Exact Arithmetic Computation of Rational Approximants and Interpolants

George Labahn
Department of Computer Science
University of Waterloo, Waterloo
Canada.

Antwerp, June 1999

Outline

- Part 1: Introduction
- Part 2: Numeric Computation
- Part 3: Symbolic Computation (preliminaries)
- Part 4: Symbolic Computation (algorithms)

Part 1: Introduction

Goal:

To compute (effectively):

- rational approximants, rational interpolants
  - Padé approximants,
  - Newton-Padé approximants,
  - etc.
- matrix-like rational approximants
  - Hermite-Pade approximants,
  - simultaneous Pade approximants,
  - M-Padé approximants,
  - etc.
- vector and matrix versions of the above
- etc.

Effective:

We want algorithms which will work correctly and efficiently when implemented.

Other related problems:

- greatest common divisors
- one-sided matrix greatest common divisors
  - using Hermite-Padé approximation bases
- minimal bases of kernels of matrix polynomials
  - using Hermite-Padé approximation bases
- normal forms of matrix polynomials
  - Hermite normal form
  - Popov normal form
  - Shifted Popov normal form

Other reasons:

- inversion of Toeplitz like systems
- factorization of differential operators
- etc.

Problems with solving our problems?

- Dependent on domain of computation
- Exact arithmetic algorithms

  $\neq$ effective algorithms in floating point environments

  – *numerical stability ?*

  ... to be continued

- Exact arithmetic algorithms

  $\neq$ effective algorithms in exact arithmetic

  – *growth of coefficients ?*

  ... to be continued

Formal Problem

Example (Hermite-Padé)

- Given:
  - $\mathbf{A}(z)$ a vector of m elements from $\mathbf{F}[[z]]$,
  - $\vec{n} = (\vec{n}^{(1)}, \ldots, \vec{n}^{(m)})$ a multi-index
  - σ a positive integer
- Determine: $\mathbf{P}(z) = [p^{(1)}(z), \ldots, p^{(m)}(z)]^T$ of polynomials
  - $\deg p^{(\ell)}(z) \leq \vec{n}^{(\ell)} - 1$ for all ℓ,
  - $\mathbf{A}(z) \cdot \mathbf{P}(z) = r_\sigma z^\sigma + r_{\sigma+1} z^{\sigma+1} + \cdots$
- Classical Hermite Padé when $\sigma = \vec{n}^{(1)} + \cdots + \vec{n}^{(m)} - 1$.

Interested in finding **all** solutions of the above problem.

- In exact arithmetic
  - Beckermann & Labahn
  - Bultheel & Van Barel
- In numeric arithmetic
  - Cabay, Jones & Labahn (but not all solutions)

Additional Information

- Problems have associate linear systems
  - matrix of coefficients have additional special structure.
  - e.g. Hermite-Padé problem

$$a(x) \cdot p(x) + b(x) \cdot q(x) + c(x) \cdot r(x) = O(x^8)$$

with deg $p(x) \le 2$, deg $q(x) \le 3$, deg $r(x) \le 1$

$$\left[\begin{array}{ccc|cccc|cc} a_0 & 0 & 0 & b_0 & 0 & 0 & 0 & c_0 & 0 \\ a_1 & a_0 & 0 & b_1 & b_0 & 0 & 0 & c_1 & c_0 \\ a_2 & a_1 & a_0 & b_2 & b_1 & b_0 & 0 & c_2 & c_1 \\ a_3 & a_2 & a_1 & b_3 & b_2 & b_1 & b_0 & c_3 & c_2 \\ a_4 & a_3 & a_2 & b_4 & b_3 & b_2 & b_1 & c_4 & c_3 \\ a_5 & a_4 & a_3 & b_5 & b_4 & b_3 & b_2 & c_5 & c_4 \\ a_6 & a_5 & a_4 & b_6 & b_5 & b_4 & b_3 & c_6 & c_5 \\ a_7 & a_6 & a_5 & b_7 & b_6 & b_5 & b_4 & c_7 & c_6 \end{array}\right] \cdot \left[\begin{array}{c} p_0 \\ p_1 \\ p_2 \\ \hline q_0 \\ q_1 \\ q_2 \\ q_3 \\ \hline r_0 \\ r_1 \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right]$$

- Efficient: find elimination process that takes advantage of additional structure
- Recursion presented in terms of multiplication of matrix polynomials.
  - e.g. the classical Euclidean algorithm.
- Normalization of components of recursion plays a large role in effective algorithms for both numeric and symbolic computation.

Part II : Numeric environment

Important problems and issues:

- inability to determine when a numerical quantity is zero
- Heuristic: do not solve ill–conditioned subproblems
- Look–ahead: jump over ill–conditioned subproblems
- Provide message if problem is ill–conditioned
- Provide (forward or backward) error analysis.

Some recent developments

- **Rational interpolation**: Thiele with pivoting and forward error analysis (Graves-Morris '80)
- **Padé approximation** and **Hankel systems**: look-ahead, estimation of condition number via inversion formulas, forward error analysis (Cabay & Meleshko '93).
- Generalization to **Hermite-Padé approximation** and **simultaneous Padé approximation** (Cabay, Jones & Labahn '95).
- Super-fast solution of **Hankel/Toeplitz systems**: presumably stable (Gutknecht, Gutknecht & Hochbruck '93).
- Fast solution of **block Toeplitz systems**: look ahead and forward error analysis (Van Barel & Bultheel '97).
- Fast detection of **relatively prime numerical polynomials** (Beckermann & Labahn '98).
- And many others, e.g.:
  - FOP's: Freund & Zha '93, Beckermann '96,...
  - Fast QR factorization: Bojanczyk, Brent, de Hoog '95
  - Via displacement rank: Gohberg, Olshevski, Golub, Heinig,...
  - Via rational interpolation on unit circle: Kravanja & Van Barel '98

... Link to Lanzcos type algorithms

Cabay-Meleshko revised

- Problem: given $f(z)$ find Padé forms $(p_n(z), q_n(z))$
  - deg $p_n(z) < n$, deg $q_n(z) \leq n$,
  - $f(z)q_n(z) - p_n(z) = \mathcal{O}(z^{2n})$
  - $||p_n(z)|| + ||q_n(z)|| \approx 1$,
    where $||\,\Sigma\, a_j z^j|| = \Sigma\,|a_j|, \quad ||f|| \approx 1$
- Basic building block $U_n(z) = \begin{bmatrix} p_n(z) & p_{n+1}(z) \\ q_n(z) & q_{n+1}(z) \end{bmatrix}$
  - $\det U_n(z) \approx \tau_n \cdot z^{2n}$, $|\tau_n| \leq 1$.
  - $U_n(z)$ called "stable" if $|\tau_n|$ not "too small"
  - Iteration:
    compute next stable $U_{n+s}(z)$ from stable $U_n(z)$
  - Method:
    build "small" system solved by classical stable method
- Why only "stable" $U_n(z)$?
  - Heuristic: all critical divisors minorized by $|\tau_n|$.
  - Heuristic: only well-conditioned subproblems

$$\frac{1}{2\sqrt{|\tau_n|}} \leq \mathrm{cond}_1\left(\begin{bmatrix} f_0 & f_1 & \cdots & f_n \\ f_1 & f_2 & \cdots & f_{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ f_n & f_{n+1} & \cdots & f_{2n} \end{bmatrix}\right) \leq \frac{1}{|\tau_n|}.$$

  - Error analysis: local errors are at most amplified by

$$||U_n(z)^{-1}U_{n+s}(z)|| \leq 2/|\tau_n|.$$

Part III: Symbolic environment

Solving problems in symbolic environments:

- **Basic coefficient domain:**
  - Quotient field: $\mathbb{Q}(\alpha_1, \ldots, \alpha_k)$
  - symbols are first class objects in CA environments.
- **Basic fact:**
  - Polynomial arithmetic easier than arithmetic with rational functions
$$\frac{a(x)}{b(x)} + \frac{c(x)}{d(x)} = \frac{a(x) \cdot d(x) + b(x) \cdot c(x)}{b(x) \cdot d(x)}$$
  - Need to normalize out gcd's at every step if want to recognize 0 at later computations.
- **Basic goal:**
  - To work with polynomial arithmetic in integral domain (e.g. in $\mathbb{Q}[\alpha_1, \ldots, \alpha_k]$) rather than in quotient field.
- We want to do our arithmetic **fraction-free** but at the same time to minimize growth of intermediate computation.

Example: Gaussian Elimination

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

- Only using cross multiplication results in exponential growth of coefficients: $\mathrm{O}(2^n \cdot N^2)$ in $n \times n$ case (since coefficients of matrix double in size at each elimination row - here N is bound for size of entries).
- Fraction-free Gaussian elimination (FFGE) : Bareiss (1968), observes a common divisor after 2 steps

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

  Simple observation allows for linear growth of coefficient size. Cost $\mathcal{O}(n^5 \cdot N^2)$.
- Important point: FFGE computes the Cramer solution of a linear problem.

Example: Euclidean Gcd

$a(x), b(x) \in \mathbf{Z}[x]$ given by (Knuth)

$$\begin{aligned} a(x) &= x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5 \\ b(x) &= 3x^6 + 5x^4 - 4x^2 - 9x + 21 \end{aligned}$$

- Euclidean Algorithm over $\mathbb{Q}[x]$ (monic)
$$\begin{aligned} r_1(x) &= x^4 - \tfrac{1}{5}\,x^2 + \tfrac{3}{5} \\ r_2(x) &= x^2 + \tfrac{25}{13}\,x - \tfrac{49}{13} \\ r_3(x) &= x - \tfrac{6150}{4663} \\ r_4(x) &= 1 \end{aligned}$$
- Euclidean-like Algorithm over $\mathbf{Z}[x]$ (Cross multiplication)
$$\begin{aligned} r_1(x) &= 15\,x^4 - 3\,x^2 + 9 \\ r_2(x) &= 15795\,x^2 + 30375\,x - 59535 \\ r_3(x) &= 1254542875143750\,x - 1654608338437500 \\ r_4(x) &= 12593338795500743100931141992187500 \end{aligned}$$
- Fraction-free Algorithm over $\mathbf{Z}[x]$ (Removing contents)
$$\begin{aligned} r_1(x) &= 5\,x^4 - x^2 + 3 \\ r_2(x) &= 13\,x^2 + 25\,x - 49 \\ r_3(x) &= 4663\,x - 6150 \\ r_4(x) &= 1 \end{aligned}$$
- Fraction-free Algorithm over $\mathbf{Z}[x]$ (Subresultants)
$$\begin{aligned} r_1(x) &= 15\,x^4 - 3\,x^2 + 9 \\ r_2(x) &= 65\,x^2 + 125\,x - 245 \\ r_3(x) &= 9326\,x - 12300 \\ r_4(x) &= 260708 \end{aligned}$$

Observations

- Gcd Examples:

$$R_0(x) = a(x), R_1(x) = b(x)$$

$$\alpha_i \cdot R_{i-1}(x) = Q_i(x) \cdot R_i(x) + \beta_i \cdot R_{i+1}(x)$$

  Let $r_i = lcoeff\ R_i(x)$ and $\delta_i = \deg\ R_{i-1}(x) - \deg\ R_i(x)$.

  - Cross multiplication: $\alpha_i = r_i^{\delta_i+1}$, $\beta_i = 1$.
  - Content removal:
$$\begin{aligned} \alpha_i &= r_i^{\delta_i+1} \\ \beta_i &= \text{content prem}(R_{i-1}(x), R_i(x)) \end{aligned}.$$
  - Reduced: $\alpha_i = r_i^{\delta_i+1}$, $\beta_i = \alpha_{i-1}$.
  - Subresultant: $\alpha_i = r_i^{\delta_i+1}$, $\beta_1 = (-1)^{\delta_1+1}, \beta_i = -r_{i-1} \cdot \phi_i^{\delta_i}$

    where $\phi_1 = -1, \phi_i = (-r_{i-1})^{\delta_{i-1}} \cdot \phi_{i-1}^{1-\delta_{i-1}}$.

- Computations kept in domain of coefficients
- Determine a common factor without any work! How?
- Some references:
  - Fraction-free Gaussian : Bareiss (1968)
  - Fraction-free GCD : Habicht (1948), Brown (1971), Collins (1967)
  - Padé approximants : Cabay and Kossowski (1990)

Aside: GCD in Maple

- Modular method:
  - GCD$(a(x), b(x)) = 1$ in $\mathbf{Z}_{23}[x]$

    $\Longrightarrow$ GCD$(a(x), b(x)) = 1$ in $\mathbf{Z}[x]$

  - Fraction-free method need to ensure modular method normalizes correctly.
- Hensel lifting
- Things are not always as expected in Symbolic Computation.

  E.g. GCD Heuristic (Char, Geddes, Gonnet '89)

$$\begin{aligned} a(x) &= 6x^4 + 21x^3 + 35x^2 + 27x + 7 \\ b(x) &= 12x^4 - 3x^3 - 17x^2 - 45x + 21 \end{aligned}$$

  - Evaluate at point:

    E.g. $a(100) = 621352707,\ b(100) = 1196825521$

  - Integer GCD $= 30607 = 3 \cdot 100^2 + 6 \cdot 100 + 7$
  - Set $c(x) = 3x^2 + 6x + 7$.
    - $c(x) \mid a(x)$ and $c(x) \mid b(x)$
    - Implies that GCD$(a(x), b(x)) = c(x)$.
  - Often works.

Part IV:

Rational Approximants and Interpolants

- Notation:
  - $\mathbb{D}$ an integral domain, $\mathbb{Q}$ its quotient field,
  - $\mathcal{V}$ infinite dimensional vector space over $\mathbb{Q}$,
  - $(c_u)_{u=0,1,\dots}$ a set of linear functionals over $\mathcal{V}$.
  - Special element z.
- Have *special rule*

  $$c_u(z \cdot f) = c_{u,0} \cdot c_0(f) + \cdots + c_{u,u} \cdot c_u(f) \text{ with } c_{u,v} \in \mathbb{D}.$$

  Special rule viewed as a type of Leibniz chain rule.
- Problem:
  - Given:
    - $f = [f^{(1)}, \dots, f^{(m)}]$ a vector of elements from $\mathcal{V}$,
    - $\vec{n} = (\vec{n}^{(1)}, \dots, \vec{n}^{(m)})$ a multi-index
    - σ a positive integer
  - Determine: $p(z) = [p^{(1)}(z), \dots, p^{(m)}(z)]^T$ of polynomials
    - $\deg p^{(\ell)}(z) \le \vec{n}^{(\ell)} - 1$
    - $c_u(f^{(1)} \cdot p^{(1)}(z) + \cdots + f^{(m)} \cdot p^{(m)}(z)) = 0$ for $u < \sigma$.
  - $p(z)$ referred to as solution of type $(\sigma, \vec{n})$.
- Interested in finding **all** solutions of the above problem.

Examples

- Hermite–Padé:
  - $\mathcal{V} = \mathbb{Q}[[z]]$,
  - $c_{i,j} = \delta_{i-1,j}$,
  - special rule = standard multplication by z
- Vector Hermite–Padé:
  - $\mathcal{V} = \mathbb{Q}^s[[z]]$,
  - $c_{i,j} = \delta_{i-s,j}$,
  - special rule = standard scalar multplication by z
- M-Padé:
  - knots $x_i \in \mathbb{D}$,
  - $\mathcal{V}$ = formal Newton series in z,
  - c_v is v-th divided difference,
  - special rule: $c_{i,j} = \delta_{i,j} \cdot x_i + \delta_{i-1,j}$
- Others: Simultaneous Padé, Power Hermite-Padé, Linearized rational interpolation problem, Generalized Richardson extrapolation process, Controller–form realizations.

Cramer Solutions

Goal: Try to find Cramer solutions

- e.g. Hermite-Padé problem

$$a(x) \cdot p(x) + b(x) \cdot q(x) + c(x) \cdot r(x) = O(x^6)$$

with deg $p(x) \leq 2$, deg $q(x) \leq 1$, deg $r(x) \leq 1$

$$\left[\begin{array}{ccc|cc|cc} a_0 & 0 & 0 & b_0 & 0 & c_0 & 0 \\ a_1 & a_0 & 0 & b_1 & b_0 & c_1 & c_0 \\ a_2 & a_1 & a_0 & b_2 & b_1 & c_2 & c_1 \\ a_3 & a_2 & a_1 & b_3 & b_2 & c_3 & c_2 \\ a_4 & a_3 & a_2 & b_4 & b_3 & c_4 & c_3 \\ a_5 & a_4 & a_3 & b_5 & b_4 & c_5 & c_4 \\ a_6 & a_5 & a_4 & b_6 & b_5 & c_6 & c_5 \end{array}\right] \cdot \left[\begin{array}{c} p_0 \\ p_1 \\ p_2 \\ \hline q_0 \\ q_1 \\ \hline r_0 \\ r_1 \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ d \end{array}\right]$$

where d is determinant of coefficient matrix.

- Solution has determinant representation in nonsingular case:

$$\text{e.g. } p(z) = \det \left[\begin{array}{ccc|ccc|c} a_0 & 0 & 0 & b_0 & 0 & c_0 & 0 \\ a_1 & a_0 & 0 & b_1 & b_0 & c_1 & c_0 \\ a_2 & a_1 & a_0 & b_2 & b_1 & c_2 & c_1 \\ a_3 & a_2 & a_1 & b_3 & b_2 & c_3 & c_2 \\ a_4 & a_3 & a_2 & b_4 & b_3 & c_4 & c_3 \\ 1 & z & z^2 & 0 & 0 & 0 & 0 \end{array}\right]$$

- Unique in nonsingular case.
- Recursively build Cramer solutions from Cramer solutions of smaller problems along offdiagonal of associated table.

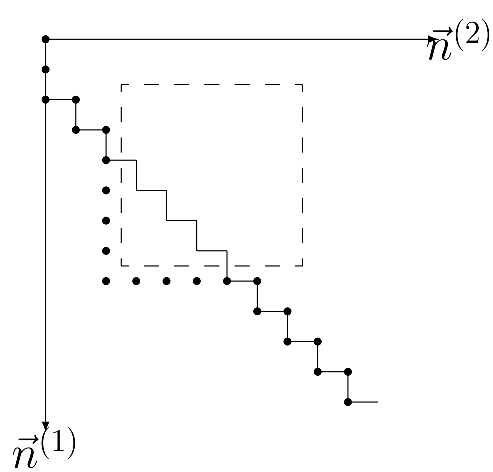
Mahler Systems

- Matrix $\mathbf{M}(z)$ of determinantal polynomials with degrees

$$\begin{bmatrix} n_1 & n_1 - 1 & \cdots & n_1 - 1 \\ n_2 - 1 & n_2 & \cdots & n_2 - 1 \\ \vdots & & \ddots & \vdots \\ n_m - 1 & \cdots & \cdots & n_m \end{bmatrix}$$

  and lcoeff of diagonal = determinant of coeff matrix.

- Unique in nonsingular case
- Basic building block of recursions.
- Method 1: via modified Schur complements
  - nonsingular location to nonsingular location in table
  - similar to look ahead
- Method 2: via determinental identities
  - works in singular cases by computing at closest nonsingular locations (look around)
  - staircase path of computation



Additional Details

- Modified Schur Complements (Look-ahead)
  - Linear systems:

$$\mathbf{T} \cdot \mathbf{S}_{large} = \begin{bmatrix} \mathbf{S}_{small} & 0 \\ \# & \mathbf{S}_{residual} \end{bmatrix}$$

    with det $\mathbf{T}$ given in terms of "small" system.
  - Recursion:

$$c \cdot \mathbf{M}_{large}(z) = \mathbf{M}_{small}(z) \cdot \mathbf{M}_{residual}(z)$$

  - c determined by det $\mathbf{T} \Longrightarrow$ easy to predict.
  - Beckermann, Cabay & Labahn '97
- Closest Normal Point (Look-around)
  - Knowledge of block structure in m-dimen. solution table
  - Mahler systems have a "shifted Popov" normal form
  - Forms basis for module of solutions of order problem
  - Going from one Mahler system to the next using
    * Sylvester's identity (as in FFGE) and
    * "structured" determinantal identity (speed up)
      For Hermite-Padé compare with Paszkowski '87
  - Includes "special rule" via block Krylov matrices
  - Cost $\mathcal{O}(m \cdot \sigma^4 \cdot N^2)$ instead of $\mathcal{O}(\sigma^5 \cdot N^2)$.
  - Beckermann & Labahn '97

Matrix GCD and Normal Forms

Basis for Matrix Hermite-Padé Problem

- $\mathbf{F}(z) \cdot \mathbf{M}(z) = r_\sigma z^\sigma + r_{\sigma+1} z^{\sigma+1} + \cdots$

Matrix GCD

- $[\mathbf{A}(z), \mathbf{B}(z)] \cdot \mathbf{U}(z) = [\mathbf{C}(z), 0]$, $\mathbf{U}(z)$ unimodular
- Reversing coefficients gives Matrix Hermite Padé problem
- Reversed $\mathbf{U}(z)$ an order basis

Matrix Normal Forms

- $\mathbf{A}(z) \cdot \mathbf{U}(z) = \mathbf{T}(z)$
- $[\mathbf{A}(z), -\mathbf{I}] \cdot \begin{bmatrix} \mathbf{U}(z) \\ \mathbf{T}(z) \end{bmatrix} = 0$
- $\mathbf{F}(z) \cdot \mathbf{M}(z) = r_\sigma z^\sigma + r_{\sigma+1} z^{\sigma+1} + \cdots = 0$
- Matrix Hermite Padé basis

$\longrightarrow$ minimal basis for kernel of $[\mathbf{A}(z), -\mathbf{I}]$

$\longrightarrow$ matrix normal form.

Continuing Work

- Additional fraction-free algorithms for
  - Matrix normal forms (Popov, Hermite)
    - Some progress: Beckermann, Labahn, Villard '99
  - Minimal Bases
- Modular algorithms (order of magnitude improvement)
  - Needs fraction-free for its normalization
- Heuristic methods (succeed or fail quickly)
- Hensel lifting methods
- Noncommutative case: differential polynomials.
  - Use in differential-algebraic systems.
- Integer, rather than polynomial, problems.
- Finding shortest vectors in a lattice.