Polynomial Matrices and Structured Matrices

Alin Bostan

Specfun, Inria

MPRI C-2-22

November 2, 2020

Hermite-Padé approximants

Definition of Hermite-Padé approximants

Definition: Given a column vector $\mathbf{F} = (f_1, \ldots, f_n)^T \in \mathbb{K}[[x]]^n$ and an n-tuple $\mathbf{d} = (d_1, \ldots, d_n) \in \mathbb{N}^n$, a Hermite-Padé approximant of type $\mathbf{d}$ for $\mathbf{F}$ is a row vector $\mathbf{P} = (P_1, \ldots, P_n) \in \mathbb{K}[x]^n$, $(\mathbf{P} \neq 0)$, such that:

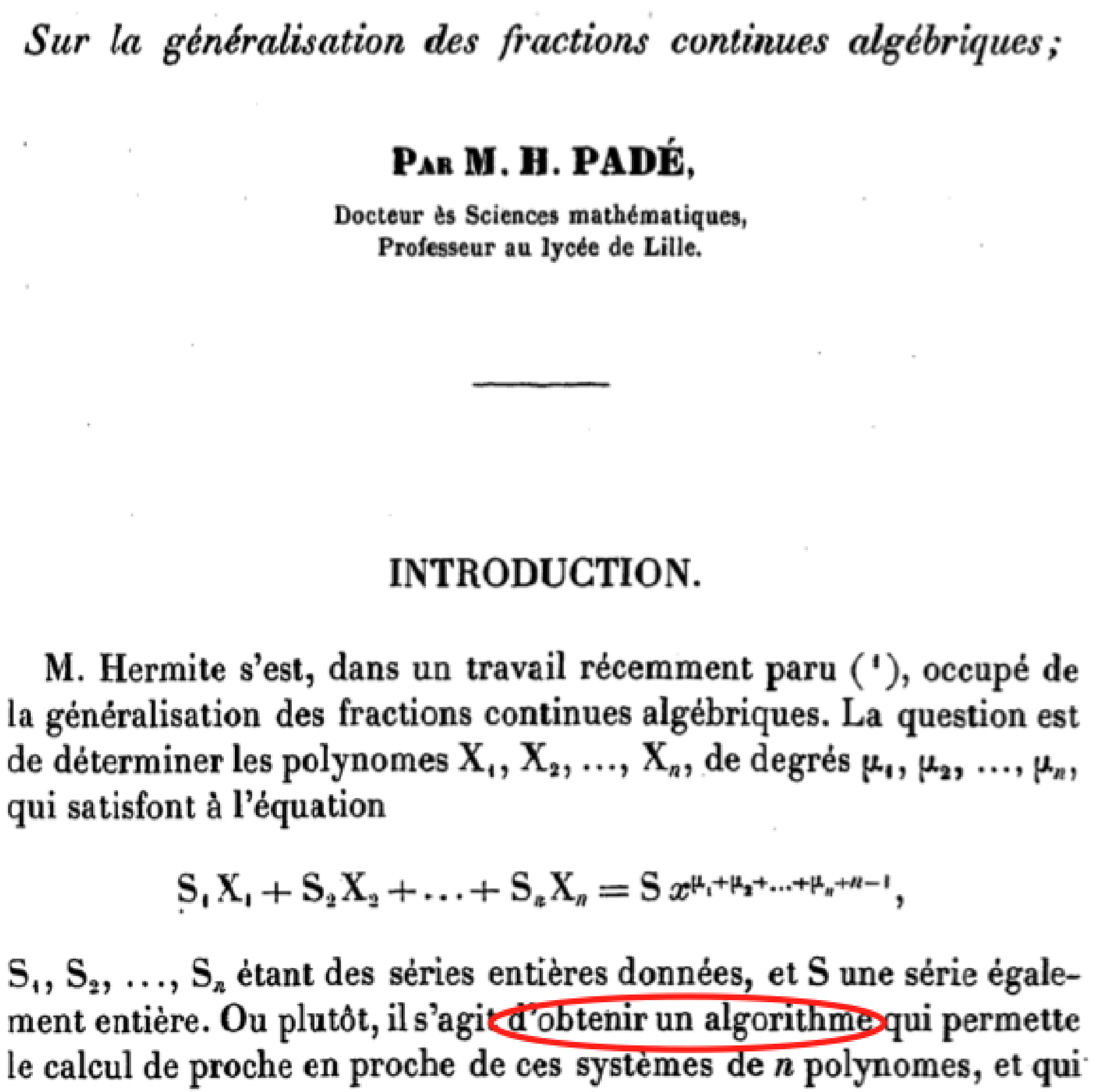
(1) $\mathbf{P} \cdot \mathbf{F} = P_1 f_1 + \cdots + P_n f_n = O(x^\sigma)$ with $\sigma = \sum_i (d_i + 1) - 1$,

(2) $\deg(P_i) \leq d_i$ for all i.

σ is called the order of the approximant $\mathbf{P}$.

▷ Very useful concept in number theory (irrationality/transcendence):

- [Hermite, 1873]: e is transcendental.
- [Lindemann, 1882]: π is transcendental; so does e^α for any $\alpha \in \overline{\mathbb{Q}} \setminus \{0\}$.
- [Apéry, 1978; Beukers, 1981]: $\zeta(3) = \sum_{n \geq 1} \frac{1}{n^3}$ is irrational.
- [Rivoal, 2000]: there exist infinitely many $k \in \mathbb{N}$ such that $\zeta(2k+1) \notin \mathbb{Q}$.



Sur la généralisation des fractions continues algébriques;

Par M. H. PADÉ,
Docteur ès Sciences mathématiques,
Professeur au lycée de Lille.

INTRODUCTION.

M. Hermite s'est, dans un travail récemment paru (1), occupé de la généralisation des fractions continues algébriques. La question est de déterminer les polynomes $X_1, X_2, \dots, X_n$, de degrés $\mu_1, \mu_2, \dots, \mu_n$, qui satisfont à l'équation

$$S_1X_1 + S_2X_2 + \dots + S_nX_n = S\,x^{\mu_1+\mu_2+\dots+\mu_n+n-1},$$

$S_1, S_2, \dots, S_n$ étant des séries entières données, et S une série également entière. Ou plutôt, il s'agit d'obtenir un algorithme qui permette le calcul de proche en proche de ces systèmes de n polynomes, et qui

[Padé, 1894]

Journal of Symbolic Computation 102 (2021) 279–303

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Algorithms for simultaneous Hermite–Padé approximations

Johan Rosenkilde[a], Arne Storjohann[b]

[a] Technical University of Denmark, Denmark
[b] University of Waterloo, Canada

ARTICLE INFO

Article history:
Received 26 November 2018
Accepted 4 July 2019
Available online 18 October 2019

Keywords:
Padé approximation
Approximant bases
Structured linear systems

ABSTRACT

We describe how to compute simultaneous Hermite–Padé approximations, over a polynomial ring $\mathsf{K}[x]$ for a field K using $O^{\sim}(n^{\omega-1}td)$ operations in K, where d is the sought precision, where n is the number of simultaneous approximations using $t < n$ polynomials, and where $O(n^{\omega})$ is the cost of multiplying $n \times n$ matrices over K. We develop two algorithms using different approaches. Both algorithms return a reduced sub-basis that generates the complete set of solutions to the input approximation problem that satisfy the given degree constraints. Previously, the cost $O^{\sim}(n^{\omega-1}td)$ has only been reached with randomized algorithms finding a single solution for the case $t < n$. Our results are made possible by recent breakthroughs in fast computations of minimal approximant bases and Hermite–Padé approximations for the case $t \geq n$.

Worked example

Let us compute a Hermite-Padé approximant of type $(1,1,1)$ for $(1,C,C^2)$, where $C(x) = 1 + x + 2x^2 + 5x^3 + 14x^4 + 42x^5 + O(x^6)$.

This boils down to finding $\alpha_0, \alpha_1, \beta_0, \beta_1, \gamma_0, \gamma_1$ (not all zero) such that

$$\alpha_0+\alpha_1 x+(\beta_0+\beta_1 x)(1+x+2x^2+5x^3+14x^4)+(\gamma_0+\gamma_1 x)(1+2x+5x^2+14x^3+42x^4)=O(x^5)$$

Identifying coefficients, this is equivalent to a homogeneous linear system:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 2 & 1 \\ 0 & 0 & 2 & 1 & 5 & 2 \\ 0 & 0 & 5 & 2 & 14 & 5 \\ 0 & 0 & 14 & 5 & 42 & 14 \end{bmatrix} \times \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \beta_0 \\ \beta_1 \\ \gamma_0 \\ \gamma_1 \end{bmatrix} = 0 \iff \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 2 & 1 & 5 \\ 0 & 0 & 5 & 2 & 14 \\ 0 & 0 & 14 & 5 & 42 \end{bmatrix} \times \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \beta_0 \\ \beta_1 \\ \gamma_0 \end{bmatrix} = -\gamma_1 \begin{bmatrix} 0 \\ 1 \\ 2 \\ 5 \\ 14 \end{bmatrix}.$$

By homogeneity, one can choose $\gamma_1 = 1$. Then, the violet minor shows that one can take $(\beta_0, \beta_1, \gamma_0) = (-1, 0, 0)$. The other values are $\alpha_0 = 1$, $\alpha_1 = 0$.

Thus the approximant is $(1, -1, x)$, which corresponds to $P = 1 - y + xy^2$ such that $P(x, C(x)) = 0 \bmod x^5$.

Algebraic and differential approximation = guessing

- Hermite-Padé approximants of $n = 2$ power series are related to Padé approximants, i.e. to approximation of series by rational functions
- algebraic approximants = Hermite-Padé approximants for $f_\ell = A^{\ell-1}$, where $A \in \mathbb{K}[[x]]$ seriestoalgeq, listtoalgeq
- differential approximants = Hermite-Padé approximants for $f_\ell = A^{(\ell-1)}$, where $A \in \mathbb{K}[[x]]$ seriestodiffeq, listtodiffeq

```
> listtoalgeq([1,1,2,5,14,42,132,429],y(x))[1];
```

$$1 - y(x) + x\,(y(x))^2$$

```
> listtodiffeq([1,1,2,5,14,42,132,429],y(x))[1][1];
```

$$-2\,y(x) + (-4\,x + 2)\,\frac{\mathrm{d}}{\mathrm{d}x}y(x) + x\,\frac{\mathrm{d}^2}{\mathrm{d}x^2}y(x)$$

Existence and naive computation

Theorem For any vector $\mathbf{F} = (f_1, \ldots, f_n)^T \in \mathbb{K}[[x]]^n$ and for any n-tuple $\mathbf{d} = (d_1, \ldots, d_n) \in \mathbb{N}^n$, there exists a Hermite-Padé approx. of type $\mathbf{d}$ for $\mathbf{F}$.

Proof: The undetermined coefficients of $P_i = \sum_{j=0}^{d_i} p_{i,j} x^j$ satisfy a linear homogeneous system with $\sigma = \sum_i (d_i + 1) - 1$ eqs and $\sigma + 1$ unknowns.

Corollary Computation in $O(\sigma^\theta)$, for $2 \leq \theta \leq 3$ (linear algebra exponent)

▷ There are better algorithms (the linear system is structured, Sylvester-like):

- Derksen's algorithm (Euclidean-like elimination) $O(\sigma^2)$
- Beckermann-Labahn algorithm (DAC) $\tilde{O}(\sigma) = O(\sigma \log^2 \sigma)$
- structured linear algebra algorithms for Toeplitz-like matrices $\tilde{O}(\sigma)$

Quasi-optimal computation

Theorem [Beckermann, Labahn, 1994] One can compute a Hermite-Padé approximant of type $(d, \ldots, d)$ for $\mathbf{F} = (f_1, \ldots, f_n)$ in $\tilde{O}(n^\theta d)$ ops. in $\mathbb{K}$.

Ideas:

- Compute a whole matrix of approximants
- Exploit divide-and-conquer

Algorithm:

1. If $\sigma = n(d+1) - 1 \leq$ threshold, call the naive algorithm
2. Else:
   (a) recursively compute $\mathbf{P}_1 \in \mathbb{K}[x]^{n \times n}$ s.t. $\mathbf{P}_1 \cdot \mathbf{F} = O(x^{\sigma/2})$, $\deg(\mathbf{P}_1) \approx \frac{d}{2}$
   (b) compute "residue" $\mathbf{R}$ such that $\mathbf{P}_1 \cdot \mathbf{F} = x^{\sigma/2} \cdot \big(\mathbf{R} + O(x^{\sigma/2})\big)$
   (c) recursively compute $\mathbf{P}_2 \in \mathbb{K}[x]^{n \times n}$ s.t. $\mathbf{P}_2 \cdot \mathbf{R} = O(x^{\sigma/2})$, $\deg(\mathbf{P}_2) \approx \frac{d}{2}$
   (d) return $\mathbf{P} := \mathbf{P}_2 \cdot \mathbf{P}_1$

▷ The precise choices of degrees is a delicate issue

▷ Corollary: Gcd, extended gcd, Padé approximants in $\tilde{O}(d)$ ops. in $\mathbb{K}$.

Application: certified algebraic guessing

Guess + Bound = Proof

Theorem. Suppose $A \in \mathbb{K}[[x]]$ is an algebraic series, and that it is a root of a (unknown) polynomial in $\mathbb{K}[x,y]$ of degree at most d in x and at most n in y.

If $\displaystyle\sum_{i=0}^{n} Q_i(x)A^i(x) = O(x^{2dn+1})$ and $\deg Q_i \le d$, then $\displaystyle\sum_{i=0}^{n} Q_i(x)A^i(x) = 0$.

In other words, A is a root of the polynomial $Q = \sum_{i=0}^{n} Q_i(x)y^i$.

Application: certified algebraic guessing

Guess + Bound = Proof

Theorem. Suppose $A \in \mathbb{K}[[x]]$ is an algebraic series, and that it is a root of a (unknown) polynomial in $\mathbb{K}[x,y]$ of degree at most d in x and at most n in y.

If $\displaystyle\sum_{i=0}^{n} Q_i(x)A^i(x) = O(x^{2dn+1})$ and $\deg Q_i \leq d$, then $\displaystyle\sum_{i=0}^{n} Q_i(x)A^i(x) = 0$.

In other words, A is a root of the polynomial $Q = \sum_{i=0}^{n} Q_i(x)y^i$.

Proof: Let $P \in \mathbb{K}[x,y]$ be an irreducible polynomial such that

$$P(x, A(x)) = 0, \text{ and } \deg_x(P) \leq d, \deg_y(P) \leq n.$$

Application: certified algebraic guessing

Guess + Bound = Proof

Theorem. Suppose $A \in \mathbb{K}[[x]]$ is an algebraic series, and that it is a root of a (unknown) polynomial in $\mathbb{K}[x,y]$ of degree at most d in x and at most n in y.

$$\text{If} \quad \sum_{i=0}^{n} Q_i(x)A^i(x) = O(x^{2dn+1}) \text{ and } \deg Q_i \leq d, \quad \text{then} \quad \sum_{i=0}^{n} Q_i(x)A^i(x) = 0.$$

In other words, A is a root of the polynomial $Q = \sum_{i=0}^{n} Q_i(x)y^i$.

Proof: Let $P \in \mathbb{K}[x,y]$ be an irreducible polynomial such that

$$P(x, A(x)) = 0, \text{ and } \deg_x(P) \leq d, \ \deg_y(P) \leq n.$$

- By definition, $R(x) = \mathrm{Res}_y(P,Q) \in \mathbb{K}[x]$ has degree at most $2dn$.
- By elimination, $R(x) = UP + VQ$ for $U, V \in \mathbb{K}[x,y]$ with $\deg_y(V) < n$.
- Evaluation at $y = A(x)$ yields

$$R(x) = U(x, A(x)) \underbrace{P(x, A(x))}_{0} + V(x, A(x)) \underbrace{Q(x, A(x))}_{O(x^{2dn+1})} = O(x^{2dn+1}).$$

- Thus $R = 0$, that is $\gcd(P,Q) \neq 1$, and thus $P \mid Q$, and A is a root of Q.

Ex. 1

Let $(a_n)_{n\geq 0}$ be a sequence with $a_0 = a_1 = 1$ satisfying the recurrence

$$(n+3)a_{n+1} = (2n+3)a_n + 3na_{n-1}.$$

Show that a_n is an integer for all n, by following the next steps:

(1) Compute the first 5 terms of the sequence, $a_0, \ldots, a_4$;

(2) Show that $[1, x-1, x^2]$ is a Hermite-Padé approximant of type $(0,1,2)$ for $(1, f, f^2)$, where $f = \sum_n a_n x^n$;

(3) Deduce that $P(x,y) := 1 + (x-1)y + x^2y^2$ satisfies $P(x, f(x)) = 0 \bmod x^5$;

(4) Show that the equation $P(x,y) = 0$ admits a root $y = g(x) \in \mathbb{Q}[[x]]$ whose coefficients satisfy the same linear recurrence as $(a_n)_{n\geq 0}$;

(5) Deduce that $a_{n+2} = a_{n+1} + \sum_{k=0}^{n} a_k \cdot a_{n-k}$ for all n, and conclude.

Polynomial Matrices

Context

▷ Linear algebra questions in $\mathcal{M}_n(\mathbb{A})$, where $\mathbb{A}$ is a ring in which multiplication does not have unit cost. Typically: $\mathbb{A} = \mathbb{K}[x]$ or $\mathbb{A} = \mathbb{Z}$.

▷ Although some algorithms (e.g., naive, or Strassen's multiplication) remain well-adapted to this framework, this is not the case for other important tasks:

- matrix inversion by Strassen's algorithm ⟶ Ex. 2
- system solving by Gaussian elimination

▷ Main reason: expression swell during intermediate computations

▷ . . . sometimes leading even to an exponential blow-up

▷ Conclusion: Need for new algorithmic ideas!

Ex. 2: Strassen's inversion for polynomial matrices

Let $M(x) \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ be an invertible polynomial matrix.

Assume one computes its inverse using Strassen's inversion algorithm for dense (scalar) matrices.

Estimate the complexity of this computation, counting operations in $\mathbb{K}$, in terms of the two parameters n and d, under the assumption that all matrices encountered during the inversion algorithm are invertible.

Using the evaluation/interpolation paradigm

$\triangleright$ The modular approach allows to recover polynomial-time complexity

$\triangleright$ ... but it usually yields non-optimal algorithms

$\triangleright$ Recall (multiplication of polynomial matrices)

If $A, B \in \mathcal{M}_n(\mathbb{K}[x]_{<d})$, then one can compute $C = AB$ by eval/interp in

$$\mathsf{MM}(n,d) = O(n^2\,\mathsf{M}(d) + \mathsf{MM}(n)d) = \tilde{O}(n^\theta d)$$

$\triangleright$ Recall (determinant of polynomial matrices)

If $A \in \mathcal{M}_n(\mathbb{K}[x]_{<d})$, then one can compute $\det(A)$ by eval/interp in

$$O(n^2\,\mathsf{M}(nd) + nd\,\mathsf{MM}(n)) = \tilde{O}(n^{\theta+1}d)$$

Main results

Theorem [complexity results for polynomial matrices]

Let $\mathbb{K}$ be a field, $n, d \in \mathbb{N}$, and θ a feasible exponent for product in $\mathcal{M}_n(\mathbb{K})$. Let $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ and $b \in \mathbb{K}[x]^n_{<d}$. Then one can compute:

(a) A^{-1} in $O(\mathsf{MM}(n, nd)) = \tilde{O}(n^{\theta+1}\, d)$ ops. in $\mathbb{K}$

(b) $y \in \mathcal{M}_n(\mathbb{K}(x))$ s.t. $Ay = b$ in $O(\mathsf{MM}(n, d) \log n) = \tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$

(c) $\det(A)$ in $O(\mathsf{MM}(n, d) \log^2 n) = \tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$

(d) $\mathrm{rk}(A)$ and a basis of $\ker(A)$ in $\tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$.

Main results

Theorem [complexity results for polynomial matrices]

Let $\mathbb{K}$ be a field, $n, d \in \mathbb{N}$, and θ a feasible exponent for product in $\mathcal{M}_n(\mathbb{K})$. Let $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ and $b \in \mathbb{K}[x]^n_{<d}$. Then one can compute:

(a) A^{-1} in $O(\mathsf{MM}(n, nd)) = \tilde{O}(n^{\theta+1}\, d)$ ops. in $\mathbb{K}$

(b) $y \in \mathcal{M}_n(\mathbb{K}(x))$ s.t. $Ay = b$ in $O(\mathsf{MM}(n, d) \log n) = \tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$

(c) $\det(A)$ in $O(\mathsf{MM}(n, d) \log^2 n) = \tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$

(d) $\operatorname{rk}(A)$ and a basis of $\ker(A)$ in $\tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$.

▷ (Generic) output sizes: $n^3 d$ for (a); $n^2 d$ for (b) and (d); nd for (c).

▷ Partic. case of (c): $A = xI_n - M$, for $M \in \mathcal{M}_n(\mathbb{K})$ $\longrightarrow$ $\chi_M(x)$ in $\tilde{O}(n^{\theta})$

▷ Open problem: is it possible to compute χ_A in $\tilde{O}(n^{\theta}\, d)$?

▷ Similar results for integer matrices: if $A \in \mathcal{M}_n(\mathbb{Z})$ with $|a_{ij}| \leq 2^{\ell}$, then product/det/system solving in $\tilde{O}(n^{\theta}\, \ell)$ binary ops. [Storjohann, 2005]

▷ [Zhou, Labahn, Storjohann, 2015] improved (a) to $\tilde{O}(n^3\, d)$.

Main results

Theorem [complexity results for polynomial matrices]

Let $\mathbb{K}$ be a field, $n, d \in \mathbb{N}$, and θ a feasible exponent for product in $\mathcal{M}_n(\mathbb{K})$. Let $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ and $b \in \mathbb{K}[x]^n_{<d}$. Then one can compute:

(a) A^{-1} in $O(\mathsf{MM}(n, nd)) = \tilde{O}(n^{\theta+1}\, d)$ ops. in $\mathbb{K}$

(b) $y \in \mathcal{M}_n(\mathbb{K}(x))$ s.t. $Ay = b$ in $O(\mathsf{MM}(n, d) \log n) = \tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$

(c) $\det(A)$ in $O(\mathsf{MM}(n, d) \log^2 n) = \tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$

(d) $\mathrm{rk}(A)$ and a basis of $\ker(A)$ in $\tilde{O}(n^{\theta}\, d)$ ops. in $\mathbb{K}$.

▷ Main new algorithmic ideas:

for (a) Newton iteration + Padé approximation (1979)

for (b) and (c) Storjohann's algorithm (2002)

- high order lifting (sort of binary powering)
- generalized Keller-Gehrig iterations

for (d) Storjohann-Villard algorithm (2005)

Inversion of polynomial matrices

[Moenck-Carter, 1979]

Input: $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ with $A(0)$ invertible

Output: $A^{-1} \in \mathcal{M}_n(\mathbb{K}(x))$

(i) Expand A^{-1} in power series to precision $N = 2nd$ using Newton iteration

$$Y_0 = A_0^{-1}, \quad Y_{\kappa+1} = Y_\kappa + Y_\kappa(I_n - AY_\kappa) \mod x^{2^{\kappa+1}}$$

(ii) Reconstruct the n^2 entries of A^{-1} in $\mathbb{K}(x)$ using Padé approximation

▷ Correctness:

- entries of A^{-1} write $Q(x)/D(x)$, with $\deg Q \leq (n-1)d$ and $\deg D \leq nd$
- If such a $Q(x)/D(x)$ is known to prec. x^{2nd}, then a Padé approx. R/V of type $((n-1)d, nd)$ will recover it: $Q/D \equiv R/V[x^{2nd}]$ implies $Q/D = R/V$

▷ Complexity:

(i) $\mathsf{C}(N) = \mathsf{C}(\frac{N}{2}) + O(\mathsf{MM}(n,N)) \implies \mathsf{C}(N) = O(\mathsf{MM}(n,N)) = \tilde{O}(n^{\theta+1}\, d)$

(ii) n^2 Padé approximants $\qquad O(n^2\mathsf{M}(N)\log N) = \tilde{O}(n^3\, d)$ (quasi-optimal)

Linear system solving, first ideas and notation

▷ If $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ invertible and $b \in \mathbb{K}[x]^n_{<d}$, then $y = A^{-1}b$ has size $n^2 d$

▷ Thus, cannot afford to compute the whole inverse A^{-1} (size $n^3 d$)

▷ Instead, we will use (a variant of) the expansion/reconstruction method

Useful compact notation for what follows:

- $P = x^d$
- $N = 2nd$
- for a matrix W in $\mathbb{K}[[x]]^{p\times q}$, write $\left\{ \, W \, \right\}$ for the coefficient of P^1 in W:

$$\text{if } \; W = W_0 + W_1\, P + W_2\, P^2 + \cdots \; \text{ for } W_i \in \mathbb{K}[x]^{p\times q}_{<d} \; \text{ then } \left\{ \, W \, \right\} := W_1$$

Linear system solving for polynomial matrices

[Moenck-Carter, 1979]

Lemma Let $P = x^d$, and write $A^{-1} = \sum_{i \geq 0} C_i P^i$ and $y = A^{-1}b = \sum_{i \geq 0} c_i P^i$, for polynomial matrices $C_i \in \mathcal{M}_n\left(\mathbb{K}[x]_{<d}\right)$ and polynomial vectors $c_i \in \mathbb{K}[x]^n_{<d}$.

Then $C_0 = A^{-1} \bmod P$, $c_0 = C_0 b \bmod P$ and for all $s \geq 0$:

$$C_{s+1} = -C_0 \cdot \left\{A \cdot C_s\right\} \mod P \quad \text{and} \quad c_{s+1} = -C_0 \cdot \left\{A \cdot c_s\right\} \mod P$$

Proof:

- $c_0 = A^{-1}b \mod P = C_0 b \mod P$
- Extracting the coefficient of P^{s+1} in

$$I_n = AC_0 + AC_1 P + \cdots + AC_s P^s + AC_{s+1} P^{s+1} \cdots$$

yields $O_n = \left\{A \cdot C_s\right\} + \left(A \cdot C_{s+1} \mod P\right)$, thus

$$C_{s+1} = -A^{-1}\left\{A \cdot C_s\right\} \mod P \;=\; -C_0\left\{A \cdot C_s\right\} \mod P.$$

Linear system solving for polynomial matrices

[Moenck-Carter, 1979]

Input: $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ invertible, with $A(0)$ invertible

Output: $y = A^{-1}b \in \mathbb{K}(x)^n$

(i) Compute C_0 and c_0 using Newton iteration

(ii) For $s = 0, 1, \ldots, \frac{N}{d} = 2n$, compute $c_{s+1} = -C_0 \cdot \left\{A \cdot c_s\right\} \mod P$

(iii) Reconstruct the n entries of $y = A^{-1}b$ in $\mathbb{K}(x)$ using Padé approximation

▷ Correctness: same argument as for inversion via expansion/reconstruction

▷ Complexity:

(i) Newton iteration $O(\mathsf{MM}(n,d))$

(ii) $O(n)$ polynomial matrix-vector products $O(n \cdot n^2\mathsf{M}(d))$

(iii) n Padé approximants $O(n\mathsf{M}(N)\log N) = \tilde{O}(n^2\, d)$ (quasi-optimal)

Total: $O(n^3\mathsf{M}(d))$ (dominant step is (ii))

High order lifting: statement

Problem: Given an invertible polynomial matrix A of degree d, compute the high order components $(C_0, C_1), (C_2, C_3), (C_6, C_7), (C_{14}, C_{15}), \dots$ in the Taylor expansion of its inverse

$$A^{-1} = \sum_{i \geq 0} C_i P^i, \qquad \text{with} \quad P = x^d, \quad C_i \in \mathcal{M}_n\left(\mathbb{K}[x]_{<d}\right)$$

Particular cases:

- If $d = 1$ and $A = I_n - xM$, then $C_i = M^i$, and the high order components can be computed fast by binary powering $O(\mathsf{MM}(n)\,\log n)$
- If $n = 1$, then high order component $=$ N-th term of a recurrent sequence $\longrightarrow$ can be computed fast by Fiduccia's algorithm $O(\mathsf{M}(d)\,\log n)$

Upcoming: Storjohann's algorithm $O(\mathsf{MM}(n, d)\,\log n)$

Generalized Newton identity

Theorem (generalized Newton identity) The following holds modulo P^{s+t+2} :

$$C_{s+1}P^{s+1}+\cdots+C_{s+t+1}P^{s+t+1} = \left(C_0+\cdots+C_tP^t\right)\cdot(I_n - A\cdot(C_0+\cdots+C_sP^s))$$

Particular case: If $s = t = 2^i$, we recover a Newton-type iteration

Proof:

$$I_n - A(C_0 + C_1P + \cdots + C_sP^s) = AP^{s+1}(C_{s+1} + C_{s+2}P + \cdots)$$

$$\implies \quad \text{RHS} \ = \underbrace{(C_0 + C_1P + \cdots + C_tP^t)A}_{I-P^{t+1}(C_{t+1}+\cdots)A}\cdot P^{s+1}(C_{s+1} + C_{s+2}P + \cdots)$$

$$= \text{LHS mod } P^{s+t+2}$$

High order lifting: algorithm

[Storjohann, 2002]

Corollary (Storjohann 2002): For all $s, t \geq 0$:

$$C_{s+t+1} = -\{\ \big(C_{t-1} + C_t P\big) \cdot \{A \cdot C_s\}\ \}$$

Recall: $\{\ B\ \}$ denotes the coefficient of P^1 in B.

Corollary (Storjohann 2002): For all $i \geq 2$, the following equalities hold

$$\begin{cases} C_{2^i-2} & = \ -\ \{\ \big(C_{2^{i-1}-2} + C_{2^{i-1}-1}P\big) \cdot \{A \cdot C_{2^{i-1}-2}\}\ \} \\ C_{2^i-1} & = \ -\ \{\ \big(C_{2^{i-1}-2} + C_{2^{i-1}-1}P\big) \cdot \{A \cdot C_{2^{i-1}-1}\}\ \}, \end{cases}$$

and they allow to compute the *high order components*

$$(C_0, C_1) \to (C_2, C_3) \to (C_6, C_7) \to (C_{14}, C_{15}) \to \dots$$

Cost: $O(\mathsf{MM}(n,d))$ ops. per arrow $\qquad O(\mathsf{MM}(n,d)\ \log n)$

Generalizes simultaneously binary powering ($d = 1$) and Fiduccia ($n = 1$)

Example (fast computation of Fibonacci numbers)

$$\frac{1}{1-x-x^2} = C_0 + C_1 P + C_2 P^2 + \cdots, \qquad P = x^2, \quad C_n = F_{2n} + F_{2n+1} x$$

The Storjohann identities become

$$\begin{cases} C_{2^i-2} & = & -\left\{ \left(C_{2^{i-1}-2} + C_{2^{i-1}-1} x^2\right) \cdot \left\{ \left(1-x-x^2\right) \cdot C_{2^{i-1}-2} \right\} \right\}, \\ C_{2^i-1} & = & -\left\{ \left(C_{2^{i-1}-2} + C_{2^{i-1}-1} x^2\right) \cdot \left\{ \left(1-x-x^2\right) \cdot C_{2^{i-1}-1} \right\} \right\}, \end{cases}$$

and allow to compute the *high order components*

$$(F_0, F_1, F_2, F_3) \to (F_4, F_5, F_6, F_7) \to (F_{12}, F_{13}, F_{14}, F_{15}) \to \ldots \quad \text{by}$$

$$\begin{cases} F_{2^{i+1}-2} = F_{2^i-2} \cdot F_{2^i} + F_{2^i-1} \cdot F_{2^i-3}, \\ F_{2^{i+1}-4} = F_{2^i-2}^2 + F_{2^i-3}^2, \\ F_{2^{i+1}-1} = F_{2^i-1} \cdot F_{2^i} + F_{2^i-2} \cdot F_{2^i-1}, \\ F_{2^{i+1}-3} = F_{2^i-1} \cdot F_{2^i-2} + F_{2^i-2} \cdot F_{2^i-3}, \end{cases} \iff \underbrace{\begin{cases} F_{2n-2} = F_{n-2}^2 + F_{n-1}^2 \\ F_{2n-1} = F_{n-1}^2 + 2F_{n-1}F_{n-2} \end{cases}}_{\text{Shortt's algorithm (1978)}}$$

Keller-Gehrig algorithm

Problem: Given $M \in \mathcal{M}_n(\mathbb{K})$ and $v \in \mathbb{K}^n$, compute the Krylov sequence

$$\mathcal{K} = \Big(v, \quad Mv, \quad M^2v, \quad \dots, \quad M^{n-1}v\Big)$$

Interest: If M is *generic*, $\mathcal{K}$ forms a basis of $\mathbb{K}^n$, and the matrix C of $v \mapsto Mv$ w.r.t. $\mathcal{K}$ is companion $\Longrightarrow$ the characteristic polynomial $\det(xI_n - M)$ reads off $C = P^{-1}MP$, where $P = \left[\, v \mid Mv \mid \cdots \mid M^{n-1}v \,\right]$.

Naive algorithm: Compute iteratively $v_{i+1} = Mv_i$, $v_0 = v$.

Cost: $O(n)$ matrix-vector products $\longrightarrow$ $O(n^3)$ ops. in $\mathbb{K}$.

Keller-Gehrig algorithm (1985) Compute

(1) $M_0 = M$, $M_1 = M^2$, $M_2 = M^4$, $M_3 = M^8, \dots$ (binary powering)

(2) $\left[\, M^{2^k}v \mid \cdots \mid M^{2^{k+1}-1}v \,\right] := M_k \times \left[\, v \mid Mv \mid \cdots \mid M^{2^k-1}v \,\right], \quad k \geq 0$

Cost: $O(\log n)$ matrix products for both (1) and (2) $\longrightarrow$ $O(\mathsf{MM}(n) \log n)$

Solving linear systems with polynomial coefficients

Problem: Given $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ invertible, and $b \in \mathbb{K}[x]^n_{<d}$, compute $A^{-1}b$

[Moenck-Carter, 1979]: One can recover the exact solution $y = A^{-1}b \in \mathbb{K}(x)^n$ of $Ay = b$ from its approximation

$$y_{2nd} = A^{-1}b \bmod x^{2nd} = c_0 + c_1 P + c_2 P^2 + \cdots + c_{2n} P^{2n}, \qquad \text{where } \ c_i \in \mathbb{K}[x]^n_{<d}$$

Conversion $y_{2nd} \to y$ by Padé approximation $\qquad O(n\,\mathsf{M}(nd)\log(nd)) = \tilde{O}(n^2 d)$

Particular case: If $d = 1$ and $A = I_n - xM$, with $M \in \mathcal{M}_n(\mathbb{K})$ and $b \in \mathbb{K}^n$, then $c_i = M^i b$, and $c_0, \ldots, c_{2n}$ can be computed fast by the Keller-Gehrig algorithm.

Storjohann's algorithm

Pb: Given $A \in \mathcal{M}_n(\mathbb{K}[x]_{\leq d})$ invertible, $b \in \mathbb{K}[x]^n_{<d}$, compute $A^{-1}b \bmod x^{2nd}$

Theorem (Storjohann 2002): For all $s, t \geq 0$:

$$c_{s+t+1} = -\{\ (C_{t-1} + C_t P) \cdot \{A \cdot c_s\}\ \}$$

Recall: $\{\ v\ \}$ denotes the coefficient of P^1 in $v \in \mathbb{K}[x]^n$.

Corollary (Storjohann 2002): For all $i \geq 2$, the following equality holds

$$\Big[\ c_{2^k} \mid \cdots \mid c_{2^{k+1}-1}\ \Big] = -\Big\{\Big(C_{2^k-2} + C_{2^k-1}P\Big) \cdot \Big\{A \cdot \Big[\ c_0 \mid \cdots \mid c_{2^k-1}\Big]\Big\}\Big\},$$

which allows to compute

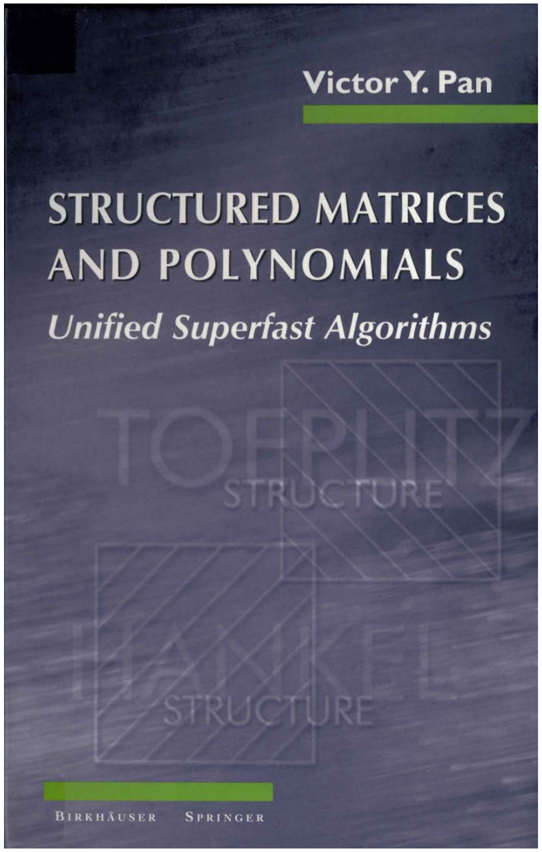
$$(c_0, c_1) \to (c_2, c_3) \to (c_4, c_5, c_6, c_7) \to (c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}) \to \ldots$$

Generalizes Keller-Gehrig's algorithm

Cost: $O(\mathsf{MM}(n,d))$ ops. per arrow $\qquad O(\mathsf{MM}(n,d)\ \log n)$

Structured Matrices

Structured Matrices

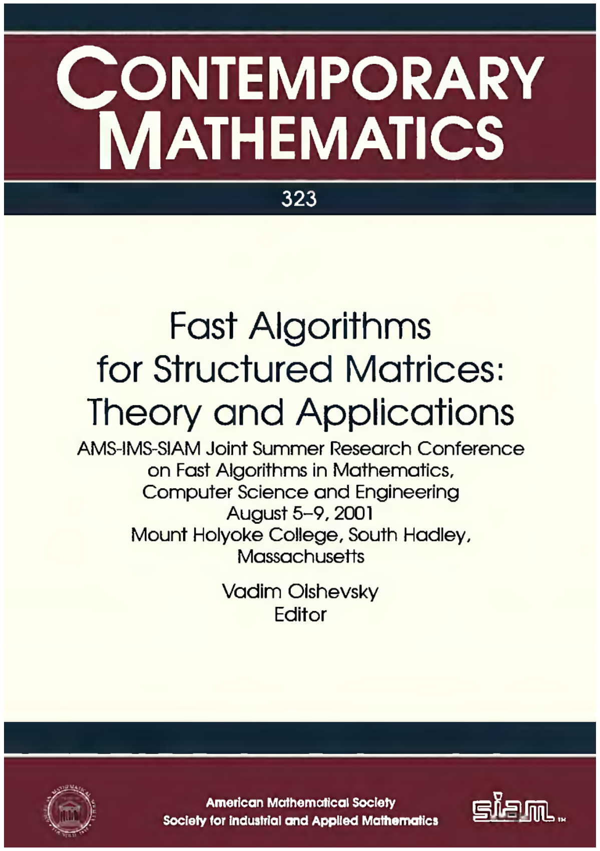


Victor Y. Pan

STRUCTURED MATRICES AND POLYNOMIALS

Unified Superfast Algorithms

BIRKHÄUSER SPRINGER



CONTEMPORARY MATHEMATICS

323

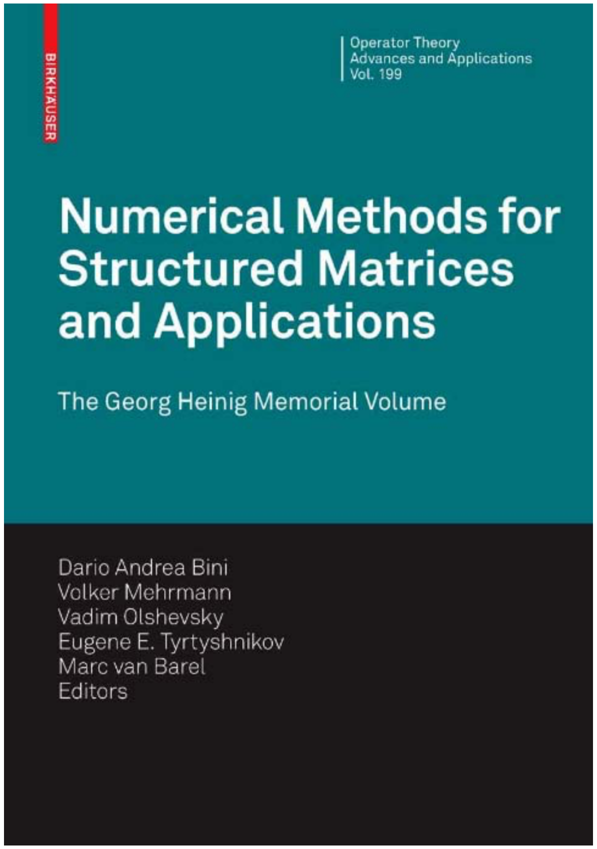
Fast Algorithms for Structured Matrices: Theory and Applications

AMS-IMS-SIAM Joint Summer Research Conference on Fast Algorithms in Mathematics, Computer Science and Engineering
August 5–9, 2001
Mount Holyoke College, South Hadley, Massachusetts

Vadim Olshevsky
Editor

American Mathematical Society
Society for Industrial and Applied Mathematics

siam



BIRKHÄUSER

Operator Theory
Advances and Applications
Vol. 199

Numerical Methods for Structured Matrices and Applications

The Georg Heinig Memorial Volume

Dario Andrea Bini
Volker Mehrmann
Vadim Olshevsky
Eugene E. Tyrtyshnikov
Marc van Barel
Editors

Motivation

Problem 1. (Hermite-Padé approximation) Recognize that

$$y = 1 + 2\,x - \frac{1}{2}x^2 + \frac{5}{24}x^4 - \frac{3}{20}x^5 + \frac{67}{720}x^6 - \frac{73}{1260}x^7 + \frac{1577}{40320}x^8 + O\left(x^9\right)$$

satisfies $(1+x)y'' + (1-x)y = 0$.

Problem 2. (Interpolation) Let $P \in \mathbb{Q}[x,y]$ of total degree ≤ 2, such that

$P(0,0)=1,\ P(0,1)=2,\ P(0,2)=1,\ P(1,4)=13,\ P(1,-1)=-2,\ P(2,3)=36.$

Find that

$$P = 1 + x + 2y + 3x^2 + 4xy - y^2.$$

These are linear algebra problems, with a lot of structure!

Basic algorithms in linear algebra

Classical approach – Gaussian elimination.

- Most questions of linear algebra in size n (matrix product, inverse, system solving, characteristic polynomial, ...) can be solved in $O(n^3)$ operations.

Faster algorithms – from Strassen (1969) to Alman & Williams (2020).

- Strassen'69: $n \times n$ matrices can be multiplied in $O(n^\theta)$ operations, $\theta < 3$.
- As of now, one can take $\theta \leq 2.37286$, even though the algorithms are quite impractical (**huge** logarithmic factors and constants hidden in the $O(\ )$).
- Most problems in linear algebra can be solved in time $O(n^\theta)$.

▷ However, none of these algorithms takes structure into account.

Toeplitz matrices

A Toeplitz matrix is invariant along its main diagonals:

$$A = \begin{bmatrix} c & d & e \\ b & c & d \\ a & b & c \end{bmatrix}.$$

Crucial remark: The Toeplitz displacement operator ϕ:

$$\phi(A) = A - (A \text{ shifted right and down by } 1) = \begin{bmatrix} c & d & e \\ b & 0 & 0 \\ a & 0 & 0 \end{bmatrix}$$

is such that $\phi(A)$ has rank $\alpha \leq 2$.

Compact representation

The matrix

$$\phi(A) = \begin{bmatrix} c & d & e \\ b & 0 & 0 \\ a & 0 & 0 \end{bmatrix}$$

can be represented in a compact way as

$$\phi(A) = GH^T, \quad \text{with} \quad G = \begin{bmatrix} c & d \\ b & 0 \\ a & 0 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & e/d \end{bmatrix}.$$

▷ This feature can be used to obtain algorithms of complexity $\tilde{O}(n)$ for solving the system $Ax = b$ ($\tilde{O}$ means that (poly-)logarithmic factors are hidden).

Def. The rank α of $\phi(A)$ is called the displacement rank of A;

Def. $G, H \in \mathbb{K}^{n\times\alpha}$ are called generators of A, of length α.

More structure . . .

Toeplitz structure: $\begin{bmatrix} c & d & e \\ b & c & d \\ a & b & c \end{bmatrix}$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

Hankel structure: $\begin{bmatrix} e & d & c \\ d & c & b \\ c & b & a \end{bmatrix}$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

Hankel structure:

$$\phi(A) = A - (A \text{ shifted left and down by } 1)$$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

Hankel structure:

$$\phi(A) = A - (A \text{ shifted left and down by } 1)$$

Vandermonde structure: $\begin{bmatrix} 1 & a & a^2 \\ 1 & b & b^2 \\ 1 & c & c^2 \end{bmatrix}$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

Hankel structure:

$$\phi(A) = A - (A \text{ shifted left and down by } 1)$$

Vandermonde structure:

$$\phi(A) = A - (\text{diagonal matrix}) \times (A \text{ shifted right by } 1)$$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

Hankel structure:

$$\phi(A) = A - (A \text{ shifted left and down by } 1)$$

Vandermonde structure:

$$\phi(A) = A - (\text{diagonal matrix}) \times (A \text{ shifted right by } 1)$$

Cauchy structure: $\begin{bmatrix} 1/(a-x) & 1/(a-y) & 1/(a-z) \\ 1/(b-x) & 1/(b-y) & 1/(b-z) \\ 1/(c-x) & 1/(c-y) & 1/(c-z) \end{bmatrix}$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

Hankel structure:

$$\phi(A) = A - (A \text{ shifted left and down by } 1)$$

Vandermonde structure:

$$\phi(A) = A - (\text{diagonal matrix}) \times (A \text{ shifted right by } 1)$$

Cauchy structure:

$$\phi(A) = A - (\text{diagonal matrix}) \times A \times (\text{diagonal matrix})'$$

More structure . . .

Toeplitz structure:

$$\phi(A) = A - (A \text{ shifted right and down by } 1)$$

Hankel structure:

$$\phi(A) = A - (A \text{ shifted left and down by } 1)$$

Vandermonde structure:

$$\phi(A) = A - (\text{diagonal matrix}) \times (A \text{ shifted right by } 1)$$

Cauchy structure:

$$\phi(A) = A - (\text{diagonal matrix}) \times A \times (\text{diagonal matrix})'$$

Def. In all these cases, the displacement rank α of A is the rank of $\phi(A)$.

Def. If $\alpha \ll n$, the matrix A is called quasi-Toeplitz, quasi-Hankel,. . .

Common features

All these classes of $n \times n$ matrices share two nice features:

- ▷ They can be represented using $O(n)$ elements
- ▷ Their product by a vector can be computed in quasi-linear time. E.g.,
  - Toeplitz × vector $\longrightarrow$ for $0 \le i \le n-1$, the entry c_i of

$$\begin{pmatrix} a_{n-1} & \cdots & a_0 \\ \vdots & \ddots & \vdots \\ a_{2n-2} & \cdots & a_{n-1} \end{pmatrix} \times \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix}$$

  is the coefficient of x^{n-1+i} in the polynomial product $O(\mathsf{M}(n))$

$$(a_0 + \cdots + a_{2n-2}x^{2n-2}) \times (b_0 + \cdots + b_{n-1}x^{n-1}).$$

  - Vandermonde × vector $\longrightarrow$ multipoint evaluation $O(\mathsf{M}(n)\log n)$
  - Cauchy × vector $\longrightarrow$ Ex. 3, next slide $O(\mathsf{M}(n)\log n)$

Ex. 3: Product of a Cauchy matrix by a vector

Show that the product Cv of a Cauchy matrix $C \in \mathcal{M}_n(\mathbb{K})$ by a vector $v \in \mathbb{K}^n$ can be performed in $O(\mathsf{M}(n)\log n)$ operations in $\mathbb{K}$.

Common features: pros and cons

All these classes of $n \times n$ matrices share two nice features:

(1) They can be represented using $O(n)$ elements

(2) Their product by a vector can be computed in quasi-linear time. E.g.,

- Toeplitz $(t_{i-j})_{i,j}$ × vector $\longrightarrow$ polynomial product $O(\mathsf{M}(n))$
- Vandermonde $(x_i^j)_{i,j}$ × vector $\longrightarrow$ multipoint evaluation $O(\mathsf{M}(n)\log n)$
- $(\frac{1}{x_i - y_j})_{i,j}$ × vector $\longrightarrow$ Ex.1, previous slide $O(\mathsf{M}(n)\log n)$

▷ Unfortunately, (1) + (2) alone *do not directly allow* quasi-linear algorithms.

▷ A Wiedemann-type approach provides algorithms in $\tilde{O}(n^2)$.

▷ Moreover, the previous classes of structured matrices (Toeplitz, Hankel, Vandermonde, Cauchy, . . .) are *not closed under inversion*.

▷ The salvation (good algorithmic definition of the structure) comes from the notion of displacement rank: it measures *how far A is from being Toeplitz*

Displacement rank and generators

Def. The *displacement operator* ϕ_+ is the map $A \mapsto A - Z \cdot A \cdot Z^T$, where

$$Z = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \ddots & \dots & \vdots \\ 0 & \cdots & 1 & 0 \end{pmatrix}.$$

▷ $ZA =$ (A shifted down by 1) and $A \cdot Z^T =$ (A shifted right by 1)

Def. The *displacement rank* of $A \in \mathcal{M}_n(\mathbb{K})$ is

$$\alpha_+(A) := \operatorname{rank}(\phi_+(A)).$$

Def. If $\alpha := \alpha_+(A) \ll n$, we say that A is *quasi-Toeplitz*.

Def. A *displacement generator* of A (for ϕ_+) is a pair $(G, H) \in \mathcal{M}_{n,\alpha}(\mathbb{K})$ s.t.

$$\phi_+(A) = G \cdot H^T.$$

Def. The integer $\alpha := \alpha_+(A)$ is called *length of the generator* (G, H).

Displacement rank and generators, cont.

One can similarly define *quasi-Vandermonde* and *quasi-Cauchy* matrices, but for different displacement operators:

- $\mathbf{V_a} \mapsto \mathbf{V_a} - \mathrm{Diag}_{\mathbf{a}} \cdot \mathbf{V_a} \cdot Z^T$ for the Vandermonde case,
- $\mathbf{C_{a,b}} \mapsto \mathbf{C_{a,b}} - \mathrm{Diag}_{\mathbf{a}}{}^{-1} \cdot \mathbf{C_{a,b}} \cdot \mathrm{Diag}_{\mathbf{b}}$ for the Cauchy case,

where $\mathbf{V_a} = (a_i^j)_{i,j}$, $\mathbf{C_{a,b}} = (1/(a_i - b_j))_{i,j}$ and $\mathrm{Diag}_{\mathbf{a}}$ has diagonal $\mathbf{a} = (a_i)_i$

▷ In all these cases, the map $\phi_{M,N}$ is defined by $\phi_{M,N}(A) = A - MAN$, with M and N well-chosen (in terms of the target structure), and the displacement rank of A is defined as the usual rank of $\phi_{M,N}(A)$.

▷ Ex.: compute the displacement ranks of Vandermonde and Cauchy matrices.

Main results

Morf, Bitmead & Anderson, Pan, Kaltofen, Gohberg & Olshevsky, . . .

Theorem. Let ϕ be one of Toeplitz, Hankel, Vandermonde, Cauchy operators.

Let A be in $\mathbb{K}^{n \times n}$, given by generators of length α, and let b be in $\mathbb{K}^n$.

One can compute $\det(A)$ and a random solution to the system $Ax = b$, or prove that no such solution exists, in (Las Vegas) time $\tilde{O}(\alpha^2 n)$.

Main results

Morf, Bitmead & Anderson, Pan, Kaltofen, Gohberg & Olshevsky, . . .

Theorem. Let ϕ be one of Toeplitz, Hankel, Vandermonde, Cauchy operators.

Let A be in $\mathbb{K}^{n\times n}$, given by generators of length α, and let b be in $\mathbb{K}^n$.

One can compute $\det(A)$ and a random solution to the system $Ax = b$, or prove that no such solution exists, in (Las Vegas) time $\tilde{O}(\alpha^2 n)$.

Remarks.

- For $\alpha = 2$ (or more generally α constant), this is $\tilde{O}(n)$, which is optimal, up to log factors $\longrightarrow$ quasi-optimal gcd, resultant, Padé approximation,. . .
- For large α, not so good: when $\alpha \simeq n$, cost $\tilde{O}(n^3)$, worse than the cost $O(n^\theta)$ of generic linear algebra algorithms.
- [B., Jeannerod, Mouilleron, Schost, 2017]: Improvement to $\tilde{O}(\alpha^{\theta-1} n)$

Some application examples

Hermite-Padé approximation. Given power series $f_1, \dots, f_m$ at precision σ, degree bounds d_i, one can find in time $\tilde{O}(m^{\theta-1}\sigma)$ polynomials $p_1, \dots, p_m$ s.t.

$$\deg(p_i) \le d_i \quad \text{and} \quad \sum p_i f_i = O(x^\sigma) \quad \text{with} \quad \sigma = \sum (d_i + 1) - 1$$

Some application examples

Hermite-Padé approximation. Given power series $f_1, \ldots, f_m$ at precision σ, degree bounds d_i, one can find in time $\tilde{O}(m^{\theta-1}\sigma)$ polynomials $p_1, \ldots, p_m$ s.t.

$$\deg(p_i) \leq d_i \quad \text{and} \quad \sum p_i f_i = O(x^\sigma) \quad \text{with} \quad \sigma = \sum (d_i + 1) - 1$$

Generalized simultaneous Hermite-Padé approximation. Given a vector of polynomials $\mathbf{P} \in \mathbb{K}[x]^s$ of degree $\leq \sigma/s$ and m vectors $\mathbf{f}_1, \ldots, \mathbf{f}_m$ of polynomials in $\mathbb{K}[x]^s$ of degree $< \sigma/s$, one can find in time $\tilde{O}(m^{\theta-1}\sigma)$ polynomials $p_1, \ldots, p_m$ such that

$$\deg(p_i) < \sigma/m \qquad \text{and} \qquad \sum p_i \mathbf{f}_i = 0 \mod \mathbf{P}.$$

Some application examples

Bivariate interpolation. Given values of a degree-d polynomial $P(x, y)$ at points

$$(a_i, b_j) \quad 0 \leq i + j \leq d,$$

one can recover its coefficients in time $\tilde{O}(d^{\theta+1})$, which is sub-quadratic in the number of terms

(generally, interpolation problems whose monomial support indexes the sample points).

Toeplitz-block-Toeplitz systems.

Let $n = pq$ and let A be block-Toeplitz, with p^2 blocks of size q that are Toeplitz. One can solve the system $Ax = b$ in $\tilde{O}(n^{\frac{\theta+1}{2}})$ operations.

Quasi-Toeplitz matrices: the strategy

▷ To be able to exploit the compact representation by generators, we need to:

- multiply a quasi-Toeplitz matrix by a vector fast $\tilde{O}(n)$

  $\longrightarrow$ as fast as Toeplitz matrices, using the $\sum LU$ formula

- prove that the sum and product of quasi-Toeplitz matrices is still quasi-Toeplitz, and that they can be computed fast $\tilde{O}(n)$

  $\longrightarrow$ algorithms relying again on the $\sum LU$ formula

- prove that the inverse of a quasi-Toeplitz matrix is still quasi-Toeplitz, and that it can be computed fast $\tilde{O}(n)$

  $\longrightarrow$ Strassen-like inversion algorithm, adapted to the structured case

Inverse of a quasi-Toeplitz is quasi-Toeplitz

Need to introduce a *dual* displacement operator ϕ_-:

$$\phi_-(A) = A - Z^T \cdot A \cdot Z = A - (A \text{ shifted up and left by } 1).$$

The dual displacement rank α_- is defined as

$$\alpha_-(A) = \text{rank}(\phi_-(A)).$$

Theorem. If A is quasi-Toeplitz and invertible, then A^{-1} is also quasi-Toeplitz.

Proof. Consequence of $|\alpha_+(A) - \alpha_-(A)| \leq 2$ (Exercise!) and

$$\begin{aligned}\alpha_-(A) &= \text{rank}(A - Z^T \cdot A \cdot Z) = \text{rank}(I_n - A^{-1} \cdot Z^T \cdot A \cdot Z) \\ &= \text{rank}(I_n - Z \cdot A^{-1} \cdot Z^T \cdot A) = \text{rank}(A^{-1} - Z \cdot A^{-1} \cdot Z^T) = \alpha_+(A^{-1}),\end{aligned}$$

where we have used that $\text{rank}(I_n - A \cdot B) = \text{rank}(I_n - B \cdot A)$ (Exercise!)

Recall: Inversion of dense matrices

[Strassen, 1969]

To invert a dense matrix $A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \in \mathbb{K}^{n \times n}$:

1. Invert $A_{1,1}$ (recursively).
2. Compute the Schur complement $\Delta := A_{2,2} - A_{2,1} A_{1,1}^{-1} A_{1,2}$.
3. Invert Δ (recursively).
4. Recover the inverse of A as

$$A^{-1} = \begin{bmatrix} I & -A_{1,1}^{-1} A_{1,2} \\ & I \end{bmatrix} \times \begin{bmatrix} A_{1,1}^{-1} & \\ & \Delta^{-1} \end{bmatrix} \times \begin{bmatrix} I & \\ -A_{2,1} A_{1,1}^{-1} & I \end{bmatrix}$$

Complexity: $C(n) = 2C(\frac{n}{2}) + \mathsf{O}(\mathsf{n}^{\theta})$.

Corollary: $A^{-1}b$ in time $O(n^{\theta})$.

Inversion of quasi-Toeplitz matrices

[Morf, 1980], [Bitmead & Anderson, 1980], [Kaltofen 1994], [Pan 2001]

To find generators of the inverse of a quasi-Toeplitz $A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \in \mathbb{K}^{n \times n}$

1. Compute generators of the inverse of $A_{1,1}$ (recursively).
2. Compute generators of Δ.
3. Compute generators of the inverse of Δ (recursively).
4. Compute generators of the inverse of A (by Strassen's formula).

Complexity: If A is given by generators of length α,

$$C(n, \alpha) = 2C\left(\frac{n}{2}, \alpha\right) + O(\mathsf{K}(n, \alpha)) + \tilde{O}(\alpha^{\theta-1} n),$$

where $\mathsf{K}(n, \alpha)$ is the cost of quasi-Toeplitz matrix multiplication, for $n \times n$ matrices given by generators of size α. Upcoming: $\mathsf{K}(n, \alpha) = \tilde{O}(\alpha^2 n)$

$\sum LU$ formula for quasi-Toeplitz matrices

Let $A \in \mathcal{M}_n(\mathbb{K})$ be quasi-Toeplitz of displacement rank $\alpha = \phi(A)$. Then,

$$\phi(A) = A - Z \cdot A \cdot Z^T = GH^T = \sum_{j=1}^{\alpha} g_j \cdot h_j^T,$$

where $g_j, h_j \in \mathbb{K}^n$ are the columns of the displacement generators G and H.

Theorem ($\sum LU$ formula). One can recover A from its generators:

$$A = \sum_{j=1}^{\alpha} L(g_j) \cdot U(h_j), \qquad \text{with}$$

$$L(g_j) = \begin{bmatrix} g_{j,1} & & & \\ g_{j,2} & g_{j,1} & & \\ \vdots & \ddots & \ddots & \\ g_{j,n} & g_{j,n-1} & \cdots & g_{j,1} \end{bmatrix} \qquad \text{and} \qquad U(h_j) = \begin{bmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ & h_{1,j} & \ddots & h_{n-1,j} \\ & & \ddots & \vdots \\ & & & h_{1,j} \end{bmatrix}$$

Proof. By linearity, it is sufficient to treat the case $\alpha = 1$. If $A = L(g) \cdot U(h)$, then $a_{i,j} = g_i h_j + g_{i-1} h_{j-1} + \cdots$, so $a_{i,j} - a_{i-1,j-1} = g_i h_j$ and $\phi_+(A) = g \cdot h^T$.

$\sum LU$ formula for quasi-Toeplitz matrices

Let $A \in \mathcal{M}_n(\mathbb{K})$ be quasi-Toeplitz of displacement rank $\alpha = \phi(A)$. Then,

$$\phi(A) = A - Z \cdot A \cdot Z^T = GH^T = \sum_{j=1}^{\alpha} g_j \cdot h_j^T,$$

where $g_j, h_j \in \mathbb{K}^n$ are the columns of the displacement generators G and H.

Theorem ($\sum LU$ formula). One can recover A from its generators:

$$A = \sum_{j=1}^{\alpha} L(g_j) \cdot U(h_j), \qquad \text{with}$$

$$L(g_j) = \begin{bmatrix} g_{j,1} & & & \\ g_{j,2} & g_{j,1} & & \\ \vdots & \ddots & \ddots & \\ g_{j,n} & g_{j,n-1} & \cdots & g_{j,1} \end{bmatrix} \qquad \text{and} \qquad U(h_j) = \begin{bmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ & h_{1,j} & \ddots & h_{n-1,j} \\ & & \ddots & \vdots \\ & & & h_{1,j} \end{bmatrix}$$

Corollary. Products Av and A^Tv can be computed in $O(\alpha\,\mathsf{M}(n))$ ops.

Proof. If $v \in \mathbb{K}^n$, then $L(g)U(h)v \equiv \mathsf{g}(x)\,(\mathsf{h}(x)\mathsf{v}(x) \bmod x^n)\ \operatorname{div}\ x^{n-1}$ $O(\mathsf{M}(n))$

$\sum LU$ formula for quasi-Toeplitz matrices

Let $A \in \mathcal{M}_n(\mathbb{K})$ be quasi-Toeplitz of displacement rank $\alpha = \phi(A)$. Then,

$$\phi(A) = A - Z \cdot A \cdot Z^T = GH^T = \sum_{j=1}^{\alpha} g_j \cdot h_j^T,$$

where $g_j, h_j \in \mathbb{K}^n$ are the columns of the displacement generators G and H.

Theorem ($\sum LU$ formula). One can recover A from its generators:

$$A = \sum_{j=1}^{\alpha} L(g_j) \cdot U(h_j), \qquad \text{with}$$

$$L(g_j) = \begin{bmatrix} g_{j,1} & & & \\ g_{j,2} & g_{j,1} & & \\ \vdots & \ddots & \ddots & \\ g_{j,n} & g_{j,n-1} & \cdots & g_{j,1} \end{bmatrix} \qquad \text{and} \qquad U(h_j) = \begin{bmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ & h_{1,j} & \ddots & h_{n-1,j} \\ & & \ddots & \vdots \\ & & & h_{1,j} \end{bmatrix}$$

Remark. Converse also holds: if $A = \sum_{j=1}^{\alpha} L(g_j) \cdot U(h_j)$, then $\phi(A) = GH^T$.

Proof. Consequence of the injectivity of ϕ (Exercise!)

$\sum LU$ formula for quasi-Toeplitz matrices

Let $A \in \mathcal{M}_n(\mathbb{K})$ be quasi-Toeplitz of displacement rank $\alpha = \phi(A)$. Then,

$$\phi(A) = A - Z \cdot A \cdot Z^T = GH^T = \sum_{j=1}^{\alpha} g_j \cdot h_j^T,$$

where $g_j, h_j \in \mathbb{K}^n$ are the columns of the displacement generators G and H.

Theorem ($\sum LU$ formula). One can recover A from its generators:

$$A = \sum_{j=1}^{\alpha} L(g_j) \cdot U(h_j), \qquad \text{with}$$

$$L(g_j) = \begin{bmatrix} g_{j,1} & & & \\ g_{j,2} & g_{j,1} & & \\ \vdots & \ddots & \ddots & \\ g_{j,n} & g_{j,n-1} & \cdots & g_{j,1} \end{bmatrix} \qquad \text{and} \qquad U(h_j) = \begin{bmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ & h_{1,j} & \ddots & h_{n-1,j} \\ & & \ddots & \vdots \\ & & & h_{1,j} \end{bmatrix}$$

Corollary (equivalent definition): $\alpha_+(A)$ is the least integer α s.t. A writes:

$$A = \sum_{i=1}^{\alpha} L_i U_i$$

for some lower Toeplitz matrices L_i and some upper Toeplitz matrices U_i.

Matrix operations in compact representation

Theorem. If A and B are quasi-Toeplitz, then $A + B$ and $A \times B$ also do.

Matrix operations in compact representation

Theorem. If A and B are quasi-Toeplitz, then $A+B$ and $A \times B$ also do.

Proof. Assume A and B given by generators (T, U), (G, H) of length α. Then:

(1) $\big([T \mid G],\ [U \mid H]\big)$ is a generator of length 2α for $A+B$.

(2) $\big([T \mid W \mid \mathbf{a}],\ [V \mid H \mid -\mathbf{b}]\big)$ is a generator of length $2\alpha+1$ for $A \times B$, with

- $V := B^T \times U$
- $W :=$ (A shifted right and down by 1) $\times\ G$
- $\mathbf{a}$ (resp. $\mathbf{b}$) is the down-shift of the last column of A (resp. B^T).

Matrix operations in compact representation

Theorem. If A and B are quasi-Toeplitz, then $A+B$ and $A\times B$ also do.

Proof. Assume A and B given by generators (T,U), (G,H) of length α. Then:

(1) $\big([T \mid G],\ [U \mid H]\big)$ is a generator of length 2α for $A+B$.

(2) $\big([T \mid W \mid \mathbf{a}],\ [V \mid H \mid -\mathbf{b}]\big)$ is a generator of length $2\alpha+1$ for $A\times B$, with

- $V := B^T \times U$
- $W :=$ (A shifted right and down by 1) $\times G$
- $\mathbf{a}$ (resp. $\mathbf{b}$) is the down-shift of the last column of A (resp. B^T).

Proof of (2). We have $T\cdot U^T = A - Z\cdot A\cdot Z^T, G\cdot H^T = B - Z\cdot B\cdot Z^T$ and

$V = B^T\cdot U,\ W = (Z\cdot A\cdot Z^T)\cdot G,\ \mathbf{a} = (ZA)\mathbf{e},\ \mathbf{b} = (ZB^T)\mathbf{e},\ \mathbf{e} = [0,\ldots,0,1]^T,$ so

$$T\cdot V^T + W\cdot H^T - \mathbf{a}\cdot\mathbf{b}^T = T\cdot(U^T\cdot B) + (Z\cdot A\cdot Z^T)\cdot G\cdot H^T - \mathbf{a}\cdot\mathbf{b}^T =$$

$$(A - Z\cdot A\cdot Z^T)\cdot B + (Z\cdot A\cdot Z^T)\cdot(B - Z\cdot B\cdot Z^T) - (ZA)\mathbf{e}\cdot\mathbf{e}^T(BZ^T) =$$

$$A\cdot B - (Z\cdot A)\cdot(I_n - D)\cdot(B\cdot Z^T) - (ZA)D(BZ^T) = \phi_+(AB),\ \text{where } D = \mathrm{Diag}(\mathbf{e})$$

Matrix operations in compact representation

Theorem. If A and B are quasi-Toeplitz, then $A + B$ and AB also do.

Proof. Assume A and B given by generators (T, U), (G, H) of length α. Then:

- $\big([T \mid G],\ [U \mid H]\big)$ is a generator of length 2α for $A + B$.
- $\big([T \mid W \mid \mathbf{a}],\ [V \mid H \mid -\mathbf{b}]\big)$ is a generator of length $2\alpha + 1$ for $A \times B$, with
  - $V := B^T \times U$
  - $W := (A$ shifted right and down by $1) \times G$
  - $\mathbf{a}$ (resp. $\mathbf{b}$) is the down-shift of the last column of A (resp. B^T).

Corollary. In compact representation, one can compute:

(1) the sum $A + B$ in $O(\alpha n)$ operations.

(2) the product $A \times B$ in $\mathsf{K}(n, \alpha) = \tilde{O}(\alpha^2 n)$ operations, using $\sum LU$ formula.

Proof. Computing $W, V, \mathbf{a}, \mathbf{b}$ amounts to α products quasi-Toeplitz $\times$ vector

Matrix operations in compact representation

Theorem. If A and B are quasi-Toeplitz, then $A+B$ and AB also do.

Proof. Assume A and B given by generators (T,U), (G,H) of length α. Then:

- $\big([T \mid G],\ [U \mid H]\big)$ is a generator of length 2α for $A+B$.
- $\big([T \mid W \mid \mathbf{a}],\ [V \mid H \mid -\mathbf{b}]\big)$ is a generator of length $2\alpha+1$ for $A \times B$, with

  (1) $V := B^T \times U$

  (2) $W := (A$ shifted right and down by $1) \times G$

  - $\mathbf{a}$ (resp. $\mathbf{b}$) is the down-shift of the last column of A (resp. B^T).

Corollary. In compact representation, one can compute:

- the sum $A+B$ in $O(\alpha n)$ operations.
- the product $A \times B$ in $\mathsf{K}(n,\alpha) = \tilde{O}(\alpha^2 n)$ operations, using $\sum LU$ formula.

$\longrightarrow$ The cost $\mathsf{K}(n,\alpha)$ for × can be lowered to $\tilde{O}(\alpha^{\theta-1} n)$ operations.

SIAM J. MATRIX ANAL. APPL.
Vol. 38, No. 3, pp. 733–775

ON MATRICES WITH DISPLACEMENT STRUCTURE: GENERALIZED OPERATORS AND FASTER ALGORITHMS[*]

A. BOSTAN[†], C.-P. JEANNEROD[‡], C. MOUILLERON[§], AND É. SCHOST[¶]

Abstract. For matrices with displacement structure, basic operations like multiplication, inversion, and linear system solving can all be expressed in terms of the following task: evaluate the product AB, where A is a structured $n \times n$ matrix of displacement rank α, and B is an arbitrary $n \times \alpha$ matrix. Given B and a so-called *generator* of A, this product is classically computed with a cost ranging from $O(\alpha^2 \mathscr{M}(n))$ to $O(\alpha^2 \mathscr{M}(n)\log(n))$ arithmetic operations, depending on the type of structure of A; here, $\mathscr{M}$ is a cost function for polynomial multiplication. In this paper, we first generalize classical displacement operators, based on block diagonal matrices with companion diagonal blocks, and then design fast algorithms to perform the task above for this extended class of structured matrices. The cost of these algorithms ranges from $O(\alpha^{\omega-1}\mathscr{M}(n))$ to $O(\alpha^{\omega-1}\mathscr{M}(n)\log(n))$, with ω such that two $n \times n$ matrices can be multiplied using $O(n^\omega)$ ring operations. By combining this result with classical randomized regularization techniques, we obtain faster Las Vegas algorithms for structured inversion and linear system solving.

Key words. structured linear algebra, matrix multiplication, computational complexity

Faster product in compact representation

Through the ΣLU formula, $\mathsf{K}(n, \alpha)$ is seen as the time of computing

$$A_\ell = \sum_{j=1}^{\alpha} G_j \,(H_j V_\ell \bmod x^n), \qquad 1 \leq \ell \leq \alpha$$

with G_j, H_j, V_ℓ in $\mathbb{K}[x]$ of degree $< n$.

Remark: the inner modulo prevents us from factoring out the V_ℓ.

Matrix reformulation: Given $\mathbf{H} \in \mathbb{K}[x]^{\alpha \times 1}$, $\mathbf{V} \in \mathbb{K}[x]^{1 \times \alpha}$ and $\mathbf{G} \in \mathbb{K}[x]^{\alpha \times 1}$, all of degree $< n$, compute $(\mathbf{HV} \bmod x^n)\,\mathbf{G}$.

$\longrightarrow$ Recast this into a polynomial matrix multiplication in size α and degree $\frac{n}{\alpha}$

$\longrightarrow$ Deduce the bound $\mathsf{K}(n, \alpha) = \tilde{O}(\alpha^{\theta-1} n)$

Short-product techniques

Idea: compute $(\mathbf{HV} \bmod x^n)\mathbf{G}$ by divide-and-conquer, as

$$\Big(\big(\mathbf{H_0}+x^{\frac{n}{2}}\mathbf{H_1}\big)\big(\mathbf{V_0}+x^{\frac{n}{2}}\mathbf{V_1}\big) \bmod x^n\Big)\big(\mathbf{G_0}+x^{\frac{n}{2}}\mathbf{G_1}\big) \quad = \quad \mathbf{H_0V_0G_0} +$$

$$x^{\frac{n}{2}}\Big(\mathbf{H_0V_0G_1} + (\mathbf{H_0V_1}+\mathbf{H_1V_0} \bmod x^{\frac{n}{2}})\mathbf{G_0}\Big)+x^n\Big((\mathbf{H_0V_1}+\mathbf{H_1V_0} \bmod x^{\frac{n}{2}})\mathbf{G_1})$$

The **desired quantities** for the recursive step read off

$$\left(\begin{bmatrix}\mathbf{H_0} & \mathbf{H_1}\end{bmatrix}\begin{bmatrix}\mathbf{V_1}\\ \mathbf{V_0}\end{bmatrix} \bmod x^{n/2}\right)\begin{bmatrix}\mathbf{G_0} & \mathbf{G_1}\end{bmatrix}$$

Let $\mathsf{K}(d,\alpha,\ell)$ be the cost of: given $\mathbf{A}\in\mathbb{K}[x]^{\alpha\times\ell}$, $\mathbf{B}\in\mathbb{K}[x]^{\ell\times\alpha}$ and $\mathbf{C}\in\mathbb{K}[x]^{\alpha\times\ell}$, of degree $< d$, compute $\big(\mathbf{AB} \bmod x^{d\ell}\big)\,\mathbf{C}$. Thus

$$\mathsf{K}(n,\alpha)=\mathsf{K}(n,\alpha,1)\leq\mathsf{K}\left(\frac{n}{2},\alpha,2\right)\leq\mathsf{K}\left(\frac{n}{4},\alpha,4\right)\leq\ldots\leq\mathsf{K}\left(\frac{n}{\alpha},\alpha,\alpha\right)=\tilde{O}(\alpha^{\theta-1}n)$$

Here $\mathsf{K}\left(\frac{n}{\alpha},\alpha,\alpha\right)=$ cost of polynomial matrix multiplication in size α, degree $\leq\frac{n}{\alpha}$

Vandermonde and Cauchy

[Pan 1990] [Gohberg-Olshevsky 1994]

One can reduce the study of Vandermonde operators

$$\phi_V(A) = A - (\text{diagonal matrix}) \times (A \text{ shifted right by } 1)$$

and Cauchy operators

$$\phi_C(A) = A - (\text{diagonal matrix}) \times A \times (\text{diagonal matrix})'$$

to that of Toeplitz operators.

▷ The reduction involves a question similar to the one before: multiply a quasi-Vandermonde (quasi-Cauchy) matrix, given by α generators, by α vectors

▷ Similar techniques apply.