Fraction-free Computation of Simultaneous Padé Approximants

George Labahn

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

Joint work with Bernd Beckermann (Lille, France)

Simultaneous Padé Approximants

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

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

- $A_1(z) \approx \frac{U_1(z)}{V(z)}, \quad \ldots \, , \;\; A_m(z) \approx \frac{U_m(z)}{V(z)}.$
- Degree bounds for $U_1(z), \ldots, U_m(z), V(z)$

Simultaneous Padé Approximants (more precise)

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

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

- $A_i(z) \approx \frac{U_i(z)}{V(z)}$ means $A_i(z)V(z) - U_i(z) = \mod z^{N+1}$
- Degree bounds for $U_1(z), \ldots, U_m(z), V(z)$

Simultaneous Padé Approximants (more precise)

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

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

- $A_i(z) \approx \frac{U_i(z)}{V(z)}$ means $A_i(z)V(z) - U_i(z) = z^{N+1}R_i(z)$
- Degree bounds : $\deg(U_i) \leq N - n_i$ $\quad \deg(V) \leq N - n_0$,

  where $N = n_0 + \cdots + n_m$. R_i called residual.

Problem : Compute these *Simultaneous Padé approximants*.

Important fact : Closely related to Hermite-Padé approximants.

Simultaneous Padé Approximants (more precise)

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

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

- $A_i(z) \approx \frac{U_i(z)}{V(z)}$ means $A_i(z)V(z) - U_i(z) = z^{N+1}R_i(z)$
- Degree bounds : $\deg\,(U_i) \leq N - n_i \quad \deg\,(V) \leq N - n_0,$

  where $N = n_0 + \cdots + n_m$. R_i called residual.

Problem : Compute these *Simultaneous Padé approximants*.

Important fact : Closely related to Hermite-Padé approximants.

Where used?

- Vector rational reconstruction problem:

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

  (e.g. linear system solving)
- Transcendence of important numbers : e.g. e, π, etc
- Inversion formulae for structured matrices
  - fast inversion of structured matrices
  - in numeric environment then used for estimating condition numbers
  - central for look ahead numeric procedures
- etc.

Where used?

- Vector rational reconstruction problem:

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

  (e.g. linear system solving)
- Transcendence of important numbers : e.g. e, π, etc
- Inversion formulae for structured matrices
  - fast inversion of structured matrices
  - in numeric environment then used for estimating condition numbers
  - central for look ahead numeric procedures
- etc.

Where used?

- Vector rational reconstruction problem:

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

  (e.g. linear system solving)
- Transcendence of important numbers : e.g. e, π, etc
- Inversion formulae for structured matrices
  - fast inversion of structured matrices
  - in numeric environment then used for estimating condition numbers
  - central for look ahead numeric procedures
- etc.

People

- 1850-1890: Hermite
  - used these (informally) to prove transcendence of e (1873)
- 1890s: H. Padé
  - student of Hermite
  - first systematic study of rational approximation (1893)
  - Padé table
- 1920-1967: K. Mahler
  - a formal study in general case
  - relationships between groups of simultaneous Padé approximants and Hermite-Padé approximants

How to compute : early example from Hermite.

If $f(x) = x^{n-1}(x-1)^n \cdots (x-m)^n$ and $N = mn + n - 1$ then

$$e^{kz}v(z) - u_k(z) = z^{N+1}R_k(z)$$

with

$$\begin{aligned} v(z) &= f(0)z^N + f'(0)z^{N-1} + \cdots + f^{(N)}(0), \\ u_k(z) &= f(k)z^N + f'(k)z^{N-1} + \cdots + f^{(N)}(k), \\ R_k(z) &= \int_0^k f(x)e^{(k-x)z}dx \end{aligned}$$

Notice

- deg $v(z) \leq N - (n-1)$
- deg $u_k(z) \leq N - n$

How to compute : early example from Hermite.

If $f(x) = x^{n-1}(x-1)^n \cdots (x-m)^n$ and $N = mn + n - 1$ then

$$e^{kz}v(z) - u_k(z) = z^{N+1}R_k(z)$$

with

$$\begin{aligned} v(z) &= f(0)z^N + f'(0)z^{N-1} + \cdots + f^{(N)}(0), \\ u_k(z) &= f(k)z^N + f'(k)z^{N-1} + \cdots + f^{(N)}(k), \\ R_k(z) &= \int_0^k f(x)e^{(k-x)z}dx \end{aligned}$$

Notice

- deg $v(z) \leq N - (n-1)$
- deg $u_k(z) \leq N - n$

How to compute : early example from Hermite.

If $f(x) = x^{n-1}(x-1)^n \cdots (x-m)^n$ and $N = mn + n - 1$ then

$$e^{kz}v(z) - u_k(z) = z^{N+1}R_k(z)$$

with

$$\begin{aligned} v(z) &= f(0)z^N + f'(0)z^{N-1} + \cdots + f^{(N)}(0), \\ u_k(z) &= f(k)z^N + f'(k)z^{N-1} + \cdots + f^{(N)}(k), \\ R_k(z) &= \int_0^k f(x)e^{(k-x)z}dx \end{aligned}$$

Notice

- deg $v(z) \leq N - (n-1)$
- deg $u_k(z) \leq N - n$

Computation

- K. Mahler(1925-1969), J. Coates(1965) , J. Della Dora (1980), . . .
  - (strong conditions assumed)
- B. Beckermann, myself, A. Bultheel, M. van Barel
- B. Salvy and P. Zimmermann (gfun), (and H. Derksen) ; M. Rubey
- M. van Hoeij (use in differential factorization)
- G. Villard (matrix normal forms)
- P. Giorgi, C-P. Jeannerod, G. Villard (fast polynomial matrix arithmetic)
- H. Cheng; E. Kaltofen and students (W. Turner,G. Yuhasz); E. Schost $\cdots$

Issues

1. Coeffs of polynomials can be solved via linear algebra.
   - seems easy
   - just use Gaussian elimination

2. Domain for power series : coeffs from integral domain
   - $A_i(z) \in \mathbb{Z}[[z]]$
   - $A_i(z) \in \mathbb{F}[\alpha_1, \ldots, \alpha_k][[z]]$
   - etc

3. Want to work efficiently in coeff domain

4. Want all approximants
   - want basis as a module for our order problem (order basis)
   - Order basis is main computational tool.

Issues

1. Coeffs of polynomials can be solved via linear algebra.
   - seems easy
   - just use Gaussian elimination
2. Domain for power series : coeffs from integral domain
   - $A_i(z) \in \mathbb{Z}[[z]]$
   - $A_i(z) \in \mathbb{F}[\alpha_1, \ldots, \alpha_k][[z]]$
   - etc
3. Want to work efficiently in coeff domain
4. Want all approximants
   - want basis as a module for our order problem (order basis)
   - Order basis is main computational tool.

Issues

1. Coeffs of polynomials can be solved via linear algebra.
   - seems easy
   - just use Gaussian elimination

2. Domain for power series : coeffs from integral domain
   - $A_i(z) \in \mathbb{Z}[[z]]$
   - $A_i(z) \in \mathbb{F}[\alpha_1, \ldots, \alpha_k][[z]]$
   - etc

3. Want to work efficiently in coeff domain

4. Want all approximants
   - want basis as a module for our order problem (order basis)
   - Order basis is main computational tool.

Issues

1. Coeffs of polynomials can be solved via linear algebra.
   - seems easy
   - just use Gaussian elimination
2. Domain for power series : coeffs from integral domain
   - $A_i(z) \in \mathbb{Z}[[z]]$
   - $A_i(z) \in \mathbb{F}[\alpha_1, \ldots, \alpha_k][[z]]$
   - etc
3. Want to work efficiently in coeff domain
4. Want all approximants
   - want basis as a module for our order problem (order basis)
   - Order basis is main computational tool.

Linear System : $A(z)$ and $B(z)$ and $C(z)$

$(n_0, n_1, n_2, n_3) = (0, 1, 1, 1)$ (so $N = 3$) gives 12×13 system

$$
\left[\begin{array}{ccc|ccc|ccc|cccc}
a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
a_1 & a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
a_2 & a_1 & a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
a_3 & a_2 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
\hline
0 & 0 & 0 & b_0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & b_1 & b_0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & b_2 & b_1 & b_0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & b_3 & b_2 & b_1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & c_0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & c_1 & c_0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & c_2 & c_1 & c_0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & c_3 & c_2 & c_1 & 0 & 0 & 0 & -1
\end{array}\right]
\left[\begin{array}{c}
u_0^{(1)} \\ u_1^{(1)} \\ u_2^{(1)} \\ \hline
u_0^{(2)} \\ u_1^{(2)} \\ u_2^{(2)} \\ \hline
u_0^{(3)} \\ u_1^{(3)} \\ u_2^{(3)} \\ \hline
v_0 \\ v_1 \\ v_2 \\ v_3
\end{array}\right] = 0.
$$

Notice that matrix is highly structured.

Example : e^z and e^{2z} and e^{3z}

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

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

Fraction-Free Gaussian Elimination

$$A = \begin{bmatrix} a & b & c & \cdots & \cdots \\ d & e & f & \cdots & \cdots \\ g & h & i & \cdots & \cdots \\ \vdots & \vdots & \vdots & & \end{bmatrix} \approx \begin{bmatrix} a & b & c & \cdots & \cdots \\ 0 & \tilde{e} & \tilde{f} & \cdots & \cdots \\ 0 & \tilde{h} & \tilde{i} & \cdots & \cdots \\ \vdots & \vdots & \vdots & & \end{bmatrix}$$

- Cross multiplication gives exponential growth of coeffs
- Fraction-free Gaussian elimination (FFGE)

$$A \approx \begin{bmatrix} a & b & c & \cdots & \cdots \\ 0 & \tilde{e} & \tilde{f} & \cdots & \cdots \\ 0 & 0 & a(..) & \cdots & a(...) \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & a(..) & \cdots & a(...) \end{bmatrix}.$$

Allows for linear growth of coefficient size.

Faster Fraction-Free Computation

- For our problems we would want to take advantage of special structure of associated linear system
- Has been successfully done for Hermite-Padé case by B-L [SIMAX 2000]
  - Computes a sequence of Cramer solutions for growing sequence of linear systems
- Gives an algorithm for Simultaneous-Padé approximants
  - Even gives fraction-free way of finding Order Bases.
- Basically our goal is to do better than above.

Faster Fraction-Free Computation

- For our problems we would want to take advantage of special structure of associated linear system
- Has been successfully done for Hermite-Padé case by B-L [SIMAX 2000]
  - Computes a sequence of Cramer solutions for growing sequence of linear systems
- Gives an algorithm for Simultaneous-Padé approximants
  - Even gives fraction-free way of finding Order Bases.
- Basically our goal is to do better than above.

Cost

If input has 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 : $O(\kappa m N)$
- B-L [2009] :
  - Bit complexity of operations: $O(\kappa^2 m^2 N^4)$
  - Size of objects : $O(\kappa N)$

Order Basis

Input : Vector of power series $\mathbf{A}(z) \in \mathbb{K}^{1\times m}[z]$ and integer σ

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

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

- All solutions $\mathbf{V}(z) \in \mathbb{K}^{m\times 1}[z]$ of $\mathbf{A}(z) \cdot \mathbf{V}(z) = O(z^{\sigma})$ is a combination of columns of $\mathbf{M}(z)$.

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

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

Mahler Systems

Input : Vector of power series $\mathbf{A}(z) \in \mathbb{K}^{1\times m}[z]$ and integer vector $\vec{v}$

Mahler System : Matrix polynomial $\mathbf{M}(z) \in \mathbb{K}m \times m[z]$

- Order basis for $\mathbf{A}(z)$ of order $\|\vec{v}\|$
- special degree constraints on columns
- matrix in special Popov form
- leading coefficient special

BL [SIMAX 2000]

For given vector of power series and given vector of degree bounds $\vec{n}$ it computes an order basis for problem

$$A_0(z)U_0(z) + \cdots + A_m(z)U_m(z) = O(z^{N+1})$$

with degree bounds

$$\deg\,(U_i) \leq n_i - 1.$$

- Order bases for Hermite-Padé approximant problem
- Fraction-free method (FFFG)
- Works for vector and matrix power series

Simultaneous-Padé

Solve as

$$\begin{bmatrix} A_1(z) \\ A_2(z) \\ \vdots \\ A_m(z) \end{bmatrix} U_0(z) + \begin{bmatrix} -A_0(z) \\ 0 \\ \vdots \\ 0 \end{bmatrix} U_1(z) + \begin{bmatrix} 0 \\ -A_0(z) \\ \vdots \\ 0 \end{bmatrix} U_2(z) + \cdots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -A_0(z) \end{bmatrix} U_m(z) = O(z^{N+1}).$$

Simultaneous-Padé becomes vector Hermite-Padé problem.

Use FFFG to solve in exact arithmetic.

Duality

Right and Left Matrix Padé duality:

$$A(z)V(z) - U(z) = O(z^K) \text{ and } \hat{V}(z)A(z) - \hat{U}(z) = O(z^K)$$

Hermite-Padé and Simultaneous-Padé duality:

$$A_0(z)U_0(z) + [A_1(z), \cdots, A_m(z)] \begin{bmatrix} U_1(z) \\ \vdots \\ U_m(z) \end{bmatrix} = O(z^{N+1})$$

$$V(z)[A_1(z), \cdots, A_m(z)] - [U_1(z), \cdots, U_m(z)]\, A_0(z) = O(z^{(N+1, \cdots, N+1)})$$

Useful for inversion formulae (L [LAA 1992])

Duality : Much Better

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

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

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

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

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

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

Computation Process

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

Process for Hermite-Padé : Computes Order bases of type

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

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

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

Computation Process

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

Process for Hermite-Padé : Computes Order bases of type

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

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

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

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

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

The Recursion (H-P)

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

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

The Recursion (S-P)

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

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

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
- Use duality to get an 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)