Fraction-free Computation of Simultaneous Padé Approximants

George Labahn

Symbolic Computation Group
Cheriton School of Computer Science
University of Waterloo, Canada

Joint work with Bernd Beckermann (Lille, France)

Vector Rational Reconstruction

Classical technique in computer algebra : rational reconstruction.

- scalar case
- vector case

Technique

$$\mathbb{K}^n(z) \to \mathbb{K}^n[[z]] \text{ - solve problem - } \mathbb{K}^n[[z]] \to \mathbb{K}^n(z)$$

For example :

Solve : $A \cdot \vec{x} = \vec{b}$ with $A \in \mathbb{K}^{n \times n}(z)$, $\vec{x}, \vec{b} \in \mathbb{K}^n(z)$.

Olesh and Storjohann, The vector rational function reconstruction problem (2007)

Vector Rational Reconstruction

Classical technique in computer algebra : rational reconstruction.

- scalar case
- vector case

Technique

$$\mathbb{K}^n(z) \rightarrow \mathbb{K}^n[[z]] \text{ - solve problem - } \mathbb{K}^n[[z]] \rightarrow \mathbb{K}^n(z)$$

For example :

Solve : $A \cdot \vec{x} = \vec{b}$ with $A \in \mathbb{K}^{n \times n}(z)$, $\vec{x}, \vec{b} \in \mathbb{K}^n(z)$.

Olesh and Storjohann, The vector rational function reconstruction problem (2007)

Vector Rational Reconstruction

Historically such a concept dates at least to 1800's.

- C. Hermite : 1873
  - Transcendence of e
- H. Padé (a student of Hermite) : 1893
  - formalized concept of Simultaneous-Padé approximant and Hermite-Padé approximant
  - invented notion of Padé table
- K. Mahler : 1920-1970 and others

This Talk

- Find solutions to a *particular* vector rational reconstruction problem
- Solve problem using *fraction-free arithmetic*
  - coefficients come from integral domain e.g. $Q[a_1, \ldots, a_n]$
  - we avoid arithmetic in quotient domain (i.e. avoids fractions)
  - however coefficient growth is controlled
- Tools used imply solutions are in fact is very general

Simultaneous Padé Approximants

Given power series $A_1(z), \ldots, A_m(z)$ and integer N

Find polynomials $U_1(z), ..., U_m(z)$ and $V(z)$ with :

$$V(z)[A_1(z), \ldots, A_m(z)] \equiv [U_1(z), \ldots, U_m(z)] \text{ mod } z^{N+1}$$

- Degree bounds for $U_1(z), \ldots, U_m(z)$, $V(z)$

Simultaneous Padé Approximants

Given power series $A_1(z), \ldots, A_m(z)$ and integer N

Find polynomials $U_1(z), ..., U_m(z)$ and $V(z)$ with :

$$V(z)[A_1(z), \ldots, A_m(z)] \equiv [U_1(z), \ldots, U_m(z)] \text{ mod } z^{N+1}$$

Models

$$[A_1(z), \ldots, A_m(z)] = [\frac{U_1(z)}{V(z)}, \ldots, \frac{U_m(z)}{V(z)}] + O(z^{N+1})$$

- Degree bounds for $U_1(z), \ldots, U_m(z)$, $V(z)$

Simultaneous Padé Approximants

Given power series $A_1(z), \ldots, A_m(z)$ and integers N, $n_0, \ldots, n_m$

Find polynomials $U_1(z)$, ..., $U_m(z)$ and $V(z)$ with :

$$V(z)[A_1(z), \ldots, A_m(z)] \equiv [U_1(z), \ldots, U_m(z)] \bmod z^{N+1}$$

- Degree bounds for $U_1(z), \ldots, U_m(z), V(z)$

$$\deg(U_i) \leq N - n_i \qquad \deg(V) \leq N - n_0,$$

$N = n_0 + \cdots + n_m$.

How to Compute Simultaneous Padé Approximants

(1) linear system of equations for coefficients of $U_i(z)$, $V(z)$

- system has ‘special’ structured coefficient matrix
- fast algorithms take advantage of ‘special’ structure

(2) represent as a vector Hermite-Padé problem

- find *order basis* for problem
- fast algorithms
- order bases find *all* solutions of these approx problems

(3) todays : solve problem via order bases using duality.

Note : all methods can be done via fraction-free arithmetic

Linear System : $[A(z), B(z), C(z)]$

Set up and solve via linear system for unknown coefficients

$$(a_0+a_1z+\cdots)(v_0+v_1z+v_2z^2+v_3z^3)-(u_0+u_1z+u_2z^2) = O(z^4)$$

$$(b_0+b_1z+\cdots)(v_0+v_1z+v_2z^2+v_3z^3)-(\hat{u}_0+\hat{u}_1z+\hat{u}_2z^2) = O(z^4)$$

$$(c_0+c_1z+\cdots)(v_0+v_1z+v_2z^2+v_3z^3)-(u^*_0+u^*_1z+u^*_2z^2) = O(z^4)$$

$$\left[\begin{array}{cccc|ccc|ccc|ccc}
a_0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_1 & a_0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_2 & a_1 & a_0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_3 & a_2 & a_1 & a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
b_0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
b_1 & b_0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
b_2 & b_1 & b_0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
b_3 & b_2 & b_1 & b_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
c_1 & c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
c_2 & c_1 & c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
c_3 & c_2 & c_1 & c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\left[\begin{array}{c}
v_0 \\ v_1 \\ v_2 \\ v_3 \\ \hline u_0 \\ u_1 \\ u_2 \\ \hline \hat{u}_0 \\ \hat{u}_1 \\ \hat{u}_2 \\ \hline u^*_0 \\ u^*_1 \\ u^*_2
\end{array}\right] = 0.$$

Notice special structure of coefficient matrix.

Example : $[e^z,\ e^{2z},\ e^{3z}]$

$$
\begin{aligned}
e^z &= \frac{-2\,z^2+6\,z-6}{6\,z^3-11\,z^2+12\,z-6}+O(z^4)\\
e^{2z} &= \frac{z^2-6}{6\,z^3-11\,z^2+12\,z-6}+O(z^4)\\
e^{3z} &= \frac{-2z^2-6\,z-6}{6\,z^3-11\,z^2+12\,z-6}+O(z^4)
\end{aligned}
$$

Here $(n_0, n_1, n_2, n_3) = (0, 1, 1, 1)$ so $N = 3$.

Degree bounds : denom $= 3$, numers $= 2,\ 2,\ 2$, order $= 4$.

Hermite-Padé Approximation

Given power series $A_1(z), \ldots, A_m(z)$, N and $n_1, \ldots, n_m$

Find polynomials $U_1(z)$, ..., $U_m(z)$ with :

$$A_1(z)U_1(z) + \cdots + A_m(z)U_m(z) \equiv 0 \bmod z^{N+1}$$

- Degree bounds $U_i(z) \leq n_i - 1$ and $N = n_1 + \cdots + n_m + 1$
- Often written as $\mathbf{A}(z) \cdot \mathbf{U}(z) = O(z^{N+1})$
- Vector versions : $A_i(z) \in \mathbb{K}^m[[z]]$, $\vec{\sigma}$ vector of orders

$$\mathbf{A}(z) \cdot \mathbf{U}(z) = O(z^{\vec{\sigma}})$$

- Most general $\mathbf{A}(z) \in \mathbb{K}^{m \times m}[[z]]$ (matrix versions)

Simultaneous-Padé as Vector Hermite-Padé

$$
\begin{aligned}
A(z)V(z) - U_1(z) &= O(z^4) \\
B(z)V(z) - U_2(z) &= O(z^4) \\
C(z)V(z) - U_3(z) &= O(z^4)
\end{aligned}
$$

same as

$$
\begin{bmatrix} A(z) \\ B(z) \\ C(z) \end{bmatrix} V(z) + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} U_1(z) + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} U_2(z) + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} U_3(z) = 0(z^4)
$$

Order Basis

Input : Matrix of power series $\mathbf{A}(z) \in \mathbb{K}^{m \times m}[z]$ and vector $\vec{\sigma}$

Order problems : $\mathbf{A}(z) \cdot \mathbf{U}(z) = O(z^{\vec{\sigma}})$ with $\mathbf{V}(z) \in \mathbb{K}^{m}[z]$

Order Basis : Matrix polynomial $\mathbf{M}(z) \in \mathbb{K}^{m \times m}[z]$

$$\mathbf{A}(z) \cdot \mathbf{M}(z) = O(z^{\vec{\sigma}})$$

- All solutions $\mathbf{V}(z)$ of order problem are a combination of columns of $\mathbf{M}(z)$ (as a module).

$$\mathbf{V}(z) = \alpha_1(z)\mathbf{M}^{(1)}(z) + \cdots + \alpha_m(z)\mathbf{M}^{(m)}(z)$$

- degree constraints become degree constraints on $\alpha_i(z)$.

Order Basis

Input : Matrix of power series $\mathbf{A}(z) \in \mathbb{K}^{m \times m}[z]$ and vector $\vec{\sigma}$

Order problems : $\mathbf{A}(z) \cdot \mathbf{U}(z) = O(z^{\vec{\sigma}})$ with $\mathbf{V}(z) \in \mathbb{K}^{m}[z]$

Order Basis : Matrix polynomial $\mathbf{M}(z) \in \mathbb{K}^{m \times m}[z]$

$$\mathbf{A}(z) \cdot \mathbf{M}(z) = O(z^{\vec{\sigma}})$$

- All solutions $\mathbf{V}(z)$ of order problem are a combination of columns of $\mathbf{M}(z)$ (as a module).

$$\mathbf{V}(z) = \alpha_1(z)\mathbf{M}^{(1)}(z) + \cdots + \alpha_m(z)\mathbf{M}^{(m)}(z)$$

- degree constraints become degree constraints on $\alpha_i(z)$.

Computation of Order Bases

Algorithms :

- Coefficient growth not an issue.
  B-L 1994, Giorgi-Jeannerod-Villard 2003, Storjohann 2006, Zhou-L 2009
- Fraction-Free computation. FFFG (B-L 2000)

FFFG (B-L 2000) Fraction-Free Computation

- Computes sequence

$$\mathbf{I} = \mathbf{M}^{(0)}(z, \vec{v}^{(0)}), \mathbf{M}^{(1)}(z, \vec{v}^{(1)}), \ldots, \mathbf{M}^{(\sigma)}(z, \vec{v}^{(\sigma)})$$

  of order bases and degrees $\vec{v}^{(i)}$.

  - use special pivot column of $\mathbf{M}^{(i)}(z, \vec{v}^{(i)})$ to increase orders of other columns
  - do special column operations to normalize next order basis

Computation of Order Bases

Algorithms :

- Coefficient growth not an issue.
  B-L 1994, Giorgi-Jeannerod-Villard 2003, Storjohann 2006, Zhou-L 2009
- Fraction-Free computation. FFFG (B-L 2000)

FFFG (B-L 2000) Fraction-Free Computation

- Computes sequence

$$\mathbf{I} = \mathbf{M}^{(0)}(z, \vec{v}^{(0)}), \mathbf{M}^{(1)}(z, \vec{v}^{(1)}), \ldots, \mathbf{M}^{(\sigma)}(z, \vec{v}^{(\sigma)})$$

  of order bases and degrees $\vec{v}^{(i)}$.

  - use special pivot column of $\mathbf{M}^{(i)}(z, \vec{v}^{(i)})$ to increase orders of other columns
  - do special column operations to normalize next order basis

Cost

- Without coefficient growth - see next talk
- With coefficient growth: if input size $O(\kappa)$ then :
  - Fraction-Free Gaussian Elimination (FFGE) :
    - Bit complexity of operations: $O(\kappa^2 m^6 N^5)$
  - B-L [SIMAX 2000] :
    - Bit complexity of operations: $O(\kappa^2 m^5 N^4)$
    - Size of objects : $O(\kappa m N)$
  - Today
    - Bit complexity of operations: $O(\kappa^2 m^2 N^4)$
    - Size of objects : $O(\kappa m N)$

Cost

- Without coefficient growth - see next talk
- With coefficient growth: if input size $O(\kappa)$ then :
  - Fraction-Free Gaussian Elimination (FFGE) :
    - Bit complexity of operations: $O(\kappa^2 m^6 N^5)$
  - B-L [SIMAX 2000] :
    - Bit complexity of operations: $O(\kappa^2 m^5 N^4)$
    - Size of objects : $O(\kappa m N)$
  - Today
    - Bit complexity of operations: $O(\kappa^2 m^2 N^4)$
    - Size of objects : $O(\kappa m N)$

Duality for Order Bases

For any order problem $\mathbf{A}(z)$, $\vec{\sigma}$, $\vec{n}$

$$\mathbf{A}(z) \cdot \mathbf{M}(z) = O(z^{\vec{\sigma}})$$

Take adjoints and transposes. Obtain order problem $\mathbf{A}^*(z)$, $|\vec{\sigma}| - \vec{\sigma}$, $|\vec{n}| - \vec{n}$

$$\mathbf{A}^*(z) \cdot \mathbf{M}^*(z) = O(z^{|\sigma|\vec{e} - \vec{\sigma}})$$

Problems are dual to each other.

- e.g. left and right (matrix) Padé approximants are dual
- duality used in special inversion formulae (L [LAA - 1992])

Duality

Hermite-Padé and Simultaneous-Padé order bases are duals to each other (B-L [JCAM 1997]) .

$$\mathbf{A}(z)\mathbf{M}(z) = \begin{bmatrix} A_0(z) & A_1(z) & \cdots & A_m(z) \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix} \begin{bmatrix} m_{00}(z) & \cdots & m_{0,m}(z) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ m_{m0}(z) & \cdots & m_{m,m}(z) \end{bmatrix} = O(z^{(N+1,0,\cdots,0)})$$

$$\mathbf{A}^*(z)\mathbf{M}^*(z) \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -A_1(z) & A_0(z) & & \\ \vdots & & \ddots & \\ -A_m(z) & & & A_0(z) \end{bmatrix} \begin{bmatrix} m^*_{00}(z) & \cdots & m^*_{0,m}(z) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ m^*_{m0}(z) & \cdots & m^*_{m,m}(z) \end{bmatrix} = O(z^{(0,N+1,\cdots,N+1)})$$

where $\mathbf{P}^*(z) = \text{adj}(\mathbf{P}(z))^T = \text{cof}(\mathbf{P}(z))$.

Computation Process

Input : $\mathbf{A}(z)$, $\vec{n}$.

Process for Hermite-Padé : Computes Order bases of type

$$\vec{v}^{(0)}, \vec{v}^{(1)}, \ldots \quad \ldots, \vec{v}^{(N+1)}$$

$$\mathbf{I} = \mathbf{M}(\vec{v}^{(0)}, z) \quad \rightarrow \quad \mathbf{M}(\vec{v}^{(1)}, z) \rightarrow \quad \cdots \quad \rightarrow \mathbf{M}(\vec{v}^{(N+1)}, z)$$

Computation Process

Input : $\mathbf{A}(z)$, $\vec{n}$.

Process for Hermite-Padé : Computes Order bases of type

$$\vec{v}^{(0)}, \vec{v}^{(1)}, \ldots \quad \ldots, \vec{v}^{(N+1)}$$

$$\mathbf{I} = \mathbf{M}(\vec{v}^{(0)}, z) \quad \rightarrow \quad \mathbf{M}(\vec{v}^{(1)}, z) \rightarrow \quad \cdots \quad \rightarrow \mathbf{M}(\vec{v}^{(N+1)}, z)$$

$$\mathbf{I} = \mathbf{M}^{*}(\vec{v}^{(0)}, z) \quad \rightarrow \quad \mathbf{M}^{*}(\vec{v}^{(1)}, z) \rightarrow \quad \cdots \quad \rightarrow \mathbf{M}^{*}(\vec{v}^{(N+1)}, z)$$

Computation Process

Input : $\mathbf{A}(z)$, $\vec{n}$.

Process for Hermite-Padé : Computes Order bases of type

$$\vec{v}^{(0)}, \vec{v}^{(1)}, \ldots \quad \ldots, \vec{v}^{(N+1)}$$

$$\mathbf{I} = \mathbf{M}(\vec{v}^{(0)}, z) \quad \rightarrow \quad \mathbf{M}(\vec{v}^{(1)}, z) \rightarrow \quad \cdots \quad \rightarrow \mathbf{M}(\vec{v}^{(N+1)}, z)$$

$$\uparrow \qquad\qquad \uparrow \qquad\qquad \uparrow$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$\mathbf{I} = \mathbf{M}^*(\vec{v}^{(0)}, z) \quad \rightarrow \quad \mathbf{M}^*(\vec{v}^{(1)}, z) \rightarrow \quad \cdots \quad \rightarrow \mathbf{M}^*(\vec{v}^{(N+1)}, z)$$

Recursion (Hermite-Padé)

$$\mathbf{M}(\vec{v}^{(i)}, z) \cdot \mathbf{A}(z) \cdot \mathbf{B}(z) \quad = \quad \mathbf{M}(\vec{v}^{(i+1)}, z)$$

$$\begin{bmatrix} m_{11}^{(i)}(z) & \cdots m_{1,m}^{(i)}(z) \\ \vdots & \vdots \\ m_{m1}^{(i)}(z) & \cdots m_{m,m}^{(i)}(z) \end{bmatrix} \begin{bmatrix} * & & \\ * & * & \\ & \ddots & \\ * & & * \end{bmatrix} \begin{bmatrix} z & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} m_{11}^{(i+1)}(z) & \cdots m_{1,m}^{(i+1)}(z) \\ \vdots & \vdots \\ m_{m1}^{(i+1)}(z) & \cdots m_{m,m}^{(i+1)}(z) \end{bmatrix}$$

Recursion (Simultaneous-Padé)

Take adjoints and then transposes to get

$$\mathbf{M}^*(\vec{v}^{(i)}, z) \cdot \mathbf{A}^*(z) \cdot \mathbf{B}^*(z) \quad = \quad \mathbf{M}^*(\vec{v}^{(i+1)}, z)$$

$$\begin{bmatrix} \hat{m}_{11}^{(i)}(z) & \cdots \hat{m}_{1,m}^{(i)}(z) \\ \vdots & \vdots \\ \hat{m}_{m1}^{(i)}(z) & \cdots \hat{m}_{m,m}^{(i)}(z) \end{bmatrix} \begin{bmatrix} * & & * \\ 0 & * & * \\ \vdots & & \\ \vdots & * & \\ 0 & * & * \end{bmatrix} \begin{bmatrix} 1 & & & \\ & z & & \\ & & \ddots & \\ & & & z \end{bmatrix} = \begin{bmatrix} \hat{m}_{11}^{(i+1)}(z) & \cdots \hat{m}_{1,m}^{(i+1)}(z) \\ \vdots & \vdots \\ \hat{m}_{m1}^{(i+1)}(z) & \cdots \hat{m}_{m,m}^{(i+1)}(z) \end{bmatrix}$$

The Algorithm

At each iteration for Hermite-Padé

- Increase order of each row using ‘special’ pivot column π
- Increase order of ‘special’ pivot column π
- Normalize order basis to get special shifted *Popov* form

At each iteration for Simultaneous-Padé

- Increase order of ‘special’ row using fraction-free Gaussian elimination on first term of residual
- Increase order of ‘special’ pivot row
- Normalize order basis to get special shifted *Popov* form

The Algorithm

At each iteration for Hermite-Padé

- Increase order of each row using 'special' pivot column π
- Increase order of 'special' pivot column π
- Normalize order basis to get special shifted *Popov* form

At each iteration for Simultaneous-Padé

- Increase order of 'special' row using fraction-free Gaussian elimination on first term of residual
- Increase order of 'special' pivot row
- Normalize order basis to get special shifted *Popov* form

Future Research

- Fraction-Free $\rightarrow$ modular methods
- Duality for alternative order basis algorithm for Hermite-Padé and vector Hermite-Padé approximation problem
- Use alternative order basis algorithm for noncommutative case of Ore matrix polynomials
- Use above algorithm to create faster algorithms for matrix polynomial and matrix Ore normal forms (Popov, Hermite)