Perfect State Transfer in Cycle Graphs ♡

Elizabeth Athaide, Leia Donaway, Sam Trombone

University of Connecticut REU 2024

November 25, 2024

Outline

What we'll show you:

Outline

What we'll show you:

- Krawtchouk paths proof

Outline

What we'll show you:

- Krawtchouk paths proof
- Splitting paths into cycles

Outline

What we'll show you:

- Krawtchouk paths proof
- Splitting paths into cycles
- Our work with orthogonal polynomials thus far

Outline

What we'll show you:

- Krawtchouk paths proof
- Splitting paths into cycles
- Our work with orthogonal polynomials thus far
- Plans for future work

Defining K_n

The Krawtchouk polynomials are defined by

$$K_n(x, N) = \sum_{k=0}^{n} \frac{(-x)_k(-n)_k}{(-N)_k} \frac{2^k}{k!}$$

Defining K_n

The Krawtchouk polynomials are defined by

$$K_n(x, N) = \sum_{k=0}^{n} \frac{(-x)_k(-n)_k}{(-N)_k} \frac{2^k}{k!}$$

We'll define a Krawtchouck-style matrix to be the matrix corresponding to

$$K_n = \frac{\sqrt{n(N+1-n)}}{2}.$$

Defining K_n

The Krawtchouk polynomials are defined by

$$K_n(x, N) = \sum_{k=0}^{n} \frac{(-x)_k(-n)_k}{(-N)_k} \frac{2^k}{k!}$$

We'll define a Krawtchouck-style matrix to be the matrix corresponding to

$$K_n = \frac{\sqrt{n(N+1-n)}}{2}.$$

The underlying Jacobi matrix is mirror symmetric, corresponding to a path weighted by Krawtchouk polynomials, with the first node being node 0 and the last being node N.

Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ *achieves PST from node* 0 *to node* N *at time* $t = \pi$.

Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

To prove, recall the theorem for conditions sufficient for PST on paths shown by Maksym in lecture.

Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

To prove, recall the theorem for conditions sufficient for PST on paths shown by Maksym in lecture.

Theorem

Let $\mathcal{J}$ be a Jacobi matrix of order N. Then there exists some $t > 0$ and φ such that

$$e^{itJ}e_1 = \varphi e_N \iff$$

1. $\mathcal{J}$ *is mirror symmetric, and*
2. $\lambda_k - \lambda_{k+1} = \frac{(2n_k+1)\pi}{t}, n_k \in \mathbb{Z}$

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

As K_N is mirror symmetric by construction, we want to show $\lambda_k - \lambda_{k+1} = -1$.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

As K_N is mirror symmetric by construction, we want to show $\lambda_k - \lambda_{k+1} = -1$. Let $\lambda_0, \lambda_1, \ldots, \lambda_N$ be eigenvalues of $\mathcal{K}_N$ such that $\lambda_0 < \lambda_1 < \cdots < \lambda_N$.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

As K_N is mirror symmetric by construction, we want to show $\lambda_k - \lambda_{k+1} = -1$. Let $\lambda_0, \lambda_1, \ldots, \lambda_N$ be eigenvalues of $\mathcal{K}_N$ such that $\lambda_0 < \lambda_1 < \cdots < \lambda_N$. This is a strict inequality because our Jacobi matrix has simple spectrum.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

As K_N is mirror symmetric by construction, we want to show $\lambda_k - \lambda_{k+1} = -1$. Let $\lambda_0, \lambda_1, \ldots, \lambda_N$ be eigenvalues of $\mathcal{K}_N$ such that $\lambda_0 < \lambda_1 < \cdots < \lambda_N$. This is a strict inequality because our Jacobi matrix has simple spectrum.

Definition

Let $\mathcal{K}'_N$ be defined as corresponding to the following recurrence relation.

$$-xK_n(x,N) = \frac{N-n}{2}K_{n+1}(x,N) - \frac{N}{2}K_n(x,N) + \frac{n}{2}K_{n-1}(x,N)$$

holding for all x if $n = 0, 1, \ldots, N-1$ and for $x = 0, 1, \ldots, N$ for $n = N$.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

As K_N is mirror symmetric by construction, we want to show $\lambda_k - \lambda_{k+1} = -1$. Let $\lambda_0, \lambda_1, \ldots, \lambda_N$ be eigenvalues of $\mathcal{K}_N$ such that $\lambda_0 < \lambda_1 < \cdots < \lambda_N$. This is a strict inequality because our Jacobi matrix has simple spectrum.

Definition

Let $\mathcal{K}'_N$ be defined as corresponding to the following recurrence relation.

$$-xK_n(x,N) = \frac{N-n}{2}K_{n+1}(x,N) - \frac{N}{2}K_n(x,N) + \frac{n}{2}K_{n-1}(x,N)$$

holding for all x if $n = 0, 1, \ldots, N-1$ and for $x = 0, 1, \ldots, N$ for $n = N$. $\mathcal{K}'_N$ is mirror symmetric and similar to $\mathcal{K}_N$.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

- $\mathcal{K}'_N$ will have eignevalues $\lambda'_0, \ldots, \lambda'_N$.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

- $\mathcal{K}'_N$ will have eignevalues $\lambda'_0, \ldots, \lambda'_N$.
- This only holds for $x \in 0, 1, \ldots, N-1, \implies 0, 1, \ldots, N$ are the eigenvalues of $\mathcal{K}'_N$.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

- $\mathcal{K}'_N$ will have eignevalues $\lambda'_0, \ldots, \lambda'_N$.
- This only holds for $x \in 0, 1, \ldots, N-1, \implies 0, 1, \ldots, N$ are the eigenvalues of $\mathcal{K}'_N$.
- We see that the eigenvalue differences for $\mathcal{K}'_N$ are equal to -1, and that $\lambda_k = \lambda'_k + a$ for some constant a, $\lambda_k - \lambda_{k+1} = \lambda'_k - \lambda'_{k+1} - a + a = -1$, so $\lambda_k - \lambda_{k+1} = -1$ is satisfied.

Proving Krawtchouk Paths Proposition

Proposition

The Krawtchouck-style matrix $\mathcal{K}_N$ achieves PST from node 0 to node N at time $t = \pi$.

- $\mathcal{K}'_N$ will have eignevalues $\lambda'_0, \ldots, \lambda'_N$.
- This only holds for $x \in 0, 1, \ldots, N-1, \implies 0, 1, \ldots, N$ are the eigenvalues of $\mathcal{K}'_N$.
- We see that the eigenvalue differences for $\mathcal{K}'_N$ are equal to -1, and that $\lambda_k = \lambda'_k + a$ for some constant a, $\lambda_k - \lambda_{k+1} = \lambda'_k - \lambda'_{k+1} - a + a = -1$, so $\lambda_k - \lambda_{k+1} = -1$ is satisfied.
- Thus, when $t = \pi, \lambda_k - \lambda_{k+1} = \frac{-1\pi}{t}$. This, combined with mirror symmetry, implies by the PST on paths theorem that $\mathcal{K}_N$ realizes PST from node 0 to node N at time $t = \pi$.

□

Orthogonal Polynomials: Review

Recall that we can write our recurrence relation using $\mathcal{K}_N$ in terms of the **orthogonal Kratchouk polynomials**:

$$\mathcal{K}_N \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix} = -x \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix}$$

Orthogonal Polynomials: Review

Recall that we can write our recurrence relation using $\mathcal{K}_N$ in terms of the **orthogonal Kratchouk polynomials**:

$$\mathcal{K}_N \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix} = -x \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix}$$

This recurrence relation holds when $K_i(x)$ is evaluated at eigenvalues λ of $\mathcal{K}_N$.

Orthogonal Polynomials: Review

Recall that we can write our recurrence relation using $\mathcal{K}_N$ in terms of the **orthogonal Kratchouk polynomials**:

$$\mathcal{K}_N \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix} = -x \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix}$$

This recurrence relation holds when $K_i(x)$ is evaluated at eigenvalues λ of $\mathcal{K}_N$. The corresponding column vector is an eigenvector of $\mathcal{K}_N$.

Orthogonal Polynomials: Review

Recall that we can write our recurrence relation using $\mathcal{K}_N$ in terms of the **orthogonal Kratchouk polynomials**:

$$\mathcal{K}_N \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix} = -x \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix}$$

This recurrence relation holds when $K_i(x)$ is evaluated at eigenvalues λ of $\mathcal{K}_N$. The corresponding column vector is an eigenvector of $\mathcal{K}_N$. Recall:

- K_i of degree i exists for every $i \geq 0$!

Orthogonal Polynomials: Review

Recall that we can write our recurrence relation using $\mathcal{K}_N$ in terms of the **orthogonal Kratchouk polynomials**:

$$\mathcal{K}_N \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix} = -x \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix}$$

This recurrence relation holds when $K_i(x)$ is evaluated at eigenvalues λ of $\mathcal{K}_N$. The corresponding column vector is an eigenvector of $\mathcal{K}_N$. Recall:

- K_i of degree i exists for every $i \geq 0$!
- K_i, K_j orthogonal with respect to positive weight function $w(t)$

Orthogonal Polynomials: Review

Recall that we can write our recurrence relation using $\mathcal{K}_N$ in terms of the **orthogonal Kratchouk polynomials**:

$$\mathcal{K}_N \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix} = -x \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix}$$

This recurrence relation holds when $K_i(x)$ is evaluated at eigenvalues λ of $\mathcal{K}_N$. The corresponding column vector is an eigenvector of $\mathcal{K}_N$. Recall:

- K_i of degree i exists for every $i \geq 0$!
- K_i, K_j orthogonal with respect to positive weight function $w(t)$
- Zeroes of K_i and K_{i+1} interlace!

Orthogonal Polynomials: Review

Recall that we can write our recurrence relation using $\mathcal{K}_N$ in terms of the **orthogonal Kratchouk polynomials**:

$$\mathcal{K}_N \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix} = -x \begin{pmatrix} K_0(x) \\ K_1(x) \\ \vdots \\ K_N(x) \end{pmatrix}$$

This recurrence relation holds when $K_i(x)$ is evaluated at eigenvalues λ of $\mathcal{K}_N$. The corresponding column vector is an eigenvector of $\mathcal{K}_N$. Recall:

- K_i of degree i exists for every $i \geq 0$!
- K_i, K_j orthogonal with respect to positive weight function $w(t)$
- Zeroes of K_i and K_{i+1} interlace!

In fact...

There exists a family of orthogonal polynomials for any Jacobi matrix!

Splitting paths: Initial Algorithm

Path on Four Nodes to Cycle on Six Nodes

If the matrix $\mathcal{P} = \begin{pmatrix} 0 & b_1 & 0 & 0 \\ b_1 & 0 & b_2 & 0 \\ 0 & b_2 & 0 & b3 \\ 0 & 0 & b_3 & 0 \end{pmatrix}$ realizes perfect state transfer then

$$\mathcal{C} = \begin{pmatrix} 0 & b_1 & 0 & 0 & 0 & b_1 \\ \frac{1}{2}b_1 & 0 & b_2 & 0 & 0 & 0 \\ 0 & b_2 & 0 & \frac{1}{2}b3 & 0 & 0 \\ 0 & 0 & b_3 & 0 & b_3 & 0 \\ 0 & 0 & 0 & \frac{1}{2}b_3 & 0 & b_2 \\ \frac{1}{2}b_1 & 0 & 0 & 0 & b_2 & 0 \end{pmatrix}$$ also realizes perfect state transfer.

Maksym showed us the process on a P_3 and we made code to generalize it to P_n.

Initial Algorithm

- Set Up Eigenvalue equation with

Initial Algorithm

- Set Up Eigenvalue equation with $\mathcal{P}$:

$$\mathcal{P}\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = \lambda \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Initial Algorithm

- Set Up Eigenvalue equation with $\mathcal{P}$:

$$\mathcal{P}\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = \lambda \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

P_0, P_1, P_2, P_3 are orthogonal polynomials and when they are evaluated at the appropriate λ they are the corresponding eigenvector. We use this notation for convenience.

Initial Algorithm

- Set Up Eigenvalue equation with $\mathcal{P}$:

$$\mathcal{P}\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = \lambda \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

  P_0, P_1, P_2, P_3 are orthogonal polynomials and when they are evaluated at the appropriate λ they are the corresponding eigenvector. We use this notation for convenience.

- We get the system of equations:

$$\begin{aligned} b_1 P_1 &= \lambda P_0 \\ b_1 P_0 + b_2 P_2 &= \lambda P_1 \\ b_2 P_1 + b_3 P_3 &= \lambda P_2 \\ b_3 P_2 &= \lambda P_3 \end{aligned}$$

Initial Algorithm

- Each P_i corresponds to a node on P_4 as seen below

Initial Algorithm

- Each P_i corresponds to a node on P_4 as seen below

- We will break this path into a cycle by splitting each inner node, P_i, into two nodes, P_i^1, P_i^2, where,

$$P_i^1 + P_i^2 = P_i$$

Initial Algorithm

- Each P_i corresponds to a node on P_4 as seen below

- We will break this path into a cycle by splitting each inner node, P_i, into two nodes, P_i^1, P_i^2, where,

$$P_i^1 + P_i^2 = P_i \text{ and } P_i^1 = P_i^2$$

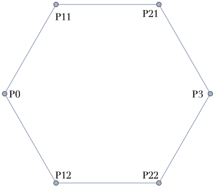
Initial Algorithm

- Each P_i corresponds to a node on P_4 as seen below

- We will break this path into a cycle by splitting each inner node, P_i, into two nodes, P_i^1, P_i^2, where,

$$P_i^1 + P_i^2 = P_i \text{ and } P_i^1 = P_i^2$$

- Our new cycle can be visualized as



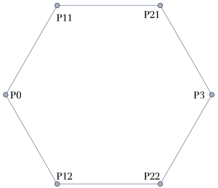
Initial Algorithm

- Each P_i corresponds to a node on P_4 as seen below

- We will break this path into a cycle by splitting each inner node, P_i, into two nodes, P_i^1, P_i^2, where,

$$P_i^1 + P_i^2 = P_i \text{ and } P_i^1 = P_i^2$$

- Our new cycle can be visualized as



- We will split the weight on the nodes such that the spectrum is preserved.

Initial Algorithm

- We split the path by splitting the weight on an inner node into two new nodes.

Initial Algorithm

- We split the path by splitting the weight on an inner node into two new nodes.
- The new system of equations is as follows:

Initial Algorithm

- We split the path by splitting the weight on an inner node into two new nodes.
- The new system of equations is as follows:

$$
\begin{aligned}
b_1 P_1 &= \lambda P_0 \\
\frac{1}{2}(b_1 P_0 + b_2 P_2) &= \lambda P_1^1 \\
\frac{1}{2}(b_2 P_1 + b_3 P_3) &= \lambda P_2^1 \\
b_3 P_2 &= \lambda P_3 \\
\frac{1}{2}(b_2 P_1 + b_3 P_3) &= \lambda P_2^2 \\
\frac{1}{2}(b_1 P_0 + b_2 P_2) &= \lambda P_1^1
\end{aligned}
$$

Initial Algorithm

- We will now relabel our polynomials,

Initial Algorithm

- We will now relabel our polynomials,

$$Q_0 = P_0, \quad Q_1 = P_1^1, \quad Q_2 = P_2^1$$
$$Q_3 = P_3, \quad Q_4 = P_2^2, \quad Q_5 = P_2^1$$

Initial Algorithm

- We will now relabel our polynomials,

$$Q_0 = P_0, \quad Q_1 = P_1^1, \quad Q_2 = P_2^1$$
$$Q_3 = P_3, \quad Q_4 = P_2^2, \quad Q_5 = P_2^1$$

- And rewrite our system,

$$b_1 Q_1 + b_1 Q_5 = \lambda Q_0$$
$$\frac{1}{2} Q_0 + b_2 Q_2 = \lambda Q_1$$
$$b_2 Q_1 + \frac{1}{2} Q_3 = \lambda Q_2$$
$$b_3 Q_2 + b_3 Q_4 = \lambda Q_3$$
$$b_2 Q_5 + \frac{1}{2} b_3 Q_3 = \lambda Q_4$$
$$\frac{1}{2} b_1 Q_0 + b_2 Q_4 = \lambda Q_5$$

Initial Algorithm

- We will now relabel our polynomials,

$$Q_0 = P_0, \quad Q_1 = P_1^1, \quad Q_2 = P_2^1$$
$$Q_3 = P_3, \quad Q_4 = P_2^2, \quad Q_5 = P_2^1$$

- And rewrite our system,

$$b_1 Q_1 + b_1 Q_5 = \lambda Q_0$$
$$\frac{1}{2} Q_0 + b_2 Q_2 = \lambda Q_1$$
$$b_2 Q_1 + \frac{1}{2} Q_3 = \lambda Q_2$$
$$b_3 Q_2 + b_3 Q_4 = \lambda Q_3$$
$$b_2 Q_5 + \frac{1}{2} b_3 Q_3 = \lambda Q_4$$
$$\frac{1}{2} b_1 Q_0 + b_2 Q_4 = \lambda Q_5$$

Initial Algorithm

- Now we have the matrix $\mathcal{C} = \begin{pmatrix} 0 & b_1 & 0 & 0 & 0 & b_1 \\ \frac{1}{2}b_1 & 0 & b_2 & 0 & 0 & 0 \\ 0 & b_2 & 0 & \frac{1}{2}b3 & 0 & 0 \\ 0 & 0 & b_3 & 0 & b_3 & 0 \\ 0 & 0 & 0 & \frac{1}{2}b_3 & 0 & b_2 \\ \frac{1}{2}b_1 & 0 & 0 & 0 & b_2 & 0 \end{pmatrix}$ which has the same eigenvalues as $\mathcal{P}$.

Example

P_5 to C_8

- We will use a P_5 that has weights from the Krawtchouk polynomials.

Example

P_5 to C_8

- We will use a P_5 that has weights from the Krawtchouk polynomials.
- Start with

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & \sqrt{\frac{3}{2}} & 0 & 0 \\ 0 & \sqrt{\frac{3}{2}} & 0 & \sqrt{\frac{3}{2}} & 0 \\ 0 & 0 & \sqrt{\frac{3}{2}} & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Example

- With the algorithm we get the cycle:

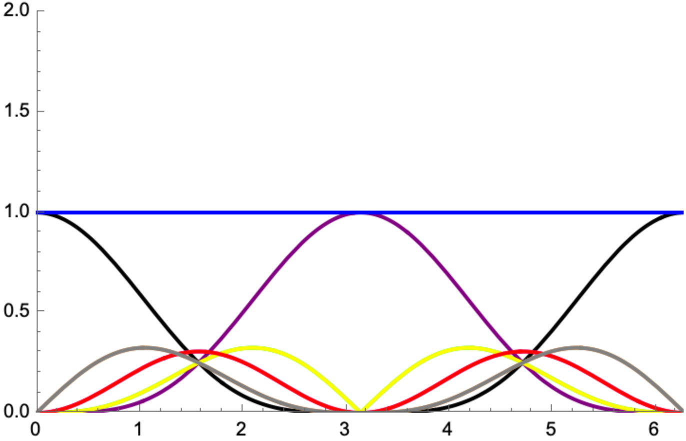
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & \sqrt{\frac{3}{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{\frac{3}{2}} & 0 & \sqrt{\frac{3}{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{3}{2}} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \sqrt{\frac{3}{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{\frac{3}{2}} & 0 & \sqrt{\frac{3}{2}} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \sqrt{\frac{3}{2}} & 0 \end{pmatrix}$$

Example

Do we get PST?

Example

Do we get PST? Yes at $t = \pi$



A Useful Pattern

We did a few examples and saw a pattern:

- $\begin{pmatrix} 0 & b_1 & 0 & b_1 \\ \frac{1}{2}b_1 & 0 & \frac{1}{2}b_2 & 0 \\ 0 & b_2 & 0 & b_2 \\ \frac{1}{2}b_1 & 0 & \frac{1}{2}b_2 & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & b_1 & 0 & 0 & 0 & b_1 \\ \frac{1}{2}b_1 & 0 & b_2 & 0 & 0 & 0 \\ 0 & b_2 & 0 & \frac{1}{2}b3 & 0 & 0 \\ 0 & 0 & b_3 & 0 & b_3 & 0 \\ 0 & 0 & 0 & \frac{1}{2}b_3 & 0 & b_2 \\ \frac{1}{2}b_1 & 0 & 0 & 0 & b_2 & 0 \end{pmatrix}$

The terms with a coefficient of $\frac{1}{2}$ are in the columns corresponding with nodes that do not have their weight split.

Given $b_1, \ldots, b_n$ this is much easier to code than the algorithm we used to derive these matrices.

Orthogonal Polynomials for a Cycle

We want to find families of orthogonal polynomials that correspond to our cycle matrices. However, we run into some issues:

Orthogonal Polynomials for a Cycle

We want to find families of orthogonal polynomials that correspond to our cycle matrices. However, we run into some issues:

$$\begin{pmatrix} 0 & b_1 & 0 & \dots & 0 & b_N \\ b_1 & 0 & b_2 & \dots & 0 & 0 \\ 0 & b_2 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & b_{n-1} \\ b_N & 0 & 0 & \dots & b_{n-1} & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_{N-1} \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_{N-1} \end{pmatrix}$$

Orthogonal Polynomials for a Cycle

We want to find families of orthogonal polynomials that correspond to our cycle matrices. However, we run into some issues:

$$\begin{pmatrix} 0 & b_1 & 0 & \dots & 0 & b_N \\ b_1 & 0 & b_2 & \dots & 0 & 0 \\ 0 & b_2 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & b_{n-1} \\ b_N & 0 & 0 & \dots & b_{n-1} & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_{N-1} \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_{N-1} \end{pmatrix}$$

The first row gives

$$b_1 P_1 + b_N P_{N-1} = x P_0.$$

Orthogonal Polynomials for a Cycle

We want to find families of orthogonal polynomials that correspond to our cycle matrices. However, we run into some issues:

$$\begin{pmatrix} 0 & b_1 & 0 & \dots & 0 & b_N \\ b_1 & 0 & b_2 & \dots & 0 & 0 \\ 0 & b_2 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & b_{n-1} \\ b_N & 0 & 0 & \dots & b_{n-1} & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_{N-1} \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_{N-1} \end{pmatrix}$$

The first row gives

$$b_1 P_1 + b_N P_{N-1} = x P_0.$$

If we assume that P_i has degree i, we get a contradiction!

A Workaround!

What if we remove the condition that P_i has degree i?

A Workaround!

What if we remove the condition that P_i has degree i? We made it work! We get:

$$\begin{pmatrix} 0 & b_1 & 0 & \dots & 0 & b_N \\ b_1 & 0 & b_2 & \dots & 0 & 0 \\ 0 & b_2 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & b_{n-1} \\ b_N & 0 & 0 & \dots & b_{n-1} & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix}$$

A Workaround!

What if we remove the condition that P_i has degree i? We made it work! We get:

$$\begin{pmatrix} 0 & b_1 & 0 & \dots & 0 & b_N \\ b_1 & 0 & b_2 & \dots & 0 & 0 \\ 0 & b_2 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & b_{n-1} \\ b_N & 0 & 0 & \dots & b_{n-1} & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix}$$

If we set $P_i = P_{N-1-i}$, we no longer have a degree contradiction ☺

A Workaround!

What if we remove the condition that P_i has degree i? We made it work! We get:

$$\begin{pmatrix} 0 & b_1 & 0 & \dots & 0 & b_N \\ b_1 & 0 & b_2 & \dots & 0 & 0 \\ 0 & b_2 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & b_{n-1} \\ b_N & 0 & 0 & \dots & b_{n-1} & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix}$$

If we set $P_i = P_{N-1-i}$, we no longer have a degree contradiction ☺
$\implies$ an NxN matrix gives us $\lceil \frac{N}{2} \rceil$ polynomials.

A Simple Example

We can explicitly solve for our polynomials given a matrix!
Let's do the simplest example: $b_i = 1$.

A Simple Example

We can explicitly solve for our polynomials given a matrix!
Let's do the simplest example: $b_i = 1$.

$$\begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 1 \\ 1 & 0 & 1 & \dots & 0 & 0 \\ 0 & 1 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix}$$

A Simple Example

We can explicitly solve for our polynomials given a matrix!
Let's do the simplest example: $b_i = 1$.

$$\begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 1 \\ 1 & 0 & 1 & \dots & 0 & 0 \\ 0 & 1 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & & & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix} = x \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_1 \\ P_0 \end{pmatrix}$$

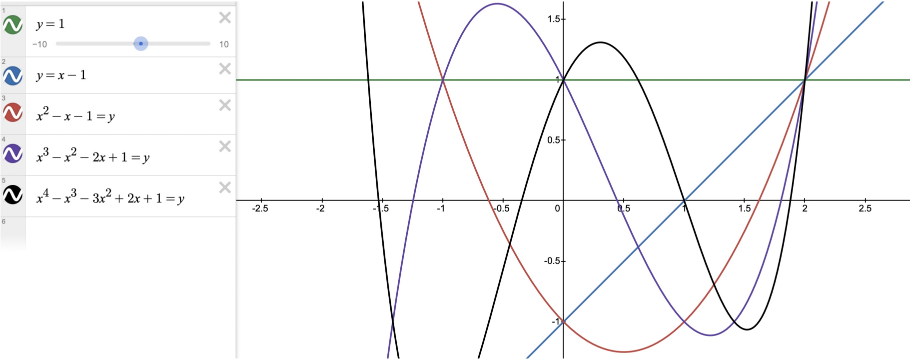
Our polynomials look like:

$$P_0 = 1, P_1 = x - 1$$
$$P_2 = x^2 - x - 1$$
$$P_3 = x^3 - x^2 - 2x + 1$$
$$P_4 = x^4 - x^3 - 3x^2 + 2x + 1$$

Our Example

It turns out to be pretty hard to check if polynomials are actually orthogonal, but we can observe some properties of these polynomials!

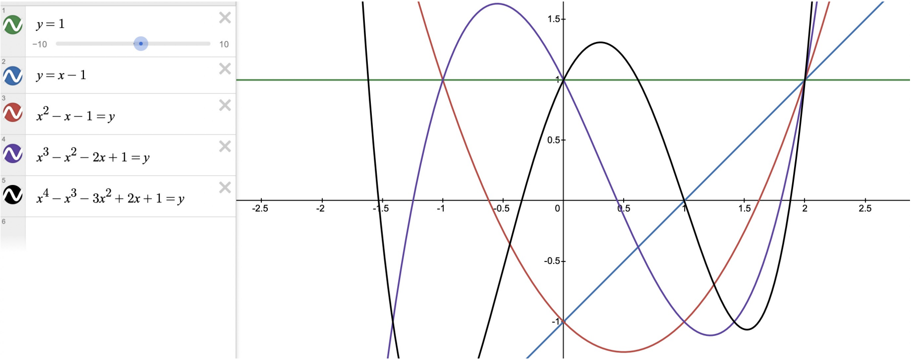
It turns out to be pretty hard to check if polynomials are actually orthogonal, but we can observe some properties of these polynomials!



The zeros of these polynomials interlace!

Our Example

It turns out to be pretty hard to check if polynomials are actually orthogonal, but we can observe some properties of these polynomials!



The zeros of these polynomials interlace!
This is evidence that we are on the right track to find some families of orthogonal polynomials.

Future Paths

- Formalizing the splitting algorithm and gaining a better understanding of how it works
- More work with orthogonal polynomials!

Thank you!

(Still to Maksym and Rachel)