Implementing sparse rational function interpolation in Maple with application to solving parametric linear systems

Michael Monagan Simon Fraser University

Joint work with Archit Scrivastava and Mantej Sokhi

Parametric Polynomial Systems

$$A = \begin{bmatrix} y_1 & y_2 & y_3 \\ y_2 & y_1 & y_2 \\ y_3 & y_2 & y_1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

The solutions of $Ax = b$ are

$$x_1 = \frac{y_1 - y_2}{y_1^2 + y_1 y_3 - 2y_2^2} \quad x_2 = \frac{y_1 - 2y_2 + y_3}{y_1^2 + y_1 y_3 - 2y_2^2} \quad x_3 = \frac{y_1 - y_2}{y_1^2 + y_1 y_3 - 2y_2^2}.$$

Problem: given an n by n matrix A and n vector b with entries in $\mathbb{Z}[y_1, y_2, \ldots, y_m]$, solve $Ax = b$ for $x_i \in \mathbb{Q}(y_1, y_2, \ldots, y_m)$. If $A^{(i)}$ is the matrix obtained by replacing column i of A with b then

$$x_i = \frac{\det(A^{(i)})}{\det(A)}.$$

CAS use variations of fraction-free Gaussian Elimination such as Lipson's algorithm.

Lipson's Fraction-Free Algorithm

S1 Set $B = [A|b]$. Triangularize B using $O(n^3)$ $\times, \div$ in $R = \mathbb{Z}[y_1, y_2, \ldots, y_m]$.
We have $B_{nn} = \pm\det(A)$.

S2 Back substitute to obtain $y_i = \pm\det(A^{(i)})$ using $O(n^2)$ $\times, \div$ in R.
We have $x_i = y_i / B_{nn}$ in $\mathbb{Q}(y_1, y_2, \ldots, y_n)$.

S3 Compute $h_i = \gcd(y_i, B_{nn})$, $f_i = y_i \div h_i$ and $g_i = B_{nn} \div h_i$ for $1 \leq i \leq n$.
We have $x_i = f_i / g_i$ in $\mathbb{Q}(y_1, y_2, \ldots, y_n)$.

Problem: Lipson's algorithm computes polynomials which are larger than $\det(A)$ and $\det(A^{(i)})$. For example,

$$B_{nn} = \det(A) = \frac{num}{B_{n-2,n-2}} \implies num = \det(A) \times B_{n-2,n-2}$$

Benchmark: Symmetric Toeplitz Systems

$A = T_n$ the $n \times n$ symmetric Toeplitz matrix in $y_1, y_2, \ldots, y_n$.
$b = [1, 1, \ldots, 1]$.

$$A = \begin{bmatrix} y_1 & y_2 & y_3 \\ y_2 & y_1 & y_2 \\ y_3 & y_2 & y_1 \end{bmatrix} \quad x_1 = \frac{y_1 - y_2}{y_1^2 + y_1 y_3 - 2y_2^2} \quad x_2 = \frac{y_1 - 2y_2 + y_3}{y_1^2 + y_1 y_3 - 2y_2^2} \quad x_3 = \frac{y_1 - y_2}{y_1^2 + y_1 y_3 - 2y_2^2}.$$

$$\det(A) = (y_1 - y_3)\left(y_1^2 + y_1 y_3 - 2y_2^2\right)$$

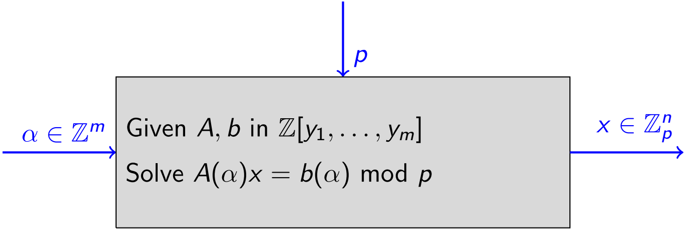
n	6	7	8	9	10	11	12	13
# det(Tn)	120	427	1628	6090	23793	90296	350726	OM
max $\#num$	1512	9206	56007	345008	2122800	13284290	82990563	–
Swell	12.6	21.5	34.4	56.6	89.2	147.2	236.6	–
max $\#f_i$	15	52	78	267	430	1623	1623	9757
max $\#g_i$	32	56	167	294	931	1730	5579	10611

Table: OM = out of memory. Means exceeded 128 gigabytes of RAM

Our Idea: interpolate the simplified rational function solutions $x_i = f_i/g_i$ from values.

Solving Parametric Linear Systems using Kaltofen-Yang

Apply Kaltofen-Yang rational function interpolation to interpolate $x_i \in \mathbb{Q}(y_1, \ldots, y_m)$.
Do it modulo primes $p_1, p_2, \ldots$ and apply CRT + RNR to recover the rational coefficients in x_i.
We first create the modular black box **B**:$(\mathbb{Z}^m, p) \to \mathbb{Z}_p^n$ as a Maple procedure:



So interpolate $x_i = f_i/g_i$ in $\mathbb{Z}_p(y_1, y_2, \ldots, y_m)$ from many values $x(\alpha)$ computed with **B**.

Kaltofen-Yang's sparse rational function interpolation 2007

Let $h = f/g$ where $f, g \in \mathbb{Z}_p[y_1, y_2, \ldots, y_m]$ and $\gcd(f, g) = 1$.
For some $\sigma \in \mathbb{Z}_p^m$, we can compute $\mathbf{B}(\sigma, p) = h(\sigma)$, how get we compute $f(\sigma)$ and $g(\sigma)$?

S1 Pick $\beta_2, \ldots, \beta_m$ from $[1, p-1]$ at random. Consider

$$T(x) = \frac{f(x, \beta_2(x-\sigma_1)+\sigma_2, \ldots, \beta_m(x-\sigma_1)+\sigma_m)}{g(x, \beta_2(x-\sigma_1)+\sigma_2, \ldots, \beta_m(x-\sigma_1)+\sigma_m)} = \frac{\hat{f}(x)}{\hat{g}(x)}.$$

with $\hat{g}(x)$ monic. Observe that $T(\sigma_1) = h(\sigma_1, \sigma_2, \ldots, \sigma_m) = \frac{\hat{f}(\sigma_1)}{\hat{g}(\sigma_1)}$. Why not use $\beta_i = 1$?

S2 Interpolate $T(x)$. Let $D = \deg(f) + \deg(g)$. Pick distinct $\alpha_0, \alpha_1, \ldots, \alpha_D \in \mathbb{Z}_p$ and use

$$T(\alpha_j) = \mathbf{B}(\alpha_1, \beta_2(\alpha_j - \sigma_1) + \sigma_2, \ldots, \beta_m(\alpha_j - \sigma_1) + \sigma_m).$$

S3 Repeat step 2 for many σ so we can interpolate f and g from $\hat{f}(\sigma_1)$ and $\hat{g}(\sigma_1)$.

BenOr-Tiwari (BOT) Sparse Polynomial Interpolation

Let $f = \sum_{i=1}^{t} a_i M_i(x_1, \ldots, x_n)$ be in $\mathbb{Z}_p[x_1, \ldots, x_n]$.
Let $m_i = M_i(2, 3, 5, \ldots, p_n)$. Let $\lambda(z) = \sum_{i=1}^{t}(z - m_i)$.
Assume we know t.

- S1 Compute $v_j = f(2^j, 3^j, \ldots, p_n^j) = \sum_{i=1}^{t} a_i m_i^j \mod p$ for $j = 0, 1, ..., 2t-1$.
- S2 Compute $\lambda(z)$ from the v_j using Berlekamp-Massey . . . . $O(t^2)$
- S3 Factor $\lambda(z)$ over $\mathbb{Z}_p$ to get the monomial evaluations m_i . . . . $O(t^2 \log p)$
- S4 Find $M_i(x_1, x_2, \ldots, x_n)$ from m_i . . . . $O(tD)$
- S5 Solve the transposed Vandermonde system $\{\sum_{i=1}^{t} a_i m_i^j = v_j : 0 \le j \le t-1\}$ . . . . $O(t^2)$

Good: BOT needs only $2t$ values of $h(y_1, \ldots, y_m)$. **Catch:** BOT requires $p > m_i$.
S2, S5 are coded in C for 63 bit primes. We use Maple's FFI to access these.
S3 Maple uses Cantor-Zassenhaus with C support for 31.5 bit primes

Benchmark continued

$A = T_n$ the n by n symmetric Toeplitz matrix in $y_1, y_2, \ldots, y_n$.
$b = [1, 1, \ldots, 1]$.

n	6	7	8	9	10	11	12	13	14
Lipson	0.036	0.128	0.728	3.741	53.06	384.5	12678.	OM	NA
$\times$	0.004	0.027	0.157	1.557	15.44	196.7	6425.	–	–
$\div$	0.000	0.001	0.034	0.357	4.45	16.0	1502.	–	–
gcd	0.032	0.100	0.537	1.827	29.11	126.3	4751.	–	–
KY	0.088	0.134	0.498	1.366	3.025	10.05	39.56	181.12	877.1
interp	0.039	0.051	0.261	0.594	1.404	3.638	16.85	42.02	241.5
factor	0.002	0.011	0.026	0.130	0.353	2.308	6.181	59.56	291.3
probe	0.016	0.033	0.098	0.339	0.582	1.419	6.391	16.24	102.6
#probes	768	1052	4124	10268	20508	49180	196636	458780	1835036

Table: Timings are in CPU seconds. OM = out of memory. NA = not attempted.

Maple's Foreign Function Interface

Purpose: Call a C/C++ function from Maple for speed
Supports: Arrays of char, float, double, int, long int

Convert a Maple polynomial f of degree d to a Array.

```
A := Array(0..d,[seq(coeff(f,x,i),i=0..d)],datatype=integer[8]);
```

Convert an Array A to a Maple polynomial.

```
f := add( A[i]*x^i, i=0..d );
```

The Maple representation for $3x + 5x^2 + 7x^3$ is

POLY	x	3	7	2	5	1	3

The add command creates

SUM	x	3	$\uparrow$ $\boxed{\text{PROD } x \ 2}$	5	$\uparrow$ $\boxed{\text{PROD } x \ 3}$	7

Maple's Foreign Function Interface continued

Interpolate $T(x)$ from $z_j = T(\alpha_j) = B(\alpha_j, \beta_2(\alpha_j - \sigma_1) + \sigma_2, \ldots, \beta_m(\alpha_j - \sigma_1) + \sigma_m)$ for $0 \le j \le D$ where $D = \deg(f) + \deg(g)$. Mantej has implemented this in C for 63 bit primes.

S1 **Newton:** interpolate $u(x)$ in $\mathbb{Z}_p[x]$ s.t. $u(\alpha_j) = z_j$ .... $O(D^2)$.

S2 **Ratrecon:** recover $T(x) = \hat{f}(x)/\hat{g}(x)$ using the EEA .... $O(D^2)$.

		Newton			Ratrecon			Conversions	
deg f	deg g	Interp	C	FFI	Ratrecon	C	FFI	seq	add
5	5	20.7	0.76	2.3	38.3	1.10	3.6	18.6	14.4
20	20	39.6	10.3	12.0	93.4	5.45	7.7	53.8	25.7
80	80	270.7	131.9	133.3	444.5	49.2	52.2	207.7	72.9
320	320	3558.	1951.	1954.	3703.4	656.3	658.9	887.3	236.6

Table: CPU timings in micro seconds using $p = 2^{31} - 1$.