Computations with Ore Polynomial Matrices

Howard Cheng

Department of Mathematics and Computer Science
University of Lethbridge, Canada

Joint work with Bernhard Beckermann, Patrick Davies, George Labahn.

Ore Polynomials

- The ring of Ore polynomials $\mathbb{Q}_{\mathbb{D}}[X;\sigma,\delta]$
  - σ: automorphism over $\mathbb{Q}_{\mathbb{D}}$
  - δ: additive homomorphism on $\mathbb{Q}_{\mathbb{D}}$
  - Polynomial multiplication: $Xa = \sigma(a)X + \delta(a)$

	$\sigma(a(t))$	$\delta(a(t))$
Polynomials	$a(t)$	0
Differential operator	$a(t)$	$a'(t)$
Difference operator	$a(t+1)$	0

- $\delta = 0 \Rightarrow$ shift polynomials
- Matrices of Ore polynomials represent systems of linear differential equations, difference equations, etc.

Problems

Given an $m \times n$ Ore polynomial matrix $\mathbf{F}(X)$ of degree N, we wish to compute:

- a basis for the left nullspace of $\mathbf{F}(X)$;

Problems

Given an $m \times n$ Ore polynomial matrix $\mathbf{F}(X)$ of degree N, we wish to compute:

- a basis for the left nullspace of $\mathbf{F}(X)$;
- a row-equivalent matrix with a non-singular trailing coefficient;

Problems

Given an $m \times n$ Ore polynomial matrix $\mathbf{F}(X)$ of degree N, we wish to compute:

- a basis for the left nullspace of $\mathbf{F}(X)$;
- a row-equivalent matrix with a non-singular trailing coefficient;
- a row-reduced form of $\mathbf{F}(X)$
  - leading row coefficient of nonzero rows have full row rank

  and associated unimodular transformation matrix $\mathbf{U}(X)$;

Problems

Given an $m \times n$ Ore polynomial matrix $\mathbf{F}(X)$ of degree N, we wish to compute:

- a basis for the left nullspace of $\mathbf{F}(X)$;
- a row-equivalent matrix with a non-singular trailing coefficient;
- a row-reduced form of $\mathbf{F}(X)$
  - leading row coefficient of nonzero rows have full row rank

  and associated unimodular transformation matrix $\mathbf{U}(X)$;
- the Popov form of $\mathbf{F}(X)$
  - leading row coefficient is triangular (weak Popov form)
  - leading entry is monic and has highest degree in its column

  and associated unimodular transformation matrix $\mathbf{U}(X)$.

Solving these problems allows us to:

- determine the rank of a matrix of Ore polynomials;

Applications

Solving these problems allows us to:

- determine the rank of a matrix of Ore polynomials;
- find rational solutions to systems of linear functional equations;

Applications

Solving these problems allows us to:

- determine the rank of a matrix of Ore polynomials;
- find rational solutions to systems of linear functional equations;
- compute greatest common right divisors (GCRD) and least common left multiples (LCLM)
  i.e. intersection and union of systems;

Applications

Solving these problems allows us to:

- determine the rank of a matrix of Ore polynomials;
- find rational solutions to systems of linear functional equations;
- compute greatest common right divisors (GCRD) and least common left multiples (LCLM)
  i.e. intersection and union of systems;
- reduce order of systems of equations;

Applications

Solving these problems allows us to:

- determine the rank of a matrix of Ore polynomials;
- find rational solutions to systems of linear functional equations;
- compute greatest common right divisors (GCRD) and least common left multiples (LCLM)
  i.e. intersection and union of systems;
- reduce order of systems of equations;
- isolate highest powers.
  e.g. convert DAE systems to first order.

Row Operations

We can obtain normal forms by elementary row operations:

1. interchange two rows

Row Operations

We can obtain normal forms by elementary row operations:

1. interchange two rows
2. multiply a row by a nonzero element (constant in most cases)

Row Operations

We can obtain normal forms by elementary row operations:

1. interchange two rows
2. multiply a row by a nonzero element (constant in most cases)
3. add a polynomial multiple of one row to another

Row Operations

We can obtain normal forms by elementary row operations:

1. interchange two rows
2. multiply a row by a nonzero element (constant in most cases)
3. add a polynomial multiple of one row to another

We also wish to compute the transformation matrix in many cases.

Issues

- Straightforward elimination may introduce coefficient growth:
  - from Gaussian elimination;
  - from multiplication by X.

Issues

- Straightforward elimination may introduce coefficient growth:
  - from Gaussian elimination;
  - from multiplication by X.
- Algorithms for polynomial matrices may not work on Ore polynomial matrices.

Issues

- Straightforward elimination may introduce coefficient growth:
  - from Gaussian elimination;
  - from multiplication by X.
- Algorithms for polynomial matrices may not work on Ore polynomial matrices.
- Proofs of correct algorithms for polynomial matrices may rely on commutativity or fractions of matrix elements.

Related Works

- GCRD and LCLM of Ore polynomials (Li, Li and Nemes):

Related Works

- GCRD and LCLM of Ore polynomials (Li, Li and Nemes):
  - normal forms of 2×1 Ore polynomial matrix;

Related Works

- GCRD and LCLM of Ore polynomials (Li, Li and Nemes):
  - normal forms of 2×1 Ore polynomial matrix;
  - subresultant (fraction-free) and modular algorithms;

Related Works

- GCRD and LCLM of Ore polynomials (Li, Li and Nemes):
  - normal forms of 2×1 Ore polynomial matrix;
  - subresultant (fraction-free) and modular algorithms;
- EG elimination and improvements (Abramov, Abramov and Bronstein);

Related Works

- GCRD and LCLM of Ore polynomials (Li, Li and Nemes):
  - normal forms of 2×1 Ore polynomial matrix;
  - subresultant (fraction-free) and modular algorithms;
- EG elimination and improvements (Abramov, Abramov and Bronstein);
- many works on polynomial matrices.

Main Tools

- Order basis

Main Tools

- Order basis
- Striped Krylov matrix

Main Tools

- Order basis
- Striped Krylov matrix
- Equivalence of Gaussian elimination and polynomial operations

Main Tools

- Order basis
- Striped Krylov matrix
- Equivalence of Gaussian elimination and polynomial operations
- Fraction-free Gaussian elimination

Main Tools

- Order basis
- Striped Krylov matrix
- Equivalence of Gaussian elimination and polynomial operations
- Fraction-free Gaussian elimination
- Modular algorithm

Order Basis

We have an elimination problem.

We want the module of solutions $\mathbf{P}(X) = [P_1(X) \cdots P_m(X)]$ of order $\vec{\omega}$ such that

$$P_1(X) \cdot \mathbf{F}_{1,\cdot}(X) + \cdots P_m(X) \cdot \mathbf{F}_{m,\cdot}(X) = \mathbf{R}(X) \cdot X^{\vec{\omega}}$$

where $\mathbf{F}_{i,\cdot}(X)$ is the i-th row of $\mathbf{F}(Z)$, and $\mathbf{R}(X)$ is a residual.

Order Basis

We have an elimination problem.

We want the module of solutions $\mathbf{P}(X) = [P_1(X) \cdots P_m(X)]$ of order $\vec{\omega}$ such that

$$P_1(X) \cdot \mathbf{F}_{1,\cdot}(X) + \cdots P_m(X) \cdot \mathbf{F}_{m,\cdot}(X) = \mathbf{R}(X) \cdot X^{\vec{\omega}}$$

where $\mathbf{F}_{i,\cdot}(X)$ is the i-th row of $\mathbf{F}(Z)$, and $\mathbf{R}(X)$ is a residual.

- A basis of the module: order basis of order $\vec{\omega}$

Order Basis

We have an elimination problem.

We want the module of solutions $\mathbf{P}(X) = [P_1(X) \cdots P_m(X)]$ of order $\vec{\omega}$ such that

$$P_1(X) \cdot \mathbf{F}_{1,\cdot}(X) + \cdots P_m(X) \cdot \mathbf{F}_{m,\cdot}(X) = \mathbf{R}(X) \cdot X^{\vec{\omega}}$$

where $\mathbf{F}_{i,\cdot}(X)$ is the i-th row of $\mathbf{F}(Z)$, and $\mathbf{R}(X)$ is a residual.

- A basis of the module: order basis of order $\vec{\omega}$
- An order basis represents row operations on $\mathbf{F}(X)$ to eliminate low-order terms.

Order Basis

We have an elimination problem.

We want the module of solutions $\mathbf{P}(X) = [P_1(X) \cdots P_m(X)]$ of order $\vec{\omega}$ such that

$$P_1(X) \cdot \mathbf{F}_{1,\cdot}(X) + \cdots P_m(X) \cdot \mathbf{F}_{m,\cdot}(X) = \mathbf{R}(X) \cdot X^{\vec{\omega}}$$

where $\mathbf{F}_{i,\cdot}(X)$ is the i-th row of $\mathbf{F}(Z)$, and $\mathbf{R}(X)$ is a residual.

- A basis of the module: order basis of order $\vec{\omega}$
- An order basis represents row operations on $\mathbf{F}(X)$ to eliminate low-order terms.
- An order basis of a particular order and row degree is unique up to a constant.

Striped Krylov Matrix

- Given a degree bound $\vec{\mu}$ for the order basis $\Rightarrow$ system of linear equations for the order basis.

Striped Krylov Matrix

- Given a degree bound $\vec{\mu}$ for the order basis $\Rightarrow$ system of linear equations for the order basis.
- Coefficient matrix is structured, called a striped Krylov matrix.

$$P(\vec{\mu}, \vec{\omega}) \cdot K(\vec{\mu}, \vec{\omega}) = \mathbf{0}$$

Expand

Striped Krylov Matrix

- Given a degree bound $\vec{\mu}$ for the order basis $\Rightarrow$ system of linear equations for the order basis.
- Coefficient matrix is structured, called a striped Krylov matrix.

$$P(\vec{\mu}, \vec{\omega}) \cdot K(\vec{\mu}, \vec{\omega}) = \mathbf{0}$$

Expand

- It is a generalization of the Sylvester matrix.

Striped Krylov Matrix

- Given a degree bound $\vec{\mu}$ for the order basis $\Rightarrow$ system of linear equations for the order basis.
- Coefficient matrix is structured, called a striped Krylov matrix.

$$P(\vec{\mu}, \vec{\omega}) \cdot K(\vec{\mu}, \vec{\omega}) = \mathbf{0}$$

Expand

- It is a generalization of the Sylvester matrix.
- The entries in the matrix are commutative—traditional linear algebra applies.

Striped Krylov Matrix

- Given a degree bound $\vec{\mu}$ for the order basis $\Rightarrow$ system of linear equations for the order basis.
- Coefficient matrix is structured, called a striped Krylov matrix.

$$P(\vec{\mu}, \vec{\omega}) \cdot K(\vec{\mu}, \vec{\omega}) = \mathbf{0}$$

Expand

- It is a generalization of the Sylvester matrix.
- The entries in the matrix are commutative—traditional linear algebra applies.
- In general, we do not know the degree bound a priori.

Fraction-free Order Basis Algorithm (FFreduce)

- Compute a sequence of order bases of increasing order and degrees:

order	$\Rightarrow$	number of columns eliminated
column degree	$\Rightarrow$	number of times a row has been used as pivot

Fraction-free Order Basis Algorithm (FFreduce)

- Compute a sequence of order bases of increasing order and degrees:

order	$\Rightarrow$	number of columns eliminated
column degree	$\Rightarrow$	number of times a row has been used as pivot

- Matrix structure is exploited by working with only one row each stripe.

Fraction-free Order Basis Algorithm (FFreduce)

- Compute a sequence of order bases of increasing order and degrees:

order	$\Rightarrow$	number of columns eliminated
column degree	$\Rightarrow$	number of times a row has been used as pivot

- Matrix structure is exploited by working with only one row each stripe.
- The elimination is done via a fraction-free recurrence.

Fraction-free Recurrence

Let $\mathbf{M}(X)$ be an order basis of order $\vec{\omega}$ and column degree $\vec{\mu}$.

Let r_j be the term of residual to be eliminated.

Let π (the pivot) be the smallest index with $r_\pi \neq 0$ and $\vec{\mu}_\pi = \min_j\{\vec{\mu}_j : r_j \neq 0\}$.

$$\widetilde{\mathbf{M}}(X)^{\ell,\cdot} = \left(r_\pi \cdot \mathbf{M}(X)^{\ell,\cdot} - r_\ell \cdot \mathbf{M}(X)^{\pi,\cdot}\right)/p_\pi \quad \text{for } l \neq \pi,$$

$$\widetilde{\mathbf{M}}(X)^{\pi,\cdot} = \left((r_\pi \cdot X - \delta(r_\pi)) \cdot \mathbf{M}(X)^{\pi,\cdot} - \sum_{\ell \neq \pi} \sigma(p_\ell) \cdot \widetilde{\mathbf{M}}(X)^{\ell,\cdot}\right)/\sigma(p_\pi),$$

where $p_j = \mathit{coefficient}(\mathbf{M}(X)^{\pi,j}, X^{\vec{\mu}_j+\delta_{\pi,j}-1})$.

Termination

- Order basis and residual of order $(mN+1)n \cdot (1, \ldots, 1)$ gives:
  - rank $\mathbf{F}(X)$
  - basis of left nullspace of $\mathbf{F}(Z)$

Termination

- Order basis and residual of order $(mN + 1)n \cdot (1, \ldots, 1)$ gives:
  - rank $\mathbf{F}(X)$
  - basis of left nullspace of $\mathbf{F}(Z)$
- For shift polynomials:
  - reverse the coefficients
  - eliminate until the trailing coefficient R_0 has full rank (row-reduced form) or is triangular (weak Popov form).

Modular Algorithm

- Design a modular version to improve performance.

Modular Algorithm

- Design a modular version to improve performance.
- Modular reductions:

$$\mathbb{Z}[t][X;\sigma,\delta] \to \mathbb{Z}_p[t][X;\sigma,\delta] \to \mathbb{Z}_p[X;\sigma,\delta]$$

Modular Algorithm

- Design a modular version to improve performance.
- Modular reductions:

$$\mathbb{Z}[t][X;\sigma,\delta] \to \mathbb{Z}_p[t][X;\sigma,\delta] \to \mathbb{Z}_p[X;\sigma,\delta]$$

- Three traditional issues:
  - definition and detection of unlucky homomorphisms
  - normalization
  - termination

Modular Algorithm

- Design a modular version to improve performance.
- Modular reductions:

$$\mathbb{Z}[t][X;\sigma,\delta] \to \mathbb{Z}_p[t][X;\sigma,\delta] \to \mathbb{Z}_p[X;\sigma,\delta]$$

- Three traditional issues:
  - definition and detection of unlucky homomorphisms
  - normalization
  - termination
- We wish to have an output-sensitive algorithm:
  - number of homomorphisms depends on the size of results
  - no need to verify the results by trial division/multiplication

Modular Algorithm

- These issues have been resolved for polynomial matrices (Cheng and Labahn).

Modular Algorithm

- These issues have been resolved for polynomial matrices (Cheng and Labahn).
- $\mathbb{Z}[t][X;\sigma,\delta] \to \mathbb{Z}_p[t][X;\sigma,\delta]$: similar to polynomial matrix case.

Modular Algorithm

- These issues have been resolved for polynomial matrices (Cheng and Labahn).
- $\mathbb{Z}[t][X;\sigma,\delta] \rightarrow \mathbb{Z}_p[t][X;\sigma,\delta]$: similar to polynomial matrix case.
- The same approach does not work for $\mathbb{Z}_p[t][X;\sigma,\delta] \rightarrow \mathbb{Z}_p[X;\sigma,\delta]$:

Modular Algorithm

- These issues have been resolved for polynomial matrices (Cheng and Labahn).
- $\mathbb{Z}[t][X;\sigma,\delta] \to \mathbb{Z}_p[t][X;\sigma,\delta]$: similar to polynomial matrix case.
- The same approach does not work for $\mathbb{Z}_p[t][X;\sigma,\delta] \to \mathbb{Z}_p[X;\sigma,\delta]$:
  - evaluation map $t \leftarrow \alpha$ is not an Ore ring homomorphism.

Reduction to $\mathbb{Z}_p[t][X; \sigma, \delta]$

- Compute order basis and residual in $\mathbb{Z}_p[t][X; \sigma, \delta]$.

Reduction to $\mathbb{Z}_p[t][X;\sigma,\delta]$

- Compute order basis and residual in $\mathbb{Z}_p[t][X;\sigma,\delta]$.
- Normalization: compute the image of the same order basis and residual as FFreduce.

Reduction to $\mathbb{Z}_p[t][X;\sigma,\delta]$

- Compute order basis and residual in $\mathbb{Z}_p[t][X;\sigma,\delta]$.
- Normalization: compute the image of the same order basis and residual as FFreduce.
- Chinese remaindering used to reconstruct the result.

- Lucky homomorphism $\Leftrightarrow$ the order and degree of order basis computed is correct

Lucky Homomorphisms

- Lucky homomorphism $\Leftrightarrow$ the order and degree of order basis computed is correct
- Sequence of pivots (the computation path) is useful:

Endpoint of path = degree of order basis

Lucky Homomorphisms

- Lucky homomorphism $\Leftrightarrow$ the order and degree of order basis computed is correct
- Sequence of pivots (the computation path) is useful:

  Endpoint of path = degree of order basis

- i.e. lucky homomorphism $\Leftrightarrow$ same endpoint as FFreduce.

Lucky Homomorphisms

- Lucky homomorphism $\Leftrightarrow$ the order and degree of order basis computed is correct
- Sequence of pivots (the computation path) is useful:

  Endpoint of path = degree of order basis

- i.e. lucky homomorphism $\Leftrightarrow$ same endpoint as FFreduce.
- Homomorphisms with different endpoints: the one that is further away from a “normal path” is unlucky.

Size of Input

For the remainder of this talk, we assume that:

$$\deg_t\left(c_k\left(X^\ell \cdot \mathbf{F}(X)_{i,j}\right)\right) \leq T$$
$$\left\|c_k\left(X^\ell \cdot \mathbf{F}(X)_{i,j}\right)\right\|_\infty \leq \kappa$$

for $1 \leq i \leq m$, $1 \leq j \leq n$, $0 \leq k < mN + 1$, and $0 \leq \ell \leq mN + 1$ where $N = \deg \mathbf{F}(X)$.

Termination

- We can apply Hadamard bound on coefficients (can be very pessimistic).

Termination

- We can apply Hadamard bound on coefficients (can be very pessimistic).
- Suppose $p_1 < p_2 < \cdots$, and τ is such that

$$((mN+1)n)\kappa(T+1) \leq p_1 \cdots p_\tau.$$

Termination

- We can apply Hadamard bound on coefficients (can be very pessimistic).
- Suppose $p_1 < p_2 < \cdots$, and τ is such that

$$((mN+1)n)\kappa(T+1) \leq p_1 \cdots p_\tau.$$

- Reconstructed results have not changed for τ additional primes $\Rightarrow$ reconstructed results are correct

Termination

- We can apply Hadamard bound on coefficients (can be very pessimistic).
- Suppose $p_1 < p_2 < \cdots$, and τ is such that

$$((mN+1)n)\kappa(T+1) \leq p_1 \cdots p_\tau.$$

- Reconstructed results have not changed for τ additional primes $\Rightarrow$ reconstructed results are correct
- τ is small in many cases (e.g. 1).

Reduction to $\mathbb{Z}_p$

- Ore rings with coefficients in $\mathbb{Z}_p$ must be commutative.

Reduction to $\mathbb{Z}_p$

- Ore rings with coefficients in $\mathbb{Z}_p$ must be commutative.
- Evaluation homomorphisms $t \leftarrow \alpha$ are not an Ore ring homomorphism in general.

Reduction to $\mathbb{Z}_p$

- Ore rings with coefficients in $\mathbb{Z}_p$ must be commutative.
- Evaluation homomorphisms $t \leftarrow \alpha$ are not an Ore ring homomorphism in general.
- We cannot simply apply the reductions and reconstruct the results as before.

Previous Work

- Modular algorithms to compute GCRDs of Ore polynomials (Li and Nemes):

Previous Work

- Modular algorithms to compute GCRDs of Ore polynomials (Li and Nemes):
    - Gaussian elimination on Sylvester matrix

Previous Work

- Modular algorithms to compute GCRDs of Ore polynomials (Li and Nemes):
  - Gaussian elimination on Sylvester matrix
  - use modular algorithm on Sylvester matrix

Previous Work

- Modular algorithms to compute GCRDs of Ore polynomials (Li and Nemes):
  - Gaussian elimination on Sylvester matrix
  - use modular algorithm on Sylvester matrix
- This is not straightforward for matrices of Ore polynomials:

Previous Work

- Modular algorithms to compute GCRDs of Ore polynomials (Li and Nemes):
  - Gaussian elimination on Sylvester matrix
  - use modular algorithm on Sylvester matrix
- This is not straightforward for matrices of Ore polynomials:
  - the computation path (degree bound) is not known a priori

Previous Work

- Modular algorithms to compute GCRDs of Ore polynomials (Li and Nemes):
  - Gaussian elimination on Sylvester matrix
  - use modular algorithm on Sylvester matrix
- This is not straightforward for matrices of Ore polynomials:
  - the computation path (degree bound) is not known a priori
  - it is not known a priori which striped Krylov matrix is needed

Our Modular Algorithm

We interleave the construction of the striped Krylov matrix with elimination steps:

- when an elimination step is performed, a new row is added (after evaluation homomorphism is applied)
- the added row is reduced with respect to **all** previous pivot rows

Our Modular Algorithm

- Normalization: same as the case $\mathbb{Z}_p[t][X;\sigma,\delta]$
- Lucky homomorphisms: similar as $\mathbb{Z}_p[t][X;\sigma,\delta]$
- Termination:

  results unchanged for T additional homomorphisms

  $\Rightarrow$ reconstructed results are correct

Example

$$K(\vec{\mu},\vec{\omega}) = \left[\begin{array}{cc|cc|cc} 6t^2 & 2 & 3t & -1 & 2 & 1 \\ 12t & 0 & 6t^2+3 & 2 & 3t & -1 \\ 12 & 0 & 24t & 0 & 6t^2+6 & 2 \\ \hline 3t^3 & t & t-1 & 3t & 0 & 0 \\ 9t^2 & 1 & 3t^3+1 & t+3 & t-1 & 3t \\ 18t & 0 & 18t^2 & 2 & 3t^3+2 & t+6 \end{array}\right].$$

- The substitution $t \leftarrow 0$ gives a completely different pivot choice (third row).

In general, pivot rows and columns correct at the end
$\Rightarrow$ the evaluation is lucky

Popov Form

- Order basis computation eliminates low-order terms.

Popov Form

- Order basis computation eliminates low-order terms.
- For shift polynomials, leading term can be eliminated by reversing coefficients.

Popov Form

- Order basis computation eliminates low-order terms.
- For shift polynomials, leading term can be eliminated by reversing coefficients.
- In general, this is not possible.

Popov Form

- Order basis computation eliminates low-order terms.
- For shift polynomials, leading term can be eliminated by reversing coefficients.
- In general, this is not possible.
- Popov form cannot be computed directly with order basis even for shift polynomials.

Popov Form

- We compute the left nullspace of the matrix:

$$\begin{bmatrix} \mathbf{F}(X) \cdot X^{b} \\ -I \end{bmatrix}$$

Popov Form

- We compute the left nullspace of the matrix:

$$\begin{bmatrix} \mathbf{F}(X) \cdot X^b \\ -I \end{bmatrix}$$

- The left nullspace can be partitioned as:

$$\mathbf{M}(X) = \begin{bmatrix} \mathbf{U}(X) & \mathbf{T}(X) \cdot X^b \end{bmatrix}$$

so

$$\mathbf{U}(X) \cdot \mathbf{F}(X) \cdot Z^b = \mathbf{T}(X) \cdot X^b$$

Popov Form

- We compute the left nullspace of the matrix:

$$\begin{bmatrix} \mathbf{F}(X) \cdot X^b \\ -I \end{bmatrix}$$

- The left nullspace can be partitioned as:

$$\mathbf{M}(X) = \begin{bmatrix} \mathbf{U}(X) & \mathbf{T}(X) \cdot X^b \end{bmatrix}$$

so

$$\mathbf{U}(X) \cdot \mathbf{F}(X) \cdot Z^b = \mathbf{T}(X) \cdot X^b$$

- If $b > \deg \mathbf{U}(\mathbf{X})$, then the leading row coefficient of $\mathbf{M}(X)$ is the leading row coefficient of $\mathbf{T}(X)$.

Popov Form

- We compute the left nullspace of the matrix:

$$\begin{bmatrix} \mathbf{F}(X) \cdot X^b \\ -I \end{bmatrix}$$

- The left nullspace can be partitioned as:

$$\mathbf{M}(X) = \begin{bmatrix} \mathbf{U}(X) & \mathbf{T}(X) \cdot X^b \end{bmatrix}$$

so

$$\mathbf{U}(X) \cdot \mathbf{F}(X) \cdot Z^b = \mathbf{T}(X) \cdot X^b$$

- If $b > \deg \mathbf{U}(\mathbf{X})$, then the leading row coefficient of $\mathbf{M}(X)$ is the leading row coefficient of $\mathbf{T}(X)$.
- $\mathbf{M}(Z)$ in Popov form $\Leftrightarrow$ $\mathbf{T}(Z)$ in Popov form.

Popov Form

- We compute the left nullspace of the matrix:

$$\begin{bmatrix} \mathbf{F}(X) \cdot X^b \\ -I \end{bmatrix}$$

- The left nullspace can be partitioned as:

$$\mathbf{M}(X) = \begin{bmatrix} \mathbf{U}(X) & \mathbf{T}(X) \cdot X^b \end{bmatrix}$$

so

$$\mathbf{U}(X) \cdot \mathbf{F}(X) \cdot Z^b = \mathbf{T}(X) \cdot X^b$$

- If $b > \deg \mathbf{U}(\mathbf{X})$, then the leading row coefficient of $\mathbf{M}(X)$ is the leading row coefficient of $\mathbf{T}(X)$.
- $\mathbf{M}(Z)$ in Popov form $\Leftrightarrow$ $\mathbf{T}(Z)$ in Popov form.
- Old idea, but proofs do not work when matrix entries are not commutative.

Popov Form

Let $\vec{\mu} = \text{rdeg}\, \mathbf{F}(X)$ and $b > |\vec{\mu}| - \min_j\{\mu_j\}$.

Suppose that $[\mathbf{U}(X) \;\; \mathbf{R}(X)]$ is a minimal polynomial basis in Popov form of the left nullspace of $\begin{bmatrix} \mathbf{F}(X) \cdot X^b \\ -\mathbf{I} \end{bmatrix}$.

Let $\mathbf{T}(X) = \mathbf{R}(X) \cdot X^{-b}$.

1. $\mathbf{U}(X)$ is unimodular;
2. $\mathbf{T}(X) = \mathbf{U}(X) \cdot \mathbf{F}(X)$ is an Ore polynomial matrix in Popov form.

Final Remarks

- Although the Ore polynomials are not commutative, the coefficients are.

Final Remarks

- Although the Ore polynomials are not commutative, the coefficients are.
- Elimination is formulated as linear systems of equations on the coefficients.

Final Remarks

- Although the Ore polynomials are not commutative, the coefficients are.
- Elimination is formulated as linear systems of equations on the coefficients.
- This allows traditional linear algebra techniques to be used to control coefficient growth.

Final Remarks

- Although the Ore polynomials are not commutative, the coefficients are.
- Elimination is formulated as linear systems of equations on the coefficients.
- This allows traditional linear algebra techniques to be used to control coefficient growth.
- Polynomial arithmetic is used to take advantange of the matrix structure.

$$\begin{array}{ccccc} & X^0 & \cdots & X^{n_k} & \\ \Big[\,\cdots\,\Big| & p_k^{(0)} & \cdots & p_k^{(n_k)} & \Big|\,\cdots\,\Big] \end{array} \begin{bmatrix} & \vdots & \\ \hline \cdots & X^0 \cdot F_{k,\cdot}(X) & \cdots \\ & \vdots & \\ \cdots & X^{n_k} \cdot F_{k,\cdot}(X) & \cdots \\ \hline & \vdots & \end{bmatrix} = \mathbf{0}$$

$$\begin{array}{cc}
\begin{array}{ccccc} & X^0 & \cdots & X^{n_k} & \end{array} & \begin{array}{ccc} X^0 & \cdots & X^{\vec{\omega}-\vec{e}} \end{array} \\
\left[\begin{array}{c|ccc|c} \cdots & p_k^{(0)} & \cdots & p_k^{(n_k)} & \cdots \end{array}\right] & \left[\begin{array}{c|c|c} & \vdots & \\ \hline \cdots & X^0 \cdot F_{k,\cdot}(X) & \cdots \\ & \vdots & \\ \cdots & X^{n_k} \cdot F_{k,\cdot}(X) & \cdots \\ \hline & \vdots & \end{array}\right]
\end{array} = \mathbf{0}$$

Example

Let $\vec{\mu} = (2, 2)$, $\vec{\omega} = (3, 3)$, and

$$\mathbf{F}(X) = \begin{bmatrix} 2X^2 + 3tX + 6t^2 & X^2 - X + 2 \\ (t-1)X + 3t^3 & 3tX + t \end{bmatrix} \in \mathbb{Z}[t][X; \sigma, \delta]^{2\times 2},$$

with $\sigma(a(t)) = a(t)$ and $\delta(a(t)) = a'(t)$.

$$\begin{array}{cc} & \begin{array}{ccc} \qquad X^0 \qquad & \qquad X^1 \qquad & \qquad X^2 \qquad \end{array} \\ K(\vec{\mu}, \vec{\omega}) = & \left[\begin{array}{cc|cc|cc} 6t^2 & 2 & 3t & -1 & 2 & 1 \\ 12t & 0 & 6t^2+3 & 2 & 3t & -1 \\ 12 & 0 & 24t & 0 & 6t^2+6 & 2 \\ \hline 3t^3 & t & t-1 & 3t & 0 & 0 \\ 9t^2 & 1 & 3t^3+1 & t+3 & t-1 & 3t \\ 18t & 0 & 18t^2 & 2 & 3t^3+2 & t+6 \end{array}\right]. \end{array}$$

Return