Sparse $\sqrt{\text{FGLM}}$ using the block Wiedemann algorithm

Seung Gyu Hyun†, Vincent Neiger⋆, Hamid Rahkooy†, Éric Schost†

† University of Waterloo, ⋆ DTU Compute

Main Problem

Input: $I \subset \mathbb{K}[x_1, \dots, x_n]$ zero-dimensional

- monomial basis of $Q = \mathbb{K}[x_1, \dots, x_n]/I$
- multiplication matrices $T_1, \dots, T_n \in \mathbb{K}^{D \times D}$ of $x_1, \dots, x_n$, with $D = \dim_{\mathbb{K}}(Q)$

Output:

- lex Gröbner basis of $\sqrt{I}$

Assumptions

- characteristic of $\mathbb{K}$ larger than D
- x_n generic coordinate

Consequence: output is

$$\begin{array}{l} x_1 - R_1(x_n), \\ \quad\vdots \\ x_{n-1} - R_{n-1}(x_n), \\ R(x_n) \end{array}$$

Previous work

[Faugère *et al.*'93] **FGLM**

- dense matrix computations

[Rouillier'99] **RUR**

- linearly generated sequence using the trace

[Bostan *et al.*'03]

- trace $\to$ random linear form

[Faugère, Mou'17] **Sparse FGLM**

- lex basis of I
- uses Berlekamp-Massey-Sakata in some cases

At a glance

Cons:

- computes a lex basis of $\sqrt{I}$ (weaker output)

Pros:

- few assumptions
- simple algorithm

Key idea: use sequences of small matrices, requires less terms than scalar sequences [Coppersmith'93]. **Correctness** from the analysis of Coppersmith's algorithm [Villard'97], [Kaltofen, Villard'04] and generating series properties [Bostan *et al.*'03]. See also [Kaltofen'95], [Kaltofen, Yuhasz'13].

Algorithm

1. choose $U, V \in \mathbb{K}^{m \times D}$
2. $s = (UT_n^i V^t)_{0 \le i < 2d}$, with $d = \frac{D}{m}$
3. $S = \text{MatrixBerlekampMassey}(s)$ and $N = S \sum_{i \ge 0} \frac{s_i}{x^{i+1}}$
4. $P =$ largest invariant factor of S and $R_n = \text{SquareFreePart}(P)$
5. $a = [0 \cdots 0 P] S^{-1}$
6. $N^* =$ first entry of aN
7. for $j = 1 \dots n-1$:
   - 7.1. $s_j = (UT_n^i T_j V^t)_{0 \le i < d}$ and $N_j = S \sum_{i \ge 0} \frac{s_{j,i}}{x^{i+1}}$
   - 7.2. $N_j^* =$ first entry of aN_j
   - 7.3. $R_j = N_j^* / N^* \bmod R_n$

Example

Input: $I = \langle f_1^2, f_2^2, f_3 \rangle \subset \mathbb{F}_{9001}[x_1, x_2, x_3]$, non radical of degree $\boldsymbol{D = 32}$, with

$$f_1 = 4979x_1^2 + 6202x_1x_2 + \dots, \quad f_2 = 4682x_1^2 + 8290x_1x_2 + \dots, \quad f_3 = 4199x_1^2 + 2325x_1x_2 + \dots$$

Step 1 with $\boldsymbol{m = 2}$

$$U = \begin{bmatrix} 1898 & 6830 & 3494 & 169 & 7991 & 3352 \dots \\ 3161 & 8858 & 8467 & 5882 & 8037 & 3726 \dots \end{bmatrix} \qquad V = \begin{bmatrix} 7595 & 8416 & 2285 & 8351 & 550 & 7012 \dots \\ 823 & 5686 & 6539 & 7884 & 7105 & 3427 \dots \end{bmatrix}^t$$

Step 2 & 3 with $\boldsymbol{d = 16}$

$$s = \left(\begin{bmatrix} 31 & 6977 \\ 1178 & 1695 \end{bmatrix}, \begin{bmatrix} 8212 & 1663 \\ 4811 & 4837 \end{bmatrix} \dots \right) \xRightarrow{\text{MatrixBerlekampMassey}} \begin{array}{l} S = \begin{bmatrix} \boldsymbol{x}^{\mathbf{16}} + \dots & 423\boldsymbol{x}^{\mathbf{15}} + \dots \\ 6426\boldsymbol{x}^{\mathbf{15}} + \dots & \boldsymbol{x}^{\mathbf{16}} + \dots \end{bmatrix} \\ N = \begin{bmatrix} 6191\boldsymbol{x}^{\mathbf{15}} + \dots & 8101\boldsymbol{x}^{\mathbf{15}} + \dots \\ 7116\boldsymbol{x}^{\mathbf{15}} + \dots & 2129\boldsymbol{x}^{\mathbf{15}} + \dots \end{bmatrix} \end{array}$$

Step 4 & 5: $a = \begin{bmatrix} 2575\boldsymbol{x}^{\mathbf{7}} + \dots & \boldsymbol{x}^{\mathbf{8}} + \dots \end{bmatrix}$ and $R_3 = \boldsymbol{x}^{\mathbf{8}} + 6990x^7 + \dots$

Step 6:

$$N^* = \begin{bmatrix} 2575\boldsymbol{x}^{\mathbf{7}} + \dots & \boldsymbol{x}^{\mathbf{8}} + \dots \end{bmatrix} \begin{bmatrix} 6191\boldsymbol{x}^{\mathbf{15}} + \dots \\ 7116\boldsymbol{x}^{\mathbf{15}} + \dots \end{bmatrix} = \begin{bmatrix} 1178\boldsymbol{x}^{\mathbf{23}} + 8727x^{22} + \dots \end{bmatrix}$$

Step 7 for $\boldsymbol{j = 1}$

$$s_1 = \left(\begin{bmatrix} 3029 & 8903 \\ 1538 & 5610 \end{bmatrix}, \begin{bmatrix} 1914 & 3734 \\ 5221 & 5431 \end{bmatrix} \dots \right) \Longrightarrow N_1 = \begin{bmatrix} 1374\boldsymbol{x}^{\mathbf{15}} + \dots & 3271\boldsymbol{x}^{\mathbf{15}} + \dots \\ 4027\boldsymbol{x}^{\mathbf{15}} + \dots & 1575\boldsymbol{x}^{\mathbf{15}} + \dots \end{bmatrix}$$

$$N_1^* = \begin{bmatrix} 2575\boldsymbol{x}^{\mathbf{7}} + \dots & \boldsymbol{x}^{\mathbf{8}} + \dots \end{bmatrix} \begin{bmatrix} 1374\boldsymbol{x}^{\mathbf{15}} + \dots \\ 4027\boldsymbol{x}^{\mathbf{15}} + \dots \end{bmatrix} = \begin{bmatrix} 1538\boldsymbol{x}^{\mathbf{23}} + 6498x^{22} + \dots \end{bmatrix}$$

$$R_1 = \frac{1538\boldsymbol{x}^{\mathbf{23}} + 6498x^{22} + \dots}{1178\boldsymbol{x}^{\mathbf{23}} + 8727x^{22} + \dots} \bmod \boldsymbol{x}^{\mathbf{8}} + 6990x^7 + \dots = 7964\boldsymbol{x}^{\mathbf{7}} + 4071x^6 + \dots$$

Parallelization

Bottleneck: computing $(UT_n^i)_{0 \le i < 2d}$

easy to parallelize!

experiments in **LinBox** modulo $p = 65537$

name	n	D	ratio (8 cores)	sparsity
katsura8	9	256	1136/213=**5.33**	0.63
katsura9	10	512	8903/1651= **5.39**	0.64
cyclic7	7	924	6585/1235=**5.33**	0.08
katsura10	11	1024	70211/13019=**5.39**	0.63
gametwo7	7	1854	35051/64673=**5.41**	0.54
schwarz11	11	2048	21699/4129=**5.25**	0.27
eco12	12	1024	61605/11296=**5.45**	0.55

(parallel speed-up for the sequence computation)

References

[1] A. Bostan, B. Salvy, and É. Schost.
Fast algorithms for zero-dimensional polynomial systems using duality.
AAECC, 14:239–272, 2003.

[2] D. Coppersmith.
Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm.
Mathematics of Computation, 62(205):330–350, 1994.

[3] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora.
Efficient computation of zero-dimensional Gröbner bases by change of ordering.
JSC, 16(4):329–344, 1993.

[4] J.-C. Faugère and C. Mou.
Sparse FGLM algorithms.
JSC, 80(3):538–569, 2017.

[5] E. Kaltofen.
Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems.
Mathematics of Computation, 64(210):777–806, 1995.

[6] E. Kaltofen and G. Villard.
On the complexity of computing determinants.
Comput. Complexity, 13(3-4):91–130, 2004.

[7] E. Kaltofen and G. Yuhasz.
On the matrix Berlekamp-Massey algorithm.
ACM Trans. on Algorithms, 9(4):33:1–33:24, 2013.

[8] F. Rouillier.
Solving zero-dimensional systems through the Rational Univariate Representation.
AAECC, 9(5):433–461, 1999.

[9] G. Villard.
Further analysis of Coppersmith's block Wiedemann algorithm for the solution of sparse linear systems.
ISSAC'97, 32–39, 1997.