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

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

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

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 '06].

Algorithm

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

Example

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

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

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

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

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

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

Step 4: $a = \begin{bmatrix} 2575\boldsymbol{x^7} + \ldots & \boldsymbol{x^8} + \ldots \end{bmatrix}$

Step 5: $[N^*] = \begin{bmatrix} 2575\boldsymbol{x^7} + \ldots & \boldsymbol{x^8} + \ldots \end{bmatrix} \begin{bmatrix} 6191\boldsymbol{x^{15}} + \ldots \\ 7116\boldsymbol{x^{15}} + \ldots \end{bmatrix} = \begin{bmatrix} 1178\boldsymbol{x^{23}} + 8727x^{22} + \ldots \end{bmatrix}$

Step 6 for $\boldsymbol{j = 1}$

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

Step 7: $[N_1^*] = \begin{bmatrix} 2575\boldsymbol{x^7} + \ldots & \boldsymbol{x^8} + \ldots \end{bmatrix} \begin{bmatrix} 1374\boldsymbol{x^{15}} + \ldots \\ 4027\boldsymbol{x^{15}} + \ldots \end{bmatrix} = \begin{bmatrix} 1538\boldsymbol{x^{23}} + 6498x^{22} + \ldots \end{bmatrix}$

Parallel Computations

- Bottleneck is computing the sequence $(UT_n^i)_{0 \le i < 2d}$
- Can parallelize by computing the sequences $(U_1T_n^i), \ldots, (U_mT_n^i)$ separately, where U_i is the i^{th} row of U
- When $m = 1$, same computation as Sparse FGLM

Conclusion

References

[1] J.-C. Faugère and C. Mou. Sparse FGLM algorithms. *Journal of Symbolic Computation*, 80(3):538–569, 2017.

[2] Don Coppersmith. Solving linear equations over GF(2): block Lanczos algorithm. *Linear Algebra and its Applications*, 192:33–60, 1993.

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

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

[5] A. Bostan, B. Salvy, and É. Schost. Fast algorithms for zero-dimensional polynomial systems using duality. *Applicable Algebra in Engineering, Communication and Computing*, 14:239–272, 2003.

[6] V. Neiger. *Bases of relations in one or several variables: fast algorithms and applications.* PhD thesis, École Normale Supérieure de Lyon, November 2016.

[7] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Polynomial Systems Solving by Fast Linear Algebra. https://hal.archives-ouvertes.fr/hal-00816724, 2013.