Relaxing order basis computation
Pascal Giorgi, Romain Lebreton

To cite this version:


HAL Id: lirmm-01372532
https://hal-lirmm.ccsd.cnrs.fr/lirmm-01372532
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.
Let \( K \) be a field, \( F = \sum_{n \geq 0} F x^n \in K[x]^{m \times n} \) a matrix of power series, \( \sigma \) a positive integer and \((F, \sigma)\) be the \( K[x] \)-module defined by the set of \( v \in K[x]^{m \times n} \) such that \( vF \equiv 0 \mod x^\sigma \).

**Definition of basis**: \( P \in K[x]^{m \times m} \) is a (left) \((\sigma, \mathcal{S})\)-order basis of \( F \) if the rows of \( P \) form a \( \mathcal{S}\)-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**: \( F \in K[x]^{m \times m}, \sigma \in \mathbb{N}^n \) and \( \mathcal{S} \in \mathbb{Z}^n \)

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

To simplify the presentation, let us assume w.l.o.g. that:
- the procedure \( \text{Basis}(F, \mathcal{S}) \) handles the \((1, \mathcal{S})\)-order basis case
- \( n = O(m) \) and the shift \( \mathcal{S} \) is balanced, as in [2]

**Algorithm 1: M-Basis \((F, \sigma, \mathcal{S})\)**

1: \( P, \vec{u} := \text{Basis}(F \mod x, \mathcal{S}) \)
2: for \( k = 1 \) to \( \sigma - 1 \) do
3: \( \vec{x} = \vec{x} \cdot \vec{u} \mod x^{k+1} \)
4: \( P_k, \vec{u} := \text{Basis}(P_k, \vec{u}) \)
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 \( O(m^2 \log \sigma) \) op. in \( K \).
- Quadratic complexity in the precision \( \sigma \)
- Not convenient for early termination
- Often requires to know coefficients of \( F \) in advance

**Algorithm 2: PM-Basis \((F, \sigma, \mathcal{S})\)**

1: if \( \sigma = 1 \) then
2: return \( \text{Basis}(F \mod x, \mathcal{S}) \)
3: else
4: \( P_0, \vec{u}_0 := \text{PM-Basis}(F, \sigma / 2, \mathcal{S}) \)
5: \( P' := (x^{\sigma/2} P_0 \cdot F) \mod x^{\sigma/2} \)
6: \( P_0, \vec{u}_0 := \text{PM-Basis}(P', \sigma / 2, \vec{u}_0) \)
7: return \( P , P_0, \vec{u}_0 \)

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 \((F, \sigma, \mathcal{S})\)**

1: \( P_0, \vec{u} := \text{Basis}(F \mod x, \mathcal{S}) \)
2: for \( k = 1 \) to \( \sigma - 1 \) do
3: \( \vec{u} := \vec{u} \cdot \vec{u} \mod x^{k+1} \)
4: \( P_k, \vec{u} := \text{Basis}(P_k, \vec{u}) \)
5: \( P := P_k \cdot P \)
6: return \( P, \vec{u} \)

Relaxing the order basis algorithm

**Problem:** At step \( k = 2^i \), Iterative-PM-Basis requires \( S \cdot [\log_2(\sigma) + i] \mod x^{2^i} \), that is \( F \mod x^{2^i} \), to perform the middle product of step 6. However, we only need the middle product modulo \( x \) at step \( k \) and therefore \( F \mod x^{1+2} \). 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.

**Defining of relaxed (or on-line) algorithm**

When computing the coefficient in \( x^\delta \) of the output, a relaxed algorithm can read at most the coefficients in \( 1, \ldots, x^\delta \) of the input.

**Relaxed middle product**

Two methods for a relaxed middle product algorithm:

1. Compute a full \( 2n \times n \) product using a relaxed multiplication algorithm on polynomial of matrices ([15])
2. Compute just the middle product as in Figure 1 to gain asymptotically a factor 2 compared to method 1.

**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 \( O(k \cdot M(\sigma) \log^2(\sigma)) \).
- **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 \( A \in GL_{kN}(K) \) with \( \Omega(N) \) non-zero elements and \( S = \sum_{\alpha \in \Omega(N)} U^\alpha V^\alpha \) for random \( U, V \in K^{n \times N} \). The block Wiedemann approach uses a \((\sigma, \mathcal{S})\)-order basis of \( F = [S]_1 \otimes [I]_1 \in K[x]^{m \times n} \) to solve sparse linear systems \( A y = b \).

**Current approach**

Computing \( S \) at precision \( \sigma \) costs \( O(n^2 \cdot N \sigma) \) operations in \( K \), 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**

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

**References**