



**HAL**  
open science

## Relaxing order basis computation

Pascal Giorgi, Romain Lebreton

► **To cite this version:**

Pascal Giorgi, Romain Lebreton. Relaxing order basis computation. ACM Communications in Computer Algebra, 2014, 47 (3/4), pp.100-101. 10.1145/2576802.2576813 . lirmm-01372532

**HAL Id: lirmm-01372532**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01372532v1>**

Submitted on 27 Sep 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Context

Let  $\mathbb{K}$  be a field,  $F = \sum_{i \geq 0} F_i x^i \in \mathbb{K}[[x]]^{m \times n}$  a matrix of power series,  $\sigma$  a positive integer and  $(F, \sigma)$  be the  $\mathbb{K}[[x]]$ -module defined by the set of  $v \in \mathbb{K}[[x]]^{1 \times m}$  such that  $vF \equiv 0 \pmod{x^\sigma}$ .

**Definition of Order basis:**  $P \in \mathbb{K}[[x]]^{m \times m}$  is a (left)  $(\sigma, \vec{s})$ -order basis of  $F$  if the rows of  $P$  form a  $\vec{s}$ -row reduced basis of  $(F, \sigma)$  (see [1]).

**Order basis are used in:** column reduction [2]; minimal nullspace basis [3]; block Wiedemann algorithm [4]; ...

## Two existing algorithms

**Input:**  $F \in \mathbb{K}[[x]]^{m \times n}$ ,  $\sigma \in \mathbb{N}^*$  and  $\vec{s} \in \mathbb{Z}^m$

**Output:**  $P \in \mathbb{K}[[x]]^{m \times m}$  a  $(\sigma, \vec{s})$ -order basis of  $F$  and  $\vec{u} \in \mathbb{Z}^m$  the shifted  $\vec{s}$ -row degree of  $P$ .

To simplify the presentation, let us assume w.l.o.g. that:

- the procedure  $\text{Basis}(F, \vec{s})$  handles the  $(1, \vec{s})$ -order basis case
- $n = O(m)$  and the shift  $\vec{s}$  is balanced, as in [2]

## M-Basis

Naive algorithm, iterative on the order  $\sigma$ , which costs  $O(m^\omega \sigma^2)$  op. in  $\mathbb{K}$ .

- ✗ Quadratic complexity in the precision  $\sigma$
- ✓ Easy to stop at any intermediate step
- ✓ Minimal knowledge on  $F$ , only coefficients  $F_0, \dots, F_k$  at step  $k$

### Algorithm 1: M-Basis( $F, \sigma, \vec{s}$ )

```

1:  $P, \vec{u} := \text{Basis}(F \bmod x, \vec{s})$ 
2: for  $k = 1$  to  $\sigma - 1$  do
3:    $F' := x^{-k} P \cdot F \bmod x^{k+1}$ 
4:    $P_k, \vec{u}_k := \text{Basis}(F', \vec{u})$ 
5:    $P := P_k \cdot P$ 
6: return  $P, \vec{u}$ 

```

## PM-Basis

Recursive variant using a divide and conquer strategy on the order  $\sigma$  which costs  $O(m^\omega M(\sigma) \log(\sigma)) = O(m^\omega \sigma)$  operations in  $\mathbb{K}$ .

- ✓ Quasi-linear complexity in the precision  $\sigma$
- ✗ Not convenient for early termination
- ✗ Often requires to know coefficients of  $F$  in advance

### Algorithm 2: PM-Basis( $F, \sigma, \vec{s}$ )

```

1: if  $\sigma = 1$  then
2:   return  $\text{Basis}(F \bmod x, \vec{s})$ 
3: else
4:    $P_1, \vec{u}_1 := \text{PM-Basis}(F, \sigma/2, \vec{s})$ 
5:    $F' := (x^{-\sigma/2} P_1 \cdot F) \bmod x^{\sigma/2}$ 
6:    $P_h, \vec{u}_h := \text{PM-Basis}(F', \sigma/2, \vec{u}_1)$ 
7:   return  $P_h \cdot P_1, \vec{u}_h$ 

```

## Our contribution

- Give an algorithm for order basis with the following properties:
  - ✓ **Quasi-optimality:** it takes a quasi-linear time in the precision  $\sigma$ ;
  - ✓ **Early termination:** easy to stop at any intermediate step;
  - ✓ **Relaxed algorithm:** minimal knowledge on the input  $F$  at each step.
- Use 1 to improve the complexity of block Wiedemann approach.

## Fast iterative algorithm

### Iterative-PM-Basis

Iterative version of PM-Basis that regroups computations step by step

- ✓ Quasi-linear complexity in the precision  $\sigma$
- ✓ Convenient for early termination
- ✗ Often requires to know coefficients of  $F$  in advance

### Algorithm 3: Iterative-PM-Basis( $F, \sigma, \vec{s}$ )

```

1:  $P_0, \vec{u} := \text{Basis}(F \bmod x, \vec{s})$ 
2:  $P := [P_0]$  and  $S := [0, \dots, 0, F]$  with  $\lceil \log_2(\sigma) \rceil$  zeros
3: for  $k = 1$  to  $\sigma - 1$  do
4:    $\ell := \nu_2(k)$  and  $\ell' := \begin{cases} \lceil \log_2(\sigma) \rceil & \text{if } k = 2^\ell \\ \nu_2(k - 2^\ell) & \text{otherwise} \end{cases}$ 
5:   Merge first  $\ell + 1$  elements of  $P$  by multiplication product tree step 7
6:    $S[\ell + 1] := (x^{-2^\ell} P[1] \cdot S[\ell' + 1]) \bmod x^{2^\ell}$  middle product step 5
7:    $P_k, \vec{u}_k := \text{Basis}(S[\ell + 1] \bmod x, \vec{u})$  recursive leafs step 2
8:   Insert  $P_k$  at the beginning of  $P$ 
9: return  $\prod_i P[i]$ 

```

## Relaxing the order basis algorithm

### Problem:

At step  $k = 2^\ell$ , Iterative-PM-Basis requires  $S[\lceil \log_2(\sigma) \rceil + 1] \bmod x^{2^{\ell+1}}$ , that is  $F \bmod x^{2^{\ell+1}}$ , to perform the middle product of step 6. However, we only need the middle product modulo  $x$  at step  $k$ , and therefore  $F \bmod x^{1+2^\ell}$ . The other coefficients of the middle product will be used in the next steps.

### Solution:

Compute the middle products gradually with the additional constraint of not using any coefficient of the input before necessary, *i.e.* using a **relaxed** algorithm.

### Definition of relaxed (or on-line) algorithm:

When computing the coefficient in  $x^k$  of the output, a *relaxed* algorithm can read at most the coefficients in  $1, \dots, x^k$  of the input.

## Relaxed middle product

### Two methods for a relaxed middle product algorithm:

- Compute a full  $2n \times n$  product using a relaxed multiplication algorithm on polynomial of matrices ([5])
- Compute just the middle product as in Figure 1 to gain asymptotically a factor 2 compared to method 1.

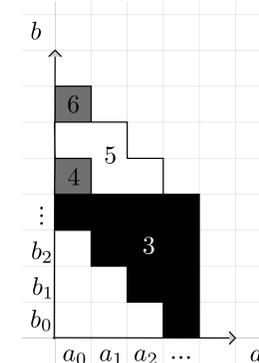


Figure 1: Relaxed middle product

### Relaxed-PM-Basis

Using this relaxed middle product within Iterative-PM-Basis, we obtain a new order basis algorithm relaxed w.r.t.  $F$ , which costs  $O(k^\omega M(\sigma) \log^2(\sigma))$ .

- ✓ Quasi-linear complexity in the precision  $\sigma$  (with an extra  $\log_2(\sigma)$ )
- ✓ Convenient for early termination
- ✓ Requires minimal knowledge on  $F$

## Application to block Wiedemann algorithm

Let  $A \in \text{GL}_N(\mathbb{K})$  with  $O(N)$  non-zero elements and  $S = \sum_{i \in \mathbb{N}} U A^i V x^i$  for random  $U, V^T \in \mathbb{K}^{n \times N}$ . The block Wiedemann approach uses a  $(\sigma, \vec{s})$ -order basis of  $F = [S^T \mid I_n]^T \in \mathbb{K}[[x]]^{2n \times n}$  to solve sparse linear systems  $Ay = b$ .

### Current approach:

Computing  $S$  at precision  $\sigma$  costs  $O(n^{\omega-1} N \sigma)$  operations in  $\mathbb{K}$ , which is dominant since  $n \ll N$ . An *a priori* bound  $\delta$  on the order  $\sigma$  is hard to find or may be loose. To circumvent this the paper [6] proposes a stopping criteria which has to be integrated into an iterative algorithm.

### Benefits of our approach:

- Iterative-PM-Basis provides the first iterative algorithm with quasi-linear time complexity that can use stopping criteria from [6].
- Relaxed-PM-Basis improves the complexity of 1 on average by a constant factor because less coefficients of  $S$  need to be computed.

## References

- [1] W. Zhou and G. Labahn, "Efficient algorithms for order basis computation," *J. Symbolic Comput.*, vol. 47, no. 7, pp. 793–819, 2012.
- [2] P. Giorgi, C.-P. Jeannerod, and G. Villard, "On the complexity of polynomial matrix computations," in *ISSAC'03*, pp. 135–142, ACM, 2003.
- [3] W. Zhou, G. Labahn, and A. Storjohann, "Computing minimal nullspace bases," in *ISSAC'12*, pp. 366–373, ACM, 2012.
- [4] W. J. Turner, *Black Box Linear Algebra with the LinBox Library*. PhD thesis, North Carolina State University, 2002.
- [5] M. J. Fischer and L. J. Stockmeyer, "Fast on-line integer multiplication," *J. Comput. System Sci.*, vol. 9, pp. 317–331, 1974.
- [6] E. Kaltofen and G. Yuhasz, "On the matrix berlekamp-massey algorithm," *ACM Trans. on Algorithms*, 2013. To appear.