ORTHOGONAL STRUCTURE

ON A TRIPOD

by

SIERRA NICOLE BATTAN

A THESIS

Presented to the Department of Mathematics
and the Robert D. Clark Honors College
in partial fulfillment of the requirements for the degree of
Bachelor of Science

March 2019

An Abstract of the Thesis of

Sierra Nicole Battan for the degree of Bachelor of Science
in the Department of Mathematics to be taken March 2019

Title: Orthogonal Structure on a Tripod

Approved: _____
Yuan Xu

This thesis establishes an orthogonal basis that accurately represents the structure of polynomials on any three-dimensional tripod. I define, restrict, and describe the contents of an inner product space for a corresponding orthogonal tripod. Then I explicitly construct a basis of the inner product space and study its transformation to an orthogonal basis, using many different algorithmic methods of increasing efficiency. Ultimately, my thesis extends the forefront of mathematical research in the numerical field and helps create a structure with which mathematicians can manipulate currently unmanageable monster polynomials that live in the three-dimensional world.

Acknowledgments

I would like to thank Professor Yuan Xu for being a kind and sympathetic advisor. Writing an undergraduate mathematics thesis is a daunting task, and without his guidance I would have really struggled to do so, especially in half of the traditional time allotted for a Clark Honors College thesis. I express my sincerest gratitude for having the privilege to work with this excellent educator who was willing to lead me through this strenuous, yet rewarding, process. I appreciate his consistent encouragement. I would also like to thank Jennifer Thorenson and Helen Southworth for their assistance with considering this topic from differing perspectives and related contexts.

A special shout out to my friends for making studying fun, celebrating small successes, helping me maintain a healthy school-work-life balance, and reminding each other to eat. Thank you for the distractions, the goofy looks, and the bear hugs. Thank you to all my peers' accomplishments for unknowingly pushing me to strive for higher standards every single day. Thank you to all my math teachers from over the years. Specifically, thank you to Mr. and Mrs. Lang for teaching me that math is cool and to Peter Dodson for reminding me to always stay to the right of zero. Many thanks to Seth Temple and Jamie Zimmerman for providing me with invaluable LaTeX thesis templates. Moo, thank you for the travel opportunities, without which my perspectives would be quite limited. Dance, thank you for teaching me to deal with making mistakes, reminding me to make fun of myself in the biggest way, and being such an active outlet for my nerves and artistic tendencies. J. K. Rowling, thank you for creating a world that has both influenced my life in more ways than I would like to admit and taught me to think of brilliant solutions to potentially problematic situations. Finally, thank you very dearly to my parents for the transfers of love, the endless essay edits, the constant encouragement, and always trusting in my ability to exceed expectations, even when I doubted it myself.

We did it!

Table of Contents

Foreword

Mathematics cumulates upon itself. It is not a subject that allows you to forget any single aspect of what you have previously learned while continuing to study additional topics. Mathematics requires a complete understanding of its foundations and the ability to quickly recall what you already know before you can move on to more complicated concepts. The exploration conducted here draws specifically from a perfect intersection of three mathematical fields — numerical analysis, linear algebra, and multivariable calculus — with a solid basis in classical analysis to prove the validity of my claims.

It has taken a long time for me to arrive at this point in my study of mathematics, and I realize none of my readers have experienced the same journey. As such, I aim to present my research in a universally-approachable manner. If, as a casual reader, you require the definition of some more elementary math terms, please start by reading Appendix A. This miniature dictionary concisely defines all the terms necessary to understand the main text. If you wish to understand my mathematical research's true complexity, then please peruse Appendix B for proofs of the utilized theorems and algorithms.

This thesis is composed of three broad sections. First, I explain the background research upon which my research builds and discuss my research's applications. Then I present the research that I conducted to create a foundation of the final section: four increasingly efficient algorithms, each with the same goal of creating an orthogonal basis accurately represents the structure of any polynomial living on a three-dimensional tripod. Ultimately, I establish one algorithm with an unprecedented level of efficiency.

Overall, my thesis is a true culmination of my education and a proof of my ability to undertake a previously-unattempted feat in the mathematical field and add it to the current wealth of knowledge. This type of success is quite rare for an undergraduate mathematician to accomplish, and I am beyond thankful for the opportunity.

1 Background

This work establishes an orthogonal basis that accurately represents the structure of any polynomial living on a three-dimensional tripod. To do so I define, restrict, and describe the contents of an inner product space on an orthogonal tripod. The original tripod upon which a polynomial lives can be linearly transformed to fit on this orthogonal tripod quite simply. To explicitly construct a basis for the inner product space and study its transformation to become an orthogonal basis, I formulate multiple algorithms of increasing efficiency. I create an orthogonal structure that describes three-dimensional orthogonal polynomials of varying degrees, which mathematicians can utilize to manipulate similar monster (by which I mean complex, messy, and difficult to manage) polynomials living within the same space. Before we continue, however, we must first understand the foundation of my research.

For over a century, mathematicians have used orthogonal polynomials to approximate functions. In 1939, Gabor Szegö published the original text, *Orthogonal Polynomials* [9], in a first official attempt to define orthogonal polynomials and what we are able to do with them. Szegö's text is often termed "The Bible" for orthogonal polynomials, because new publications that situate the origins of orthogonal polynomials often dance around, emulating the main ideas of this text. Szegö's writing also sets a solid foundation for further research, but as it was published so long ago, unfortunately it does not contain any recently conducted research. That said, it is very lucky to have remained relevant for so long. For my research specifically, this text does not have many direct applications. Szegö discusses orthogonal polynomials of only one variable, and here we investigate orthogonal polynomials of three variables.

Published in 1975, Davis J. Philip's text *Interpolation and Approximation* [7] discusses the construction of a basis of orthogonal polynomials, also known as an orthogonal basis, and describes the specifics of inner product spaces in terms of their contents and constraints. This book records many fundamental theorems that elucidate why mathematicians may

use property-maintaining transformations to manipulate bases of an inner product space to determine more manageable structures for the polynomials they house. Discussing these concepts clearly requires a solid foundation in some more basic branches of mathematics.

As aforementioned, my research is the perfect intersection of the three mathematical fields I studied the most in college (multivariable calculus, linear algebra, and numerical analysis) with a foundation in the logic behind all basic mathematics: classical analysis. I must therefore understand the technicalities of the four subjects' corresponding textbooks, and how each specifically relates to my research. Let us discuss these four branches now.

Multivariable Calculus [8], by James Stewart, defines both multivariate and three-dimensional functions, explores their integration, and conceptualizes orthogonality. In the context of my thesis, orthogonality for three-variable polynomials is essentially a fancy classification of three polynomials that are perpendicular to all other polynomials of lower degree. Formally, elements of a set of polynomials $\{P_n(x, y, z), Q_n(x, y, z), R_n(x, y, z)\}$ are orthogonal to any other lower-degree polynomial $S_m(x, y, z)$, where m is less than or equal to $n - 1$, if the inner product of an element and S_m is equal to zero:

Definition 1.1. *P_n is an orthogonal polynomial* $\iff \forall\ S_m \mid m \leq n - 1,\ \langle P_n, S_m \rangle = 0.$

As is often the case in mathematics, there exist additional ways to classify orthogonal polynomials. Here we define three specific variations: monic, mutually-orthogonal, and orthonormal polynomials. Generally, monic polynomials have a single, unweighted term of highest-degree variables. In this thesis' specific context, given later restrictions, the first, highest-degree component of a monic polynomial is simplified into a single, unweighted, isolated variable. For example, if a polynomial P_n is monic, then it is of the form $P_n = x^n +$ $\langle$ a lower-degree linear combination of $x, y, z\rangle$. These monic polynomials' inner product only equals zero when the degree of S_m is of degree $m \leq n - 1$, as mentioned before. On the other hand, mutually-orthogonal polynomials are constructed such that they are orthogonal to all other polynomials in a comparing set of the same degree n or less. When

their identity inner product equals one, we can also deem them orthonormal polynomials. Utilizing orthonormal polynomials turn complicated computations trivial, as their inner products often equal zero, which fosters a great number of simplifications. While monic polynomials are easier to manipulate, orthonormal polynomials are better at improving an algorithm's efficiency and an approximation's accuracy. Luckily, transforming a monic basis into an orthonormal basis is trivial. Multivariable calculus ultimately provides all the techniques necessary to define my three-dimensional structures.

David C. Lay teaches readers of *Linear Algebra and Its Applications* [4] about not only linearity, but also matrices, what it means for a matrix to be invertible and unstable, vector spaces, inner product spaces, how to solve linear systems, and property-conserving transformations. This textbook also describes algorithms vital to my research, like the Gram-Schmidt process, which produces an orthogonal basis for any nonzero subspace:

Theorem 1.1. *Given a basis* $\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ *for a nonzero subspace of* $\mathbb{R}^n$, $\{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$ *is an orthogonal basis of equal dimension, defined as follows:*

$$\begin{aligned}
\mathbf{v}_1 &= \mathbf{x}_1 \\
\mathbf{v}_2 &= \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 \\
\mathbf{v}_3 &= \mathbf{x}_3 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_2}{\mathbf{v}_2 \cdot \mathbf{v}_2}\mathbf{v}_2 \\
&\vdots \\
\mathbf{v}_p &= \mathbf{x}_p - \frac{\mathbf{x}_p \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 - \frac{\mathbf{x}_p \cdot \mathbf{v}_2}{\mathbf{v}_2 \cdot \mathbf{v}_2}\mathbf{v}_2 - \cdots - \frac{\mathbf{x}_p \cdot \mathbf{v}_{p-1}}{\mathbf{v}_{p-1} \cdot \mathbf{v}_{p-1}}\mathbf{v}_{p-1}.
\end{aligned}$$

Gram-Schmidt demonstrates that nonzero subspaces always have orthogonal bases, since all spaces are implicitly defined by any ordinary basis $\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$. Linear algebra's property-conserving transformations allow mathematicians to perform structural shifts, and as such, these techniques allow me to manipulate my three-dimensional structures.

In *Numerical Mathematics and Computing* [2], the authors Ward Cheney and David Kincaid describe how to minimize the accumulated error in any approximation in order

to avoid accuracy loss. Exploiting orthogonality, computerized mathematics, and iterative and recursive algorithms can be used to reduce error. Three-term recursion is a specific form of a recursive algorithm that is extremely useful when using computers to reduce the amount of accumulated error for approximations. Specifically, this algorithm uses the two previous iterations' results to produce the next result: the third result. This process is normally straightforward, but, like any other process, it becomes messy as it enters higher dimensions. An iconic example of a one-dimensional three-term recurrence relation is the Fibonacci sequence: $F_n = F_{n-1} + F_{n-2}$ with initial values $F_0 = 0$ and $F_1 = 1$. Note that three-term recursion requires two initial states. Using this technique greatly contributes to the improved efficiency of my novel algorithms. This text also expresses the importance of algorithmic efficiency. Because the scientific industry now lives in a world of "Big Data" and computations are applied on very large scales, efficiency has become the driving force that determines which algorithms remain relevant. All these numerical techniques are the inspiration behind my thesis, which ultimately improves algorithmic efficiency by combining existing numerical methods.

Finally, *Understanding Analysis* [1], by Stephen Abbot, sets a strong foundation for conducting and proving research in any mathematical field by using definitions, theorems, propositions, proofs, and more. I utilize these methods to identify the problem that this thesis addresses and to prove my solution's validity. Professional mathematicians also use these methods to evaluate the validity of new ideas or completed works.

Other than my college textbooks, which provide extensive background research, there exist only a few other texts relevant to my thesis' foundation. Yuan Xu, a mathematician here at the University of Oregon with whom I have had the pleasure of working closely, has been researching orthogonal structure for many years. Along with Charles F. Dunkl, Xu published a book in 2001, *Orthogonal Polynomials of Several Variables* [3], which describes the current status of multivariate orthogonal polynomials. Xu also produced an identically titled, higher-level, and abridged chapter of this work [10] on his own a few

years later. These publications lay the foundation for Xu's specialized work with another colleague: Sheehan Olver. Together they construct a basis of orthogonal polynomials in two-variables on a wedge in their October 2017 article entitled *Orthogonal Structure on a Wedge and on the Boundary of a Square* [6]. Olver and Xu extend their research in another article from July 2018 entitled *Orthogonal Structure on a Quadratic Curve* [5]. Sitting at the forefront of mathematical research, these publications deem Olver and Xu leaders in the study of orthogonal polynomials and orthogonal structure in this context. Altogether, these research pieces neatly summarize the present state of the field and demonstrate the true novelty of my research topic.

In their 2017 paper, Olver and Xu study classical orthogonal polynomials with respect to weight functions. They represent their wedged inner product space in bilinear form:

$$\langle f, g \rangle = \int_0^1 f(x,1) \times g(x,1) \times w_1(x) dx + \int_0^1 f(1,y) \times g(1,y) \times w_2(y) dy.$$

Here f and g are mutually-orthogonal polynomials and w_1 and w_2 are weight functions. Together, they investigate two large classes of weight functions and explicitly construct an orthogonal basis for their inner product space. Olver and Xu establish analogous results for orthogonal polynomials on the boundary of a square by applying affine transformations to their orthogonal basis. In my personal discussions with Xu, he argues that their findings can define orthogonal structure for any two-dimensional quadratic curve, which they proved together in their 2018 article. He also reveals some additional extensions of their research, such as the implications of taking these univariate orthogonal polynomials and orienting them on curves, or converting them into multivariate orthogonal polynomials. My research more deeply explores the latter.

Now we can begin to discuss the specifics of my research. This paper extends results from Olver and Xu's 2017 article. I imitate its methodologies and physical structure as a means to establish an orthogonal basis for the inner product space of any polynomial living

on a basic three-dimensional tripod. This involves a study of orthogonal polynomials and their corresponding orthogonal structures, which I investigate by drawing parallels to the beginning sections of Olver and Xu's work to similarly establish *Orthogonal Structure on a Tripod*. Accomplishing such a feat requires many steps. First, I must justify why we use an orthogonal tripod to define our inner product space. We now know that any polynomial in $\mathbb{R}[x, y, z]$ is automatically accompanied by some kind of three-dimensional structure. Without much difficulty, special linear transformations can be used to shift this arbitrary structure onto my orthogonal tripod, which is easier to handle on a numerical front. Then this orthogonal tripod can be used to construct a simple inner product space. I must also define this inner product space's structure, describe its appearance, give it a name, establish its contents, determine its size, produce one of its bases, and, finally, transform its basis into one that is orthogonal, efficiently.

Currently, there do exist techniques that can determine the coefficients necessary to take any basis and transform it into an orthogonal basis, but none are specific enough to this context to be particularly useful. Nor have they been combined in the way that I do here. Given recent developments, none of these algorithms are sufficiently efficient either. These methods lack efficiency, so mathematicians consider them naïve and abandon them in modern-day use. I construct a comprehensive algorithm with greater efficiency, which will cause it to survive in the world of "Big Data." Xu and his colleagues will also be able to use my findings to help further research in this specific niche of numerical mathematics.

2 Applications

This research helps mathematicians manipulate unmanageable polynomials in $\mathbb{R}[x, y, z]$ that live on a tripodal shape. Property-conserving linear transformations can be used to fit these monster polynomials onto a more manageable, orthogonal tripod structure of three isolated axes. My findings add to the very forefront of mathematical research that Xu specifically conducts. This research is evidently novel because the most recent and relevant publication on this topic was only published a year before the conclusion of this thesis.

Mathematics aims to broaden all knowledge, sometimes despite the existence of any real-world applications. The real-world application of our research finds itself, however, in the physics field, which is outside the realm of my expertise. Consequently, I cannot comment deeply upon the traditional applicability of orthogonal structure. This does not imply that my work is not useful. As research topics in mathematics become more and more abstract, there exist fewer and fewer real-world applications. Today mathematicians conduct most of their research simply for the purpose of extending knowledge and I am grateful to have the opportunity to add a previously-unattempted mathematical truth to the research pool. Synthesizing relevant ideas from three fields into one concise paper is also extremely useful to the field. For all these reasons, this thesis is ideally situated in the forefront of the mathematical field.

Xu and I discovered a particularly useful application. Generally, functions are written in their traditional form, but they can also be written in an expanded infinite series. These expansions are often used to approximate monster polynomials, and we can improve the approximations' precision by utilizing orthogonal infinite series, or Fourier series. Every function f on our tripod can be expanded as an infinite series of orthonormal polynomials, also known as the Fourier series of a Hilbert Space:

$$f = \sum_{i=1}^{\infty} a_i P_i + \sum_{i=1}^{\infty} b_i Q_i + \sum_{i=1}^{\infty} c_i R_i + d_0.$$

My research will show that if P_i, Q_i, and R_i are orthonormal polynomials, then all of the coefficients a_i, b_i, and c_i are directly determined by $a_i = \langle f, P_i \rangle, b_i = \langle f, Q_i \rangle$, and $c_i = \langle f, R_i \rangle$. If numerical mathematicians ever have the opportunity to use orthonormal polynomials to approximate functions, they do. Orthonormal polynomials are particularly convenient, and, in this case, they make each coefficient's calculation trivial. With this reformation, we can approximate any monster function $f(x, y, z)$ by using a simpler sum of polynomials: the first n terms of its orthogonal expansion. This application can also interpret irregular signal readings by using a sum of the points to approximate the signal using orthonormal equations. Generally, orthonormality simplifies any approximation and improves the efficiency of any algorithm.

Future research in other areas could be conducted using my results as well. One could explore using a change of variables on this basis to arrive at an orthogonal structure for any of the corners of a cube, as Olver and Xu do in their paper of a two-dimensional parallel. This would require affine transformations, which are outside the realm of my expertise. One could also use my research to extend orthogonal structures into the fourth dimension. The potential research extensions are truly endless. Olver and Xu's most-recent paper is a great demonstration this. They succinctly discuss five different types of quadratic curves, define their orthogonal structure, and present the technical applications of this research, those about which I do not have the ability to speak given my current knowledge and grasp of the complexities of these problems. These discoveries are groundbreaking. Ultimately, current research in this very specific niche of numerical mathematics attempts to map out all the orthogonal structures of polynomials living on different shapes of varying dimensions in the polynomial world. My research is another small step of this long journey.

3 Foundations

Before I can present the algorithms that produce orthogonal bases for any polynomial living on a tripod space, I must first establish the foundational research I conducted. My three-dimensional space that will be restricted to become an inner product space is defined as a bilinear sum of each isolated variable component's integral from zero to one:

$$\langle f, g\rangle = \int_0^1 f(x,0,0)g(x,0,0)dx + \int_0^1 f(0,y,0)g(0,y,0)dy + \int_0^1 f(0,0,z)g(0,0,z)dz.$$

You may notice some differences between the structure of my target inner product space and Olver and Xu's inner product space. First of all, I use three variables, x, y, and z, which is justified as I am studying a three-dimensional tripod and not a two-dimensional wedge. Also, there is an absence of weighted functions because my research investigates a simplified problem that removes the need to provide additional transformations. To be considered a true inner product space, the space must always satisfy these conditions:

1. $\langle f, g\rangle$ is linear in each component,
2. $\langle f, g\rangle = \langle g, f\rangle$, and
3. $\langle f, f\rangle \geq 0$ where $\langle f, f\rangle = 0$ if and only if f itself equals 0.

For a more descriptive definition of what it means for a space to be an inner product space, please see Appendix A. At the moment there exist no restrictions on my bilinear space, and though it is commutative and linear in each component, it is not always positive definite. Therefore, this space is not an inner product space in its current domain $\mathbb{R}[x, y, z]$. We must restrict the space so that all the necessary conditions hold.

Let us call this newly restricted space $\mathbb{R}[x, y, z]|\tau$; the three-dimensional polynomial space constrained by an orthogonal tripod τ of isolated variable components. Given the formulation of my inner product space, I have discovered that the easiest way to define its domain $\mathbb{R}[x, y, z]|\tau$ is to set τ equal to a particular polynomial ideal. A polynomial ideal

is represented by a list of polynomials within overloaded angle brackets $\langle p_1, p_2, \ldots, p_n \rangle$, where each polynomial p need not be of the degree by which it is subscripted. Here, the subscript simply represents its ordering. Polynomial ideals describe a larger family of polynomials. For a function f to be a member of $\langle p_1, p_2, \ldots, p_n \rangle$, $f = \sum_{i=1}^{n} \{ \, p_i \times q_i \, \}$ must be true for any other polynomials q_i in the same variables as the ideal polynomials. Since we must maintain positive-definite structure — which requires that we avoid any instance of $\langle p, p \rangle = 0$ when $p \neq 0$ — it is imperative that the space on which it is defined, $\mathbb{R}[x, y, z]|\tau$, excludes a polynomial ideal. The existing multiplication within my inner product space of isolated variables with others set to zero clearly implies that $\langle xy, xz, yz \rangle$ must restrict the original $\mathbb{R}[x, y, z]$ space. Multiplication of any variable's value by zero results in a zero function, which therefore invalidates the space as an inner product space. Formally, I propose the following, and prove this assessment in Appendix B:

Theorem 3.1. *The domain that restricts our inner product*

$$\langle f, g \rangle = \int_0^1 f(x, 0, 0) g(x, 0, 0) dx + \int_0^1 f(0, y, 0) g(0, y, 0) dy + \int_0^1 f(0, 0, z) g(0, 0, z) dz$$

such that it becomes a proper inner product space is $\mathbb{R}_n[x, y, z] \setminus \langle xy, xz, yz \rangle$.

It is important to discuss what this domain looks like for each degree n in order to truly understand how to create an orthogonal basis for the inner product space. Please note that we can define a subspace of degree n within my inner product space because the original space, defined in terms of the sum of three integrals, can be rewritten as the sum of three derived sums from zero to infinity, where n is the highest dimension of each derived sum's polynomials. This allows me to use an iterative approach. If we recall aspects of what numerical analysis teaches us we will remember that this starts us on the path of creating a more-efficient algorithm, which calculates an orthogonal basis for my defined orthogonal structure. Let us take a look at a basis for each n of the original $\mathbb{R}_n[x, y, z]$:

$$
\begin{aligned}
n = 0: &\quad 1 \\
n = 1: &\quad x \quad y \quad z \\
n = 2: &\quad x^2 \quad xy \quad xz \quad y^2 \quad yz \quad z^2 \\
n = 3: &\quad x^3 \quad x^2y \quad xy^2 \quad x^2z \quad xz^2 \quad y^3 \quad y^2z \quad yz^2 \quad z^3 \quad xyz.
\end{aligned}
$$

Now let us look at the restricted $\mathbb{R}_n[x, y, z]|\tau$ to see which basic polynomials are removed once we consider the constraining polynomial ideal $\langle xy, xz, yz\rangle$:

$$
\begin{aligned}
n = 0: &\quad 1 \\
n = 1: &\quad x \quad y \quad z \\
n = 2: &\quad x^2 \quad y^2 \quad z^2 \\
n = 3: &\quad x^3 \quad y^3 \quad z^3.
\end{aligned}
$$

Evidently, each n^{th}-degree basis of non-orthogonal polynomials for my inner product space is of the form $\{x^n, y^n, z^n\}$. These are the polynomials needed to create an orthogonal basis.

Now let us determine the appropriate dimension of the n^{th}-degree inner product space. In the second list above with a restricted domain, we see that the 0^{th}-degree basis contains only one polynomial (1) and so its inner product space has dimension one. Similarly, the first-degree basis contains three polynomials ($x, y,$ and z) and so its inner product space has dimension three, as does the second- and third-degree inner product space. I prove that this pattern follows for each degree of $\mathbb{R}_n[x, y, z] \setminus \langle xy, xz, yz\rangle$ in Appendix B:

Theorem 3.2. *Our inner product space's restrictive domain has the following dimensions:*

$$\textit{When } n = 0, dim(\mathbb{R}_n[x, y, z] \setminus \langle xy, xz, yz\rangle) = 1.$$

$$\forall n \in \mathbb{N} : n > 0, dim(\mathbb{R}_n[x, y, z] \setminus \langle xy, xz, yz\rangle) = 3.$$

If we used the unrestricted domain, $\mathbb{R}_n[x, y, z]$, for our inner product space, then we would have much larger bases to deal with, especially for a large n. Our dimension would have been of size $\binom{n+2}{2}$, which grows quickly for increasing n. If this was the case (and

there was a week when Xu and I were convinced that it was) then this thesis would likely not have come to fruition. Instead, we can use $\mathbb{R}_n[x, y, z] \setminus \langle xy, xz, yz \rangle$ as our domain, and we are left with convenient bases of three elements that simplify many future manipulations.

Finally, I can present this thesis' true focus: four algorithms that establish orthogonal bases that accurately represent the orthogonal structure of any polynomial situated on a three-dimensional tripod. These algorithms are calculated techniques that ultimately define which bases of my inner product space satisfy all constraints and determine how to calculate the necessary coefficients to make the bases orthogonal, efficiently.

4 Naïve Algorithm

The first, simplistic algorithm transforms a regular basis into an orthogonal basis by framing the defined problem in linear algebra terms. Let the n^{th}-degree inner product space domain of the restricted tripod be $V_n = \mathbb{R}_n[x, y, z] \setminus \langle xy, xz, yz \rangle$ and call its basis of three monic polynomials $B_n = \{P_n, Q_n, R_n\}$. By the definition of a monic polynomial and the restriction on each V_i, each element of a basis B_n is of the form

$$P_n(x, y, z) = x^n + \sum_{i=1}^{n-1} a_i^P x^i + \sum_{i=1}^{n-1} b_i^P y^i + \sum_{i=1}^{n-1} c_i^P z^i + d_0^P$$

where the a_i^Ps, b_i^Ps, c_i^Ps and d_0^P are all constants in $\mathbb{R}$ and where the subscript delineates that these coefficients belong to P_n. Both Q_n and R_n are similarly formed with respective leading components y^n and z^n. The three sums represent a lower-degree polynomial in terms of all the allowable variables. Since P_n is monic, we can determine it by using $\langle P_n, x^j \rangle = \langle P_n, y^j \rangle = \langle P_n, z^j \rangle = 0$ for each j from $0, 1, \ldots, n-1$, which holds true for both Q_n and R_n too. With simple substitutions and inner product manipulations, we can rewrite P_n without "even doing anything" (as Xu likes to say):

$$\langle P_n, x^j \rangle = 0, \langle P_n, y^j \rangle = 0, \langle P_n, z^j \rangle = 0$$

$$\Updownarrow$$

$$\langle x^n + \sum_{i=1}^{n-1} a_i^P x^i + \sum_{i=1}^{n-1} b_i^P y^i + \sum_{i=1}^{n-1} c_i^P z^i + d_0^P, x^j \rangle = 0$$

$$\langle x^n + \sum_{i=1}^{n-1} a_i^P x^i + \sum_{i=1}^{n-1} b_i^P y^i + \sum_{i=1}^{n-1} c_i^P z^i + d_0^P, y^j \rangle = 0$$

$$\langle x^n + \sum_{i=1}^{n-1} a_i^P x^i + \sum_{i=1}^{n-1} b_i^P y^i + \sum_{i=1}^{n-1} c_i^P z^i + d_0^P, z^j \rangle = 0$$

$$\Updownarrow$$

$$\langle d_0^P, x^j \rangle + \sum_{i=1}^{n-1} a_i^P \langle x^i, x^j \rangle + \sum_{i=1}^{n-1} b_i^P \langle y^i, x^j \rangle + \sum_{i=1}^{n-1} c_i^P \langle z^i, x^j \rangle = -\langle x^n, x^j \rangle$$

$$\langle d_0^P, y^j\rangle + \sum_{i=1}^{n-1} a_i^P \langle x^i, y^j\rangle + \sum_{i=1}^{n-1} b_i^P \langle y^i, y^j\rangle + \sum_{i=1}^{n-1} c_i^P \langle z^i, y^j\rangle = -\langle x^n, y^j\rangle$$

$$\langle d_0^P, z^j\rangle + \sum_{i=1}^{n-1} a_i^P \langle x^i, z^j\rangle + \sum_{i=1}^{n-1} b_i^P \langle y^i, z^j\rangle + \sum_{i=1}^{n-1} c_i^P \langle z^i, z^j\rangle = -\langle x^n, z^j\rangle.$$

We can similarly construct the other monic polynomials, Q_n and R_n.

This results in a large system of linear equations that we must solve to determine which coefficients of a_i^Ps, b_i^Ps, c_i^Ps, and d_0^P create a monic polynomial for the new basis of a larger degree. I want to briefly take the time to explain how tedious this can become. For a small basis of the first degree, yes, calculating only one set of a_1^P, b_1^P, c_1^P, and d_0^P is simple. But then we have to use a sum of these newly determined coefficients to construct the second degree basis, resulting in two sets of a_i^P, b_i^P, c_i^P for $i = \{1, 2\}$ and one d_0^P. By the time we get to any intriguing basis of a higher degree, we must calculate many sets of a_i^P, b_i^P, c_i^P, and d_0^P. Executing these inefficient algorithms is a banal process that must be completed before we even reach the main goal of our algorithm.

Let us not forget that right now we are only discussing the coefficients that need to be determined for P_n. We must conduct this tedious process of solving for the coefficients of three equations for each monic polynomial of the desired basis $B_n = \{P_n, Q_n, R_n\}$ for each lower-degree j from $0, 1, \ldots, n-1$. We are thus met with three enormous $(3n-2)\times(3n-2)$ independent systems of equations for each basis of degree $n = 1, 2, 3, \ldots$. These large matrices are also unstable, which complicates the orthogonalization process even further.

Only after we produce the entirety of this new B_n are we allowed to tackle our goal: turning the basis into an orthogonal basis, appropriately named OB_n. Right now we can only conduct this transformation by using traditional linear algebra techniques, which adds on even more complexity to this already tedious process. Yes, this algorithm is clearly defined, but it requires a massive amount of computational power and is very unorganized. It is also terribly inefficient and implementing this algorithm is not worth the trouble. Thus, we call this method naïve, and for good reason.

5 Basic Algorithm

The next method implements the current standard for creating an orthogonal basis from any other basis. Here, we are introducing the concept of induction. Generally, inductive algorithms are convenient because they introduce recursion, which increases algorithmic efficiency. We will use the same structure as we did previously, starting with a monic basis $B_n = \{P_n, Q_n, R_n\}$. However, we were using linear algebra techniques to solve the three sets of $(3n-2) \times (3n-2)$ system of equations from before, and now we will use induction to simplify these equations further.

Before, we simply defined each element of B_n using an arbitrary linear weighted sum of isolated variable components. In this more-efficient algorithm, we will slightly adapt this basis. Now that we are using induction, let us assume that we also know each basis B_i for $i = 0, 1, \ldots, n-1$. This means that the algorithm has access to all the previous P_is, Q_is, and R_is. So let us utilize them to construct the next basis $B_n = \{P_n, Q_n, R_n\}$. Let us now take a combination of the previous elements into our newly-formed polynomials,

$$P_n(x, y, z) = x^n + \sum_{i=1}^{n-1} a_i^P P_i(x, y, z) + \sum_{i=1}^{n-1} b_i^P Q_i(x, y, z) + \sum_{i=1}^{n-1} c_i^P R_i(x, y, z) + d_0^P,$$

and similarly, for each other polynomial Q_n and R_n. Using this inductive procedure, we can enforce that a new basis $OB_n = \{P_n, Q_n, R_n\}$ will, in fact, be orthogonal. However, we must still determine the values of all the weights a_i, b_i, and c_i, as well as the initial d_0, for each monic polynomial. First, let us significantly reduce these equations.

Recall that before P_n was determined by $\langle P_n, x^j \rangle = \langle P_n, y^j \rangle = \langle P_n, z^j \rangle = 0$ for each lower-degree j from $0, 1, \ldots, n-1$. Now we can similarly determine P_n by $\langle P_n, P_j \rangle = \langle P_n, Q_j \rangle = \langle P_n, R_j \rangle = 0$ for the same fixed js. This is allowable because each element of the basis that we are now attempting to build is, by its monic polynomial definition, orthogonal to the lower-degree bases we can access. Therefore, using the same logic we discussed above, the formulation of P_n can now be simplified to the following three equations:

$$\sum_{i=1}^{n-1} a_i^P \langle P_i, P_j \rangle + \sum_{i=1}^{n-1} b_i^P \langle Q_i, P_j \rangle + \sum_{i=1}^{n-1} c_i^P \langle R_i, P_j \rangle = -\langle x^n, P_j \rangle$$

$$\sum_{i=1}^{n-1} a_i^P \langle P_i, P_j \rangle + \sum_{i=1}^{n-1} b_i^P \langle Q_i, P_j \rangle + \sum_{i=1}^{n-1} c_i^P \langle R_i, P_j \rangle = -\langle x^n, P_j \rangle$$

$$\sum_{i=1}^{n-1} a_i^P \langle P_i, P_j \rangle + \sum_{i=1}^{n-1} b_i^P \langle Q_i, P_j \rangle + \sum_{i=1}^{n-1} c_i^P \langle R_i, P_j \rangle = -\langle x^n, P_j \rangle$$

Again, both Q_n and R_n are similarly simplified. Notice the inner product of each lower-degree basis element and d_0^P is absent. Induction gives us a benefit here. Since $\langle P_n, P_0 \rangle = 0$, properties of inner products tell us that $d_0^P \langle P_0, P_0 \rangle = 0$ too, and therefore $d_0^P = 0$ as well. Furthermore, we now need to pay attention to each fixed j from $1, 2, \ldots, n-1$.

Monic orthogonality tells us that $\langle P_i, P_j \rangle = \langle P_i, Q_j \rangle = \langle P_i, R_j \rangle = 0$ whenever $i \neq j$. So, most of the elements we sum in these equations equal zero. Ultimately, again using the same logic as before, we arrive at a matrix of these three mutually-orthogonal polynomials that will eventually construct P_n by determining all its coefficients, for the fixed j only:

$$\begin{aligned}
a_j^P \langle P_j, P_j \rangle + b_j^P \langle Q_j, P_j \rangle + c_j^P \langle R_j, P_j \rangle &= -\langle x^n, P_j \rangle \\
a_j^P \langle P_j, Q_j \rangle + b_j^P \langle Q_j, Q_j \rangle + c_j^P \langle R_j, Q_j \rangle &= -\langle x^n, Q_j \rangle \\
a_j^P \langle P_j, R_j \rangle + b_j^P \langle Q_j, R_j \rangle + c_j^P \langle R_j, R_j \rangle &= -\langle x^n, R_j \rangle.
\end{aligned}$$

With P_n's property of orthogonality, we know that this 3×3 matrix is invertible and we can calculate all of its constants (all the $a_j^P, b_j^P, c_j^P \in \mathbb{R}$) for each lower-degree j from $1, 2, \ldots, n-1$. To finally produce our next basis B_n, we must also construct Q_n and R_n by replacing x^n with their respective y^n and z^n.

Now we can attempt to turn this basis B_n into OB_n. Since $B_n = \{P_n, Q_n, R_n\}$, let us again define $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$. Rather than following the elementary linear algebra techniques from before in the naïve algorithm, let us implement Gram-Schmidt:

1. $\hat{P}_n = P_n$
2. $\hat{Q}_n = Q_n + a\hat{P}_n$. Use orthogonality to determine a.
3. $\hat{R}_n = R_n + b\hat{P}_n + c\hat{Q}_n$. Use orthogonality to determine b and c.

Note that these coefficients are a different set of $a, b,$ and c than before. Gram-Schmidt allows us to calculate the constant coefficients of these variables relatively quickly, but not simply. At least not yet. Due to magnitude of these coefficients, calculating higher degrees of these orthogonal bases could become very unorganized and tedious.

By using this algorithm's iterative process, we are able to induce that each step needs to solve $3(n-1)$ sets of the 3×3 system of equations from above. This algorithm is messy and still takes a long time, and the matrices could also be unstable, but it is nevertheless a great improvement over the naïve algorithm's three sets of a $(3n-2) \times (3n-2)$ matrix. Solving small systems of equations is much easier than solving large systems of equations, especially those that grow in a linear fashion. Computing these coefficients will be much faster than before, even though we have $3(n-1)$ matrices to solve instead of the original three. Clearly, this demonstrates just how naïve the first algorithm is. Alas, this algorithm itself cannot be made more efficient, as there exist too many dependencies that we cannot isolate, and it thus requires sub-algorithms to calculate each orthogonal basis. Although this algorithm's organization is messy, it does the job satisfactorily, so many mathematicians still use it. This is why we consider this algorithm basic.

6 Clever Algorithm

We are close to arriving at an intelligent algorithm. The previous algorithm is nearly there, but we utilize a general and isolated highest-degree component when formulating our monic polynomials. So, when it comes to calculating all of the coefficients, we do not make use of all the available advantages. Now that we have used Gram-Schmidt to reduce the initial algorithm of constructing a basis into a polynomial time complexity, it is time to add another improvement to this algorithm by implementing a key technique from numerical mathematics: three-term recursion. Instead of writing each monic polynomial of the basis B_n in the form from the previous algorithm, let us slightly modify its structure. Let us replace each corresponding isolated variable component of the highest-degree with a recursive version that maintains the same degree. For P_n we now have the following form:

$$P_n(x,y,z) = xP_{n-1}(x,y,z) + \sum_{i=1}^{n-1} a_i^P P_i(x,y,z) + \sum_{i=1}^{n-1} b_i^P Q_i(x,y,z) + \sum_{i=1}^{n-1} c_i^P R_i(x,y,z) + d_0^P.$$

Q_n and R_n are similarly formed using their respective $yQ_{n-1}(x,y,z)$ and $zR_{n-1}(x,y,z)$ to replace the original highest-degree component y^n and z^n. Remember that three-term recursion requires two initial states. In our case, we have already established that $B_0 = \{1\}$, where 1 is the value of each P_0, Q_0, and R_0. Then we know that the first-degree basis will be generally of the form $B_1 = \{x + d_0^P, y + d_0^Q, z + d_0^R\}$, where each d_0 need not be the same value, as each is dependent on which P_1, Q_1, and R_1 we are calculating. By running Gram-Schmidt I have determined that $B_1 = \{x - \frac{1}{6}, y - \frac{1}{6}, z - \frac{1}{6}\}$. So in this case, each d_0 does have the same value, but that is simply coincidental. Now that we have established the iterative algorithm, we can continue along the execution of this clever algorithm.

If you can recall, the basic algorithm resulted in $3(n-1)$ sets of a 3×3 matrix. Let us again commit the above modification to arrive at these matrices of a slightly different form:

$$a_j\langle P_j, P_j\rangle + b_j\langle Q_j, P_j\rangle + c_j\langle R_j, P_j\rangle = -\langle xP_{n-1}, P_j\rangle$$
$$a_j\langle P_j, Q_j\rangle + b_j\langle Q_j, Q_j\rangle + c_j\langle R_j, Q_j\rangle = -\langle xP_{n-1}, Q_j\rangle$$
$$a_j\langle P_j, R_j\rangle + b_j\langle Q_j, R_j\rangle + c_j\langle R_j, R_j\rangle = -\langle xP_{n-1}, R_j\rangle.$$

Remember that we are creating a recursive algorithm that implements Gram-Schmidt, and therefore, each polynomial is already automatically a mutually-orthogonal polynomial. By definition, we know that many of the above 3×3 matrices will now equal zero for different values of our fixed j. In fact, with this reformation, I propose a new theorem:

Theorem 6.1. *If our algorithm implements three-term recursion, then for* $i = 0, 1, \ldots n-3$*, each monic polynomial's coefficients* $a_i = b_i = c_i = 0$.

This bold statement is hard to immediately comprehend. If you need to be convinced of its validity, please see my formal proof in Appendix B. In particular, our equations that directly compute each element of our bases are reduced to

$$\begin{aligned} P_n(x,y,z) = (x - a^P_{n-1})P_{n-1}(x,y,z) + a^P_{n-2}P_{n-2}(x,y,z) \\ + b^P_{n-1}Q_{n-1}(x,y,z) + b^P_{n-2}Q_{n-2}(x,y,z) \\ + c^P_{n-1}R_{n-1}(x,y,z) + c^P_{n-2}R_{n-2}(x,y,z). \end{aligned}$$

Q_n and R_n are similarly formed with y and z associated with b^P_{n-1} and c^P_{n-1}, respectively.

With this discovery, we greatly reduce the number of coefficients that we must compute. Before we had $3(n-1)$ 3×3 matrices. Now we only have six 3×3 matrices — two for each P_n, Q_n and R_n — since the theorem tells us that the only values of j that we must consider are $n-2$ and $n-1$. Clearly this demonstrates the benefit of three-term recursion.

All we must do now is run Gram-Schmidt on the defined B_n to obtain our desired OB_n. Luckily this algorithm's reformation makes Gram-Schmidt even easier. As before, define $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$. Now we can run Gram-Schmidt and reveal the existence of some nice simplifications along the way:

1. Set $\hat{P}_n = P_n$.
2. Set $\hat{Q}_n = Q_n + a\hat{P}_n$. Which, by definition, is equivalent to $\hat{Q}_n = Q_n + aP_n$.

   Calculating a would normally be difficult, however, this formulation has fewer dependencies between the inner products than before in the basic algorithm. So, since $\langle \hat{Q}_n, \hat{P}_n \rangle = 0$, by definition we know that $\langle Q_n + aP_n, P_n \rangle = 0$. We can therefore use inner product properties to see that $-\langle Q_n, P_n \rangle = a\langle P_n, P_n \rangle$. Thus, $a = -\frac{\langle Q_n, P_n \rangle}{\langle P_n, P_n \rangle}$, which is a simple, constant-time computation that we can substitute in to the original equation without much additional work.
3. Set $\hat{R}_n = R_n + b\hat{P}_n + c\hat{Q}_n$.

   Using the same logic as in step two, we can determine b fairly easily following the normal rules of Gram-Schmidt. Since $\langle \hat{R}_n, \hat{P}_n \rangle = 0$, it is implied that $-\langle R_n, P_n \rangle = b\langle P_n, P_n \rangle$, so $b = -\frac{\langle R_n, P_n \rangle}{\langle P_n, P_n \rangle}$. Given the simple definition of Gram-Schmidt, we also know that $c = -\frac{\langle R_n, \hat{Q}_n \rangle}{\langle \hat{Q}_n, \hat{Q}_n \rangle}$. More specifically, because we know the definition of $\hat{Q}_n$, we can see that $c = -\frac{\langle R_n, Q_n + aP_n \rangle}{\langle Q_n + aP_n, Q_n + aP_n \rangle}$. Calculating c is more complicated than calculating the other two coefficients.

Therefore, after running Gram-Schmidt, we are met with one 3×3 diagonalized matrix and all our computations are near trivial. Furthermore, we implemented recursion, and so this clever algorithm can be calculated with greater efficiency than any before. This algorithm has come a long way from the original naïve algorithm, and for that, we should be proud, but not yet satisfied.

7 Brilliant Algorithm

We have arrived at the final algorithm. It utilizes inductive and recursive methods to obtain maximal efficiency: the underlying goal of numerical mathematics and this thesis. This algorithm incorporates all the small improvements I made to the first three algorithms.

In the previous algorithm, we had to rely upon tightly-bound dependencies between each P_n, Q_n, and R_n. This is not ideal because as the number of dependencies increases, efficiency decreases, especially since we have to make more computations overall. So, let us now abandon the traditional monic polynomial structure that we have been using. In each of the prior algorithms, we construct a generic B_n and transform it into an OB_n, while still using the polynomials from B_n to construct the next higher-degree basis. Here we will instead use the polynomials from OB_n to construct the next basis. Monic polynomials are useful because they are easy to manipulate, but mutually-orthogonal polynomials (and orthonormal polynomials) help improve algorithmic efficiency. So let us begin to use those mutually-orthogonal polynomials in the formulation of each higher-degree basis.

Suppose we are trying to produce $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$ and we know the orthogonal basis for all $\mathbb{R}_i[x, y, z] \setminus \langle xy, xz, yz \rangle$ of degrees $0 \leq i < n$. We must first focus on creating our $B_n = \{P_n, Q_n, R_n\}$, as we cannot simply construct an orthogonal basis without some initial work. Let each element of B_n be of the following form:

$$P_n(x, y, z) = x\hat{P}_{n-1}(x, y, z) + \sum_{i=1}^{n-1} a_i^P \hat{P}_i(x, y, z) + \sum_{i=1}^{n-1} b_i^P \hat{Q}_i(x, y, z) + \sum_{i=1}^{n-1} c_i^P \hat{R}_i(x, y, z) + d_0^P.$$

Q_n and R_n are formed similarly using their respective $y\hat{Q}_{n-1}(x, y, z)$ and $z\hat{R}_{n-1}(x, y, z)$. Now we simply continue following the smart algorithm for P_n to attain a 3×3 matrix of undetermined coefficients, except this time, a lot more of these coefficients will equal zero. Given the definition of a monic mutually-orthogonal polynomial, we know that for all of

the smaller OB_is, each of its elements are orthogonal to each other. Therefore, to define P_n for each fixed $j \in \{n-1, n-2\}$, we previously had a matrix that looked like this:

$$\begin{aligned}
a_j^P\langle \hat{P}_j, P_j\rangle + b_j^P\langle \hat{Q}_j, P_j\rangle + c_j^P\langle \hat{R}_j, P_j\rangle &= -\langle x\hat{P}_{n-1}, P_j\rangle \\
a_j^P\langle \hat{P}_j, Q_j\rangle + b_j^P\langle \hat{Q}_j, Q_j\rangle + c_j^P\langle \hat{R}_j, Q_j\rangle &= -\langle x\hat{P}_{n-1}, Q_j\rangle \\
a_j^P\langle \hat{P}_j, R_j\rangle + b_j^P\langle \hat{Q}_j, R_j\rangle + c_j^P\langle \hat{R}_j, R_j\rangle &= -\langle x\hat{P}_{n-1}, R_j\rangle,
\end{aligned}$$

Instead, we now have matrix that is simplified down to this:

$$\begin{aligned}
a_j^P\langle P_j, P_j\rangle + 0 + 0 &= -\langle xP_{n-1}, P_j\rangle \\
0 + b_j^P\langle Q_j, Q_j\rangle + 0 &= -\langle xP_{n-1}, Q_j\rangle \\
0 + 0 + c_j^P\langle R_j, R_j\rangle &= -\langle xP_{n-1}, R_j\rangle.
\end{aligned}$$

One glance at these systems reveals what each coefficient's value should be. For example, $b_{n-2} = -\frac{\langle x\hat{P}_{n-1}, \hat{Q}_{n-2}\rangle}{\langle \hat{Q}_{n-2}, \hat{Q}_{n-2}\rangle}$. If these mutually-orthogonal polynomials are also orthonormal, then each coefficient is simplified further. To continue our example, we now have that $b_{n-2} = -\langle x\hat{P}_{n-1}, \hat{Q}_{n-2}\rangle$. Given this simplification, we can uniquely determine P_n as such:

$$\begin{aligned}
P_n(x,y,z) = (x - a_{n-1}^P)\hat{P}_{n-1}(x,y,z) + a_{n-2}^P\hat{P}_{n-2}(x,y,z) \\
+ b_{n-1}^P\hat{Q}_{n-1}(x,y,z) + b_{n-2}^P\hat{Q}_{n-2}(x,y,z) \\
+ c_{n-1}^P\hat{R}_{n-1}(x,y,z) + c_{n-2}^P\hat{R}_{n-2}(x,y,z).
\end{aligned}$$

Of course, similarly for Q_n and R_n. Though this equation looks similar to that in the clever algorithm, the mutually-orthogonal polynomials' circumflexes indicate its true brilliance.

Recall that the next step in this process is running Gram-Schmidt. If you have a good grasp of Gram-Schmidt, you'll remember that nothing changes during the conversion of P_n to $\hat{P}_n$. Thus, this algorithm can construct $\hat{P}_n$ incredibly concisely. This is not true for

$\hat{Q}_n$ nor $\hat{R}_n$, however, because both $\hat{Q}_j$ and $\hat{R}_j$ need not be monic in previous lower-degree orthogonal bases. We must still perform Gram-Schmidt to determine the coefficients of $\hat{Q}_n$ and $\hat{R}_n$ that transform B_n to OB_n. As such, this method is not entirely recursive. Yes, it implements three-term recursion, and that is all $\hat{P}_n$'s construction requires. However, there still exist dependencies for $\hat{Q}_n$ and $\hat{R}_n$ that need to be isolated by using a Gram-Schmidt subroutine. That additional step does not require much more work, though, since the simplicity of this algorithm's structure determines each coefficient immediately. This is the best mathematicians can do, at least for right now.

Conclusion

This thesis establishes an orthogonal basis that accurately represents the structure of polynomials on any three-dimensional tripod. I demonstrate that an orthogonal tripod to regulate these orthogonal structures is defined by an inner product space restricted from its original domain by an orthogonal tripod $\mathbb{R}_n[x, y, z] \setminus \langle xy, xz, yz \rangle$:

$$\langle f, g \rangle = \int_0^1 f(x, 0, 0)g(x, 0, 0)dx + \int_0^1 f(0, y, 0)g(0, y, 0)dy + \int_0^1 f(0, 0, z)g(0, 0, z)dz.$$

My research explicitly constructs a basis of this inner product space and studies its transformation into an orthogonal basis, using different algorithms of increasing efficiency. I start with an algorithm that creates a basis of monic polynomials with an isolated highest-degree variable plus an arbitrary lower-degree polynomial by using naïve linear algebra techniques. This method requires solving three huge $(3n-2) \times (3n-2)$ matrices, where n is the basis' degree. I improve the algorithm by iteratively using each lower-degree basis' elements to produce new monic polynomials and using Gram-Schmidt to solve the resulting $3(n-1)$ systems of 3×3 equations. This basic algorithm is what most mathematicians today would likely use to produce such a basis. I then introduce the idea of using three-term recursion by replacing each arbitrary power of x^n, y^n, and z^n with a multiplication of each variable, powerless, and their respective previous polynomial P_{n-1}, Q_{n-1}, and R_{n-1}. This algorithm is clever in its formulation, as recursion is common and preferred, but not always immediately available to the initial problem. Finally, I construct our bases using mutually-orthogonal polynomials for greater efficiency and orthonormal polynomials for additional simplification. This brilliant method exploits all the available advantages and computes most of the coefficients in constant time.

Ultimately, my thesis extends the forefront of research in this very specific niche of numerical mathematics and creates a structure that mathematicians can use to manipulate currently unmanageable monster polynomials living in the three-dimensional world.

Appendix A — Definitions

- A real-valued constant a is a number in $\mathbb{R}$, the real coordinate space of one dimension.
- The three-dimensional space $\mathbb{R}^3 = \{(x, y, z) : x, y, z \in \mathbb{R}\}$, the Euclidian coordinate space of three real dimensions, distinguishes corresponding directions with x, y, z.
- The polynomial space of three variables $\mathbb{R}[x, y, z] = \{p(x, y, z) : x, y, z \in \mathbb{R}\}$ is a space of nothing other than polynomials. This is a fairly abstract concept, briefly introduced in linear algebra, but is similar to the concept of a vector space.
- If there exists a polynomial p in terms of a variable x in the form $p(x) = a_0x^0 + a_1x^1 + \cdots + a_{n-1}x^{n-1} + a_nx^n$ where each $a_0, a_1, \ldots, a_{n-1}, a_n$ are in $\mathbb{R}$, then that polynomial p has degree n, where n is the highest power of variable x.
- For two vectors u and v, their dot product $u \bullet v$ multiplies each element of u with the corresponding element of v and sums them together. If $u \bullet v = 0$, then the vectors are orthogonal. Orthogonality can be thought of as a specific version of perpendicularity.
- When generalized to polynomials, the dot product becomes an inner product space, $\langle p, q\rangle$, for two polynomials p and q, with an orthogonal structure. An inner product space must satisfy three specific qualities:
  1. $\langle p, q\rangle$ is linear in each component. If either polynomial is multiplied by a real-valued constant a or added to another polynomial r, they can be moved outside for simplicity. Formally,$\langle a \times p, q\rangle = a \times \langle p, q\rangle$ and $\langle p, q + r\rangle = \langle p, q\rangle + \langle p, r\rangle$.
  2. $\langle p, q\rangle$ is commutative. So order does not matter and $\langle p, q\rangle = \langle q, p\rangle$.
  3. $\langle p, p\rangle$ is positive-definite, so $\langle p, p\rangle \geq 0$ must always be true. An inner product must "vanish" (i.e., if $\langle p, p\rangle = 0$) if and only if the polynomial p is itself 0.
- An orthogonal polynomial is a polynomial "perpendicular" to all other lower-degree polynomials. For a n^{th}-degree orthogonal polynomial p_n, $\langle p_n, q_i\rangle = 0$ for any other polynomial q_i of degree $i < n$. If a set of polynomials are mutually orthogonal, then for the same p_n, $\langle p_n, q_i\rangle = 0$ for any other polynomial q_i of degree $i \leq n$.

- An abstract space $\mathbb{S}$ is a set of elements with some added structure.
- In this paper, we call our restricting space an orthogonal tripod τ, which lays on the boundary of a cube. A non-orthogonal tripod is the intersection of three lines. Think of the bottom half of a camera tripod, or the right-hand rule in physics.
- The cardinality of a set $|\mathbb{S}|$ is the set's finite or infinite size.
- A basis $\mathbb{B}$ is a set of the defining members for a space. From a basis, you can use a linear combination of the members to obtain anything that exists in the space.
- The dimension of a space is the cardinality of a set's basis.
- An orthogonal basis contains mutually orthogonal elements.
- A linear transformation $\mathbb{S} \rightarrow \mathbb{T}$ is a spatial shift that maintains all linear and structural properties. A specialized transformation, affine, is explored in abstract algebra.
- An integral $\int_a^b p(x)dx$ can sometimes be interpreted as the area under a non-negative polynomial p in terms of a variable x, from point a and point b, two real-valued constants in $\mathbb{R}$. Think of the statistics probability curves for continuous variables.
- An approximation of a function often means to use an easily-manageable function to approach a more complicate function, with some small error term ϵ. More formally, if a function f is approximated by a polynomial p, then $f \simeq p$ and $f = p + \epsilon$. Ideally, these approximations will be accurate, which requires minimizing the error term ϵ.
- An algorithm is a set of steps that produce a result. We can think of this as a grouped sequence of functions or a "black box" that takes input x and returns output y.
- Efficiency is a reflection of a computation's speed or the amount of "memory" it requires. In the field of computer science, an algorithm's efficiency is described using Big O notation, which looks at the algorithm's scaled timing over time in terms of the n variables involved. We call $O(1)$ constant time, $O(n)$ linear time, and $O(n^2)$ n-squared time. Algorithms with faster-growing Big O notations are less efficient.

Appendix B — Proofs

Theorem 2.1: *The domain that restricts our inner product*

$$\langle f, g\rangle = \int_0^1 f(x,0,0)g(x,0,0)dx + \int_0^1 f(0,y,0)g(0,y,0)dy + \int_0^1 f(0,0,z)g(0,0,z)dz$$

such that it becomes a proper inner product space is $\mathbb{R}_n[x,y,z] \setminus \langle xy, xz, yz\rangle$.

Proof. As is, $\mathbb{R}_n[x,y,z]$ is not a valid domain for $\langle f, g\rangle$ to be considered an inner product space. We must remove any polynomials p that cause this $\langle p, p\rangle$ to equal zero when p itself is not zero. Given its definition, we can see that $\langle p, p\rangle$ will be calculated in three parts, isolating each variable, with the other variables set to zero. Therefore, if any polynomial is exclusively of the form $xyz \times q$ for any other polynomial q, then clearly this polynomial always equals zero. We can break this down further, however, into the multiplication of any two variables. The three different permutations of x, y, and z are xy, xz, and yz. Therefore the space of each of these polynomial ideals must be removed the original domain, and a domain of $\mathbb{R}_n[x,y,z] \setminus \langle xy, xz, yz\rangle$ makes $\langle f, g\rangle$ a valid inner product space. □

Theorem 2.2: *Our inner product space's restrictive domain has the following dimensions:*

$$\textit{When } n = 0, dim(\mathbb{R}_n[x,y,z] \setminus \langle xy, xz, yz\rangle) = 1.$$
$$\forall n \in \mathbb{N} : n > 0, dim(\mathbb{R}_n[x,y,z] \setminus \langle xy, xz, yz\rangle) = 3.$$

Proof. Given our inner product space's restricted domain, we arrive at the following bases:

$$\begin{aligned}
n = 0 : &\quad 1 \\
n = 1 : &\quad x \quad y \quad z \\
n = 2 : &\quad x^2 \quad y^2 \quad z^2 \\
n = 3 : &\quad x^3 \quad y^3 \quad z^3.
\end{aligned}$$

These are the only allowable variable combinations that avoid the three constraining ideals. Any other variable combination would cause the inner product space to equal zero when using a non-zero polynomial, which would violate its definition as an inner product space.

Therefore, since we can only have an isolated variable as each element in our basis and we are only working with three variables, then the dimension of each higher degree is also three. This comes naturally when we see that our bases are defined by $\{x^n, y^n, z^n\}$. □

Theorem 6.1: *If our algorithm implements three-term recursion, then for* $i = 0, 1, \ldots n-3$*, each monic polynomial's coefficients* $a_i = b_i = c_i = 0$*.*

Proof. Currently, we have $\langle P_n, P_j\rangle = \langle xP_{n-1}, P_j\rangle + a_j\langle P_j, P_j\rangle + b_i\langle Q_j, P_j\rangle + c_i\langle R_j, P_j\rangle$ and we want this to equal zero. So, set $\langle xP_{n-1}, P_j\rangle + a_j\langle P_j, P_j\rangle + b_i\langle Q_j, P_j\rangle + c_i\langle R_j, P_j\rangle = 0$ and let us continue making manipulations until this is in fact true.

First, we can consider the first term on the right and know that this equals zero because, using an inner product property, we can move the x to the right side of the inner product without changing anything. Then, we are guaranteed that for any $j \neq n-1 \neq n-2$, the degree of P_j is less than the degree of P_{n-1}. This is crucial because due to the definition of Ps orthogonality, we know that $P_n - 1$ is orthogonal to any polynomial of a lesser degree. Therefore, if the two polynomials are orthogonal, then their inner product equals zero.

Now we are left with the three remaining terms. If we combine these inner products into a single inner product, we can arrive at the following three equations:

$$\langle a_iP_j + b_iQ_i + c_iR_i, P_j\rangle = 0$$

$$\langle a_iP_j + b_iQ_i + c_iR_i, Q_j\rangle = 0$$

$$\langle a_iP_j + b_iQ_i + c_iR_i, R_j\rangle = 0.$$

Using a clever trick, we can multiply each of these three equations by their respective a_i, b_i, and c_i to create a single inner product of a weighted sum of each line: $\langle a_iP_j + b_iQ_i + c_iR_i, a_iP_j + b_iQ_i + c_iR_i\rangle = 0$. Given the definition of an inner product, we know that if the identity inner product equals zero, then the involved equation must also be zero. Thus, the remaining three components of the original equation that we started this proof with also equal zero, and so $\langle P_n, P_j\rangle = 0$ is in fact true. □

Bibliography

[1] Stephen Abbott. *Understanding Analysis*. Ed. by Sheldon Axler and Kenneth Ribet. 2nd ed. Springer, 2016.

[2] Ward Cheny and David Kincaid. *Numerical Mathematics and Computing*. 6th ed. Thomson Brooks/Cole, 2008.

[3] Charles K. Dunkl and Yuan Xu. *Orthogonal Polynomials of Several Variables*. Cambridge University Press, 2001.

[4] David C. Lay. *Linear Algebra and Its Applications*. Ed. by Steven R. Lay and Judi J. McDonald. 5th ed. Pearson, 2016.

[5] Sheehan Olver and Yuan Xu. *Orthogonal Structure on a Quadratic Curve*. July 2018. URL: arxiv.org/pdf/1807.04195.pdf.

[6] Sheehan Olver and Yuan Xu. *Orthogonal Structure on a Wedge and on the Boundary of a Square*. Oct. 2017. URL: arxiv.org/abs/1710.08654.

[7] Davis J. Philip. *Interpolation and Approximation*. Dover ed. Blaisdell Publishing, 1975.

[8] James Stewart. *Multivariable Calculus*. Ed. by Liz Covello et al. 7th ed. Brooks/Cole, 2012.

[9] Gabor Szegö. *Orthogonal Polynomials*. Vol. XXIII. Colloquium Publications and American Mathematical Society Providence, 1939.

[10] Yuan Xu. "Orthogonal Polynomials of Several Variables". In: *Multivariable Special Functions*. Vol. 2. Askey-Bateman Project, 2017.