Inverse Problems for Polynomial and Rational Structural Data

Richard Hollister

Department of Mathematics
University at Buffalo, SUNY

June 14, 2022

Polynomial inverse problem

Definition

Structural data for a polynomial matrix includes

- finite elementary divisors (or e-vals with partial multiplicities over $\mathbb{C}$),
- infinite elementary divisors,
- left and right minimal indices,
- size and degree.

We consider the following

Inverse Problem

Given a list of structural data, construct a polynomial matrix realization.

Polynomial inverse problem

Definition

Structural data for a polynomial matrix includes

- finite elementary divisors (or e-vals with partial multiplicities over $\mathbb{C}$),
- infinite elementary divisors,
- left and right minimal indices,
- size and degree.

We consider the following

Inverse Problem

Given a list of structural data, construct a polynomial matrix realization.

Polynomial inverse problem

Definition

Structural data for a polynomial matrix includes

- finite elementary divisors (or e-vals with partial multiplicities over $\mathbb{C}$),
- infinite elementary divisors,
- left and right minimal indices,
- size and degree.

We consider the following

Inverse Problem

Given a list of structural data, construct a polynomial matrix realization.

Polynomial inverse problem

Definition

Structural data for a polynomial matrix includes

- finite elementary divisors (or e-vals with partial multiplicities over $\mathbb{C}$),
- infinite elementary divisors,
- left and right minimal indices,
- size and degree.

We consider the following

Inverse Problem

Given a list of structural data, construct a polynomial matrix realization.

Polynomial inverse problem

Definition

Structural data for a polynomial matrix includes

- finite elementary divisors (or e-vals with partial multiplicities over $\mathbb{C}$),
- infinite elementary divisors,
- left and right minimal indices,
- size and degree.

We consider the following

Inverse Problem

Given a list of structural data, construct a polynomial matrix realization.

Background results

Some recent results...

Polynomial

- *general existance (De Terán, Dopico, Van Dooren, 2015)*
- *Kronecker-like quadratic (De Terán, Dopico, Mackey)*
- *structured quadratic (Perovic, Mackey, 2022)*
- *quasi-triangular (Anguas, Dopico, Hollister, Mackey)*

Rational

- *general existance (Anguas, Dopico, Hollister, Mackey, 2019)*

Background results

Some recent results...

Polynomial

- *general existance (De Terán, Dopico, Van Dooren, 2015)*
- *Kronecker-like quadratic (De Terán, Dopico, Mackey)*
- *structured quadratic (Perovic, Mackey, 2022)*
- *quasi-triangular (Anguas, Dopico, Hollister, Mackey)*

Rational

- *general existance (Anguas, Dopico, Hollister, Mackey, 2019)*

Background results

Some recent results...

Polynomial

- *general existance (De Terán, Dopico, Van Dooren, 2015)*
- *Kronecker-like quadratic (De Terán, Dopico, Mackey)*
- *structured quadratic (Perovic, Mackey, 2022)*
- *quasi-triangular (Anguas, Dopico, Hollister, Mackey)*

Rational

- *general existance (Anguas, Dopico, Hollister, Mackey, 2019)*

Background results

Some recent results...

Polynomial

- *general existance (De Terán, Dopico, Van Dooren, 2015)*
- *Kronecker-like quadratic (De Terán, Dopico, Mackey)*
- *structured quadratic (Perovic, Mackey, 2022)*
- *quasi-triangular (Anguas, Dopico, Hollister, Mackey)*

Rational

- *general existance (Anguas, Dopico, Hollister, Mackey, 2019)*

Background results

Some recent results...

Polynomial

- *general existance (De Terán, Dopico, Van Dooren, 2015)*
- *Kronecker-like quadratic (De Terán, Dopico, Mackey)*
- *structured quadratic (Perovic, Mackey, 2022)*
- *quasi-triangular (Anguas, Dopico, Hollister, Mackey)*

Rational

- *general existance (Anguas, Dopico, Hollister, Mackey, 2019)*

Index Sum Theorem

Theorem

Let $P(\lambda)$ have rank r and degree d. Then

$$\sum\{\textit{inv. poly. degs.}\}+\sum\{\textit{inf. e-val mult.}\}+\sum\{\textit{min. inds.}\}=dr.$$

- The only constraint for realizing a list of structural data.
- Rational counterpart (Van Dooren, 1979; Anguas, Dopico, Hollister, Mackey, 2019)

Index Sum Theorem

Theorem

Let $P(\lambda)$ have rank r and degree d. Then

$$\sum\{\textit{inv. poly. degs.}\}+\sum\{\textit{inf. e-val mult.}\}+\sum\{\textit{min. inds.}\} = dr.$$

- The only constraint for realizing a list of structural data.
- Rational counterpart (Van Dooren, 1979; Anguas, Dopico, Hollister, Mackey, 2019)

Index Sum Theorem

Theorem

Let $P(\lambda)$ have rank r and degree d. Then

$$\sum\{\textit{inv. poly. degs.}\}+\sum\{\textit{inf. e-val mult.}\}+\sum\{\textit{min. inds.}\}=dr.$$

- The only constraint for realizing a list of structural data.
- Rational counterpart (Van Dooren, 1979; Anguas, Dopico, Hollister, Mackey, 2019)

Strictly-regular inverse problem

Consider only data lists without infinite elementary divisors and without minimal indices.

Goal

Given a list containing finite elementary divisors and a choice of size and degree, construct a strictly-regular realization such that the structural data can be recovered without numerical calculations.

- direct-sum-of-blocks seems unreasonable
- focus on invariant polynomials instead of elementary divisors
- new types of blocks
- new ways of combining blocks

Strictly-regular inverse problem

Consider only data lists without infinite elementary divisors and without minimal indices.

Goal

Given a list containing finite elementary divisors and a choice of size and degree, construct a strictly-regular realization such that the structural data can be recovered without numerical calculations.

- direct-sum-of-blocks seems unreasonable
- focus on invariant polynomials instead of elementary divisors
- new types of blocks
- new ways of combining blocks

Strictly-regular inverse problem

Consider only data lists without infinite elementary divisors and without minimal indices.

Goal

Given a list containing finite elementary divisors and a choice of size and degree, construct a strictly-regular realization such that the structural data can be recovered without numerical calculations.

- direct-sum-of-blocks seems unreasonable
- focus on invariant polynomials instead of elementary divisors
- new types of blocks
- new ways of combining blocks

Strictly-regular inverse problem

Consider only data lists without infinite elementary divisors and without minimal indices.

Goal

Given a list containing finite elementary divisors and a choice of size and degree, construct a strictly-regular realization such that the structural data can be recovered without numerical calculations.

- direct-sum-of-blocks seems unreasonable
- focus on invariant polynomials instead of elementary divisors
- new types of blocks
- new ways of combining blocks

Strictly-regular inverse problem

Consider only data lists without infinite elementary divisors and without minimal indices.

Goal

Given a list containing finite elementary divisors and a choice of size and degree, construct a strictly-regular realization such that the structural data can be recovered without numerical calculations.

- direct-sum-of-blocks seems unreasonable
- focus on invariant polynomials instead of elementary divisors
- new types of blocks
- new ways of combining blocks

Strategy for building a realization

Let $\mathcal{L}$ be a list of fully factored invariant polynomials over $\mathbb{C}$ and choice of degree d.

1. Construct the Smith form.
2. Construct a "target diagonal" by re-arranging the diagonal factors.
3. Realize the target diagonal as a combination of two block types,
   - bidiagonal chains,
   - recombinant matrices

   put together using three operations
   - embedding (bidiag chain inside recombinant),
   - smoosh sum,
   - direct sum.

Strategy for building a realization

Let $\mathcal{L}$ be a list of fully factored invariant polynomials over $\mathbb{C}$ and choice of degree d.

1. Construct the Smith form.
2. Construct a "target diagonal" by re-arranging the diagonal factors.
3. Realize the target diagonal as a combination of two block types,
   - bidiagonal chains,
   - recombinant matrices

   put together using three operations
   - embedding (bidiag chain inside recombinant),
   - smoosh sum,
   - direct sum.

Strategy for building a realization

Let $\mathcal{L}$ be a list of fully factored invariant polynomials over $\mathbb{C}$ and choice of degree d.

1. Construct the Smith form.
2. Construct a "target diagonal" by re-arranging the diagonal factors.
3. Realize the target diagonal as a combination of two block types,
   - bidiagonal chains,
   - recombinant matrices

   put together using three operations
   - embedding (bidiag chain inside recombinant),
   - smoosh sum,
   - direct sum.

Strategy for building a realization

Let $\mathcal{L}$ be a list of fully factored invariant polynomials over $\mathbb{C}$ and choice of degree d.

1. Construct the Smith form.
2. Construct a "target diagonal" by re-arranging the diagonal factors.
3. Realize the target diagonal as a combination of two block types,
   - bidiagonal chains,
   - recombinant matrices

   put together using three operations
   - embedding (bidiag chain inside recombinant),
   - smoosh sum,
   - direct sum.

Strategy for building a realization

Let $\mathcal{L}$ be a list of fully factored invariant polynomials over $\mathbb{C}$ and choice of degree d.

1. Construct the Smith form.
2. Construct a "target diagonal" by re-arranging the diagonal factors.
3. Realize the target diagonal as a combination of two block types,
   - bidiagonal chains,
   - recombinant matrices

   put together using three operations
   - embedding (bidiag chain inside recombinant),
   - smoosh sum,
   - direct sum.

Strategy for building a realization

Let $\mathcal{L}$ be a list of fully factored invariant polynomials over $\mathbb{C}$ and choice of degree d.

1. Construct the Smith form.
2. Construct a "target diagonal" by re-arranging the diagonal factors.
3. Realize the target diagonal as a combination of two block types,
   - bidiagonal chains,
   - recombinant matrices

   put together using three operations
   - embedding (bidiag chain inside recombinant),
   - smoosh sum,
   - direct sum.

Strategy for building a realization

Let $\mathcal{L}$ be a list of fully factored invariant polynomials over $\mathbb{C}$ and choice of degree d.

1. Construct the Smith form.
2. Construct a "target diagonal" by re-arranging the diagonal factors.
3. Realize the target diagonal as a combination of two block types,
   - bidiagonal chains,
   - recombinant matrices

   put together using three operations
   - embedding (bidiag chain inside recombinant),
   - smoosh sum,
   - direct sum.

Arrangement algorithm

Goal

Re-arrange the invariant polynomials so that the degree is "spread out".

- Start with an invariant polynomial of degree higher than d.
- If the average degree excedes d, include an invariant poly with degree lower than d.
- If the average degree is less than d, include an invariant polynomial of higher degree.
- The result is a permuted Smith form that can be partioned into blocks whose average degree is close to d.

Arrangement algorithm

Goal

Re-arrange the invariant polynomials so that the degree is "spread out".

- Start with an invariant polynomial of degree higher than d.
- If the average degree excedes d, include an invariant poly with degree lower than d.
- If the average degree is less than d, include an invariant polynomial of higher degree.
- The result is a permuted Smith form that can be partioned into blocks whose average degree is close to d.

Arrangement algorithm

Goal

Re-arrange the invariant polynomials so that the degree is "spread out".

- Start with an invariant polynomial of degree higher than d.
- If the average degree excedes d, include an invariant poly with degree lower than d.
- If the average degree is less than d, include an invariant polynomial of higher degree.
- The result is a permuted Smith form that can be partioned into blocks whose average degree is close to d.

Arrangement algorithm

Goal

Re-arrange the invariant polynomials so that the degree is "spread out".

- Start with an invariant polynomial of degree higher than d.
- If the average degree excedes d, include an invariant poly with degree lower than d.
- If the average degree is less than d, include an invariant polynomial of higher degree.
- The result is a permuted Smith form that can be partioned into blocks whose average degree is close to d.

Arrangement algorithm

Goal

Re-arrange the invariant polynomials so that the degree is "spread out".

- Start with an invariant polynomial of degree higher than d.
- If the average degree excedes d, include an invariant poly with degree lower than d.
- If the average degree is less than d, include an invariant polynomial of higher degree.
- The result is a permuted Smith form that can be partioned into blocks whose average degree is close to d.

Arrangement algorithm in action

Define the *running degree excess*

$$e_0 := 0$$

$$e_j := \sum_{i=1}^{j} \deg(d_i) - jd.$$

After t steps:

$$\mathcal{L}_h = \{x_1, x_2, \ldots, x_r\}$$
$$\mathcal{L}_\ell = \{y_1, y_2, \ldots, y_s\}$$

$$D = \mathrm{diag}\{d_1, d_2, \ldots, d_t, \_\_\_, \_\_\_, \ldots, \_\_\_\}$$

Arrangement algorithm in action

Define the *running degree excess*

$$e_0 := 0$$
$$e_j := \sum_{i=1}^{j} \deg(d_i) - jd.$$

After t steps: If $e_t \leq 0$, then...

$$\mathcal{L}_h = \{x_1, x_2, \ldots, x_r\}$$
$$\mathcal{L}_\ell = \{y_1, y_2, \ldots, y_s\}$$

$$D = \operatorname{diag}\{d_1, d_2, \ldots, d_t, \underline{\quad}, \underline{\quad}, \ldots, \underline{\quad}\}$$

Arrangement algorithm in action

Define the *running degree excess*

$$e_0 := 0$$

$$e_j := \sum_{i=1}^{j} \deg(d_i) - jd.$$

After t steps: If $e_t \leq 0$, then...

$$\mathcal{L}_h = \{x_1, x_2, \ldots, x_r\}$$
$$\mathcal{L}_\ell = \{y_1, y_2, \ldots, y_s\}$$

$$D = \operatorname{diag}\{d_1, d_2, \ldots, d_t, \underline{\quad}, \underline{\quad}, \ldots, \underline{\quad}\}$$

Arrangement algorithm in action

Define the *running degree excess*

$$e_0 := 0$$
$$e_j := \sum_{i=1}^{j} \deg(d_i) - jd.$$

After t steps: If $e_t \leq 0$, then...

$$\mathcal{L}_h = \{x_1, x_2, \ldots, \cancel{x_r}\}$$
$$\mathcal{L}_\ell = \{y_1, y_2, \ldots, y_s\}$$

$$D = \operatorname{diag}\{d_1, d_2, \ldots, d_t, \underline{\ x_r\ }, \underline{\quad}, \ldots, \underline{\quad}\}$$

Arrangement algorithm in action

Define the *running degree excess*

$$e_0 := 0$$
$$e_j := \sum_{i=1}^{j} \deg(d_i) - jd.$$

After t steps: If $e_t > 0$, then...

$$\mathcal{L}_h = \{x_1, x_2, \ldots, x_r\}$$
$$\mathcal{L}_\ell = \{y_1, y_2, \ldots, y_s\}$$

$$D = \operatorname{diag}\{d_1, d_2, \ldots, d_t, \_\_\_, \_\_\_, \ldots, \_\_\_\}$$

Arrangement algorithm in action

Define the *running degree excess*

$$e_0 := 0$$
$$e_j := \sum_{i=1}^{j} \deg(d_i) - jd.$$

After t steps: If $e_t > 0$, then...

$$\mathcal{L}_h = \{x_1, x_2, \ldots, x_r\}$$
$$\mathcal{L}_\ell = \{y_1, y_2, \ldots, y_s\}$$

$$D = \mathrm{diag}\{d_1, d_2, \ldots, d_t, \_\_\_, \_\_\_, \ldots, \_\_\_\}$$

Arrangement algorithm in action

Define the *running degree excess*

$$e_0 := 0$$
$$e_j := \sum_{i=1}^{j} \deg(d_i) - jd.$$

After t steps: If $e_t > 0$, then...

$$\mathcal{L}_h = \{x_1, x_2, \ldots, x_r\}$$
$$\mathcal{L}_\ell = \{y_1, y_2, \ldots, \cancel{y_s}\}$$

$$D = \mathrm{diag}\{d_1, d_2, \ldots, d_t, \underline{\ y_s\ }, \underline{\qquad}, \ldots, \underline{\qquad}\}$$

Bidiagonal chains

Definition

A polynomial matrix $\mathcal{B}$ is called a *bidiagonal chain* if it is of the form

$$\mathcal{B} = \begin{bmatrix} r_n & s_{n-1} & & & \\ & r_{n-1}s_{n-1} & s_{n-2} & & \\ & & \ddots & \ddots & \\ & & & r_2 s_2 & s_1 \\ & & & & r_1 s_1 \end{bmatrix},$$

where $s_1 | s_2 | \cdots | s_{n-1} | r_n$

- Data can be stored as two vectors of sizes n and $n-1$.

Bidiagonal chains

Definition

A polynomial matrix $\mathcal{B}$ is called a *bidiagonal chain* if it is of the form

$$\mathcal{B} = \begin{bmatrix} r_n & s_{n-1} & & & \\ & r_{n-1}s_{n-1} & s_{n-2} & & \\ & & \ddots & \ddots & \\ & & & r_2 s_2 & s_1 \\ & & & & r_1 s_1 \end{bmatrix},$$

where $s_1 | s_2 | \cdots | s_{n-1} | r_n$

- Data can be stored as two vectors of sizes n and $n-1$.

Diagonalizing a bidiagonal chain

Lemma

A bidiagonal chain in the form of $\mathcal{B}$ is equivalent to

$$\operatorname{diag}\left\{\prod_{i=1}^{n} r_i, s_{n-1}, \ldots, s_1\right\}.$$

- Start in the bottom right and use row and column operations.
- The divisibility chain $s_1 | s_2 | \cdots | s_{n-1} | r_n$ plays a key role.
- Diagonalized matrix is in Smith form.
- Can read off diagonalization from original matrix.

Diagonalizing a bidiagonal chain

Lemma

A bidiagonal chain in the form of $\mathcal{B}$ is equivalent to

$$\operatorname{diag}\left\{\prod_{i=1}^{n} r_i, s_{n-1}, \ldots, s_1\right\}.$$

- Start in the bottom right and use row and column operations.
- The divisibility chain $s_1|s_2|\cdots|s_{n-1}|r_n$ plays a key role.
- Diagonalized matrix is in Smith form.
- Can read off diagonalization from original matrix.

Diagonalizing a bidiagonal chain

Lemma

A bidiagonal chain in the form of $\mathcal{B}$ is equivalent to

$$\operatorname{diag}\left\{\prod_{i=1}^{n} r_i, s_{n-1}, \ldots, s_1\right\}.$$

- Start in the bottom right and use row and column operations.
- The divisibility chain $s_1|s_2|\cdots|s_{n-1}|r_n$ plays a key role.
- Diagonalized matrix is in Smith form.
- Can read off diagonalization from original matrix.

Diagonalizing a bidiagonal chain

Lemma

A bidiagonal chain in the form of $\mathcal{B}$ is equivalent to

$$\operatorname{diag}\left\{\prod_{i=1}^{n} r_i, s_{n-1}, \ldots, s_1\right\}.$$

- Start in the bottom right and use row and column operations.
- The divisibility chain $s_1 | s_2 | \cdots | s_{n-1} | r_n$ plays a key role.
- Diagonalized matrix is in Smith form.
- Can read off diagonalization from original matrix.

Diagonalizing a bidiagonal chain

Lemma

A bidiagonal chain in the form of $\mathcal{B}$ is equivalent to

$$\operatorname{diag}\left\{\prod_{i=1}^{n} r_i, s_{n-1}, \ldots, s_1\right\}.$$

- Start in the bottom right and use row and column operations.
- The divisibility chain $s_1 | s_2 | \cdots | s_{n-1} | r_n$ plays a key role.
- Diagonalized matrix is in Smith form.
- Can read off diagonalization from original matrix.

Recombinant matrices

Definition

A *recombinant matrix* is a polynomial matrix $\mathcal{R}$ of the form

$$\mathcal{R} = \begin{bmatrix} p_n & p_{n-1} & \cdots & p_2 & p_1 \\ & p_{n-1}q_n & \cdots & p_2 q_n & p_1 q_n \\ & & \ddots & \vdots & \vdots \\ & & & p_2 q_3 & p_1 q_3 \\ & & & & p_1 q_2 \end{bmatrix}$$

where $p_1 | p_2 | \cdots | p_n$.

- Data can be stored as two vectors of sizes n and $n+1$.
- Upper triangular part of a rank one matrix.

Recombinant matrices

Definition

A *recombinant matrix* is a polynomial matrix $\mathcal{R}$ of the form

$$\mathcal{R} = \begin{bmatrix} p_n & p_{n-1} & \cdots & p_2 & p_1 \\ & p_{n-1}q_n & \cdots & p_2 q_n & p_1 q_n \\ & & \ddots & \vdots & \vdots \\ & & & p_2 q_3 & p_1 q_3 \\ & & & & p_1 q_2 \end{bmatrix}$$

where $p_1 | p_2 | \cdots | p_n$.

- Data can be stored as two vectors of sizes n and $n-1$.
- Upper triangular part of a rank one matrix.

Recombinant matrices

Definition

A *recombinant matrix* is a polynomial matrix $\mathcal{R}$ of the form

$$\mathcal{R} = \begin{bmatrix} p_n & p_{n-1} & \cdots & p_2 & p_1 \\ & p_{n-1}q_n & \cdots & p_2q_n & p_1q_n \\ & & \ddots & \vdots & \vdots \\ & & & p_2q_3 & p_1q_3 \\ & & & & p_1q_2 \end{bmatrix}$$

where $p_1 | p_2 | \cdots | p_n$.

- Data can be stored as two vectors of sizes n and $n-1$.
- Upper triangular part of a rank one matrix.

Diagonalizing a recombinant matrix

Lemma

A recombinant matrix in the form of $\mathcal{R}$ is equivalent to

$$\operatorname{diag}\left\{p_n q_n,\ p_1,\ p_{n-1} q_{n-1}, \ldots, p_2 q_2\right\}.$$

- Use row and column operations and the divisibility chain $p_1 | p_2 | \cdots | p_n$.
- Can read off diagonalization from original matrix.
- Can construct p's and q's so that diagonalized matrix in Smith form.

Diagonalizing a recombinant matrix

Lemma

A recombinant matrix in the form of $\mathcal{R}$ is equivalent to

$$\operatorname{diag}\left\{p_n q_n,\ p_1,\ p_{n-1} q_{n-1}, \ldots, p_2 q_2\right\}.$$

- Use row and column operations and the divisibility chain $p_1 | p_2 | \cdots | p_n$.
- Can read off diagonalization from original matrix.
- Can construct p's and q's so that diagonalized matrix in Smith form.

Diagonalizing a recombinant matrix

Lemma

A recombinant matrix in the form of $\mathcal{R}$ is equivalent to

$$\operatorname{diag}\{p_n q_n,\ p_1,\ p_{n-1}q_{n-1}, \dots, p_2 q_2\}.$$

- Use row and column operations and the divisibility chain $p_1|p_2|\cdots|p_n$.
- Can read off diagonalization from original matrix.
- Can construct p's and q's so that diagonalized matrix in Smith form.

Diagonalizing a recombinant matrix

Lemma

A recombinant matrix in the form of $\mathcal{R}$ is equivalent to

$$\operatorname{diag}\left\{p_n q_n,\ p_1,\ p_{n-1}q_{n-1}, \ldots, p_2 q_2\right\}.$$

- Use row and column operations and the divisibility chain $p_1 | p_2 | \cdots | p_n$.
- Can read off diagonalization from original matrix.
- Can construct p's and q's so that diagonalized matrix in Smith form.

Embedding a bidiagonal chain inside a recombinant matrix

A bidiagonal chain and recombinant matrix pair can be embedded as follows.

$$\left[\begin{array}{cc|ccc|ccc}
p_n & p_{n-1} & & & & p_{n-2} & \cdots & p_1 \\
0 & p_{n-1}q_n & s_m & & & p_{n-2}q_n & \cdots & p_1 q_n \\
\hline
 & & r_m s_m & \ddots & & & & \\
 & & & \ddots & s_1 & & & \\
 & & & & r_1 s_1 & & & \\
\hline
0 & 0 & & & & p_{n-2}q_{n-1} & \cdots & p_1 q_{n-1} \\
\vdots & \vdots & & & & & \ddots & \vdots \\
0 & 0 & & & & & & p_1 q_2
\end{array}\right]$$

- To diagonalize, need $s_m | p_n q_n$.

Embedding a bidiagonal chain inside a recombinant matrix

A bidiagonal chain and recombinant matrix pair can be embedded as follows.

$$\left[\begin{array}{cc|ccc|ccc}
p_n & p_{n-1} & & & & p_{n-2} & \cdots & p_1 \\
0 & p_{n-1}q_n & s_m & & & p_{n-2}q_n & \cdots & p_1q_n \\
\hline
 & & r_m s_m & \ddots & & & & \\
 & & & \ddots & s_1 & & & \\
 & & & & r_1 s_1 & & & \\
\hline
0 & 0 & & & & p_{n-2}q_{n-1} & \cdots & p_1q_{n-1} \\
\vdots & \vdots & & & & & \ddots & \vdots \\
0 & 0 & & & & & & p_1q_2
\end{array}\right]$$

- To diagonalize, need $s_m | p_n q_n$.

Smoosh sum

Definition

The *smoosh sum* of $A \in M_{m,n}(\mathbb{C})$ and $B \in M_{p,\ell}(\mathbb{C})$ is the $(m+p-1) \times (n+\ell-1)$ matrix

$$A \vee B := \left[\begin{array}{ccc|ccc} a_{1,1} & \cdots & a_{1,n} & & & \\ \vdots & & \vdots & & & \\ a_{m,1} & \cdots & a_{m,n} & b_{1,2} & \cdots & b_{1,\ell} \\ \hline & & b_{2,1} & b_{2,2} & \cdots & b_{2,\ell} \\ & & \vdots & \vdots & & \vdots \\ & & b_{p,1} & b_{p,2} & \cdots & b_{p,\ell} \end{array}\right].$$

- Used to join two or more embedded pairs together.

Smoosh sum

Definition

The *smoosh sum* of $A \in M_{m,n}(\mathbb{C})$ and $B \in M_{p,\ell}(\mathbb{C})$ is the $(m+p-1) \times (n+\ell-1)$ matrix

$$A \vee B := \left[\begin{array}{ccc|ccc} a_{1,1} & \cdots & a_{1,n} & & & \\ \vdots & & \vdots & & & \\ a_{m,1} & \cdots & a_{m,n} & b_{1,2} & \cdots & b_{1,\ell} \\ \hline & & b_{2,1} & b_{2,2} & \cdots & b_{2,\ell} \\ & & \vdots & \vdots & & \vdots \\ & & b_{p,1} & b_{p,2} & \cdots & b_{p,\ell} \end{array}\right].$$

- Used to join two or more embedded pairs together.

Main result

Theorem

Any list of invariant polynomials (completely factored) together with a choice of degree consistant with the index sum theorem can be realized by

- *a direct sum of blocks,*
- *each block is a smoosh sum of embedded pairs.*

- Can be diagonalized without any numerical computations.
- The amount of data contained in the matrix is about $2n$ polynomials of degree d or less.

Main result

Theorem

Any list of invariant polynomials (completely factored) together with a choice of degree consistant with the index sum theorem can be realized by

- *a direct sum of blocks,*
- *each block is a smoosh sum of embedded pairs.*

- Can be diagonalized without any numerical computations.
- The amount of data contained in the matrix is about $2n$ polynomials of degree d or less.

Main result

Theorem

Any list of invariant polynomials (completely factored) together with a choice of degree consistant with the index sum theorem can be realized by

- *a direct sum of blocks,*
- *each block is a smoosh sum of embedded pairs.*

- Can be diagonalized without any numerical computations.
- The amount of data contained in the matrix is about $2n$ polynomials of degree d or less.

Additional features of the realization

1. The realization is not unique, not even up to permutation of blocks.
   - Some uniqueness can be salvaged during the arrangement algorithm.
2. Size of blocks is determined by choices during arrangement algorithm.
3. When target degree is 1, the Weierstrass canonical form can be easily recovered.
   - Requires reading the diagonal entries of the realization and deleting off-diagonal entries when the diagonal entry changes value.
   - Results in the Weierstrass form, up to permutation of blocks.
4. Extended in my thesis to regular polynomial matrix, general polynomial matrices, and general rational matrices.

Additional features of the realization

1. The realization is not unique, not even up to permutation of blocks.
   - Some uniqueness can be salvaged during the arrangement algorithm.
2. Size of blocks is determined by choices during arrangement algorithm.
3. When target degree is 1, the Weierstrass canonical form can be easily recovered.
   - Requires reading the diagonal entries of the realization and deleting off-diagonal entries when the diagonal entry changes value.
   - Results in the Weierstrass form, up to permutation of blocks.
4. Extended in my thesis to regular polynomial matrix, general polynomial matrices, and general rational matrices.

Additional features of the realization

1. The realization is not unique, not even up to permutation of blocks.
   - Some uniqueness can be salvaged during the arrangement algorithm.
2. Size of blocks is determined by choices during arrangement algorithm.
3. When target degree is 1, the Weierstrass canonical form can be easily recovered.
   - Requires reading the diagonal entries of the realization and deleting off-diagonal entries when the diagonal entry changes value.
   - Results in the Weierstrass form, up to permutation of blocks.
4. Extended in my thesis to regular polynomial matrix, general polynomial matrices, and general rational matrices.

Additional features of the realization

1. The realization is not unique, not even up to permutation of blocks.
   - Some uniqueness can be salvaged during the arrangement algorithm.
2. Size of blocks is determined by choices during arrangement algorithm.
3. When target degree is 1, the Weierstrass canonical form can be easily recovered.
   - Requires reading the diagonal entries of the realization and deleting off-diagonal entries when the diagonal entry changes value.
   - Results in the Weierstrass form, up to permutation of blocks.
4. Extended in my thesis to regular polynomial matrix, general polynomial matrices, and general rational matrices.

Additional features of the realization

1. The realization is not unique, not even up to permutation of blocks.
   - Some uniqueness can be salvaged during the arrangement algorithm.
2. Size of blocks is determined by choices during arrangement algorithm.
3. When target degree is 1, the Weierstrass canonical form can be easily recovered.
   - Requires reading the diagonal entries of the realization and deleting off-diagonal entries when the diagonal entry changes value.
   - Results in the Weierstrass form, up to permutation of blocks.
4. Extended in my thesis to regular polynomial matrix, general polynomial matrices, and general rational matrices.

Additional features of the realization

1. The realization is not unique, not even up to permutation of blocks.
   - Some uniqueness can be salvaged during the arrangement algorithm.
2. Size of blocks is determined by choices during arrangement algorithm.
3. When target degree is 1, the Weierstrass canonical form can be easily recovered.
   - Requires reading the diagonal entries of the realization and deleting off-diagonal entries when the diagonal entry changes value.
   - Results in the Weierstrass form, up to permutation of blocks.
4. Extended in my thesis to regular polynomial matrix, general polynomial matrices, and general rational matrices.

Additional features of the realization

1. The realization is not unique, not even up to permutation of blocks.
   - Some uniqueness can be salvaged during the arrangement algorithm.
2. Size of blocks is determined by choices during arrangement algorithm.
3. When target degree is 1, the Weierstrass canonical form can be easily recovered.
   - Requires reading the diagonal entries of the realization and deleting off-diagonal entries when the diagonal entry changes value.
   - Results in the Weierstrass form, up to permutation of blocks.
4. Extended in my thesis to regular polynomial matrix, general polynomial matrices, and general rational matrices.

The future of these realizations

Working on modifying these results for

- Regular, structured (Hermitian, palindromic, alternating) polynomial matrices,
- General, structured polynomial matrices,
- Structured rational matrices.

The plan is to

- Refine the bidiagonal chains and recombinant matrices to reflect the given structure,
- Put them together in ways that preserve the structure.

The future of these realizations

Working on modifying these results for

- Regular, structured (Hermitian, palindromic, alternating) polynomial matrices,
- General, structured polynomial matrices,
- Structured rational matrices.

The plan is to

- Refine the bidiagonal chains and recombinant matrices to reflect the given structure,
- Put them together in ways that preserve the structure.

The future of these realizations

Working on modifying these results for

- Regular, structured (Hermitian, palindromic, alternating) polynomial matrices,
- General, structured polynomial matrices,
- Structured rational matrices.

The plan is to

- Refine the bidiagonal chains and recombinant matrices to reflect the given structure,
- Put them together in ways that preserve the structure.

The future of these realizations

Working on modifying these results for

- Regular, structured (Hermitian, palindromic, alternating) polynomial matrices,
- General, structured polynomial matrices,
- Structured rational matrices.

The plan is to

- Refine the bidiagonal chains and recombinant matrices to reflect the given structure,
- Put them together in ways that preserve the structure.

The future of these realizations

Working on modifying these results for

- Regular, structured (Hermitian, palindromic, alternating) polynomial matrices,
- General, structured polynomial matrices,
- Structured rational matrices.

The plan is to

- Refine the bidiagonal chains and recombinant matrices to reflect the given structure,
- Put them together in ways that preserve the structure.

The future of these realizations

Working on modifying these results for

- Regular, structured (Hermitian, palindromic, alternating) polynomial matrices,
- General, structured polynomial matrices,
- Structured rational matrices.

The plan is to

- Refine the bidiagonal chains and recombinant matrices to reflect the given structure,
- Put them together in ways that preserve the structure.

Questions???

?

Richard Hollister
raholis@buffalo.edu