18.338 Project- Methods for Computing Multivariate Orthogonal Bases: A Short Survey

Cecilia Chen

December 16, 2024

1 Introduction

Orthogonal polynomials are pivotal in a variety of applications. For example, the Chebyshev polynomials appear seemingly everywhere, including applications in polynomial interpolation [10], approximation of psuedospectra [9], distributed signal processing [8], and graph neural networks [2]. A natural extension from the study of orthogonal polynomials is to multivariate orthogonal polynomials, where generalizing to several variables can better express different systems. For example, consider the case of orthogonal polynomials in image processing, where multiple variables are used to represent different colors in [7].

Since representing data in an orthogonal polynomial basis is a powerful tool for numerical methods and approximation, the question we ask is how to compute these bases. In the univariate case, the answer has been already established for many years. The Lanczos method [5] is a highly popular algorithm for efficiently approximating a variety of linear algebra problems such as eigenvalue estimation and least squares. However, in the multivariate case, the answer is no longer as clear due to increased computational complexity, greater freedom for choosing arbitrary canonical bases, and different advantages of current known algorithms. Furthermore, a lack of both unified and explicit notation for cases with more than two variables makes it unclear how different methods compare to each other.

The goal of this project is to overview current methods for computing multivariate orthogonal polynomial bases and express them in a directly comparable manner. In Section 2, we review notation and well-known results in orthogonal polynomial theory. In Section 3, we go over three algorithms for computing multivariate orthogonal polynomial bases. In addition, I implemented Julia code for the method of Huhtanen and Larson which can be found at `https://github.com/Cece2520/MultivariateLanczos.jl`.

2 Background and Notation

Let $\mathcal{P}_n^1$ be the span of all univariate polynomials of degree at most n. Let $S \subseteq \mathbb{C}$ be some compact support set and $w(x)$ a weight function over S. Then, we say polynomials $p_0 \in \mathcal{P}_0^1, ..., p_n \in \mathcal{P}_n^1$ form an orthonormal basis with respect to w if

$$\langle p_i, p_j \rangle_w := \int_S p_i(x) p_j(x) w(x) dx = \delta_{ij}$$

where δ_{ij} is the Kronecker delta function and $p_0, ..., p_n$ span $\mathcal{P}_n^1$.

In the multivariate case, it is useful to define the monomials indexed by their powers. For some multi-index $\alpha = (\alpha_1, ..., \alpha_d)$ and variables $x_1, ..., x_d$, let $x^\alpha = \prod_{i=1}^n x_i^{k_i}$. Then we say x^k has degree n if $|\alpha| = \alpha_1 + ... + \alpha_d = n$ and $\mathcal{P}_n^d$ is the space of all d-variable polynomials of degree at most n. To count the polynomials of each degree, let $R_n^d = \dim \mathcal{P}_n^d = \binom{n+d}{n}$ and $r_n^d = \dim (\mathcal{P}_n^k \setminus \mathcal{P}_{n-1}^d) = \binom{n+d-1}{n}$. Let $S \subseteq \mathbb{C}^d$ be our support set and $w(x)$ a weight function over S. Then $p_1, ..., p_{R_n^d}$ form an orthonormal basis with respect to w if the same condition holds as the univariate case.

In the univariate case, it is well-known that orthogonal polynomials can be expressed by a tridiagonal recurrence. This fact can be proven very cleanly by the following proposition

Proposition 2.1. *Given orthogonal polynomials $p_1, ..., p_n$ such that $p_i \in \mathcal{P}_i \setminus \mathcal{P}_{i-1}$ for all $i \in [n]$, we have*

$$x p_i(x) = \alpha_{i+1} p_{i+1}(x) + \beta_i p_i(x) + \alpha_i p_{i-1}(x)$$

for constants $\alpha_1, ..., \alpha_n, \beta_1, ..., \beta_n$ and $p_{-1}(x) = p_{n+1}(x) = 0$.

Proof. To show that each recurrence contains at most three terms, we see for any $i < j-1$, $x p_i(x)$ has degree less than j. Hence, $\langle p_j(x), x p_i(x) \rangle_w = 0$. By the simple fact that $\langle x p_j(x), p_i(x) \rangle_w = \langle p_j(x), x p_i(x) \rangle_w$,

we also have $\langle xp_j(x), p_i(x)\rangle_w = 0$ for $j < i-1$. The symmetry is also apparent from the same fact, as $\alpha_i = \langle xp_i(x), p_{i-1}(x)\rangle$ from the recurrence of $xp_i(x)$ is the same as $\alpha_i = \langle xp_{i-1}(x), p_i(x)\rangle$ from the recurrence of $xp_{i-1}(x)$. □

A multivariate analogue also exists as follows

Proposition 2.2. *[1, Theorem 3.3.2] Take some orthonormal polynomial basis $p_1, ..., p_{R_n^d}$ in d variables. Let $\mathbb{P}_i = [p_{R_{i-1}^d+1}, ..., p_{R_i^d}]^T$ for $i \in [n]$ such that $\text{span}(\mathbb{P}_i) \subseteq \mathcal{P}_i^d$. Then for all $i \in [n]$ and $j \in [d]$,*

$$x_j\mathbb{P}_i = A_{i,j}\mathbb{P}_{i+1} + B_{i,j}\mathbb{P}_i + A_{i-1,j}^T\mathbb{P}_{i-1}$$

for matrices $A_{i,j} : r_i^d \times r_{i+1}^d$ and $B_{i,j} : r_i^d \times r_i^d$.

Proof. The same flavor of proof holds as in Proposition 2.1 where for degree $i < j-1$ and any $k \in [d]$, $x_k\text{span}(\mathbb{P}_i)$ has degree less than j and must be orthogonal to all of $\mathbb{P}_j$, and symmetry follows by comparing difference recurrences. □

From Proposition 2.2, it should now be clear how having multiple variables introduces significant underdetermination in how to express these recurrences. Each subspace $\text{span}(\mathbb{P}_i)$ for $i \in [n]$ has $r_i^d - 1$ degrees of freedom and arbitrary decisions on the representation of each subspace produces very different recurrences. The goal of the algorithms given in the next section, will be to compute some recurrence matrices A, B and their associated basis over a finite support set.

3 Methods for Computing Multivariate Orthogonal Bases

In this section, we overview three algorithms for computing multivariate orthogonal bases. The first is a Lanczos-like algorithm by Huhtanen and Larsen [4], followed by a pivoting algorithm by Van Barel and Chesnokov [11] with the advantage of iteratively introducing new support points. Finally, we cover a more recent Stieljes-type algorithm by Liu and Narayan [6], which attempts to circumvent ill-conditioning from naive Gram-Schmidt orthogonalization but suffers significant computation complexity for numbers of variables above three.

The first two algorithms as published are defined explicitly only for the bivariate case because they are simple to extend beyond two variables. However the lack of explicit notation and differences in the algorithm output make it unclear how they compare. For each algorithm, we convert output to the A, B notation described in the Section 2.

3.1 Huhtanen and Larsen

The earliest method for multivariate orthogonal bases, this algorithm [4] from 2002 was inspired by previous work by one of its authors for producing recurrences from nonnormal input matrices [3] (in contrast to the Lanczos method, which requires a Hermitian matrix as input). For variables $x_1, ..., x_d$, the support set is represented by a set of simultaneously diagonalizable matrices $M_1, ..., M_d \in \mathbb{C}^{n\times n}$. For each point $s_i = (s_i^{(1)}, ..., s_i^{(d)}) \in S$ where the value $s_i^{(j)}$ is associated with variable x_j, we take some eigenvector ρ_i and let $\lambda_i^{(j)}$ the corresponding eigenvalue of M_j equal $s_i^{(j)}$ for all j. Finally, a weight vector $b \in \mathbb{C}^n$ defines the discrete weight ${\rho_i}^T b$ to s_i for all i.

We can write all of the definitions above succinctly from some unitary basis U such that for all i

$$M_1 = U\begin{bmatrix} s_1^{(1)} & & & \\ & s_2^{(1)} & & \\ & & \ddots & \\ & & & s_n^{(1)} \end{bmatrix} U^*, \qquad M_2 = U\begin{bmatrix} s_1^{(2)} & & & \\ & s_2^{(2)} & & \\ & & \ddots & \\ & & & s_n^{(2)} \end{bmatrix} U^*, \qquad \dots$$

From this definition, it is clear that taking multivariate polynomials over the matrices $M_1, ..., M_d$ is equivalent to evaluating the polynomial at each support point (with some unitary change of basis). What is left is to determine how to iteratively generate higher degree polynomials by successively multiplying the input M matrices.

Huhtanen and Larsen define an ordering on the process for building monomials by drawing a pyramid where each level i contains all monomials of degree i.

1

$M_1 \quad \dots \quad M_d$

$$M_1^2 \quad M_1M_2 \quad M_2^2 \quad M_1M_3 \quad M_2M_3 \quad M_3^2 \quad \dots \quad M_d^2$$
$$\vdots$$

We avoid redundancy while generating each level by "activating" variables one at a time. Suppose we have finished level i and are just starting to build level $i+1$. First, we activate x_1, and multiply M_1 by against all monomials in level i containing only active variables. In this case, only M_1^i is valid in level i, giving us M_1^{i+1}. Then, we activate x_2, and obtain $M_1^i \to M_1^iM_2$, $M_1^{i-1}M_2 \to M_1^{i-1}M_2^2$, and so on. We continue in this manner until all variables are activated, and we have produced all monomials of degree $i+1$ is an ordered (in this case, graded lexicographic) manner.

Combining this ordering scheme with Gram-Schmidt orthogonalization, similar to the univariate Lanczos method, gives us the algorithm sketch below, where $(A)_{k,\ell}$ refers to the k,ℓ entry of matrix A. We also use e_j to denote the jth elementary basis vector and use $k+e_j$ to denote incrementing the j-th entry of a multi-index k by 1. Finally, we associate each multi-index k with an integer determined by graded lexicographic order and use k interchangeably to mean the integer and multi-index.

Algorithm 1 Huhtanen and Larsen

Input: $M_1, ..., M_d \in \mathbb{C}^{n\times n}$ simultaneously diagonalizable, $b \in \mathbb{C}^n$, and $m \in \mathbb{N}$ the maximum degree desired.
$p_{(0,\dots,0)} = b/\|b\|$
for $i = 0 : m-1$ **do**
  for $j = 1 : d$ **do**
    for all multi-indices k of degree i and nonzeros only in the first j entries **do**
      $p_{k+e_j} = M_j p_k$
      for all multi-indices ℓ of degree $i-1$ **do**
        $(A_{i-1,j})_{k+e_j,\ell} = \langle p_{k+e_j}, p_\ell \rangle$
        $p_{k+e_j} = p_{k+e_j} - (A_{i-1,j})_{k+e_j,\ell} p_\ell$
      end for
      for all multi-indices ℓ of degree i **do**
        $(B_{i,j})_{k+e_j,\ell} = \langle p_{k+e_j}, p_\ell \rangle$
        $p_{k+e_j} = p_{k+e_j} - (B_{i,j})_{k+e_j,\ell} p_\ell$
      end for
      for all previously computed multi-indices ℓ of degree $i+1$ **do**
        $(A_{i,j})_{k+e_j,\ell} = \langle p_{k+e_j}, p_\ell \rangle$
        $p_{k+e_j} = p_{k+e_j} - (A_{i,j})_{k+e_j,\ell} p_\ell$
      end for
      $(B_{i,j})_{k+e_j,k+e_j} = \|p_{k+e_j}\|$
      $p_{k+e_j} = p_{k+e_j}/\|p_{k+e_j}\|$
    end for
  end for
end for

Note that this algorithms gains its efficiency from two main reasons. First, due to the multiplication scheme, redundant recurrences are skipped. For example, in a bivariate case, xy is generated by multiplying $x * p_{(0,1)}(x,y)$ while skipping $y * p_{(1,0)}(x,y)$. This results in an incomplete recurrence, assuming that the polynomial will be obtained another way. The second reason is due to a bounded recurrence growth factor, which we will address in the following theorem.

Theorem 3.1. *[4, Theorem 2.3] Let N denote the total number of orthogonal polynomials generated by Algorithm 1. For any single orthogonal polynomial p of degree n, its computed recurrence contains at most $O(N^{1-1/d})$ terms.*

Proof. This fact is readily apparent from the size of each recurrence matrix. The width of each matrix $A_{n,j}, B_{n,j}, B_{n-1,j}$ determines the maximum number of terms in the recurrence. We know this to be $2 * r_n^d + r_{n-1}^d$ as well as $N \geq R_n^d$, which gives us $\frac{2*r_n^d + r_{n-1}^d}{R_n^d} = O(1/n)$ and we can manipulate variables from here. □

From Theorem 3.1, we see that by leveraging the structure of polynomial degree in the order of orthogonalization, we can reduce the total number of orthogonalizations compared to a naive Gram-Schmidt orthogonalization scheme, which would result in $O(N)$ terms. While the savings diminish as the number of variables increases, the Gram matrix remains polynomially sparser.

3.2 Van Barel and Chesnokov

The next algorithm we will discuss is from 2010, which improves on the previous method by allowing support points to be added iteratively instead of fixing the support set from the start [11]. This method relies on pivoting using Givens rotations to generate recurrence matrices in generalized Hessenberg form.

This method takes as input support points $s^{(1)}, ..., s^{(n)} \in \mathbb{C}^d$ over d variables $x_1, ..., x_d$ and a weight vector $w \in \mathbb{C}^n$. It produces an orthonormal basis Φ and a set of matrices $H_1, ..., H_d$ such that $M_i\Phi = \Phi H_i$ for all $i \in [d]$.

In order to extract the A and B recurrence matrices from the H_is, it is necessary to know which rows correspond to which degree of polynomial. Van Barel and Chesnokov use the same ordering scheme as the previous method, namely graded lexicographic ordering, which means they produce the same recurrence matrices except with all rows completed. While Huhtanen and Larsen enforce one specific order to evaluate the polynomials, the method by Van Barel and Chesnokov allows for multiple evaluation orders at the cost of extra storage.

The general idea of Van Barel and Chesnokov's algorithm is to concatenate the matrices $\left[w|S_1|...|S_d\right]$ where $S_i = \text{diag}(s_i^{(1)}, ..., s_i^{(n)})$ for all i. Then, transform

$$\left[\|w\|e_1|H_1|...|H_d\right] = Q^* \left[w|S_1|...|S_d\right] \begin{bmatrix} I & & \\ & Q & \\ & & Q \end{bmatrix}$$

by successive Givens rotations at predetermined pivots. We refer the reader to [11] for the details of the pivots as there is no simple description of the pivot locations even for the bivariate case other than by construction.

3.3 Liu and Narayan

This last algorithm was a much more recent development from 2023, which aims to generate an orthogonal basis with an oracle for generalized polynomial moments ($\int_S p(x)w(x)dx$ for any polynomial p) [6]. In contrast to the previous methods, this algorithm allows for infinite sized support sets. However, methods to efficiently compute generalized polynomial moments for the infinite case in general are not known, so for practical consideration at the time, the benefits are less noticeable.

The main idea for this Stieltjes-like method, is to compute the recurrence matrices directly. The following definitions and theorems characterize necessary properties of the recurrence matrices. (Recall that $\mathbb{P}_i$ are matrices that span all the monomials of degree i).

Definition 3.2. *[6, Definition 3.1] Given an orthonormal basis $\mathbb{P}_1, ..., \mathbb{P}_n$ in d variables with recurrence matrices $A_{i,j}, B_{i,j}$ for $i \in [n]$, $j \in [d]$, the basis is in* canonical form *if for every i,*

$$\sum_{j=1}^{d} B_{i,j}^T B_{i,j} = \Lambda_i$$

such that Λ_i is diagonal and elements are nondecreasing down the diagonal.

Theorem 3.3. *[6, Section 3.2, Theorem 3.3] Given matrices $A_{i,j}, B_{i,j}$, there exists a basis $\{\mathbb{P}_n\}$ in canonical form which is orthonormal with respective some positive-definite bilinear form if and only if the following hold:*

- *$A_{i,j}$ is symmetric for all i and $j \in [d]$.*
- *$B_{i,j}$ is full rank for al i and $j \in [d]$.*
- *The following commuting conditions hold For every i and $j, k \in [d]$:*
  - $B_{i+1,j}^T B_{i+1,k} + A_{i+1,j}^T A_{i+1,k} + B_{i,j}^T B_{i,k} = B_{i+1,k}^T B_{i+1,j} + B_{i+1,k}^T B_{i+1,j} + B_{i+1,k}^T B_{i+1,j}$
  - $B_{i,j} A_{i+1,k} + A_{i,j} B_{i,k} = B_{i,k} A_{i+1,j} + A_{i,k} B_{i,j}$
  - $B_{i,j} B_{i+1,k} = B_{i,k} B_{i+1,j}$.

Then, the algorithm proceeds inductively by increasing the maximum degree in the basis and looking for new recurrence matrices which satisfy the constraints of Theorem 3.3. Supposing the basis has already been computed up to degree i, we can compute specific moment matrices which satisfy the first two conditions, requiring an SVD and reshuffling of eigenvectors to achieve canonical form. The main computational bottleneck for this algorithm comes from satisfying the commuting conditions. Using some

clever tricks, the author were able to simplify the computation for $d = 2, 3$, but for $d > 3$, we are forced to solve a nonconvex optimization problem. Again, we refer to [6] for the exact algebraic manipulations.

In comparison to a naive Gram-Schmidt orthogonalization scheme, Liu and Narayan were able to show that their algorithm performs significantly better in terms of numerical stability. However, it is unclear how their algorithm compares to the Huhtanen-Larsen and Van Barel-Chesnokov methods. In particular, the Huhtanen-Larsen method also contains a Gram-Schmidt orthogonalization component, but the basis is built iteratively and computed against previously orthonormalized vectors. As a result, I would expect the numerical stability of the Huhtanen-Larsen to be far superior to the naive Gram-Schmidt, similar to the case for the univariate Lanczos method. Unfortunately, I did not have the time to implement this experiment numerically.

References

[1] Charles F. Dunkl and Yuan Xu. *General Properties of Orthogonal Polynomials in Several Variables*, page 57–113. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2014.

[2] Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 7264–7276. Curran Associates, Inc., 2022.

[3] Marko Huhtanen. Orthogonal polyanalytic polynomials and normal matrices. *Mathematics of Computation*, 72(241):355–373, 2003.

[4] Marko Huhtanen and Rasmus Munk Larsen. On generating discrete orthogonal bivariate polynomials. *BIT Numerical Mathematics*, 42(2):393–407, 2002.

[5] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.

[6] Zexin Liu and Akil Narayan. A stieltjes algorithm for generating multivariate orthogonal polynomials. *SIAM Journal on Scientific Computing*, 45(3):A1125–A1147, 2023.

[7] Zhe Liu, Yu-Qing Song, Jian-Mei Chen, Cong-Hua Xie, and Feng Zhu. Color image segmentation using nonparametric mixture models with multivariate orthogonal polynomials. *Neural Computing and Applications*, 21(4):801–811, 2012.

[8] David I Shuman, Pierre Vandergheynst, and Pascal Frossard. Chebyshev polynomial approximation for distributed signal processing. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pages 1–8, 2011.

[9] Kim-Chuan Toh and Lloyd N. Trefethen. The chebyshev polynomials of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 20(2):400–419, 1998.

[10] Lloyd N. Trefethen. *Chapter 2. Chebyshev Points and Interpolants*, pages 7–12.

[11] Marc Van Barel and Andrey Chesnokov. A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations. *Numerical Algorithms*, 55(2):383–402, 2010.