Relaxing order basis computation
Pascal Giorgi, Romain Lebreton

To cite this version:

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

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

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

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

#### M-Basis

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

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

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

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

#### PM-Basis

**Recursive variant using a divide and conquer strategy on the order $\sigma$ which costs $O(m^2 \log(\sigma)/\log(\sigma)) = O(m^2 \sigma)$ operations in $K$.**

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

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

### 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;
  - Relaxation: minimal knowledge on the input $F$ at each step.
- Use 1 to improve the complexity of block Wiedemann approach.

#### Fast iterative algorithm

**Iterative version of PM-Basis that regroups computations 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. $P_0, \overline{u} := \text{Basis}(F \mod x, \mathcal{S})$
2. $P := [P_0^\ell] \text{ and } S = [0, \ldots, 0, F]$ with $[\log_2(\sigma)]$ zeros
3. for $k = 1$ to $\sigma - 1$
4. $\mathcal{E} := \nu_2(k)$ and $\mathcal{E}' := \left\{ \begin{array}{ll} \log_2(\sigma) & \text{if } k = 2^t \\ \nu_2(k) - 2 & \text{otherwise} \end{array} \right.$
5. Merge first $\mathcal{E}$ elements of $P$ by multiplication
6. set product tree step 7
7. $S[\ell + 1] := (x^{-\mathcal{E}} P[1] \cdot S[\ell + 1]) \mod x^{\mathcal{E}'}$
8. middle product step 5
9. $P_{k-1} \cdot \overline{u} := \text{Basis}(S[\ell + 1] \mod x, \overline{u})$
10. Insert $P_{k-1}$ at the beginning of $P$
11. return $[1, P^\ell]$

### Relaxing the order basis algorithm

**Problem:** At step $k = 2^t$. Iterative-PM-Basis requires $S[\log_2(\sigma)] + 1 \mod x^{\mathcal{E}'}$, that is $F \mod x^{\mathcal{E}'}$, to perform the middle product of step 6. However, we only need the middle product modulo $x$ at step $k$, and therefore $F \mod x^{k+2}$.
- 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 $x^k$ of the output, a relaxed algorithm can read at most the coefficients in $1, \ldots, x^{k}$ 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 ([5])
2. Compute just the middle product as in Figure 1 to gain asymptotically a factor 2 compared to method 1.

### Application to block Wiedemann algorithm

Let $A \in GL_\mathbb{K}(K)$ with $O(N)$ non-zero elements and $S = \sum_{i=0}^{n-1} U^i V^j \in \mathbb{K}^{n \times N}$. The block Wiedemann approach uses a $(1, \mathcal{S})$-order basis of $F = [S[i]]$, $\mathcal{S} \in K[x]^{m \times n}$ to solve sparse linear systems $Ay = b$.

**Current approach**:
Computing $S \cdot \sigma$ costs $O(m^2 \cdot N \sigma)$ operations in $K$, which is dominant since $\sigma \ll N$. 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