chapter.

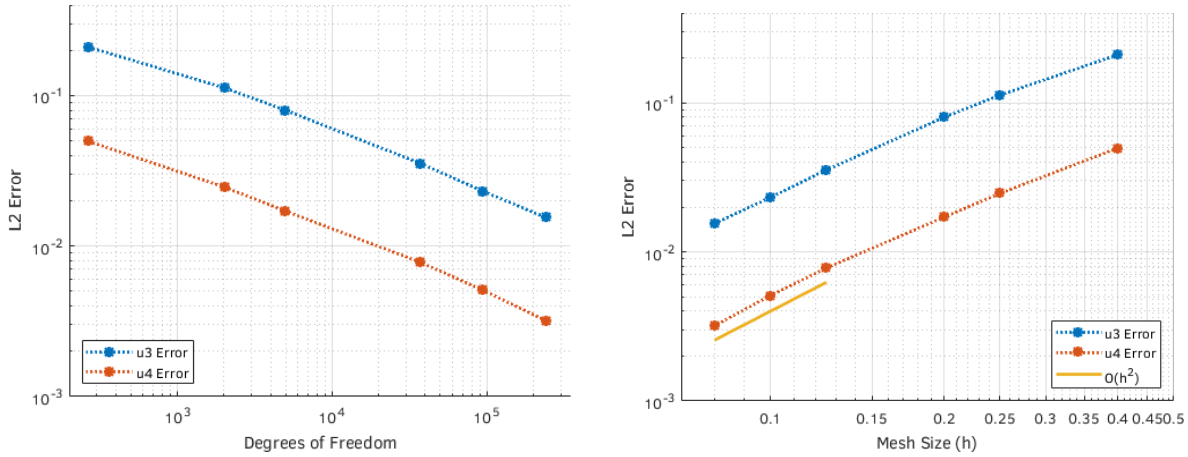

Figure 2.2: Convergence of finite element error in $L^2(Q)$ for the solution of the heat equation on a four-dimensional space-time domain.

Chapter 2, in part, is currently being prepared for submission for publication of the material. The dissertation author was the sole investigator and author of this material.

Chapter 3

Four-Dimensional Space-Time Meshes

In one way or another, every numerical algorithm that approximates the solution of a time-dependent PDE relies on a discretization of the space-time domain $Q = \Omega \times [0, T]$. However, this discretization is often implicit, serving a minor role in the overall description of the method. Very often, numerical methods maintain two separate discretizations: a d-dimensional tesselation of the spatial domain Ω and a one-dimensional subdivision of the time interval $[0, T]$. These two discretizations are, of course, the spatial mesh and the time steps which are ubiquitous in time-stepping algorithms.

By maintaining discretizations of space and time separately, time-stepping methods avoid the need to handle a $(d + 1)$-dimensional discretization of Q directly. For this reason, four-dimensional discretizations have been largely unstudied in the finite element literature. As we have discussed, however, a unified treatment of space and time means that high-dimensional meshes are unavoidable.

As we shall see in the present chapter, once the work has been done to set up four-dimensional unstructured meshes, there are a variety of benefits immediately at our fingertips. In the language of time-stepping methods, these includes generalizations of moving meshes and adaptive-time stepping, as well as time-dependent refinement and coarsening. What is paid for in abstraction is made up in flexibility.

The two major contributions of this chapter are a method for creating $(d + 1)$-dimensional space-time meshes from d-dimensional spatial meshes ($d \leq 3$) and a description of simplex bisection in four dimensions, which is adapted from Stevenson's method[45] for space-time meshes. Our mesh generation procedure is one of only a very limited number which appear in the literature, and we believe that it holds several advantanges over existing methods.

Before describing our proposed method for space-time mesh generation, it will be helpful to discuss a number of different structures that space-time meshes can possess. Some of these, like the implicit discretizations used in time-stepping methods, are highly structured. Our goal is to create a simplex mesh with as little imposed structure as possible, which will allow for the greatest level of freedom to refine and/or coarsen the mesh.

Types of Space-Time Meshes

A classification of space-time mesh elements was proposed by Behr[13], which delineates three types of space-time meshes (see Figure 3.1 for an illustration). In order of decreasing structure, these are:

1. *Flat Space-Time Meshes (FST)*: FST meshes are defined by a spatial discretization and a series of time intervals $[t_i, t_{i+1}]$, which are referred to as "time slabs." The space-time elements in an FST mesh are tensor product-type elements; for a spatial element $\tau \in \mathbb{R}^d$, an associated space-time element is

$$P_{\tau,i} = \{(x,t) : x \in \tau, t \in [t_i, t_{i+1}]\} = \tau \times [t_i, t_{i+1}]$$

   Classical time-stepping methods implicitly use FST meshes as a discretization of the space-time domain, where continuity is enforced within time slabs, but not across time slabs.

2. *Simplex Space-Time Meshes (SST)*: As the name implies, SST meshes are comprised of simplex-type elements, in contrast to the tensor product elements in FST meshes. SST meshes can be thought of as refinements of FST meshes, where each time slab $[t_i, t_{i+1}]$ has been tesselated into a collection of simplices. A defining feature of SST meshes is that the simplex elements do not cross time slab boundaries, and therefore there is an alignment of the simplex elements to the times t_i. We emphasize that SST meshes can contain new vertices with time coordinates in the interval (t_i, t_{i+1}), which allows for local temporal refinement.

3. *Unstructured Space-Time Meshes (UST)*: In a UST mesh, no distinction is made between the time and spatial dimensions, and the space-time discretization is a conforming collection of $(d+1)$-simplices in $\mathbb{R}^{d+1}$. In this setting, there are no time slabs and the boundaries of these simplex elements are not aligned in any particular way.

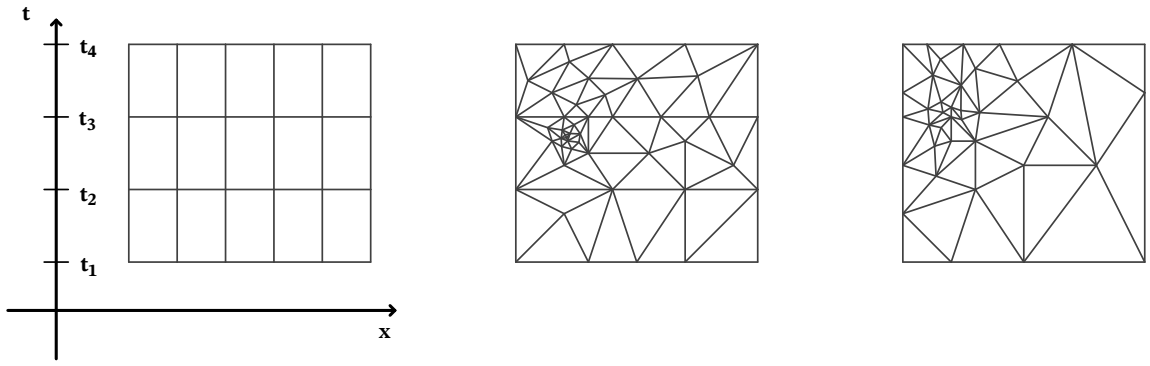

Figure 3.1: Types of space-time meshes associated to a 1D spatial domain. From left to right: Flat Space-Time, Simplex Space-Time, Unstructured Space-Time.

Each of the space-time mesh formats described above have strengths and weaknesses that affect their application to different kinds of problems. FST meshes, for instance, are often the simplest to implement. Due to the tensor product structure of FST elements, methods employing FST discretizations can maintain a d-dimensional spatial discretization and a 1-dimensional temporal discretization separately. This avoids the need to manage $(d+1)$-dimensional elements explicitly. In practice, many time-stepping methods can be formulated in terms of FST discretization. This ability to separate space and time discretizations in FST meshes, together with the widespread usage of time-stepping methods, is one reason that unified treatments of space-time discretizations have received relatively little study until recently.

Due to their structure with respect to the time dimension, FST meshes are well-suited to problems which evolve discontinuously in time. For instance, in applications of fracture mechanics with multiple prescribed fractures at different crack points, it is often desirable to allow for discontinuities at each time of interest.

On the other hand, FST meshes are severely limited by their inability to handle local temporal refinements. Because of the tensor product structure of FST elements, the temporal resolution $\Delta_i = t_{i+1} - t_i$ is global with respect to space; that is, the height of every element within each time slab is the same. In many numerical simulations of physical processes, there is some (often moving) region of the domain that requires a high time resolution to achieve a desired accuracy, while the rest of the domain can be modeled with a lower resolution. In the FST paradigm, the smallest time increment Δ_i must be applied to every spatial element τ no matter its location. This frequently generates space-time meshes

with far more elements than is necessary.

In the finite element literature, one technique for achieving local temporal refinement is adaptive time-stepping, in which the time evolution of different spatial elements can be performed on different temporal scales. Successful usage of adaptive time-stepping can dramatically reduce the computational work of a solver based on time steps, but the implementation and analyses of such methods can often be difficult to transfer from one application to another.

A major strength of SST discretizations is the ability to achieve local temporal refinement in a natural way; that is, without relying on any specific equation or spatial discretization. Since every time slab in an SST mesh is tesselated with simplex elements, the mesh elements within each time slab can be refined adaptively to vary mesh resolution in both time and space simultaneously. The cost of this flexibility, however, is that the elements in SST meshes are truly spatiotemporal - they cannot be described by independent spatial and temporal discretizations. Thus the primary trade-off in moving from an FST to a SST mesh is the addition of local space-time adaptivity at the expense of a higher-dimensional representation.

Simplex Space-Time methods can also be considered as a generalization of moving-mesh methods. In moving-mesh methods, a time-stepping strategy is employed, but the spatial position of mesh nodes can differ at each time step, essentially "moving the mesh" as the time t increases. In SST meshes, the location of mesh vertices lying on the hyperplanes $t = t_i$ and $t = t_{i+1}$ need not match; in effect, the spatial position of degrees of freedom "move" from time t_i to t_{i+1}. It should also be noted that the total number of mesh nodes lying on the hyperplanes $t = t_i$ and t_{i+1} need not be the same, which allows the overall spatial mesh size (and even mesh topology) to change from time t_i to t_{i+1}. While moving mesh methods can be combined with h-refinement schemes to accommodate the insertion and deletion of nodes, creating a robust decision criterion to guide the h-refinement in concert with the mesh movement poses a number of challenges [16].

While the difference between FST and SST meshes is evident from the shapes of their constituent elements, the difference between SST and UST meshes is more subtle, and even somewhat philosophical. Since UST meshes are any simplicial subdivision of a space-time domain, there are no time slabs to speak of, and therefore no analogy to time-stepping methods. UST meshes are most appropriate for problems where space-time elements need not be aligned to temporal boundaries, or those temporal boundaries are not known ahead of time.

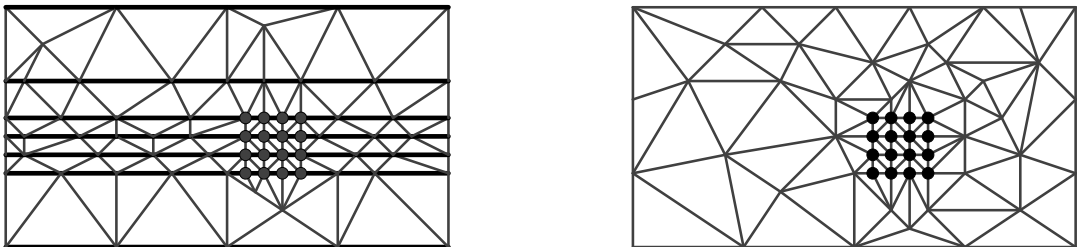

Figure 3.2: Examples of SST (left) and UST (right) meshes containing a subset of closely-packed vertices. The bold horizontal lines at left represent time slab boundaries.

The primary benefit of UST discretizations is that no restrictions are imposed on the temporal coarseness of the mesh. In SST meshes, by contrast, no space-time element can span more than one time slab. This can be a notable constraint if, for instance, several short time slabs are specified, but the solution is smooth in the majority of the spatial domain during these times (by "short" time slabs we mean Δ_i is small).

As an example, consider an earthquake propagation simulation based on a kinematic source model (see, e.g., [39, 46, 53]). In these applications, the nucleation (essentially, the "start") of an earthquake is described as a tight cluster of prescribed movements at pre-specified times and locations. As a result, mesh nodes near the nucleation site and time must be aligned to these tightly-spaced spatiotemporal points, even though the vast majority of the domain (away from the seismic fault) is in a constant state. If an SST mesh were used to enforce time steps at each of these points, the temporal resolution far from the nucleation would be very fine, since space-time elements cannot cross the boundaries of the prescribed time slabs. However, a UST mesh for the same application could contain very coarse elements away from the nucleation zone, which taper to a fine, regular mesh in the immediate vicinity of nucleation. See Figure 3.2 for an illustration of these two scenarios.

Of course, the problematic SST mesh in Figure 3.2 could be avoided if the time slabs were not aligned to the prescribed times of the source model; it is equally possible to enclose the entire nucleation region in a large time slab and then prescribe a locally-refined mesh in the interior in the same manner as a UST mesh. This brings us to the "philosophical" distinction between SST and UST meshes. As a limiting case, UST meshes are just SST meshes where there is only one time slab (that is, the slab between the initial and final time).

While UST meshes may be considered to be a special subclass of SST meshes, it is the author's

view that there should be some distinction made between meshes which bear no global dependence on a sequence of "notable times" associated to the underlying application, and those which do. As such, we consider UST meshes to be discretizations in which no attempt is made to align simplex boundaries to specific times.[1] By contrast, SST meshes are characterized by elements which possess some structure in the time dimension that is not present in the spatial dimensions.

In this dissertation, we will consider space-time meshes with simplicial elements. Our presentation will require a set of "time steps" $t_0, t_1, ...$ for the construction of a coarse space-time mesh only, and these time steps may be chosen almost arbitrarily. Therefore, the initial coarse mesh construction described in this chapter yields an SST mesh. The given time steps will be ignored after the construction of the initial mesh, at which point we investigate some methods for producing a truly unstructured space-time mesh.

3.1 Construction of Space-Time Meshes

Some of the first work to address unstructured space-time mesh generation arose in the context of spacetime-discontinuous Galerkin (SDG) methods. In 2000, Ungör and Scheffer proposed the Tent Pitcher algorithm [49], which is an advancing front method for producing $(d+1)$-dimensional meshes from d-dimensional meshes. At the time of its introduction, mesh generation based on Tent Pitcher only handled problems in two spatial dimensions. In 2012, Mont[36] extended the algorithm to three spatial dimensions; nevertheless, their methodology avoids treating four-dimensional mesh elements explicitly.

Space-time meshes described explicitly in terms of four-dimensional elements were explored briefly in the 2004 thesis of Sathe[42], but no systematic method for producing four-dimensional meshes was presented. In their work, the space-time refinement scheme was dubbed "Enhanced-Discretization Space-Time Technique - Single Mesh," or EDSTT-SM.

In 2008, Behr[5] introduced a method for creating simplex space-time meshes, starting from a given spatial mesh. Behr's method is based on an extrusion of the spatial mesh into a series of time slabs, which are then refined into simplices by adding new vertices and performing a Delaunay triangulation.

[1] A reasonable exception to this rule are meshes which are partitioned and distributed to parallel processors along temporal boundaries. In this setting, the mesh on each processor is unstructured; furthermore, the partition along temporal boundaries is an implementation artifact, not an *a priori* time constraint.

Another method for constructing four-dimensional meshes is to create a simple, coarse initial mesh and then adaptively refine the mesh until a given criteria is met. An example of this method is demonstrated in the 2019 thesis of Caplan[11], where the hypercube is initially triangulated with Kuhn simplices and then adaptively refined with Caplan's adaptive mesher (which is the primary subject of the thesis).

The mesh construction procedure proposed in this dissertation is most closely related to that of Behr. The present method was developed independently, with the connection to Behr's construction being discovered after the initial research was complete. In addition, there are a few key differences between these methods, which we highlight here. The primary difference in methodology comes from the process of incorporating local temporal refinement into the space-time mesh. In both methods, the construction begins by extruding a spatial mesh into a sequence of space-time prism elements. At this point, the method we present here immediately subdivides each prism into simplicial elements in a deterministic way. Temporal refinement is achieved in a post-processing step, using a method based on element bisection. In contrast, the method of Behr adds new vertices to the space-time prism elements to achieve temporal refinement. These prisms are then independently triangulated via a Delaunay criterion; however, in order to avoid producing nonconforming meshes, this method requires a vertex perturbation and sliver removal process to be carried out alongside the Delaunay triangulation.

The method contained in the present work is characterized by its simplicity - the subdivision of prism elements into simplices is deterministic, combinatorial in nature, always produces conforming meshes, and does not require additional operations like vertex perturbation and sliver elimination. It may be argued that Behr's method has the benefit of controlling temporal refinement within the mesh generation process. However, a similar result can be achieved with our method by immediately following the coarse mesh generation with a sequence of element bisections.

Finally, we remark that on an even more general level, these two methods approach temporal refinement in different ways. We say that our method utilizes *element-based refinement* in the sense that a particular simplex element is specified for bisection, and the mesh is adapted from there. In contrast, we say that Behr's approach uses *vertex-based refinement*, in which a particular vertex is selected for insertion into the mesh, and a series of additional elements are created in order maintain consistency. One benefit of element-based refinement is a greater control over the similarity classes of elements produced by refinement. The element bisection method used in this dissertation has been shown to

produce a finite number of similarity classes, which gives us *a priori* guarantees on the shape regularity of the adapted mesh (see Section 3.2 for an expanded discussion). In contrast, the vertex-based refinement utilized in [5] does not make explicit reference to a coarse triangulation, and conclusions regarding shape regularity of the resulting mesh are not apparent. Although Behr directly addresses the problem of sliver removal for extremely degenerate elements, bounds on the shape regularity of non-degenerate elements remains an open problem.

Overview of the Method

The construction of a space-time triangulation from a spatial triangulation will be achieved through a sequence of two smaller operations: Extrusion and Subdivision. These steps are described in a general sense below and will be defined explicitly in the next section.

In the extrusion step, a d-dimensional triangulation will be converted into a $(d+1)$-dimensional cell complex. The resulting cell complex will consist of $(d+1)$-dimensional prisms, where the base of each prism is congruent to a d-simplex belonging to the original triangulation. These prisms may be viewed as a kind of tensor product element, where each prism is the tensor product of a simplex with an interval. In the case of two spatial dimensions, this procedure "extrudes" a triangle into a triangular prism. The extrusion step can be repeated arbitrarily many times to produce a sequence of prisms where the top base of one prism is the bottom base of the next. In two dimensions, this would appear as a "tower" comprised of triangular prisms stacked on top of each other.

In the subdivision step, each $(d+1)$-dimensional prism constructed during extrusion is partitioned into $d+1$ simplices of dimension $d+1$. Due to the regular nature of simplex prisms, this construction can always be carried out in such a way that the resulting set of $(d+1)$-simplices is conforming (that is, the set of $(d+1)$-simplices forms a triangulation). At the end of the subdivision step, a space-time triangulation has been produced which covers the space-time cylinder $Q = \Omega \times [0, T]$. For an illustration of the extrusion-subdivision process applied to a single triangle, see Figure 3.3.

The main goal of this section is to define a procedure for the construction of four-dimensional space-time meshes which correspond to a given three-dimensional spatial mesh. However, much of the following discussion is equally valid for space-time meshes constructed over d-dimensional spatial domains. When appropriate, we will state definitions and basic properties for d-dimensional spatial

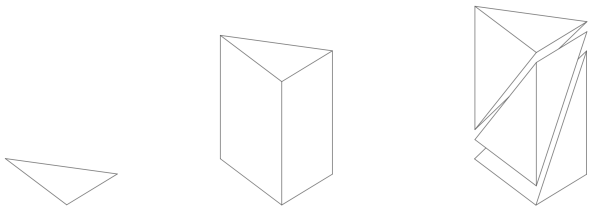

Figure 3.3: Left: Triangle in a two-dimensional mesh. Center: Extruded space-time triangular prism. Right: Subdivision of space-time prism into tetrahedra.

domains, where d is arbitrary. This is done to emphasize extensibility of this theory to even higher dimensions; however, the special case of $d = 3$ remains the primary object of study, and we shall not prove every result for arbitrary dimension d.

3.1.1 Basic Operations and Notation

As a matter of notation, when $p \in \mathbb{R}^d$ is some point in Euclidean space, we will denote its j^{th} coordinate by $p^{(j)}$. The canonical basis vectors are denoted $e_1, e_2, e_3, \dots, e_d$.

It is often useful to identify points in $\mathbb{R}^d$ with points in $\mathbb{R}^{d+1}$, where the $(d+1)^{th}$ coordinate is constant. Informally, this considers $\mathbb{R}^d$ as a hyperplane in $\mathbb{R}^{d+1}$ which is orthogonal to the canonical basis vector e_{d+1}. Formally speaking, we make this identification via an embedding $\phi_r : \mathbb{R}^d \hookrightarrow \mathbb{R}^{d+1}$, where

$$\phi_r\left(p^{(1)}, p^{(2)}, \dots, p^{(d)}\right) = \left(p^{(1)}, p^{(2)}, \dots, p^{(d)}, r\right).$$

When $d = 2$, the parameter r can be thought of as the "height" of a 2D object lying in $\mathbb{R}^3$, where the third dimension is considered as the vertical direction.

This embedding may be extended to subsets of $\mathbb{R}^d$ as well. For any $D \subset \mathbb{R}^d$, we define

$$\phi_r(D) = \{\phi_r(p) : p \in D\}. \tag{3.1}$$

Although trivial to prove, we will also record a few properties about this embedding so that they may be referred to later. All properties follow immediately from the definition of $\phi_r(p)$.

Lemma 3.1. *Let $\alpha, \beta, r, s \in \mathbb{R}$ and $p, q \in \mathbb{R}^d$. Then*

i) $\phi_r(p) + \phi_s(q) = \phi_{r+s}(p+q)$

ii) $\alpha\phi_r(p) = \phi_{\alpha r}(\alpha p)$

iii) *If* $\phi_r(p) = \phi_s(q)$, *then* $r = s$ *and* $p = q$.

We remark that when considering polytopes embedded in Euclidean space, there are two "dimensionalities" at play. The first is the dimension of the polytope itself (see Section 1.1.1); in the following discussion we denote the polytope dimension by k. The second type of dimensionality is the ambient dimension, or the dimension of the Euclidean space in which the polytope is embedded; we denote this dimension by d. Thus a triangle in $\mathbb{R}^3$ would have polytope dimension $k = 2$ and ambient dimension $d = 3$.

Additionally, throughout this chapter we will adopt the notation of Definition 1.15 for the description of convex polytopes, which are the convex hull of their extremal points. Let us briefly describe an example of this notation to illustrate its usage. Suppose that $p_1, p_2, p_3, p_4 \in \mathbb{R}^3$ form the vertices of a tetrahedron τ. Then $\tau = \{\{p_1, p_2, p_3, p_4\}\}$. Suppose now that we would like to enumerate all of the triangular faces of τ. Each triangular face is the convex hull of three vertices of τ. Thus the triangular faces of τ are

$$\{\{p_1, p_2, p_3\}\}, \quad \{\{p_1, p_2, p_4\}\}, \quad \{\{p_1, p_3, p_4\}\}, \quad \text{and} \quad \{\{p_2, p_3, p_4\}\}.$$

Since every face of a convex polytope is itself a convex polytope, the above notation provides a convenient mechanism for describing operations on polytopes in a purely combinatorial way.

3.1.2 Three-Dimensional Constructions

In order to gain some geometric intuition behind the four-dimensional operations described in Section 3.1.3, we shall describe first the process of constructing space-time meshes in three dimensions (corresponding to two spatial dimensions). Due to our natural inability to understand four-dimensional geometry in a spatial sense, a methodical treatment of four-dimensional operations must rely on something other than spatial reasoning. In the following sections, we will rely on two general techniques to make reasoning in four dimensions more tractable.

The first of these is combinatorics. In a simplex, every face can be represented uniquely by the vertices it contains, meaning that operations on faces can considered to be operations on combina-

tions of vertices. In addition, adjacency relations among faces can be deduced from the sets of vertices bounding those faces. This means that almost any geometric operation on simplices is actually a purely combinatorial operation, with no dependence on the ambient space $\mathbb{R}^4$.

The second technique we will employ is dimension reduction. As much as possible, it will be beneficial to define operations on d-dimensional objects in terms of operations on $(d-1)$-dimensional objects. For example, we will show in the following section that the triangulation of a triangular prism requires each rectangular face to be triangulated. Thus, an operation (triangulation) on a three-dimensional object can be broken into sub-operations on two-dimensional objects. By emphasizing this structure, we can then make an analogy to four-dimensional geometry. As it turns out, the triangulation of a four-dimensional prism is executed in part by a series of triangulations of three-dimensional prisms.

There is an extraordinary amount that can be said about three-dimensional geometry and tetrahedral meshes. We make no attempt to be exhaustive here; instead, our intent is solely to lay a foundation for the four-dimensional constructions defined in Section 3.1.3. Namely, we shall walk through the construction of tetrahedra via the extrusion of triangles and subdivision of the resulting prisms.

For the remainder of Section 3.1.2, let $\Omega \subset \mathbb{R}^2$ be a 2D domain with polygonal boundary and $\mathcal{T} = \{\tau_j\}_{j=1}^M$ a triangulation that covers Ω. We define the set $\mathcal{V}(\mathcal{T}) = \{v_k\}_{k=1}^N$ to be the collection of vertices of the triangulation.

Extrusion into 3D Space-Time

For each triangle τ_j, the 2D extrusion operation on a triangle creates an associated triangular prism P_j which has τ_j as one of its bases. Recalling the definition of the convex hull (denoted Conv) from Definition 1.9, we define the (r,s)-*extrusion* of $\tau \in \mathcal{T}$ to be

$$\begin{aligned} \mathrm{Extr}_{r,s} : \mathbb{R}^2 &\to \mathbb{R}^3 \\ \tau &\mapsto \mathrm{Conv}(\phi_r(\tau), \phi_s(\tau)) \end{aligned} \tag{3.2}$$

In other words, the (r,s)-extrusion of a triangle τ is the set of all points between two copies of τ embedded in $\mathbb{R}^3$ at heights $z = r$ and $z = s$. Clearly, $P := \mathrm{Extr}_{r,s}(\tau)$ is a triangular prism.

A quick remark on the face structure of triangular prisms is in order, since this structure will

be important to the study of four-dimensional prisms as well. Consider an arbitrary triangle $\tau = \{\{p_1, p_2, p_3\}\}$ and let P be the prism formed by extruding τ. That is, let

$$P = \mathrm{Extr}_{r,s}\left(\{\{p_1, p_2, p_3\}\}\right) = \{\{a_1, a_2, a_3, b_1, b_2, b_3\}\}. \tag{3.3}$$

There are two kinds of 2-faces of P: rectangular faces and triangular faces. Written in vertex notation, these are

$$\begin{aligned} &\{\{a_1, a_2, b_1, b_2\}\} \qquad &&\{\{a_1, a_2, a_3\}\} \\ &\{\{a_1, a_3, b_1, b_3\}\} \qquad &&\{\{b_1, b_2, b_3\}\} \\ &\{\{a_2, a_3, b_2, b_3\}\} \end{aligned} \tag{3.4}$$

Now, each triangular face is simply an embedded copy of the extruded triangle:

$$\{\{a_1, a_2, a_3\}\} = \phi_r(\tau), \qquad \{\{b_1, b_2, b_3\}\} = \phi_s(\tau). \tag{3.5}$$

Furthermore, each rectangular 2-face of P is the extrusion of a 1-face of τ; for example

$$\{\{a_1, a_2, b_1, b_2\}\} = \mathrm{Extr}_{r,s}\left(\{\{p_1, p_2\}\}\right). \tag{3.6}$$

Thus all of the 2-faces of P can be described explicitly in terms of the original 2-simplex τ: They are either an embedded copy of τ or the extrusion of a 1-face of τ. We shall see shortly that every 3-face of a four-dimensional prism can (in the same way) be described solely in terms of the original 3-simplex.

It is worth noting that the above discussion can be carried out for the 1-faces of P as well. Every edge (1-face) of P is either a vertically shifted edge of the underlying triangle or the extrusion of a 0-face of the underlying triangle (see Figure 3.4).

In particular, this means that if τ_1 and τ_2 are two triangles that intersect on edge e, then the intersection of $\mathrm{Extr}_{r,s}(\tau_1)$ and $\mathrm{Extr}_{r,s}(\tau_2)$ is $\mathrm{Extr}_{r,s}(e)$. We conclude the rather plain fact that when two triangles intersect along a j-face, their corresponding extrusions intersect along a $j+1$ face. However, this observation will be important in the four-dimensional case, when the geometry is more difficult to visualize.

In order to create a space-time mesh from the triangular mesh $\mathcal{T}$, each triangle in the spatial mesh will be repeatedly extruded into a collection of different prisms, each stacked one atop another.

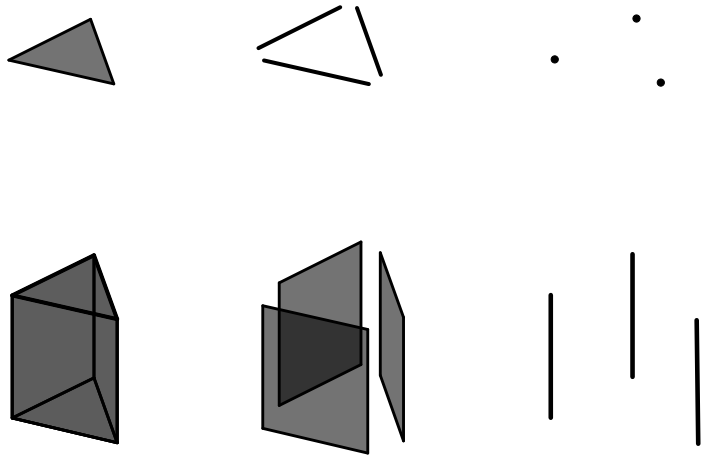

Figure 3.4: Relationship between faces of a triangle and its corresponding triangular prism. At top, from left to right: faces of the underlying triangle of dimension 2,1,0. At bottom, the extrusions of each face at top. Note that the extrusion of each face from the triangle at top is a face of the prism at bottom.

Let

$$W = \{w_0, w_1, \dots, w_{N_w}\} \qquad \text{where} \quad w_{i-1} < w_i \text{ for } 1 \le i \le N_w. \tag{3.7}$$

be a sequence of real numbers. We call each w_i a *time slice*, but the values of each w_i need not take on any physical significance.

Next, we define an extrusion operation on the entire spatial triangulation $\mathcal{T}$, such that each triangle is extruded into multiple prisms with heights given by W.

Definition 3.2. Let $\mathcal{T} = \{\tau_j\}_{j=1}^{M}$ be a two-dimensional triangulation and $W = \{w_i\}_{i=0}^{N_w}$ a sequence of time slices. The *extrusion of $\mathcal{T}$ over W* is defined to be

$$\begin{aligned} \mathrm{Extr}_W(\mathcal{T}) = \{P_{ij} \,:\, 1 \le i \le N_w,\ 1 \le j \le M\}, \\ \text{where } P_{ij} = \mathrm{Extr}_{w_{i-1}, w_i}(\tau_j) \end{aligned} \tag{3.8}$$

As would be expected from an operation that increases the problem dimension, this construction notably increases the size of the geometric discretization. Let $N_0(\mathcal{T})$ be the number of vertices in $\mathcal{T}$, $N_1(\mathcal{T})$ the number of edges, and $N_2(\mathcal{T})$ the number of triangular faces (in general $N_j(\mathcal{C})$ is the number of j-dimensional cells in the complex $\mathcal{C}$).

Furthermore, let

$$\mathcal{P} = \mathrm{Extr}_W(\mathcal{T}) \tag{3.9}$$

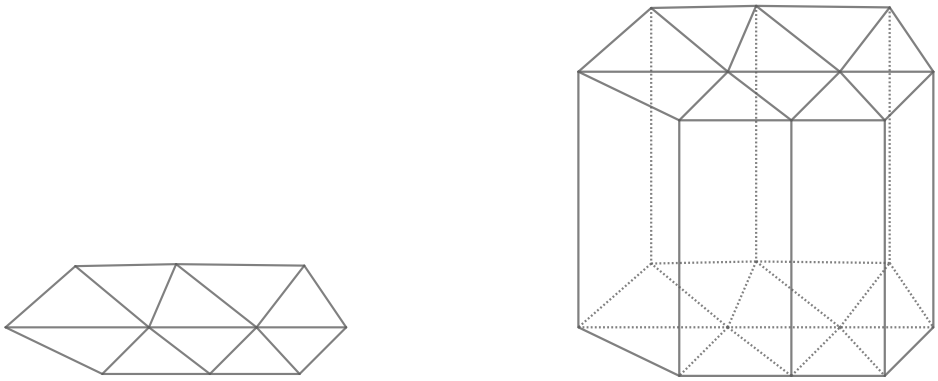

Figure 3.5: Illustration of a triangular mesh in two dimensions, and its corresponding prism mesh in three dimensions. Due to the conforming nature of the triangular mesh, the prism mesh is also conforming.

be the space-time prism mesh produced by extruding $\mathcal{T}$ over W. Then:

$$N_0(\mathcal{P}) = (N_w + 1) \cdot N_0(\mathcal{T})$$

$$N_1(\mathcal{P}) = (N_w + 1) \cdot N_1(\mathcal{T}) + N_w \cdot N_0(\mathcal{T})$$

$$N_2(\mathcal{P}) = (N_w + 1) \cdot N_2(\mathcal{T}) + N_w \cdot N_1(\mathcal{T})$$

$$N_3(\mathcal{P}) = N_w \cdot N_2(\mathcal{T})$$

Subdivision of Triangular Prisms

The output from the extrusion operation is a collection $\mathcal{P}$ of triangular prisms which covers the space-time domain Q. Next, in order to produce a conforming mesh of tetrahedra which covers Q, each prism is subdivided into tetrahedra. In order transform the full collection of prisms into a conforming triangulation, the subdivision of each prism must be carried out such that neighboring prisms are divided in a matching way.

There are exactly six ways to subdivide a triangular prism into tetrahedra without introducing new vertices, and all six divide the prism into 3 tetrahedra. Let us consider a general triangular prism P, where the vertices of the lower base are $\{a_1, a_2, a_3\}$, the vertices of the upper base are $\{b_1, b_2, b_3\}$, and for each i, b_i lies directly above a_i.

One triangulation of P is then given by the three tetrahedra:

$$\{\{a_1, b_1, b_2, b_3\}\}, \qquad \{\{a_1, a_2, b_2, b_3\}\}, \qquad \{\{a_1, a_2, a_3, b_3\}\} \tag{3.10}$$

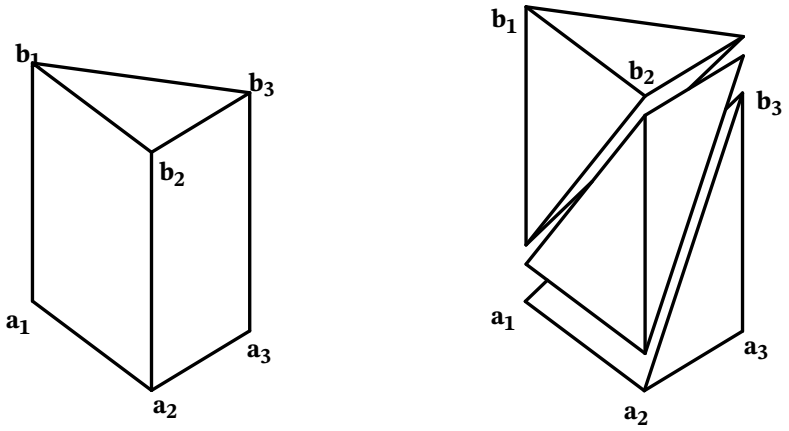

Figure 3.6: One possible subdivision of a triangular prism into tetrahedra. Note that each tetrahedron contains exactly one vertical edge.

which is illustrated in Figure 3.6. One crucial feature of this triangulation is that each tetrahedron contains exactly one vertical edge connecting $a_i b_i$ for some i. In fact, *every* triangulation of the triangular prism has this property, and all six triangulations are equivalent up to a permutation of the vertex labels.

The problem of prism subdivision is the following: for each triangular prism element, which of the six subdivision methods should be chosen such that the resulting collection of tetrahedra form a conforming triangulation?

To guarantee conformity, we define the following process. First, let $\mathcal{V}(\mathcal{T})$ be the vertex set of $\mathcal{T}$, and let $\mathcal{V}(\mathcal{T}) = \{v_j\}_{j=1}^{N}$ be some ordering on the vertices. Next, let $\mathcal{V}(\mathcal{P})$ be the vertex set of the prism mesh $\mathcal{P} = \text{Extr}_W(\mathcal{T})$. Since $\mathcal{P}$ was produced by the extrusion operation of the previous section, we know that every vertex in $\mathcal{P}$ is some shifted copy of a vertex in $\mathcal{T}$. Thus for every $v \in \mathcal{V}(\mathcal{P})$, $v = \phi_{w_i}(v_j)$ for some $0 \leq i \leq N_w$ and $1 \leq j \leq N$. This naturally establishes the indexing

$$v_{iN+j} = \phi_{w_i}(v_j), \tag{3.11}$$

which is uniquely determined for every vertex in $\mathcal{V}(\mathcal{P})$. For any two vertices a and b, we say that

$$a \prec b \qquad \text{if and only if} \qquad a = v_k,\ b = v_{k'},\ \text{and } k < k'. \tag{3.12}$$

Note that under this ordering, the index of every vertex differs by N from those vertices immediately above and below them.

Now, for each prism $P \in \mathcal{P}$, let

$$P = \{\{a_1, a_2, a_3, b_1, b_2, b_3\}\} \quad \text{where} \quad a_1 \prec a_2 \prec a_3 \prec b_1 \prec b_2 \prec b_3. \tag{3.13}$$

By the definition of $\prec$, this means that bottom base is $\{\{a_1, a_2, a_3\}\}$, the top base is $\{\{b_1, b_2, b_3\}\}$, and each b_i vertex lies directly above a_i. To subdivide P, we always choose the triangulation specified in Equation 3.10, applied to this particular ordering of the vertices.

The collection of tetrahedra obtained by this process is always conforming. Suppose P and P' are two prism which share a common 2-face. If the common face is a triangle, then the subdivisions of P and P' will be conforming because this triangular face is never subdivided (see Figure 3.6). If the common face is a rectangle, the situation is slightly move complex, since the rectangular face may be triangulated in two different ways. We will see that under our vertex ordering and subdivision method, a matching triangulation is always independently chosen by both adjacent prisms.

As a consequence of the vertex ordering in Equation 3.13 and the subdivision given in Equation 3.10, every new tetrahedron has the form $\{\{a_1, \dots, a_i, b_i, \dots, b_3\}\}$ for some $i = 1, 2, 3$. In particular, this means that for a tetrahedron χ,

$$\text{If } a_i, b_i \in \chi, \quad \text{then} \quad a_j \notin \chi \text{ for } j > i \quad \text{and} \quad b_j \notin \chi \text{ for } j < i. \tag{3.14}$$

Next, let

$$P = \{\{a_1, a_2, a_3, b_1, b_2, b_3\}\} \quad \text{and} \quad P' = \{\{a_1, a_2, a_3', b_1, b_2, b_3'\}\}, \tag{3.15}$$

where each b_i is directly above the corresponding a_i. Without loss of generality we may assume that $a_1 \prec a_2$ (and thus $b_1 \prec b_2$); however, the ordering of a_3 and a_3' with respect to the other vertices is not assumed.

The intersection of P and P' is the rectangular face $\{\{a_1, a_2, b_1, b_2\}\}$. Let $\chi \subset P$ be a new tetrahedron produced by subdividing P which has a triangular face contained in $\{\{a_1, a_2, b_1, b_2\}\}$. By the condition in Equation 3.14, this triangular face must be either $\{\{a_1, b_1, b_2\}\}$ or $\{\{a_1, a_2, b_2\}\}$. By the same logic, if $\chi' \subset P'$ is a new tetrahedron which has a triangular face on the intersection, that triangular face must be either $\{\{a_1, b_1, b_2\}\}$ or $\{\{a_1, a_2, b_2\}\}$.

Table 3.1: Summary of the type and quantity of lower-dimensional faces in a pentatope.

Dimension	Type	Count
0	Point	5
1	Segment	10
2	Triangle	10
3	Tetrahedron	5

Therefore, any two new tetrahedra formed during subdivision of neighboring prisms either overlap on a triangular face (on which they always coincide) or on a rectangular face. If they intersect on a rectangular face, then the triangular faces of the tetrahedra contained within this rectangle either coincide, or they meet along the diagonal $a_1 b_2$. In any case, their intersection is conforming.

We have shown that by imposing a global ordering on vertices, each prism can be subdivided independently (and indeed, in parallel) from the rest, and the resulting collection of tetrahedra will always be conforming. No post-processing checks or adjustments are needed.

3.1.3 Four-Dimensional Constructions

Basic Four-Dimensional Geometry

The four-dimensional simplex has been given a variety of names throughout the literature without any one term becoming standard. We will refer to such simplices as *pentatopes*, following the trend of recent papers in space-time finite element analysis (e.g. [5, 30, 38]). Other names for this shape include *pentachoron, 5-cell*, and *pentahedroid*.

The pentatope, being a simplex, possesses all of the properties of simplices laid out in Section 1.1.1. In particular, a pentatope is the convex hull of 5 affinely independent points. In this section, we will always assume that a pentatope is embedded in $\mathbb{R}^4$ (non-degenerate pentatopes may be embedded in higher dimensional space, but not lower).

We recall from Section 1.1.1 that the boundary of any k-simplex is the union of its $(k-1)$-faces. In the case of a pentatope, each 3-face is a tetrahedron. Therefore, the boundary of a pentatope is the union of five tetrahedra (embedded in $\mathbb{R}^4$). In general, the number of l-faces in a k-simplex is $\binom{k+1}{l+1}$; see Table 3.1 for a complete summary of the proper faces of the pentatope.

Another four-dimensional object that plays a central role in the construction of space-time meshes is the simplex prism. The *k-simplex prism* is a generalization of the triangular prism to other

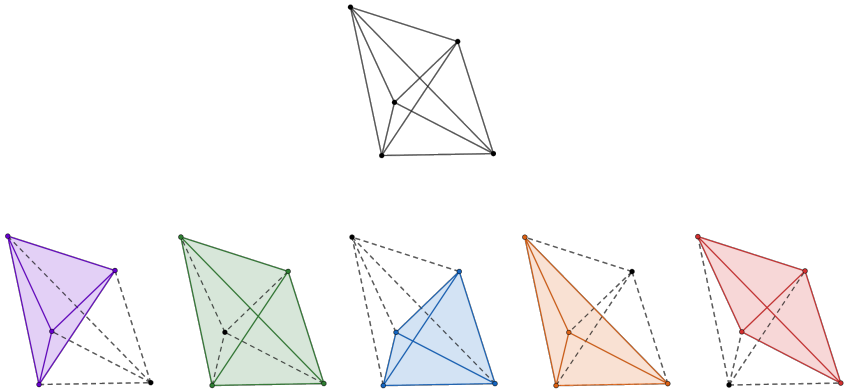

Figure 3.7: The five tetrahedral faces of a pentatope.

dimensions.

Definition 3.3. Let $K \subset \mathbb{R}^d$ be a k-simplex, and let $v \in \mathbb{R}^d$ be a direction which is orthogonal to every face of K (0-simplices are considered to be orthogonal to all vectors in $\mathbb{R}^d$). If T_v is a translation in the direction v, then the polytope

$$P = \text{Conv}\,(K, T_v(K)) \tag{3.16}$$

is a *k-simplex prism (over K)*. Furthermore, K is called the *bottom base* of P, $T_v(K)$ is called the *top base* of P, and the remaining k-faces are called the *lateral faces* of P.

Remark 3.4. The requirement that the translation direction v be orthogonal to the simplex faces ensures that the prism is a *right* prism. For instance, a skew triangular prism can be described by Equation 3.16 with $k = 2$ and v not orthogonal to the edges of the bottom base.

Remark 3.5. Clearly, any k-simplex prism may be described by its top and bottom bases. Since these bases are simplices, they in turn may be described by their vertices. As such, we can describe any k-simplex prism as the convex hull of its vertices. Writing

$$P = \{\{a_1, a_2, \dots, a_{k+1}, b_1, b_2, \dots, b_{k+1}\}\} \tag{3.17}$$

we will always take this to mean that the bottom simplex base is $\{\{a_1, a_2, \dots, a_{k+1}\}\}$, the top simplex base is $\{\{b_1, b_2, \dots, b_{k+1}\}\}$, and b_i is the image of a_i under the translation T_v defining the prism.

An important property of k-simplex prisms is that every lateral face is a $(k-1)$-simplex prism. This can be proven in a variety of ways; see, [14], Chapter 6 for a treatment of simplex prisms via point

Table 3.2: Summary of the type and quantity of lower-dimensional faces in a tetrahedral prism.

Dimension	Type	Count		Dimension	Type	Count
0	Point	8		2	Rectangle	6
1	Segment	16		3	Tetrahedron	2
2	Triangle	8		3	Tri. Prism	4

configurations. Figure 3.8 exhibits k-simplex prisms for $k = 1, 2, 3, 4$. It is clear that the lateral faces of the triangular (2-simplex) prism are rectangles, which are 1-simplex prisms. Likewise, the lateral faces of the rectangle (1-simplex prism) are segments, which can be thought of as 0-simplex prisms.

In addition, every lateral face of a k-simplex prism corresponds to a $(k-1)$-face of the base of the prism. To illustrate this point, consider the 2-simplex prism (i.e. triangular prism). Every lateral face of the triangular prism is a rectangle, or 1-simplex prism. Each of these 1-simplex prisms is a prism over a particular 1-simplex. Indeed, the base of each rectangular face is a line segment forming the perimeter of the triangular base. In this way, we may identify each lateral (rectangular) face with its base, which must be a $(k-1)$-face of the base of the original prism. We summarize the above discussion by stating

Proposition 3.6. *Let P be a k-simplex prism and K a base of P. Then*

i) Every lateral k-face of P is a $(k-1)$-simplex prism

ii) If K' is a $(k-1)$-face of K, then one of the lateral faces of P is a $(k-1)$-simplex prism over K'.

iii) P has $k+1$ lateral faces

The remainder of this chapter will consider the manipulation of pentatopes (4-simplices) and tetrahedral prisms (3-simplex prisms). As such, it will be useful to record a few specific facts about tetrahedral prisms.

By Proposition 3.6, every tetrahedral prism has six 3-faces: two tetrahedra and four triangular prisms. In addition, for every triangular-prismatic face, the upper and lower triangular faces coincide with a pair of triangular faces on the upper and lower tetrahedral bases. Furthermore, the above characterization of 3-faces implies that if two tetrahedral prisms intersect in a 3-dimensional region, their intersection is either a tetrahedron or a triangular prism. A full accounting of the various faces in a tetrahedral prism is given in Table 3.2.

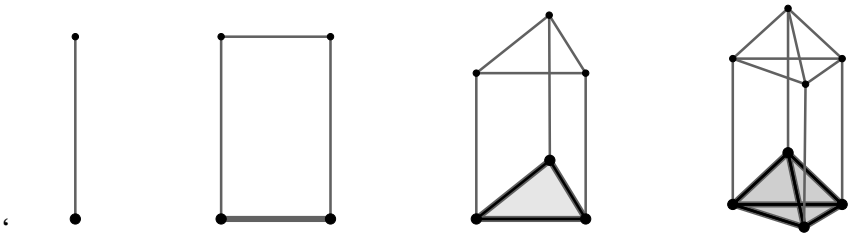

Figure 3.8: k-Simplex prisms. From left to right: $k = 0, 1, 2, 3$. In each case, the bottom base is highlighted.

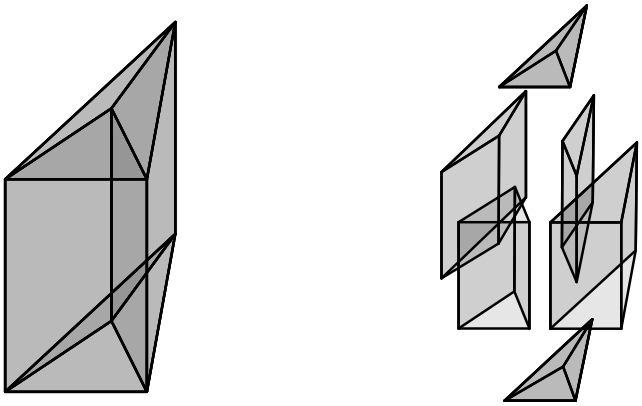

Figure 3.9: Exploded view of a tetrahedral prism. Every triangular prism is a lateral face of the tetrahedral prism. Furthermore, every triangular face on the top or bottom tetrahedron coincides with a triangular face of a triangular prism.

Extrusion into 4D Space-Time

The construction of a four-dimensional space-time mesh from a three-dimensional spatial mesh proceeds analogously to the construction from two- to three-dimensional space. In the three-dimensional case, the extrusion step created a mesh of triangular prisms from the spatial triangular mesh. In the four-dimensional case, a mesh of tetrahedral prisms will be created from the spatial tetrahedral mesh.

We define the (r, s)-*extrusion* of a convex set $S \subset \mathbb{R}^d$ to be

$$\begin{aligned} \text{Extr}_{r,s} : \mathbb{R}^d &\to \mathbb{R}^{d+1} \\ S &\mapsto \text{Conv}(\phi_r(S), \phi_s(S)) \end{aligned} \tag{3.18}$$

Remark 3.7. If S is a k-simplex, then $P = \text{Extr}_{r,s}(S)$ is k-simplex prism. To check this fact, we simply verify that $\phi(S)$ is a translation of $\phi_r(S)$ in the direction of e_{d+1}, which is orthogonal to any face of $\phi_r(S)$

(since $\phi_r(S)$ is contained in a hyperplane orthogonal to e_{d+1}).

In contrast to the definition of $\mathrm{Extr}_{r,s}$ for the special case of $d = 2$ (see Equation 3.2), this general definition applies to arbitrary dimension d and arbitrary convex subsets $S \subset \mathbb{R}^d$. This is of practical importance for the construction of space-time meshes, since it will often be necessary to consider the extrusion of individual faces of an element. In particular, when a convex a set S can be decomposed into a collection of convex subsets $S_1, \ldots, S_n$, we can relate the extrusion of the set S to the extrusion of its constituent subsets.

Proposition 3.8. *Let $S \subset \mathbb{R}^d$ be a convex set, and*

$$S = \bigcup_{j=1}^{n} S_j,$$

where each $S_j \subset S$ is also convex. Then

i) The extrusion of S is the extrusion of its constituent sets; that is,

$$\mathrm{Extr}_{r,s}(S) = \bigcup_{j=1}^{n} \mathrm{Extr}_{r,s}(S_j). \tag{3.19}$$

ii) The intersection of the extrusions of two constituent sets is the extrusion of their intersection:

$$\mathrm{Extr}_{r,s}(S_i) \cap \mathrm{Extr}_{r,s}(S_j) = \mathrm{Extr}_{r,s}(S_i \cap S_j) \tag{3.20}$$

Proof. To prove (i), first consider an arbitrary $p \in \mathrm{Extr}_{r,s}(S)$. By Lemma 1.10, $p = \alpha p_1 + (1-\alpha)p_2$, where $p_1 \in \phi_r(S)$, $p_2 \in \phi_s(S)$, and $0 \leq \alpha \leq 1$. Now, since $p_1 \in \phi_r(S)$ and $p_2 \in \phi_s(S)$, there must be points $q_1, q_2 \in S$ such that $p_1 = \phi_r(q_1)$ and $p_2 = \phi_s(q_2)$.

Since S is convex, the point $q = \alpha q_1 + (1-\alpha)q_2 \in S$. By the definition of ϕ, we have

$$\begin{aligned}\alpha p_1 + (1-\alpha)p_2 = \alpha\phi_r(q_1) + (1-\alpha)\phi_s(q_2) &= \phi_{\alpha r}(\alpha q_1) + \phi_{(1-\alpha)s}((1-\alpha)q_2) \\ &= \phi_{\alpha r+(1-\alpha)s}(\alpha q_1 + (1-\alpha)q_2) \\ &= \phi_{\alpha r+(1-\alpha)s}(q) \\ &= \phi_{\alpha r+(1-\alpha)s}(\alpha q + (1-\alpha)q) \\ &= \alpha\phi_r(q) + (1-\alpha)\phi_s(q).\end{aligned} \tag{3.21}$$

Since $q \in S$, it follows that $q \in S_j$ for some j. Therefore Equation 3.21 implies that $p \in \mathrm{Extr}_{r,s}(S_j) \subset \bigcup_1^n \mathrm{Extr}_{r,s}(S_j)$, establishing set inclusion from left to right.

Showing the reverse inclusion is much simpler. Suppose $p \in \bigcup_1^n \mathrm{Extr}_{r,s}(S_j)$. Then there is some j such that $p \in \mathrm{Extr}_{r,s}(S_j)$. Hence $p = \alpha p_1 + (1-\alpha)p_2$ by Lemma 1.10, where $p_1 \in \phi_r(S_j)$ and $p_2 \in \phi_s(S_j)$. Since $S_j \subset S$, this immediately implies that $p \in \mathrm{Extr}_{r,s}(S)$.

To prove (ii), we begin by considering an arbitrary element $p \in \mathrm{Extr}_{r,s}(S_i) \cap \mathrm{Extr}_{r,s}(S_j)$. Therefore for some $p_1 \in \phi_r(S_i)$, $p_2 \in \phi_s(S_i)$, $p_1' \in \phi_r(S_j)$, $p_2' \in \phi_s(S_j)$ and $0 \le \alpha, \alpha' \le 1$, we have

$$\begin{aligned}p = \alpha p_1 + (1-\alpha)p_2 = \alpha\phi_r(q_1) + (1-\alpha)\phi_s(q_2) \\ p = \alpha' p_1' + (1-\alpha')p_2' = \alpha'\phi_r(q_1') + (1-\alpha')\phi_s(q_2')\end{aligned} \tag{3.22}$$

where $q_1, q_2 \in S_i$ and $q_1', q_2' \in S_j$ are chosen in the same manner as in the proof of part (i). Setting the two right hand sides of Equation 3.22 equal to each other and considering only the $(d+1)^{th}$ coordinate of each expression, we obtain the relation

$$\alpha r + (1-\alpha)s = \alpha' r + (1-\alpha')s. \tag{3.23}$$

Simplifying this relation (assuming $r \neq s$) yields the condition that $\alpha = \alpha'$.

Now, applying the same argument as in the proof of the first inclusion of part (i), there must be points $q_i \in S_i$ and $q_j \in S_j$ such that

$$p = \alpha\phi_r(q_i) + (1-\alpha)\phi_s(q_i) = \alpha'\phi_r(q_j) + (1-\alpha')\phi_s(q_j). \tag{3.24}$$

Rearranging terms and using the fact that $\alpha = \alpha'$, we deduce

$$\alpha\phi_0(q_i - q_j) = (\alpha - 1)\phi_0(q_i - q_j).$$

Thus $\alpha(q_i - q_j) = (\alpha - 1)(q_i - q_j)$, and consequently $q_i = q_j$. Therefore, $q_i, q_j \in S_i \cap S_j$, and by Equation 3.24 it follows that $p \in \mathrm{Extr}_{r,s}(S_i \cap S_j)$, thus establishing set inclusion in part (ii) from left to right.

To show set inclusion in (ii) from right to left, suppose that $p \in \mathrm{Extr}_{r,s}(S_i \cap S_j)$. Then $p = \alpha\phi_r(q_1) + (1 - \alpha)\phi_s(q_2)$, where $q_1, q_2 \in S_i \cap S_j$. This immediately implies that $p \in \mathrm{Extr}_{r,s}(S_i)$ and $p \in \mathrm{Extr}_{r,s}(S_j)$, which establishes the set inclusion in (ii) from right to left, closing the proof. □

When a collection of elements in a spatial mesh are extruded into space-time prisms, Proposition 3.8 can be used to describe properties of the space-time mesh in terms of its underlying spatial mesh. For instance, if τ_i and τ_j are two elements of the spatial mesh and $P_i = \mathrm{Extr}_{r,s}(\tau_i)$, $P_j = \mathrm{Extr}_{r,s}(\tau_j)$ are their corresponding space-time prisms, then any interface between P_i and P_j will have the structure $\mathrm{Extr}_{r,s}(\tau_i \cap \tau_j)$. In particular, this means that if τ_i and τ_j intersect along an edge, P_i and P_j will intersect along a rectangle. If the interface between τ_i and τ_j is a triangular face, the interface between P_i and P_j will be a triangular prism.

We have now established the necessary preliminaries to define the extrusion of a tetrahedral mesh into a four-dimensional prism mesh.

Definition 3.9. Let $\mathcal{T} = \{\tau_j\}_{j=1}^{N}$ be a triangulation of a spatial domain $\Omega \subset \mathbb{R}^3$ with polytopal boundary. Additionally, let $W = \{w_i\}_{i=0}^{N_w}$ be a series of time steps, where $w_0 < w_1 < ... < w_{N_w}$. The *extrusion of* $\mathcal{T}$ *over* W is

$$\begin{aligned} \mathrm{Extr}_W(\mathcal{T}) = \{P_{ij} : 1 \le i \le N_2, 1 \le j \le N\}, \\ \text{where } P_{ij} = \mathrm{Extr}_{w_{i-1},w_i}(\tau_j) \end{aligned} \tag{3.25}$$

In other words, the space-time extrusion of $\mathcal{T}$ over the time steps W is the collection of all 4D tetrahedral prisms formed by extruding the spatial tetrahedra at different times. Let $\mathcal{P} = \mathrm{Extr}_W(\mathcal{T})$ denote this mesh of prisms. Then each $P_{ij} \in \mathcal{P}$ is a tetrahedral prism with several known characteristics. Firstly, each P_{ij} has upper and lower bases which are congruent to τ_j. In addition, P_{ij} is bounded in the

fourth dimension by the planes $e_4 = w_{i-1}$ and $e_4 = w_i$; put another way, every point $p \in P_{ij}$ satisfies $w_{i-1} \leq p^{(4)} \leq w_i$.

The adjacency relations among the tetrahedral prisms can be deduced from the adjacency relations on the spatial triangulation. The four lateral faces of a tetrahedral prism P_{ij} are shared with four other tetrahedral prisms (unless P_{ij} is near the boundary). Furthermore, if $\tau_{j'}$ shares a triangular face with τ_j, then P_{ij} is adjacent to $P_{ij'}$ and their interface is a triangular prism. In addition, for $i \neq 1, N_w - 1$, each prism P_{ij} is adjacent "above" and "below" to the prisms $P_{i+1,j}$ and $P_{i-1,j}$, and their interface is a tetrahedron.

Prism Subdivision

The goal of the prism subdivision step is to create a conforming mesh of pentatopes from the space-time prism mesh in a way that does not introduce any new vertices. To do this we will first describe how any k-simplex prism can be subdivided into a collection of $k+1$ $(k+1)$-simplices. If this subdivion is applied to every prism in the mesh produced during the extrusion step, the result will be a collection of k-simplices which covers the entire space-time domain. However, this collection of simplices may not form a *conforming* triangulation. To address this issue, we introduce simple criterion which ensure that the prism subdivision produces a conforming mesh.

In Section 3.1.2, we described the six methods to subdivide a triangular prism and then showed how to choose a particular method for each triangular prism so that the resulting tetrahedral mesh is conforming. When dealing with tetrahedral prisms, the situation is slighly more complex, but the methodology for choosing conforming subdivisions is the same.

The set of all possible subdivisions of a k-simplex prism can be precisely described. The following proposition is a restatement of Proposition 6.2.3 in [14] (using the notational conventions of Remark 3.5 and Definition 1.15).

Proposition 3.10. *Let $P = \{\{a_1, \dots, a_{k+1}, b_1, \dots, b_{k+1}\}\}$ be a k-simplex prism, and let σ be a permutation on $\{1, 2, \dots, k+1\}$. Then the collection of simplices*

$$\mathcal{C}_\sigma := \left\{\tau_{\sigma,i} = \{\{a_{\sigma(1)}, \dots, a_{\sigma(i)}, b_{\sigma(i)}, \dots, b_{\sigma(k+1)}\}\} : 1 \leq i \leq k+1\right\} \tag{3.26}$$

is a triangulation of P. Furthermore, every triangulation of P has the form of Equation 3.26, and there are

Table 3.3: List of all possible triangulations of a triangular prism, enumerated by parameters i and σ from Proposition 3.10.

	$\sigma = \mathbf{e}$	$\sigma = (\mathbf{2\,3})$	$\sigma = (\mathbf{1\,2})$	$\sigma = (\mathbf{1\,3\,2})$	$\sigma = (\mathbf{1\,2\,3})$	$\sigma = (\mathbf{1\,3})$
$i = 1$	$a_1b_1b_2b_3$	$a_1b_1b_3b_2$	$a_2b_2b_1b_3$	$a_2b_2b_3b_1$	$a_3b_3b_1b_2$	$a_3b_3b_2b_1$
$i = 2$	$a_1a_2b_2b_3$	$a_1a_3b_3b_2$	$a_2a_1b_1b_3$	$a_2a_3b_3b_1$	$a_3a_1b_1b_2$	$a_3a_2b_2b_1$
$i = 3$	$a_1a_2a_3b_3$	$a_1a_3a_2b_2$	$a_2a_1a_3b_3$	$a_2a_3a_1b_1$	$a_3a_1a_2b_2$	$a_3a_2a_1b_1$

precisely $(k+1)!$ *distinct triangulations of* P.

Proof. See [14], Proposition 6.2.3. □

Proposition 3.10 states that, up to a (consistent) reordering of vertices, there is only one way to triangulate a simplex prism P. By "consistent reordering" we mean a permutation of vertices within each base, such that the same permutation is applied to the top and bottom bases simultaneously. In addition, the above proposition shows that every $(k+1)$-simplex in a triangulation of P contains exactly one pair corresponding vertices in the top and bottom bases. This is analogous to the property we observed for triangular prisms, in which each of tetrahedra produced by subdivision contained exactly one vertical edge.

This structure can be verified in the case of triangular prisms which were analyzed in the previous section. Table 3.3 lists each of the six possible triangulations of the triangular prism, organized by the permutation σ in Proposition 3.10. Examining the column labeled "$\sigma = e$," we note that for each tetrahedron, there is exactly one index i such that both a_i and b_i are contained in the vertex set. Upon further examination of Table 3.3, we note that each column (and therefore each triangulation) is the same as the first column with the appropriate permutation applied to the vertex labels.

We shall not enumerate the $4! = 24$ possible triangulations of the tetrahedral prism, since the collection of all admissible triangulations has the same pattern as that of the triangular prism. We reiterate once more, however, that the structure of each possible triangulation is the same; the only difference is the labeling, or permutation, of vertices.

After triangulating each tetrahedral prism in a space-time prism mesh, the result is a collection of pentatopes which cover the space-time domain. However, this collection may not form a conforming triangulation, since no guarantees have been made that Property (ii) of Definition 1.19 is satisfied. In Section 3.1.2, the key to producing a conforming triangulation was imposing a global ordering on the vertices of $\mathcal{T}$. We shall see that same condition suffices in four-dimensional space.

The proof of conformity is simplified by recognizing that a collection of $(k+1)$-simplices intersect properly in the sense of Definition 1.19 if and only if every pair of simplices which intersect over a k-volume have a shared k-face. When $k = 2$, this is the statement that a tetrahedral mesh is conforming if any only if every pair of tetrahedra with a two-dimensional intersection overlap along a shared triangular face.

Theorem 3.11. *Let X be a collection of $(k+1)$-simplices covering some domain $Q \subset \mathbb{R}^{k+1}$. Then the following are equivalent.*

1. *X is a triangulation of Q.*
2. *For every $\chi, \chi' \in X$, if $\chi \cap \chi'$ has affine dimension k, then $\chi \cap \chi'$ is a shared face of both.*

Proof. For a proof in the context of finite element methods, see [45], Theorem 3.2 and the preceding discussion in Remark 3.1. For a proof in the context of computational geometry, see [14], Theorem 4.5.8. □

It is also necessary to extend the vertex ordering introduced in Equation 3.11 and Equation 3.13 to higher dimensions.

Definition 3.12. Let $\mathcal{T}$ be a spatial triangulation in $\mathbb{R}^d$ and $\mathcal{V}(\mathcal{T}) = \{v_j\}_{j=1}^{N_v}$ its vertex set. Additionally, let $W = \{w_i\}_{i=0}^{N_w}$ be a set of time steps and set $\mathcal{P} = \mathrm{Extr}_W(\mathcal{T})$. The *extruded-vertex ordering* on $\mathcal{V}(\mathcal{P})$ is

$$\text{For } v \in \mathcal{V}(\mathcal{P}), \quad v = v_{iN_v+j} \quad \text{when} \quad v = \phi_{w_i}(v_j) \tag{3.27}$$

and we say that

$$v \prec v' \qquad \text{if and only if} \qquad v = v_k \quad \text{and} \quad v' = v_{k'} \quad \text{where} \quad k < k'. \tag{3.28}$$

The following theorem describes a prism subdivision method which always produces conforming triangulations. The structure is essentially the same as the procedure used to subdivide triangular prisms in the earlier discussion. At the root of this method is the global ordering that we impose on vertices of the prism mesh. The subdivision procedure is defined entirely in terms of the ordering on vertices, which means that two adjacent prisms always subdivide their common faces in the same way.

Practically, this means that all prisms can be subdivided in a single pass and the subdivision of a prism does not depend on the subdivision of its neighbors.

Theorem 3.13. *Let $\Omega \subset \mathbb{R}^3$ be a polyhedral domain and $Q = \Omega \times (0, T)$ a corresponding space-time domain. Suppose $\mathcal{T}$ is a triangulation of Ω, and let $\mathcal{P} = \mathrm{Extr}_W(\mathcal{T})$ be the space-time extrusion of $\mathcal{T}$ over time steps W. Let $\prec$ be the extruded-vertex ordering on $\mathcal{V}(\mathcal{P})$.*

For each tetrahedral prism $P \in \mathcal{P}$, order the vertices incident to P such that $P = \{\{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}\}$ and

$$a_1 \prec a_2 \prec a_3 \prec a_4 \prec b_1 \prec b_2 \prec b_3 \prec b_4. \tag{3.29}$$

Finally, for each P, define the pentatopes

$$\begin{aligned} \chi_{P,1} &= \{\{a_1, b_1, b_2, b_3, b_4\}\} \quad \chi_{P,3} = \{\{a_1, a_2, a_3, b_3, b_4\}\} \\ \chi_{P,2} &= \{\{a_1, a_2, b_2, b_3, b_4\}\} \quad \chi_{P,4} = \{\{a_1, a_2, a_3, a_4, b_4\}\} \end{aligned} \tag{3.30}$$

Then the collection

$$X = \left\{\chi_{P,j} \,:\, P \in \mathcal{P} \text{ and } 1 \le j \le 4\right\} \tag{3.31}$$

forms a conforming triangulation of Q.

Proof. In order to prove that X is a triangulation of Q, we must establish the two constitutive properties of Definition 1.19; that is,

i) $Q = \bigcup_{P \in \mathcal{P}} \bigcup_{j=1}^{4} \chi_{P,j}$, and

ii) For $\chi, \chi' \in X$, the intersection $\chi \cap \chi'$ is a face of both χ and χ'.

The preceding discussion has established Property (i) already. In particular, Proposition 3.10 states that for each $P \in \mathcal{P}$,

$$P = \bigcup_{j=1}^{4} \chi_{P,j}, \tag{3.32}$$

and thus

$$\bigcup_{P \in \mathcal{P}} \bigcup_{j=1}^{4} \chi_{P,j} = \bigcup_{P \in \mathcal{P}} P = Q. \tag{3.33}$$

To prove Property (ii), will show the equivalent condition described in Theorem 3.11. That is, it suffices to show that if $\chi, \chi' \in X$ have a three-dimensional intersection, then $\chi \cap \chi'$ is a tetrahedron.

First, suppose that $\chi, \chi' \subset P$ for some $P \in \mathcal{P}$. Then both pentatopes were produced during the subdivision of P; by Proposition 3.10 their intersection must be a common face, and the intersection condition is satisfied.

For the remainder of the proof, we may assume that there are prisms $P, P' \in \mathcal{P}$ such that $\chi \subset P$ and $\chi' \subset P'$, with $P \neq P'$. Since $P \neq P'$, we may deduce that $P \cap P' = \partial P \cap \partial P'$. Consequently,

$$\begin{aligned} \chi \cap \chi' = (\chi \cap P) \cap (\chi' \cap P') &= (\chi \cap \chi') \cap (P \cap P') \\ &= (\chi \cap \chi') \cap (\partial P \cap \partial P') \\ &= (\chi \cap \partial P) \cap (\chi' \cap \partial P') \\ &= (\partial \chi \cap \partial P) \cap (\partial \chi' \cap \partial P'), \end{aligned} \tag{3.34}$$

which means that the intersection of χ and χ' is the same as the intersection of the faces of χ, χ' which lie on the boundary of each prism.

To prove the necessary intersection condition, suppose that $\chi \cap \chi'$ is three-dimensional. By Equation 3.34, it must also be the case that $\partial P \cap \partial P'$ is three-dimensional. By the face structure of tetrahedral prisms (as described in Proposition 3.6 and the subsequent discussion), this means that $P \cap P'$ is either a tetrahedron or a triangular prism.

If $P \cap P'$ is a tetrahedron, the prisms intersect at either their top and bottom base and without loss of generality we can write

$$P = \{\{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}\} \quad \text{and} \quad P' = \{\{b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4\}\}, \tag{3.35}$$

and therefore

$$P \cap P' = \{\{b_1, b_2, b_3, b_4\}\}. \tag{3.36}$$

Under the presecribed prism subdivision rule, the only pentatopes which have a three-dimensional intersection with $P \cap P'$ are

$$\chi = \{\{a_1, b_1, b_2, b_3, b_4\}\} \quad \text{and} \quad \chi' = \{\{b_1, b_2, b_3, b_4, c_4\}\}. \tag{3.37}$$

Since $a_i^{(4)} < b_i^{(4)} < c_i^{(4)}$, we conclude that $\chi \cap \chi' = \{\{b_1, b_2, b_3, b_4\}\}$, which is a 3-face of both pentatopes.

Finally, we consider the case where $P \cap P'$ is a triangular prism. In this scenario, P and P' have the form

$$P = \{\{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}\} \quad \text{and} \quad P' = \{\{a_1, a_2, a_3, a_4', b_1, b_2, b_3, b_4'\}\}, \tag{3.38}$$

where each a_i and b_i differ only in their fourth coordinate and $a_1 \prec a_2 \prec a_3$. In particular, we do not make assumptions on the ordering of a_4, b_4, a_4', b_4' with respect to the other vertices. Using this notation, the intersection between P and P' is the triangular prism

$$P \cap P' = \{\{a_1, a_2, a_3, b_1, b_2, b_3\}\}. \tag{3.39}$$

Again, we are assuming that $\chi \cap \chi'$ is three-dimensional, and thus $\chi \cap \mathcal{P}$ and $\chi' \cap P'$ are 3-faces of χ and χ'. By Equation 3.30, the only pentatopes contained in P that have three-dimensional faces in $\{\{a_1, a_2, a_3, b_1, b_2, b_3\}\}$ are

$$\{\{a_1, b_1, b_2, b_3, b_4\}\}, \qquad \{\{a_1, a_2, b_2, b_3, b_4\}\}, \quad \text{and} \quad \{\{a_1, a_2, a_3, b_3, b_4\}\}. \tag{3.40}$$

Therefore, $\chi \cap P$ must take the form:

$$T_1 = \{\{a_1, b_1, b_2, b_3\}\}, \quad \text{or} \quad T_2 = \{\{a_1, a_2, b_2, b_3\}\}, \quad \text{or} \quad T_3 = \{\{a_1, a_2, a_3, b_3\}\}. \tag{3.41}$$

The same argument can be applied to χ' and P' to show that $\chi' \cap P' = T_i$ for some i.

Therefore, $\chi \cap \chi' = (\chi \cap P) \cap (\chi' \cap P') = T_i \cap T_j$ for some $1 \leq i, j \leq 3$. However, these three tetrahedra form a conforming triangulation of the triangular prism (see Proposition 3.10)! Thus $T_i \cap T_j = T_i$ (if $i = j$) or has dimension less than three. Since we have assumed that $\chi \cap \chi'$ is three-dimensional, we conclude that $\chi \cap \chi' = T_i$ for some i. Since T_i is a face of both χ and χ', the intersection condition is satisfied and the proof is complete. □

Despite its involved proof, Theorem 3.13 gives simple criterion and preconditions for the creation of a conforming 4-simplex mesh. As long as the extruded-vertex ordering is established, the subdivision of a 3-simplex mesh is achieved by looping through each prism P in the mesh, sorting the vertices of P according to the global ordering, and then adding the pentatopes $\chi_{P,i}$ to the new space-time mesh.

Algorithm 3.1 provides a pseudocode description of this process.

Algorithm 3.1 Pseudocode algorithm for creating a pentatopal mesh from a mesh of tetrahedral prisms.

```
 1: procedure SUBDIVIDEPRISMS(PList)        ▷ Tesselate all prisms in PList
 2:     for each P in PList do
 3:         BVList ← BOTTOMVERTICESOF(P)
 4:         TVList ← TOPVERTICESOF(P)
 5:         {a_1, a_2, a_3, a_4} ← SORT(BVList)
 6:         {b_1, b_2, b_3, b_4} ← SORT(TVList)
 7:
 8:         Pent1 ← CREATEPENTATOPE(a_1, b_1, b_2, b_3, b_4)
 9:         Pent2 ← CREATEPENTATOPE(a_1, a_2, b_2, b_3, b_4)
10:         Pent3 ← CREATEPENTATOPE(a_1, a_2, a_3, b_3, b_4)
11:         Pent4 ← CREATEPENTATOPE(a_1, a_2, a_3, a_4, b_4)
12:
13:         ADDTOMESH(Pent1,Pent2,Pent3,Pent4)
14:     end for
15: end procedure
```

We also remark that the subdivision routine in Algorithm 3.1 is an inherently parallel workload. The construction of pentatopes within different steps of the for-loop are completely independent of each other, and therefore may be done in any order. The only serial step is an accumulation at the end where the global mesh data structure is updated with the new adjacency information. In addition, a similar property hold for the space-time mesh extrusion algorithm - each new prism element is defined independently of its neighbors. Therefore, the entire pipeline of space-time mesh generation may be implemented in parallel.

3.2 Bisection of 4D Mesh Elements

As we have seen, the full power of unstructured space-time finite element methods relies on the simultaneous refinement of the mesh in space and time. Furthermore, the use of simplicial mesh elements allows for the shape and size of elements to vary gradually from areas of low to high resolution. Our goal is a refinement process that is truly local in space-time. We contrast this with spatial refinement schemes, where a spatial mesh is refined and then used at for every time step, as well as adaptive time-stepping schemes, where the temporal resolution is adjusted but applies to every spatial element simultaneously. In these two examples, the size of a space-time element was determined solely by its spatial location or solely by its temporal location, respectively. In this section we describe a refinement

process for space-time elements so that the size of an element depends on its space and time positions simultaneously, not just one or the other.

In the previous section, we introduced a method for creating space-time meshes over a given spatial domain. To do this, a series of time steps $W = \{w_i\}_{i=0}^{N_w}$ was specified, and the resulting triangulation was a simplex space-time mesh with aligned time-slabs at each w_i. At present, we shall assume that the initial spatial mesh $\mathcal{T}$ has roughly equally-sized elements (that is, the ratio of largest to smallest element size is close to 1). Then, we take $\Delta_i = w_i - w_{i-1}$ to be the average diameter of the spatial elements for all i. These assumptions are necessary to avoid the construction of wedge or needle elements during space-time mesh generation.

In general, we presume that the spatial mesh $\mathcal{T}$ is the coarsest shape-regular mesh which accurately fits the geometry of the spatial domain Ω. This allows for the greatest degree of freedom in any subsequent refinement steps. For instance, suppose there was some subdomain $S \subset \Omega$ that was covered by a collection of very fine spatial elements, $S = \bigcup_1^k \tau_i$. After mesh extrusion and subdivision, this high spatial resolution will propagate through the entire patch $\Omega \times [0, T]$. In this case, the spatial resolution of the entire space-time patch is bounded below uniformly, which is precisely the situation we are trying to avoid.

Some applications utilize pre-set triangulations that have been constructed specifically for that application. In this case, the size of mesh elements may vary widely, which is not suitable for the current approach. The present method of space-time mesh generation *relies* on a simultaneous discretization of space and time. Thus if the input triangulation has been previously refined in space, this construction may not be well-behaved.

The local refinement of simplex meshes has been studied in depth for meshes of dimension two and three, and a smaller collection of results exists for arbitrary-dimension simplex meshes. Generally speaking, these methods may be classified as one of three types: uniform, red-green, and bisection.

3.2.1 Uniform Refinement

Uniform refinement is perhaps one of the most well-known methods of simplex subdivision. If τ is a k-simplex, then τ may be divided into 2^k subsimplices by adding vertices at the midpoint of each edge of τ and adding new edges as needed. Due to the fact that every edge of τ is subdivided,

mesh refinement schemes that rely solely on uniform subdivision must subdivide every mesh element simultaneously in order to maintain a conforming mesh. Nevertheless, uniform refinement is useful when a mesh is so coarse that multiple rounds of global mesh refinement are needed.

It is important to note that the simplicity of uniform refinement in two dimensions obscures some of the complexity in three dimensions, and the situation in higher dimensions is even more intricate. In two dimensions, there is only one way to uniformly refine a triangle. However, the uniform refinement of the tetrahedron can be achieved in multiple ways (essentially, the "interior children" of the refined tetrahedron can be configured in a number of ways, see [7]). As a result, successive uniform refinement of tetrahedra can produce degenerate elements if the method of subdivision is not chosen correctly[51].

Furthermore, uniform refinement of pentatopes is not necessarily conforming unless a consistent scheme is defined. This follows from the fact that every uniform refinement of a simplex uniformly refines each of its faces. In the case of pentatopes, every 3-face is a tetrahedron, and any two adjacent pentatopes intersect in this tetrahedron. Since the uniform refinement of tetrahedra is not unique, neighboring pentatopes must be refined in a way which is consistent with their neighbors.

As discussed by Bey, Freudenthal's algorithm[20] can be used to generate conforming uniform refinements of simplex meshes[6]. Applied to pentatopes, Freudenthal's method generates 12 similarity classes upon repeated refinements (and therefore does not produce degenerate child elements). In [38], Neumüller and Steinbach generalized Bey's discussion with a criterion for consistent uniform refinement of pentatopes, for which Freudenthal's algorithm is a special case.

3.2.2 Red-Green Refinement

A major drawback of uniform simplex refinement is that it is not a local operation - in order to maintain consistency, every mesh element must be uniformly refined simultaneously. Red-green refinement methods combine the uniform refinement ("red refinement") of a simplex with a closure operation ("green refinement") on its neighbors. The green refinement is chosen in such a way that a neighbor element is subdivided in a conforming way without introducing new vertices. Red-green refinement techniques were introduced for triangular elements in 1983 by Bank, Sherman, and Weiser[3], and extended to tetrahedral elements by Bey[7], Zhang[52], and Liu and Joe[31] in the mid-1990s. His-

torically, two approaches have been taken to the construction of red-green refinement rules.

In its original formulation[3], meshes produced by red-green refinement may contain irregular (nonconforming) vertices, so long as the *1-irregular* rule is satisfied. This rule states that every edge of a triangular element may have at most one irregular vertex from a refined neighbor. A simple post-processing step allows one to produce accurate finite element discretizations from 1-irregular meshes. Therefore, it is enough to apply green refinement only to ensure that the 1-irregular rule is satisfied. In this paradigm, further rules can be imposed to reduce the number of irregular vertices, but they are not strictly necessary.

Other red-green refinement schemes[7, 22], however, impose total conformity on the refined mesh. In these methods, every neighbor of a red-refined element must be refined in order to maintain mesh conformity. If the neighboring element is not also red-refined, then a closure operation must be chosen which depends on the pattern of irregular vertices incident to the neighbor element. Now, an arbitrary element marked for green refinement can contain irregular vertices in a number of configurations if several of its neighbors are red-refined. A full green-refinement rule in this setting must produce a consistent triangulation for any possible pattern. As the dimension of the mesh in question increases, the number and complexity of these edge refinement patterns increases as well, which can make higher-dimensional green rules difficult to formulate in practice.

Due to the difficulty in defining closure operations that eliminate all irregular vertices in high-dimensional meshes, it is often simpler to utilize the 1-irregular rule and allow slightly nonconforming meshes. While methods of this type require an additional step to manage irregular vertices, this is often preferable to implementing high-dimensional conforming closure operations.

3.2.3 Bisection Refinement

The last general class of simplex subdivision techniques is bisection. In these refinement schemes, simplices are divided in two by inserting a vertex at the midpoint of one edge. In order to maintain conformity, any neighboring elements which share this edge are refined via bisection as well. This rather large class of refinement schemes can be further subdivided into what we call “geometric bisection rules” and “combinatorial bisection rules.”

We define geometric bisection rules to be any bisection method which uses the geometric prop-

erties of a simplex element to determine how the element is bisected. A classical example of this is the longest-edge bisection[41] popularized by Rivara. In their simplest forms, geometry bisection rules require limited additional data structures to implement, since they rely on the mesh data explicitly. However, the shape regularity of meshes produced by longest-edge bisection is difficult to study. In two dimensions, it has been proven that the smallest interior angles of child elements are bounded away from zero. This phenomenon has been observed for the dihedral angles of tetrahedra in three dimensions, but has not been proven.

On the other hand, combinatorial bisection rules are easier to study theoretically but can be more challenging to implement. In a combinatorial rule, a simplex mesh is first tagged with a collection of labels (typically corresponding to vertex orderings and bisection generations). Then, the bisection of all elements *and* all of their child elements are completely determined by these labels. A prototypical example of this is the newest vertex bisection method[35] of Mitchell. In this method, the edges of a two-dimensional triangulation are tagged for refinement such that:

1. There is exactly one tagged edge per triangle, and
2. Any two adjacent triangles agree on the tag of their shared edge

After this tagging is complete, triangles marked for refinement are bisected along their tagged edge, and their neighbor is also bisected along this edge. Then, for every child triangle, the edge opposite the new vertex (at the midpoint) is designated as the new tagged edge. It has been proved that the child elements produced by newest vertex bisection belong to a finite set of similarity classes, which implies that the shape regularity of elements does not degenerate.

The key components of newest vertex bisection are the initial tagging scheme, and the rule for tagging child simplices. By choosing these rules correctly, the refinement scheme continues in a controlled way *ad infinitum*. It is also worth noting that, given a rule for tagging child elements, the refinement scheme is completely determined by the initial tagging of the coarse mesh and the adjacency structure of the elements.

A number of generalizations of newest vertex bisection have been proposed for simplex meshes in dimension greater than three. In all cases, the cornerstone of these methods is the rule for tagging child elements. A comprehensive overview of newest vertex bisection methods may be found in [34]. However, while a number of methods have been proposed and studied for three-dimensional triangu-

lation, the theory for simplices in higher dimensions is limited. We note the works of Maubach[32] and Traxler[48], who constructed combinatorial bisection methods for d-dimensional simplex meshes in the mid-1990s. Unfortunately, a critical roadblock for handling high-dimensional meshes is the initial tagging of the coarse mesh. For instance, Maubach's tagging scheme is proved only for meshes generated by reflections, while Traxler's method requires every edge to incident to an even number of elements.

Finally, in 2008, Stevenson[45] further analyzed the methods of Maubach and Traxler, proving an upper bound on the number additional bisections needed to retain conformity and relaxing the required conditions on the intial coarse mesh. In the remainder of this section we shall apply Stevenson's theory to four-dimensional simplex meshes and detail the refinement pattern of pentatopes. We shall also discuss the conditions required to set an initial tagging of the mesh.

3.2.4 Local Bisection of Pentatopes

The following definitions are proposed by Stevenson in [45] in order to describe the refinement rules of Maubach and Traxler. Here, we adapt the definitions for four-dimensional simplices, which simplifies the analysis significantly.

Definition 3.14. Let $\tau = \{\{a, b, c, d, e\}\}$ be a pentatope. The *type* of τ is and integer $\gamma(\tau)$ between 0 and 4. The *order* of τ is the ordered tuple

$$\sigma(\tau) = (v_0, v_1, v_2, v_3, v_4), \qquad \text{where } v_i \in \{a, b, c, d, e\},$$

that is, $\sigma(\tau)$ is a local ordering on the vertices of τ. When τ is a pentatope of type γ with ordering $(v_1, v_2, v_3, v_4, v_5)$, we write:

$$\tau = \{\gamma \mid v_0, v_1, v_2, v_3, v_4\}.$$

We say that a pentatope τ is *tagged* when τ has been given a choice of $\gamma(\tau)$ and $\sigma(\tau)$. A 4-simplex mesh is tagged if all of its elements are tagged.

The behavior of the refinement algorithm is completely determined by the tag on each element. In particular, if $\tau = \{\gamma \mid v_0, v_1, v_2, v_3, v_4\}$ is an arbitrary element of $\mathcal{T}$, then τ will be bisected along the edge v_0v_4, and the ordering of its child elements is determined by γ. The two children of τ are uniquely

determined to be:

$$\begin{aligned}\tau_a &= \left\{\gamma + 1 (mod 4) \mid v_0, \frac{v_0 + v_4}{2}, v_1, v_2, v_3\right\} \\ \tau_b &= \left\{\gamma + 1 (mod 4) \mid v_4, \frac{v_0 + v_4}{2}, v_1, \dots, v_\gamma, v_3, \dots, v_{\gamma+1}\right\}\end{aligned} \tag{3.42}$$

(where in abuse of notation, we take the term $v_1, \dots, v_\gamma$ to disappear when $\gamma = 0$, and $v_3, \dots, v_{\gamma+1}$ to disappear when $\gamma = 3$. Let us examine the structure of these two child elements more closely. In the case of τ_a, the vertex ordering of the parent element is preserved, with the new midpoint being given the second-lowest index. However, the case of τ_b is more complex. Here, the ordering of vertices is permuted from that of the parent element. Table 3.4 shows the possible orderings of child vertices for various types γ.

Table 3.4: Local vertex orderings of child elements formed by the bisection of the tagged pentatope $\tau = \{\gamma \mid v_0, v_1, v_2, v_3, v_4\}$. The new vertex m is the midpoint of the edge $v_0 v_4$.

	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$
$\sigma(\tau_a)$	(v_0, m, v_1, v_2, v_3)	(v_0, m, v_1, v_2, v_3)	(v_0, m, v_1, v_2, v_3)	(v_0, m, v_1, v_2, v_3)
$\sigma(\tau_b)$	(v_4, m, v_3, v_2, v_1)	(v_4, m, v_1, v_3, v_2)	(v_4, m, v_1, v_2, v_3)	(v_4, m, v_1, v_2, v_3)

The permutation of vertices described in Equation 3.42 is crucial to the success of the bisection method. It can be derived by mapping each simplex element to a Kuhn simplex, which is a kind of reference element contained in the unit cube[48]. The sequence of bisections and vertex shuffles then corresponds to Freudenthal's partition of the unit cube. We refer to [6] for a general exposition of Freudenthal's algorithm applied to simplex meshes.

In a similar vein, the *reflection* of a tagged pentatope $\tau = \{\gamma \mid v_0, v_1, v_2, v_3, v_4\}$ is

$$\tau_R = \begin{cases} \{\gamma \mid v_4, v_3, v_2, v_1, v_0\} & \text{if } \gamma = 0 \\ \{\gamma \mid v_4, v_1, v_3, v_2, v_0\} & \text{if } \gamma = 1 \\ \{\gamma \mid v_4, v_1, v_2, v_3, v_0\} & \text{if } \gamma = 2 \\ \{\gamma \mid v_4, v_1, v_2, v_3, v_0\} & \text{if } \gamma = 3 \end{cases} \tag{3.43}$$

That is, the reflection of τ is another tagging of τ which produces the same children upon bisection. Since the bisection rule is completely determined by the tag on each pentatope, we may consider the tagged pentatopes τ and τ_R to be equivalent.

In addition, we say that two pentatopes τ and τ' that intersect on a tetrahedron are *reflected*

neighbors if the orderings on their vertices match in a particular way. In particular, if $\sigma(\tau')$ matches $\sigma(\tau)$ or $\sigma(\tau_R)$ on all but one position, then τ and τ' are reflected neighbors (note that it is impossible for the two ordering to match completely because τ and τ' have only 4 vertices in common).

Finally, a pentatopal mesh is said to be *consistently tagged* if every two pentatopes sharing a tetrahedral face are reflected neighbors, OR the adjacent children of this pair of pentatopes are reflected neighbors. In practice, it is hard to impose this property except in simple cases. However, we shall see below that there are a number of workarounds to guarantee that a mesh is consistently tagged.

Algorithm 3.2 Psuedocode description of Stevenson's bisection scheme.

```
1:  procedure BISECTPENT(𝒯, τ)          ▷ Outputs new mesh after bisection
2:      BisectList ← ∅
3:      TempBisectList ← τ
4:      while TempBisectList ≠ ∅ do
5:          CheckNext ← ∅
6:          for all τ′ ∈ TempBisectList do
7:              for all τ″ neighbors of τ′, not in BisectList or TempBisectList do
8:                  if τ and τ′ share refinement edge then
9:                      CheckNext ← CheckNext ∪ τ″
10:                 else
11:                     𝒯 ← BISECTPENT(𝒯,τ″)        ▷ Refine τ″ and update 𝒯
12:                 end if
13:             end for
14:         end for
15:         BisectList ← TempBisectList
16:         TempBisectList ← CheckNext
17:     end while
18:     Bisect all element in BisectList along refinement edge
19:     return 𝒯
20: end procedure
```

We describe the main refinement algorithm in [45] in Algorithm 3.2. While the algorithm is recursive, it is proved in [45] that the algorithm always terminates, and the generation of any new pentatope is no more than one greater than the generation of the input τ. Furthermore, if $M \subset \mathcal{T}$ is a collection of pentatopes marked for refinement, then sequentially applying BISECTPENT to all elements in M produces the smallest conforming refinement of $\mathcal{T}$ which bisects every element in M.

The characteristics of this scheme are in many ways optimal. Unfortunately, the behavior of the method relies strongly on the fact that the initial coarse mesh is consistently tagged, which is a nontrivial condition. In fact, it is an open problem as to whether any simplex mesh can be consistently tagged. For triangular meshes, a condition equivalent to consistent tagging was proved in [8] to hold for all meshes. However, no analogous results hold for dimension ≥ 3.

In [24], Kossaczky demonstrated a method or refining an arbitrary two- or three-dimensional mesh into one which is generated by reflections; meshes of this type can always be consistently tagged. This method was generalized into higher dimensions by Stevenson, and we state the particular case for $d = 4$ here.

Let $\mathcal{T}$ be an arbitrary pentatopal mesh. We shall subdivide every element in $\mathcal{T}$ into a series of sub-pentatopes by introducing new vertices within each element. Given a pentatope τ, let R_τ be the barycenter of τ, let S_τ^i be the barycenter of the i^{th} tetrahedral face of τ, and let T_τ^j the barycenter of the j^{th} triangular face of τ (the ordering on the faces of τ may be chosen arbitrarily). In addition, let v_0, v_1, v_2, v_3, v_4 denote the vertices of τ.

For each $\tau \in \mathcal{T}$, refine τ into the collection of pentatopes which have the form

$$\tau_{ij}^{mn} = \{\{R_\tau, S_\tau^i, T_\tau^j, v_m, v_n\}\}$$

where S_τ^j is the barycenter of a tetrahedron containing v_m and v_n, and T_τ^j is the barycenter of a triangle containing v_m and v_n. Furthermore, we tag each new pentatope as:

$$\tau_{ij}^{mn} = \{4 \mid v_m, T_\tau^j, S_\tau^i, R_\tau, v_n\}.$$

where the ordering v_m and v_n may be arbitrary.

Under this tagging scheme, every original edge of the original triangulation is now marked for refinement (since every new pentatope tags v_m and v_n to be its first and last vertices). In addition, by Equation 3.43, we find that the reflected neighbor of every new pentatope has the same vertex ordering with just the first and last vertices transposed. This makes it easy to check that all adjacent pentatopes are reflected in this new mesh.

First, suppose that τ_{ij}^{mn} and $\tau_{i'j'}^{m'n'}$ are two new pentatopes produced from the same ancestor, τ. If these two pentatopes are neighbors, then their intersection is a common tetrahedral face. Therefore, R_τ and S_τ^i must coincide with their primed ($'$) versions. Then, if (v_m, v_n) and $(v_{m'}, v_{n'})$ also match (up to reordering) these two elements are reflected neighbors. If they do not match, then the two pentatopes must share a common triangular face and T_τ^i matches $T_\tau^{i'}$. In this case, a direct application of the bisection rule shows that the adjacent children of these two pentatopes will always be reflected neighbors.

Finally, we may consider the scenario where two new pentatopes were generated from adjacent pentatopes in the coarse mesh. In this setting, $R_{\tau'} \neq R_\tau$, and therefore the remaining four vertices of each pentatope much coincide (otherwise the elements would not be adjacent). In particular, this means that (v_m, v_n) and $(v_{m'}, v_{n'})$ match up to reordering, and thus the two elements are reflected neighbors.

This brings us to our concluding remark, which summarizes the above discussion.

Remark 3.15. The refinement scheme described in Algorithm 3.2 produces conforming refinements of pentatope meshes with a sequence of bisection operations. The number of new elements introduced by this sequence is minimal, and the shape regularity of new elements does not degenerate upon repeated applications of the elements. In order to satisfy the precondition for Algorithm 3.2 (that is, a consistent tagging of the coarse mesh), it is possible to refine the mesh by introducing vertices at face barycenters. After an appropriate tagging of the newly-created elements, the mesh will always be consistently tagged.

Chapter 3, in part, is currently being prepared for submission for publication of the material. The dissertation author was the sole investigator and author of this material.

Chapter 4

Conclusion

Space-time finite element methods based on conforming, unstructured meshes possess a number of useful properties that support a much broader class of discretizations than traditional time-stepping methods. High-dimensional simplex elements can be used to achieve local spatiotemporal refinement in a manner analogous to spatial refinement of triangular and tetrahedral meshes. Because the finite element discretization in these methods is continuous, the resulting numerical schemes are fully implicit and stable. Furthermore, the huge linear system that results from this implicit method can be solved in parallel by applying standard domain decomposition methods. While space-time methods possess an added computational overhead compared to time-stepping methods, parallel implementations have been shown to scale well and outperform time-stepping methods on highly parallel computers[19].

In this dissertation, we addressed two major challenges to the efficient implementation of unstructured space-time finite element methods. Firstly, we examined the behavior of space-time methods applied to general linear parabolic PDEs. At present, we know of no other research which has treated parabolic equations in this general setting. Specifically, we introduce the first analysis of second-order linear parabolic PDEs with non-autonomous convection and reaction coefficients. This work also considers general elliptic operators in the principal term, in contrast to existing work which focuses solely on constant or scalar-valued diffusion operators.

Considered in its space-time formulation, the parabolic equations we consider possess a singular space-time diffusion term, which introduces a numerical instability in traditional Galerkin methods. To stabilize these methods, we defined a streamline-upwind Petrov-Galerkin (SUPG) method for the

space-time setting. In our scheme, we simultaneously upwind in the direction of time and any spatial convection. This choice of upwinding term eliminates the previously-mentioned numerical instability, as well as any instability stemming from convection-dominated flow.

In addition, we have proved that the space-time SUPG method converges at near optimal rates, with the error of the solution and its derivatives following asymptotic decay rates which are standard for upwinded methods. These results hold even for relatively non-smooth solutions possessing $1 + \epsilon$ weak derivatives.

This work establishes some of the fundamental properties of our upwinded space-time method, but research into this scheme can be extended in a number of ways.

Firstly, it would be instructive to prove analogous results for parabolic PDEs posed with Neumann or Robin boundary conditions. Our analysis exclusively considers Dirichlet initial-boundary problems, as does the space-time finite element literature as a whole. Having established the basic theory for Dirichlet conditions, we may observe what modifications are necessary to produce a scheme for other boundary conditions.

Secondly, an extension of this work to problems with moving domains would illustrate the full generality of all-at-once space-time discretizations. In this dissertation, we consider space-time domains with a Cartesian product structure, $Q = \Omega \times [0, T]$. However, this condition is not required. The main change which arises when considering moving spatial boundaries comes from the interaction of the time convection term with the movement of the spatial boundary. When Q is a tensor product, the normal vector to spatial boundary is always orthogonal to the time dimension; that is, $b \cdot n = b_x \cdot n_x$. When the boundary of Ω moves with time, this outward pointing normal will no longer be orthogonal to the time component of the convection.

Finally, we would like to study the effect of upwinding strength on the finite element error. Just like in the traditional analysis of steady-state convection diffusion problems, choosing the upwinding strength too large unnecessarily smooths the solution, while choosing it too small decreases the numerical stability. In [27], a heuristic for element-wise upwinding strength based on a small generalized eigenvalue problem was used. The cited research studied the upwinding scheme in the context of heat equations; however, we see no reason that this cannot be extended to the general parabolic setting.

The second main component of this dissertation was a study of four-dimensional unstructured meshes for space-time methods. Since space-time mesh elements always exist in Euclidean space one

dimension higher than the spatial domain, four-dimensional meshes are required in order to implement space-time finite element methods for PDEs in three-dimensional space. The existing literature on four-dimensional finite element methods is quite sparse, which makes the issue of four-dimensional meshing one of the greatest impediments to applying space-time methods to large-scale problems.

Much of the existing research into four-dimensional triangulations comes from disciplines outside of numerical PDEs. To bridge this gap, we extended foundational concepts from computational geometry into the setting of finite element meshes. These concepts were then used to define a new method for space-time mesh construction in four dimensions. Our meshing algorithm takes as input a spatial mesh in d dimensions and outputs a corresponding space-time mesh in $(d+1)$-dimensions. In practice, this means that any mesh that could be used for a time-stepping scheme can be extended for use in a space-time scheme. Furthermore, this mesh construction process can be instantly parallelized and only one communication step is required at the very end in order to share adjacency information.

As an add-on to our mesh construction algorithm, we also discussed methods of mesh refinement on four-dimensional simplex meshes. Local refinement of simplex meshes increases in complexity in dimensions greater than three, but we were able to apply the bisection procedure of Stevenson to the special case of 4-simplex meshes. While this method is proven to maintain consistency and shape regularity of mesh elements, it requires than any initial mesh satisfy a strict precondition. To address this issue, a modification of the previously introduced meshing algorithm was defined which guarantees this condition is satisfied.

Moving forward, there are a host of open questions regarding 4-simplex meshes which are of direct importance to space-time finite element methods. One of the greatest open challenges at the moment is a general-purpose four-dimensional mesher which can produce a mesh from a four-dimensional skeleton. At the moment, existing mesh generation techniques (our own included) require a tensor product structure on the space-time domain. Algorithms based on Delaunay mesh construction have been proven to work in theory, but may be unacceptably slow in four dimensions. It is possible that a mix of coarse Delaunay mesh generation mixed with local adaptive operations can speed up this process.

This brings us to the next major open problem, which is four-dimensional mesh smoothing. In addition to mesh refinement, it is often helpful to change the structure of a simplicial mesh through vertex movements, edge collapses, and the like. The very recent dissertation of Caplan[11] exhibits a

number of promising operations of this type. It would certainly be helpful to incorporate these refinement operations into the context of space-time meshers.

The field of unstructured space-time finite element methods is a young and active one, and the list of promising research directions could go on for some while. In addition to the questions posed and answered here, the sphere of inquiry into these methods is advancing rapidly on several fronts. Exciting research is ongoing to construct space-time adaptive methods which can target spatiotemporal refinement in a flexible way. Preconditioners and multigrid methods for space-time problems continue to improve the efficiency of the linear system solution. Recently, space-time methods were introduced into the literature of high-performance computing, where the parallel scalability of these methods was highlighted. There is certainly a great deal to continue learning.

Appendix A

Proof of Lemma 2.7

Lemma 2.7 was used in order to prove an estimate on the coefficient for uniform positive-definiteness of $D_{h,\theta}$ in Chapter 2. However, the proof of this lemma is somewhat intricate and the details provide no additional insight into the discussion of space-time methods, so the proof was omitted from Chapter 2. For completeness, we include the full proof here, along with a restatement of Lemma 2.7.

Lemma. *Given constants $A, C > 0$ and $B \geq 0$,*

$$\min_{0\leq z\leq 1} Az^2 + C\left(Bz - \sqrt{1-z^2}\right)^2 \geq \min\left(A + B^2C, \frac{AC}{A + C\,(B^2+B)}\right).$$

Proof. First, note that the objective function $f(z)$ can be simplified as

$$\begin{aligned} f(z) := Az^2 + C\left(Bz - \sqrt{1-z^2}\right)^2 &= Az^2 + CB^2z^2 + C - Cz^2 - 2BCz\sqrt{1-z^2} \\ &= Gz^2 - 2BCz\sqrt{1-z^2} + C, \end{aligned}$$

where we have defined the constant $G = A - C + CB^2$. Note that G is not necessarily positive.

Next, we treat an edge case where $G = 0$ and $B = 0$. By the definition of G, this implies that $A = C$. Then $f(z) = Az^2 + A(1-z^2) = A$, the minimum is A, and this satisfies the statement of the lemma. For the remainder of the proof, we assume that $B^2 + G^2 > 0$.

Since $f(z)$ is smooth on $(0,1)$, the minimum of f is obtained at $z = 0, z = 1$, or at a critical point

of f. The critical points of f lying in $(0,1)$ satisfy

$$\begin{aligned} 0 = f'(z) &= 2Gz - 2BC\sqrt{1-z^2} + 2BC\frac{z^2}{\sqrt{1-z^2}} \\ &= 2Gz + 2BC\frac{2z^2-1}{\sqrt{1-z^2}}, \end{aligned}$$

which implies that

$$(-Gz)^2(1-z^2) = B^2C^2(2z^2-1)^2$$

and consequently

$$\left(G^2 + 4B^2C^2\right)z^4 - \left(G^2 + 4B^2C^2\right)z^2 + B^2C^2 = 0.$$

This equation is quadratic in z^2; if we set $\alpha := G^2 + 4B^2C^2$ and $\beta := B^2C^2$, then the solutions to $\alpha z^4 - \alpha z^2 + \beta = 0$ are:

$$z^2 = \frac{\alpha \pm \sqrt{\alpha^2 - 4\alpha\beta}}{2\alpha} = \frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{\alpha^2 - 4\alpha\beta}{\alpha^2}} = \frac{1}{2}\left(1 \pm \sqrt{\frac{\alpha - 4\beta}{\alpha}}\right).$$

Substituting the values of α and β, we conclude that

$$\begin{aligned} z^2 &= \frac{1}{2}\left(\pm\sqrt{\frac{G^2 + 4B^2C^2 - 4B^2C^2}{G^2 + 4B^2C^2}}\right) \\ &= \frac{1}{2}\left(1 \pm \sqrt{\frac{G^2}{G^2 + 4B^2C^2}}\right). \end{aligned}$$

Since the term under the radical is less than 1, both of these solutions lie in the interval $[0,1]$. The quartic equation above gives has four potential candidates for the critical points (the positive and negative square roots of the two values of z^2 above); however, since the critical points must be positive, we can disregard the negative square roots. Thus we are left with two potential critical points of f:

$$z_- = \sqrt{\frac{1}{2}\left(1 - \sqrt{\frac{G^2}{G^2 + 4B^2C^2}}\right)}, \qquad z_+ = \sqrt{\frac{1}{2}\left(1 + \sqrt{\frac{G^2}{G^2 + 4B^2C^2}}\right)}.$$

It is possible that one or both of these points is not a critical point. Nevertheless, it suffices to bound $f(z_\pm)$.

Substituting the values of $z_\pm$ back into the objective function, we have

$$\begin{aligned}
f(z_\pm) &= G\left(\frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{G^2}{G^2+4B^2C^2}}\right) - 2BC\sqrt{\frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{G^2}{G^2+4B^2C^2}}}\sqrt{\frac{1}{2} \mp \sqrt{\frac{G^2}{G^2+4B^2C^2}}} + C \\
&= \frac{G}{2} + C \pm \frac{G}{2}\sqrt{\frac{G^2}{G^2+4B^2C^2}} - 2BC\sqrt{\frac{1}{4} - \frac{1}{4}\left(\frac{G^2}{G^2+4B^2C^2}\right)} \\
&= \frac{G}{2} + C \pm \frac{G}{2}\sqrt{\frac{G^2}{G^2+4B^2C^2}} - BC\sqrt{\frac{4B^2C^2}{G^2+4B^2C^2}} \\
&= \frac{G}{2} + C \pm \frac{G|G|}{2}\sqrt{\frac{1}{G^2+4B^2C^2}} - 2B^2C^2\sqrt{\frac{1}{G^2+4B^2C^2}} \\
&= \frac{G}{2} + C - \frac{1}{2}\left(\mp G|G| + 4B^2C^2\right)\sqrt{\frac{1}{G^2+4B^2C^2}}.
\end{aligned}$$

The second term will be minimized when the factor $(\mp G|G| + 4B^2C^2)$ is maximized. No matter the sign of G, this maximum is $G^2 + 4B^2C^2$, and it will be attained either at z_+ or z_-. Hence

$$\begin{aligned}
f(z_\pm) &\geq \frac{G}{2} + C - \frac{1}{2}\left(G^2 + 4B^2C^2\right)\sqrt{\frac{1}{G^2+4B^2C^2}} \\
&= \frac{G}{2} + C - \frac{1}{2}\sqrt{G^2+4B^2C^2}.
\end{aligned}$$

This expression may be simplified by multiplying by a fraction to cancel some terms.

$$\begin{aligned}
\frac{G}{2} + C - \frac{1}{2}\sqrt{G^2+4B^2C^2} &= \left(\frac{G}{2} + C - \frac{1}{2}\sqrt{G^2+4B^2C^2}\right) \cdot \frac{\frac{G}{2} + C + \frac{1}{2}\sqrt{G^2+4B^2C^2}}{\frac{G}{2} + C + \frac{1}{2}\sqrt{G^2+4B^2C^2}} \\
&= \frac{\frac{G^2}{4} + GC + C^2 - \frac{1}{4}\left(G^2+4B^2C^2\right)}{\frac{G}{2} + C + \frac{1}{2}\sqrt{G^2+4B^2C^2}} \\
&= \frac{(A - C + B^2C)C + C^2 - B^2C^2}{\frac{G}{2} + C + \frac{1}{2}\sqrt{G^2+4B^2C^2}} \\
&= \frac{AC}{\frac{G}{2} + C + \frac{1}{2}\sqrt{G^2+4B^2C^2}}.
\end{aligned}$$

Finally, by the subadditivity of the square root function, we have that $\sqrt{G^2+4B^2C^2} \leq G + 2BC$. There-

fore,

$$\begin{aligned} f(z_{\pm}) &\geq \frac{AC}{\frac{G}{2} + C + \frac{1}{2}\sqrt{G^2 + 4B^2C^2}} \\ &\geq \frac{AC}{\frac{G}{2} + C + \frac{1}{2}(G + 2BC)} \\ &= \frac{AC}{G + C + BC} \\ &= \frac{AC}{A + B^2C + BC}. \end{aligned}$$

Now, since f is continuous on $[0, 1]$, the global minimum must occur at $z = 0$, $z = 1$, or at a critical point. We have just shown that the value of f at any potential critical point is bounded below by $(AC)(A + B^2C + BC)^{-1}$. Hence

$$\min_{0 \leq z \leq 1} f(z) \geq \min\left(f(0), f(1), \frac{AC}{A + C(B^2 + B)}\right).$$

By direct substitution, $f(0) = C$ and $f(1) = A + CB^2$. Noting that

$$\frac{AC}{A + C(B^2 + B)} = C \cdot \frac{A}{A + C(B^2 + B)} \leq C,$$

we conclude that

$$\min_{0 \leq z \leq 1} f(z) \geq \min\left(f(0), f(1), \frac{AC}{A + C(B^2 + B)}\right) = \min\left(A + CB^2, \frac{AC}{A + C(B^2 + B)}\right),$$

which completes the proof. □

Bibliography

[1] Thomas Apel and Jens M Melenk. "Interpolation and Quasi-Interpolation in h-and hp-Version Finite Element Spaces". *Encyclopedia of Computational Mechanics Second Edition* (2017), pp. 1–33.

[2] Randolph Bank. *Multigraph User's Guide*. Version 2.1. Department of Mathematics, University of California San Diego. 2017.

[3] Randolph E. Bank, Andrew H. Sherman, and Alan Weiser. "Some Refinement Algorithms And Data Structures For Regular Local Mesh Refinement". *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences* 1 (1983), pp. 3–17.

[4] Randolph E Bank, Panayot S Vassilevski, and Ludmil T Zikatanov. "Arbitrary dimension convection–diffusion schemes for space–time discretizations". *Journal of Computational and Applied Mathematics* 310 (2017), pp. 19–31.

[5] Marek Behr. "Simplex space–time meshes in finite element simulations". *International Journal for Numerical Methods in Fluids* 57.9 (2008), pp. 1421–1434.

[6] Jürgen Bey. "Simplicial grid refinement: on Freudenthal's algorithm and the optimal number of congruence classes". *Numerische Mathematik* 85.1 (2000), pp. 1–29.

[7] Jürgen Bey. "Tetrahedral grid refinement". *Computing* 55.4 (1995), pp. 355–378.

[8] Peter Binev, Wolfgang Dahmen, and Ron DeVore. "Adaptive finite element methods with convergence rates". *Numerische Mathematik* 97.2 (2004), pp. 219–268.

[9] Susanne Brenner and Ridgway Scott. *The mathematical theory of finite element methods*. Vol. 15. Springer Science & Business Media, 2007.

[10] Alexander N. Brooks and Thomas J.R. Hughes. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations". *Computer Methods in Applied Mechanics and Engineering* 32.1 (1982), pp. 199–259.

[11] Philip Claude Delhaye Caplan. "Four-dimensional anisotropic mesh adaptation for spacetime numerical simulations". PhD thesis. Massachusetts Institute of Technology, 2019.

[12] P. G. Ciarlet. "Basic error estimates for elliptic problems". In: vol. II. Handbook of Numerical Analysis. North-Holland, Amsterdam, 1991, pp. 17–351.

[13] Max von Danwitz, Violeta Karyofylli, Norbert Hosters, and Marek Behr. “Simplex space-time meshes in compressible flow simulations”. *International Journal for Numerical Methods in Fluids* 91.1 (2019), pp. 29–48.

[14] Jesus A. De Loera, Jorg Rambau, and Francisco Santos. *Triangulations: Structures for Algorithms and Applications*. 1st. Springer Publishing Company, Incorporated, 2010.

[15] Daniele Di Pietro and Alexandre Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*. Vol. 69. 2012.

[16] Michael G. Edwards, J. Tinsley Oden, and Leszek Demkowicz. “An h-r-Adaptive Approximate Riemann Solver for the Euler Equations in Two Dimensions”. *SIAM Journal on Scientific Computing* 14.1 (1993), pp. 185–217.

[17] *Eigen v3*. http://eigen.tuxfamily.org. 2010.

[18] L.C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 2010.

[19] R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, J. B. Schroder, and S. Vandewalle. “Multigrid methods with space–time concurrency”. *Computing and Visualization in Science* 18.4 (2017), pp. 123–143.

[20] Hans Freudenthal. “Simplizialzerlegungen von Beschrankter Flachheit”. *Annals of Mathematics* 43.3 (1942), pp. 580–582.

[21] Martin J. Gander. “50 Years of Time Parallel Time Integration”. In: *Multiple Shooting and Time Domain Decomposition Methods*. Ed. by Thomas Carraro, Michael Geiger, Stefan Körkel, and Rolf Rannacher. Cham: Springer International Publishing, 2015, pp. 69–113.

[22] Jörg Grande. “Red–green refinement of simplicial meshes in d dimensions”. *Mathematics of Computation* 88.316 (2019), pp. 751–782.

[23] Claes Johnson and Jukka Saranen. “Streamline Diffusion Methods for the Incompressible Euler and Navier-Stokes Equations”. *Mathematics of Computation* 47.175 (1986), pp. 1–18.

[24] Igor Kossaczkỳ. “A recursive approach to local mesh refinement in two and three dimensions”. *Journal of Computational and Applied Mathematics* 55.3 (1994), pp. 275–288.

[25] O. A. Ladyzhenskaia. *The boundary value problems of mathematical physics*. English. Springer-Verlag New York, 1985.

[26] U. Langer, M. Neumüller, and I. Toulopoulos. “Multipatch Space-Time Isogeometric Analysis of Parabolic Diffusion Problems”. In: *Large-Scale Scientific Computing*. Ed. by Ivan Lirkov and Svetozar Margenov. Cham: Springer International Publishing, 2018, pp. 21–32.

[27] Ulrich Langer, Martin Neumüller, and Andreas Schafelner. “Space-Time Finite Element Methods for Parabolic Evolution Problems with Variable Coefficients”. In: *Advanced Finite Element Methods with Applications: Selected Papers from the 30th Chemnitz Finite Element Symposium 2017*. Ed. by Thomas Apel, Ulrich Langer, Arnd Meyer, and Olaf Steinbach. Cham: Springer International Publishing, 2019, pp. 247–275.

[28] Ulrich Langer and Andreas Schafelner. “Adaptive space-time finite element methods for non-autonomous parabolic problems with distributional sources”. *arXiv e-prints*, arXiv:2003.09248 (2020).

[29] Ulrich Langer and Andreas Schafelner. “Space-Time Finite Element Methods for Parabolic Evolution Problems with Non-smooth Solutions”. *arXiv e-prints*, arXiv:1903.02350 (2019).

[30] Christoph Lehrenfeld. “On a space-time extended finite element method for the solution of a class of two-phase mass transport problems”. PhD thesis. Universitätsbibliothek der RWTH Aachen, 2015.

[31] Anwei Liu and Barry Joe. “Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision”. *Mathematics of computation* 65.215 (1996), pp. 1183–1200.

[32] Joseph M Maubach. “Local bisection refinement for n-simplicial grids generated by reflection”. *SIAM Journal on Scientific Computing* 16.1 (1995), pp. 210–227.

[33] *MFEM: Modular Finite Element Methods Library*. `mfem.org`.

[34] William F. Mitchell. “30 Years of Newest Vertex Bisection”. *Journal of Numerical Analysis, Industrial and Applied Mathematics* 11 (2017), pp. 11–22.

[35] William F. Mitchell. “Adaptive refinement for arbitrary finite-element spaces with hierarchical bases”. *Journal of Computational and Applied Mathematics* 36.1 (1991), pp. 65–78.

[36] Alexander D Mont. “Adaptive unstructured spacetime meshing for four-dimensional spacetime discontinuous Galerkin finite element methods”. MA thesis. University of Illinois at Urbana-Champaign, 2012.

[37] Stephen Edward Moore. “A stable space–time finite element method for parabolic evolution problems”. *Calcolo* 55.2 (2018), p. 18.

[38] Martin Neumüller and Olaf Steinbach. “Refinement of flexible space–time finite element meshes and discontinuous Galerkin methods”. *Computing and Visualization in Science* 14.5 (2011), pp. 189–205.

[39] K. Olsen, S. Day, Bernard Minster, Yanghao Cui, Amit Chourasia, Marcio Faerman, Reagan Moore, Y. Hu, J. Zhu, Y. Li, Philip Maechling, and Thomas Jordan. “TeraShake: Strong Shaking in Los Angeles Expected From Southern San Andreas Earthquake”. *AGU Fall Meeting Abstracts* -1 (2005), p. 03.

[40] Alfio M. Quarteroni and Alberto Valli. *Numerical Approximation of Partial Differential Equations*. 1st ed. 1994. 2nd printing. Springer Publishing Company, Incorporated, 2008.

[41] María-Cecilia Rivara. “New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations”. *International Journal for Numerical Methods in Engineering* 40.18 (1997), pp. 3313–3324.

[42] Sunil V. Sathe. “Enhanced-discretization and solution techniques in flow simulations and parachute fluid -structure interactions”. PhD thesis. Rice University, 2004.

[43] L. Ridgway Scott and Shangyou Zhang. “Finite Element Interpolation of Nonsmooth Functions Satisfying Boundary Conditions”. *Mathematics of Computation* 54.190 (1990), pp. 483–493.

[44] Igor R Shafarevich and Alexey O Remizov. *Linear algebra and geometry*. Springer Science & Business Media, 2012.

[45] Rob Stevenson. “The completion of locally refined simplicial partitions created by bisection”. *Mathematics of computation* 77.261 (2008), pp. 227–241.

[46] Hiroshi Takenaka and Yushiro Fujii. “A compact representation of spatio-temporal slip distribution on a rupturing fault”. *Journal of seismology* 12.2 (2008), pp. 281–293.

[47] The CGAL Project. *CGAL User and Reference Manual*. 5.0.2. CGAL Editorial Board, 2020.

[48] C. T. Traxler. “An algorithm for adaptive mesh refinement in N dimensions”. *Computing* 59.2 (1997), pp. 115–137.

[49] Alper Üngör and Alla Sheffer. “Tent-Pitcher: A meshing algorithm for space-time discontinuous Galerkin methods”. In: *In proc. 9th int’l. meshing roundtable*. 2000.

[50] Rüdiger Verfürth. “Error estimates for some quasi-interpolation operators”. *ESAIM: Mathematical Modelling and Numerical Analysis* 33.4 (1999), pp. 695–713.

[51] Shangyao Zhang. “Multi-level Iterative Techniques”. PhD thesis. The Pennsylvania State University, 1988.

[52] Shangyou Zhang. “Successive subdivisions of tetrahedra and multigrid methods on tetrahedral meshes”. *Houston J. Math* 21.3 (1995), pp. 541–556.

[53] Xiaozhi Zhang, Jinjun Hu, Lili Xie, and Haiyun Wang. “Kinematic source model for simulation of near-fault ground motion field using explicit finite element method”. *Earthquake Engineering and Engineering Vibration* 5.1 (2006), pp. 19–28.