Orthogonal Structure on a Tripod

Sierra Nicole Battan

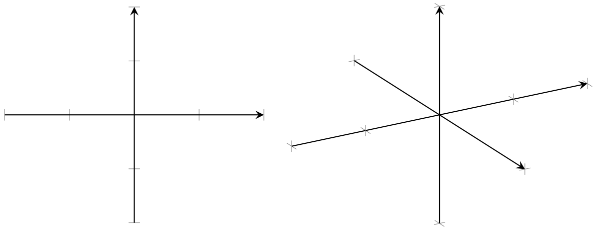
Robert D. Clark Honors College
Department of Mathematics
University of Oregon

4 March 2019

What Is Going On?

- Intersection of multivariable calculus, linear algebra, and numerical mathematics.
- Extension of the forefront of numerical research that defines a structure with which mathematicians can manipulate monster polynomials.
- Presentation of four increasingly efficient algorithms that construct an orthogonal basis that accurately represents the structure of polynomials on any 3D tripod.

What Is Orthogonality?



- Orthogonality between two functions: $\langle f, g \rangle = 0$.
  - Orthogonal means 90° and $\cos(90) = 0$.
  - Euclidean: $\cos(\theta) = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{||\mathbf{x}||\ ||\mathbf{y}||}$.
- Higher dimensions use sums of functions $f = \sum a_n P_n(x)$.
- Inner product $\langle f, g \rangle$ must be commutative, linear in each component, and positive definite.

What Are Iterative And Recursive Algorithms?

- Solving a larger problem in a series of smaller steps that may or may not depend on the previous step.
- The solution to our current execution of the algorithm(s) utilizes results from the execution of previous algorithm(s).
- Specifically, three-term recursion:
  $F_n = F_{n-1} + F_{n-2}$ with initial values $F_0 = 0$ and $F_1 = 1$.
- Minimizes dependencies and redundant computations
  $\Longrightarrow$ improves algorithmic efficiency.

Research Goals & Applications

- Exploit orthogonality, iteration, recursion, and computerized mathematics to improve algorithmic efficiency, which is crucial in the world of "big data."
- Approximate polynomials on a 3D tripod by using orthogonal infinite sums.
- Fourier orthogonal series and signal readings:

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

- Expand the current study of all orthogonal structures.

Desired Inner Product Space

$$
\begin{aligned}
\langle f, g\rangle = &\int_0^1 f(x,0,0)g(x,0,0)dx \\
&+ \int_0^1 f(0,y,0)g(0,y,0)dy \\
&+ \int_0^1 f(0,0,z)g(0,0,z)dz
\end{aligned}
$$

Commutative, linear in each component, and positive definite: $\langle f, f\rangle \geq 0$ where $\langle f, f\rangle = 0$ if and only if f itself equals 0.

Domain Restriction With Polynomial Ideals

General $\mathbb{R}_n[x, y, z]$ bases:

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

Polynomial Ideals: $f \in \langle p_1, \ldots, p_n \rangle \Leftrightarrow \exists q_i : f = \sum_{i=1}^{n} \{p_i \times q_i\}$.

Our inner product space cannot contain $\langle xy, xz, yz \rangle$.

Inner Product Space's Domain Dimension

Restricted $\mathbb{R}[x,y,z]/\langle xy,xz,yz\rangle$ bases:

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

Theorem

Our inner product space's restrictive domain has dimension:

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

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

Naïve Algorithm

$V_n = \mathbb{R}_n[x,y,z]/\langle xy, xz, yz\rangle$ is the n^{th}-degree inner product space of the restricted tripod.

Monic: $P_n = x^n + \langle$ lower-degree combination of$x, y, z\rangle$.

Construct the monic polynomial basis $B_n = \{P_n, Q_n, R_n\}$:

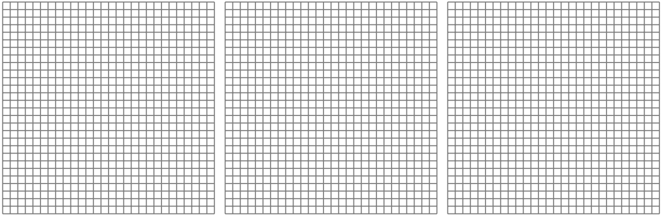
$$P_n(x,y,z) = x^n + \sum_{i=1}^{n-1} a_i^P x^i + \sum_{i=1}^{n-1} b_i^P y^i + \sum_{i=1}^{n-1} c_i^P z^i + d_0^P.$$

Since each basic element is monic, we can determine them by $\langle P_n, x^j\rangle = \langle P_n, y^j\rangle = \langle P_n, z^j\rangle = 0$.

$\ldots$

Nav̈e Algorithm Complexity

Three $(3n-2)\times(3n-2)$ matrices:



.

Given $B_n = \{P_n, Q_n, R_n\}$ we can use elementary linear algebra techniques to construct $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$.

Basic Algorithm

Introduce iteration to similarly construct $B_n = \{P_n, Q_n, R_n\}$:

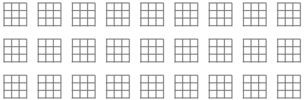
$$P_n(x,y,z) = x^n + \sum_{i=1}^{n-1} a_i^P P_i(x,y,z) + \sum_{i=1}^{n-1} b_i^P Q_i(x,y,z) + \sum_{i=1}^{n-1} c_i^P R_i(x,y,z) + d_0^P.$$

Determine each element for fixed $j = 1, 2, \ldots, n-1$:

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

Basic Algorithm Complexity

$3(n-1)\ \ 3 \times 3$ matrices:


.

Given $B_n = \{P_n, Q_n, R_n\}$ we can use Gram-Schmidt to construct $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$.

Clever Algorithm

Introduce recursion to similarly construct $B_n = \{P_n, Q_n, R_n\}$ with base cases $B_0 = \{1\}$ and $B_1 = \{x - \frac{1}{6}, y - \frac{1}{6}, z - \frac{1}{6}\}$:

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

Directly compute each element after coefficient determination:

$$P_n(x,y,z) = (x - a_{n-1}^P)P_{n-1}(x,y,z) + a_{n-2}^P P_{n-2}(x,y,z) + b_{n-1}^P Q_{n-1}(x,y,z) + b_{n-2}^P Q_{n-2}(x,y,z) + c_{n-1}^P R_{n-1}(x,y,z) + c_{n-2}^P R_{n-2}(x,y,z).$$

Clever Algorithm Complexity

Six 3×3 matrices:  .

Given $B_n = \{P_n, Q_n, R_n\}$ we can use Gram-Schmidt to construct $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$.

Brilliant Algorithm

Mutually-Orthogonal: $P_n = \langle$any linear combination of $x, y, z\rangle$.

Orthonormal: mutually-orthogonal polynomials $\langle P_n, P_n\rangle = 1$.

Again construct $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$ with base cases $OB_0 = \{1\}$ and $OB_1 = \{x - \frac{1}{6}, y - \frac{1}{6}, z - \frac{1}{6}\}$:

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

Determine each element directly, but with more simplifications!

Brilliant Algorithm

Given mutual-orthogonality, we obtain diagonal matrices:

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

Given orthonormality, we obtain direct definitions:

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

Thus, the computation of each element is trivial:

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

Brilliant Algorithm Complexity

Zero matrices.

Gram-Schmidt immediately constructs $OB_n = \{\hat{P}_n, \hat{Q}_n, \hat{R}_n\}$.

Research Recap

- Establish an orthogonal basis of our inner product space on $\mathbb{R}[x,y,z]/\langle xy, xz, yz\rangle$ that accurately represents the structure of [monster] polynomials on any 3D tripod.
- Naïve: arbitrary monic polynomials.
  - Three $(3n-2)\times(3n-2)$ matrices.
- Basic: iterative monic polynomials.
  - $3(n-1)$ 3×3 matrices.
- Clever: recursive monic polynomials.
  - Six 3×3 matrices.
- Brilliant: orthonormal polynomials.
  - Zero matrices.

Questions?