Relaxing Order Basis Computation

Pascal Giorgi and Romain Lebreton
LIRMM, CNRS-UM2 France
pascal.giorgi@lirmm.fr, romain.lebreton@lirmm.fr

The computation of an order basis (also called sigma basis in [3]) is a fundamental tool for linear algebra with polynomial coefficients. Such computation is one of the key ingredient to provide algorithms which reduce to polynomial matrix multiplication. This has been the case for column reduction [3] or minimal nullspace basis [12] of polynomial matrix over a field. In this poster, we are interested in the application of order basis to compute minimal matrix generators of a linear matrix sequence (see [10]). In particular, we focus on the linear matrix sequence used in Block Wiedemann algorithm [1].

As of today, the fast order basis algorithm PM-Basis from [3] suffers from two issues. In our applications, the bound σ on its degree may be pessimistic and therefore we need to use early termination. However the recursive aspect of PM-Basis is unhelpful to implement such an early termination. Also PM-Basis may require to know more coefficients of F than necessary. This can hinder the complexity when the cost of computing coefficients of the entry is dominant. This is the case for instance for the block Wiedemann algorithm which motivates this work.

Main results In this work we propose a relaxed variant of the PM-Basis algorithm. The property of relaxed algorithms is that they do not require more knowledge on the input than necessary while keeping a quasi-optimal complexity in the order σ.
We first propose an iterative variant Iterative-PM-Basis of PM-Basis which is more suited to the relaxed model and also to early termination. Then we show how to relax Iterative-PM-Basis *via* the use of a relaxed polynomial matrix multiplication algorithm. Thus we obtain our relaxed order basis computation within the complexity of PM-Basis with only an extra logarithmic factor in σ. Finally, we show the benefit of this algorithm to gain a constant factor on average on the block Wiedemann algorithm.

Order basis algorithms Let $\mathbb{K}$ be a field, $F = \sum_{i \geqslant 0} F_i x^i \in \mathbb{K}[[x]]^{m \times n}$ a matrix of power series, σ a positive integer and (F, σ) be the $\mathbb{K}[x]$-module defined by the set of $v \in \mathbb{K}[x]^{1 \times m}$ such that $vF \equiv 0 \bmod x^\sigma$. A polynomial matrix P is a (left) order basis of F of order σ and shift $\vec{s}$ if the rows of P form a basis of (F, σ) and P is $\vec{s}$-row reduced (see [11] for details). Without loss of generality we only consider in this poster the case $n = O(m)$ with a balanced shift $\vec{s}$ as in [3]. Indeed the techniques of [11] allow to reduce the general case to our particular case.

Two different algorithms presented in [3] compute an order basis P of F. The M-Basis algorithm works iteratively on the order σ to compute the order basis P. It is a lazy algorithm that costs $O(m^\omega \sigma^2)$ arithmetic operations in $\mathbb{K}$, *i.e.* it only requires the coefficients F_j of F for $0 \leqslant j \leqslant (i-1)$ for computing the intermediate order basis of order i. The PM-Basis algorithm uses a divide-and-conquer approach on the order σ to reduce the arithmetic complexity to $O(m^\omega \mathsf{M}(\sigma) \log(\sigma)) = O\tilde{}(m^\omega \sigma)$, where M denotes the arithmetic complexity of polynomial multiplication. Roughly speaking, the algorithm is made of four steps: 1) a recursive call to compute an order basis P_{low} of F of order $\sigma/2$, 2) an update of the problem *via* the middle product $F' := (x^{-\sigma/2} P_{\text{low}} F) \bmod x^{\sigma/2}$, 3) a recursive call to compute an order basis P_{high} of F' of order $\sigma/2$ and 4) return the order basis $P_{\text{high}} P_{\text{low}}$ of F of order σ. Step 2) implies that one may need at most twice as much coefficients of the input series than necessary to go from an intermediate order basis of order i to $i+1$.

Fast iterative order basis Let us give an iterative version of PM-Basis. Our algorithm performs exactly the same operations on matrices as PM-Basis when σ is a power of two. This iterative presentation of PM-Basis is original. Let us denote $\nu_2(k)$ the valuation in 2 of any integer k and index our lists from 1.

Algorithm 1: Iterative-PM-Basis

Input: $F \in \mathbb{K}[[x]]^{m\times n}$, $\sigma > 0$, $\vec{s} \in \mathbb{N}^m$
Output: $P \in \mathbb{K}[x]^{m\times n}$ such that P is a $\vec{s}$-row reduced order basis of (F, σ)

1: $P_0, \vec{u} :=$ M-Basis$(F \bmod x,\ 1,\ \vec{s}\ )$; $P := [P_0]$; $S := [0, \ldots, 0, F]$ with $\lceil \log_2(\sigma) \rceil$ zeros
2: **for** $k = 1$ **to** $\sigma - 1$ **do**
3: $\quad \ell := \nu_2(k)$; $\ell' := \begin{cases} \lceil \log_2(\sigma) \rceil & \text{if } k = 2^\ell \\ \nu_2(k - 2^\ell) & \text{otherwise} \end{cases}$
4: $\quad$ Update P by merging its first $\ell + 1$ elements by multiplication //Product tree of step 4)
5: $\quad S[\ell + 1] :=$ MiddleProduct$(P[1], S[\ell' + 1], 2^\ell)$ //Update of the series of step 2)
6: $\quad P_k, \vec{u} :=$ M-Basis$(S[\ell + 1] \bmod x, 1, \vec{u}\ )$ //Recursive calls on leafs of steps 1) and 3)
7: $\quad$ Insert P_k at the beginning of P
8: **return** $\prod_i P[i]$

Relaxing order basis algorithm In algorithm PM-Basis, we have noticed that only the middle product of step 5 reads more entries of F than necessary at step k. Let us perform this step differently so that it reads at most the coefficients $F_0, \ldots, F_{k-1}$ of F at step k. This property is called a *relaxed* (or on-line) algorithm w.r.t. F.

A naive approach would be to compute a full $2n \times n$ product using a relaxed multiplication algorithm on polynomial of matrices ([2, 6, 5, 7, 9]) in time $\mathsf{R}(n) = O(\mathsf{M}(n)\log(n))$ [2]. We propose another relaxed algorithm that gains asymptotically a factor 2 compared to the full $2n \times n$ relaxed product. We illustrate our algorithm in Figure 1. We decompose the relaxed middle product in a normal high product (in black) followed by a multiplication (in white and gray) relaxed w.r.t. only b using [7] in this example.

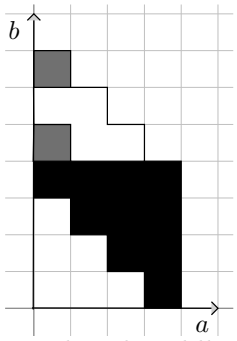

Figure 1: Relaxed middle product

Using this relaxed middle product algorithm within Iterative-PM-Basis we obtain an order basis algorithm Relaxed-PM-Basis relaxed w.r.t. F. This relaxed order basis algorithm costs $O(k^\omega \mathsf{R}(\sigma)\log(\sigma)) = O(k^\omega \mathsf{M}(\sigma)\log^2(\sigma))$ operations in $\mathbb{K}$.

Application to block Wiedemann algorithm Let $A \in \mathrm{GL}_N(\mathbb{K})$ with $O(N)$ non-zero elements. Block Wiedemann approach uses a minimal matrix generator of the matrix series $S = \sum_{i\in\mathbb{N}} UA^iVx^i$ for any random $U, V^T \in \mathbb{K}^{m\times N}$ in order to solve a linear system $Ax = b \in \mathbb{K}^N$. As described in [10], this matrix generator can be obtained from an order basis of $F = [S \mid I_m]^T \in \mathbb{K}[[x]]^{2m\times m}$. We can derive a bound on the maximal degree δ of this order basis using the stopping criteria of [8, Th. 4.19]. Since this bound may be loose, a constant factor in the complexity can often be saved using an early termination in the order basis algorithm.

We compare the complexity of Iterative-PM-Basis and Relaxed-PM-Basis in this setting. Computing S at precision σ costs $O(k^{\omega-1}N\sigma)$. In practice $k \ll N$ so that the cost of computing S always dominates the cost of (relaxed) order basis algorithm.

Assume that δ is uniformely distributed between $2^p + 1$ and 2^{p+1} for $p \in \mathbb{N}$. Iterative-PM-Basis requires the coefficients $F_0, \ldots, F_{2^{p+1}-1}$ whereas Relaxed-PM-Basis only asks for $F_0, \ldots, F_{\delta-1}$. Therefore our relaxed approach improves the dominant cost of computing F in block Wiedemann by a factor 2 at most and 4/3 on average.

References

[1] Don Coppersmith. "Solving Homogeneous Linear Equation Over GF(2) via Block Wiedemann Algorithm". In: *Mathematics of Computation* 62.205 (1994), pp. 333–350.

[2] M. J. Fischer and L. J. Stockmeyer. "Fast on-line integer multiplication". In: *J. Comput. System Sci.* 9 (1974), pp. 317–331. ISSN: 0022-0000.

[3] P. Giorgi, C.-P. Jeannerod, and G. Villard. "On the complexity of polynomial matrix computations". In: *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*. ISSAC '03. ACM, 2003, pp. 135–142.

[4] G. Hanrot, M. Quercia, and P. Zimmermann. "The middle product algorithm. I". In: *Appl. Algebra Engrg. Comm. Comput.* 14.6 (2004), pp. 415–438. ISSN: 0938-1279.

[5] J. van der Hoeven. *Faster relaxed multiplication*. Tech. rep. HAL-00687479, 2012.

[6] J. van der Hoeven. "New algorithms for relaxed multiplication". In: *J. Symbolic Comput.* 42.8 (2007), pp. 792–802. ISSN: 0747-7171.

[7] J. van der Hoeven. "Relaxed multiplication using the middle product". In: *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*. New York: ACM, 2003, 143–147 (electronic).

[8] E. Kaltofen and G. Yuhasz. "On the matrix Berlekamp-Massey algorithm". In: *ACM Transaction on Algorithms* (2013). to appear.

[9] R. Lebreton and É. Schost. "Relaxed power series multiplication using middle and short product". In preparation.

[10] W. J. Turner. "Black Box Linear Algebra with the LinBox Library". PhD thesis. North Carolina State University, 2002.

[11] W. Zhou and G. Labahn. "Efficient algorithms for order basis computation". In: *Journal of Symbolic Computation* 47.7 (2012), pp. 793 –819.

[12] W. Zhou, G. Labahn, and A. Storjohann. "Computing minimal nullspace bases". In: *Proceedings of the 2012 International Symposium on Symbolic and Algebraic Computation*. ISSAC '12. ACM, 2012, pp. 366–373.