HAL
open science

Computing Sparse Representations of Systems of Rational Fractions

François Lemaire, Alexandre Temperville

To cite this version:

François Lemaire, Alexandre Temperville. Computing Sparse Representations of Systems of Rational Fractions. Computer Algebra in Scientific Computing, Sep 2016, Bucharest, Romania. pp.349 - 366, 10.1007/978-3-319-45641-6_23 . hal-01407919

HAL Id: hal-01407919

https://hal.science/hal-01407919v1

Submitted on 2 Dec 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.

Computing Sparse Representations of Systems of Rational Fractions

François Lemaire, Alexandre Temperville

Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL -
Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille,
France francois.lemaire@univ-lille1.fr, a.temperville@ed.univ-lille1.fr

Abstract. We present new algorithms for computing sparse representations of systems of parametric rational fractions by means of change of coordinates. Our algorithms are based on computing sparse matrices encoding the degrees of the parameters occurring in the fractions. Our methods facilitate further qualitative analysis of systems involving rational fractions. Contrary to symmetry based approaches which reduce the number of parameters, our methods only increase the sparsity, and are thus complementary. Previously hand made computations can now be fully automated by our methods.

Keywords: Parametric systems ; simplification of rational fractions ; sparse basis of vector spaces

1 Introduction

This article presents new algorithms for computing sparse representations of systems of parametric rational fractions by means of change of variables. The goal of these algorithms is to help the analysis of parametric systems of rational fractions by producing sparser and equivalent formulations. Simplifying parametric systems is a central task, since many qualitative analyses (such as steady point analysis, bifurcation analysis, ...) rely on quite costly computations in real algebraic geometry (see [1] and references therein).

Symmetry based approaches [2,3,4,5,6] reduce the number of parameters and as a consequence usually help the analysis of parametric systems. On the contrary, our approach keeps the number of parameters (in the same spirit as [2, Algorithm SemiRectifySteadyPoints]) and makes the systems sparsest (in the sense of Algorithms getSparsestFraction and getSparsestSumOfFractions given later).

This work was motivated by the differential system (see Example 14):

$$\begin{cases} G' = \theta(1-G) - \alpha k_1 k_2 k_3 P^4 G \\ M' = \rho_b(1-G) + \rho_f G - \delta_M M \\ P' = \dfrac{4\theta(1-G) - 4\alpha k_1 k_2 k_3 P^4 G - \delta_P P + \beta M}{16 k_1 k_2 k_3 P^3 + 9 k_1 k_2 P^2 + 4 k_1 P + 1} \end{cases} \tag{1}$$

where the unknown functions are $G = G(t)$, $M = M(t)$ and $P = P(t)$, and where the parameters are θ, α, k_1, k_2, k_3, ρ_b, ρ_f, δ_M, δ_P and β.

Equation (1) was considered in [7], and rewritten with the guessed change of variables $\bar{k}_3 = k_1k_2k_3$, $\bar{k}_2 = k_1k_2$, $\bar{k}_1 = k_1$, in order to obtain [7, Equation (3.4)] (in the case $n = 4$ and $\gamma_0 = 1$). Our new algorithm getSparsestSumOfFractions (see Section 4) was designed to automatically compute this change of variables, which yields the following simpler system:

$$\begin{cases} G' = \theta(1-G) - \alpha\bar{k}_3P^4G \\ M' = \rho_b(1-G) + \rho_f G - \delta_M M \\ P' = \dfrac{4\theta(1-G) - 4\alpha\bar{k}_3P^4G - \delta_P P + \beta M}{16\bar{k}_3P^3 + 9\bar{k}_2P^2 + 4\bar{k}_1P + 1}. \end{cases} \tag{2}$$

The relatively small improvement between (1) and (2) in terms of degrees proves useful while searching for a Hopf Bifurcation. Indeed, the Routh-Hurwitz criterion applied on systems (1) and (2) leads to semi-algebraic systems of the form $h_1, h_2, h_3, h_4 = 0$, $h_5, h_6, h_7 > 0$, with respective degrees 9, 2, 9, 42, 12, 20, 23 in the case of (1), and smaller degrees 7, 2, 7, 32, 8, 14, 19 in the case of (2).

Consider a system of parametric rational fractions, involving parameters of a set U. To this system, we associate the so-called *matrix representation* encoding the degrees of the monomials in U. We then consider an invertible monomial map ϕ (i.e. an application which sends each parameter of U to a monomial in the elements of a set $\bar{U}$). The monomial map ϕ acts linearly on the matrix representation above, which allows us to look for a sparsest matrix representation using only linear algebra techniques. However, the use of fractions brings some difficulty since fractions are invariant when both numerators and denominators are multiplied by the same value. Example 10 shows how to automatically rewrite the fraction $\frac{x}{1+x+ix\tau_1 w}$ into the sparser fraction $\frac{1}{y+1+i\tau_1 w}$ by first dividing both numerators and denominators by x and then introducing $y = 1/x$.

Section 2 introduces the basic concepts. Section 3 introduces two new algorithms. Roughly speaking, the first one called CSBmodulo computes a sparsest representation of a vector space modulo another vector space, where both vector spaces are given by their basis. The second one called getSparsestFraction is a direct application of CSBmodulo for computing a sparsest representation of a fraction. Section 4 describes Algorithm getSparsestSumOfFractions which is a generalization of getSparsestFraction for systems or sums of rational fractions. Section 5 details the (technical) proof of CSBmodulo, relying on Corollary 4 which clarifies the structure of the sparsest bases.

2 Preliminaries

Consider $\mathbb{K}$ a commutative field and $U = \{u_1, \ldots, u_n\}$ a set of variables. We consider monomials in U of the form $u_1^{\alpha_1} \cdots u_n^{\alpha_n}$ where the α_i are in $\mathbb{Z}$. Since negative integer exponents are allowed, $u_1u_2^{-1}u_3$ is considered as a monomial. The row vector $(1, \ldots, 1)$ of length ℓ is denoted by $\mathbb{1}_\ell$ (or simply $\mathbb{1}$ when the context is clear).

2.1 Matrix Representation of a Fraction

Definition 1. *Consider a matrix $N = (\alpha_{i,j})$ in $\mathbb{Z}^{n\times\ell}$, a set $T = \{t_1, \dots, t_\ell\}$ of elements of $\mathbb{K}$, and an integer g with $1 \le g < \ell$. By definition, the triple (N, T, g) written in the form*

$$\begin{array}{c} \left(\begin{array}{ccc|ccc} \alpha_{1,1} & \dots & \alpha_{1,g} & \alpha_{1,g+1} & \dots & \alpha_{1,\ell} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,1} & \dots & \alpha_{n,g} & \alpha_{n,g+1} & \dots & \alpha_{n,\ell} \end{array}\right) \begin{array}{c} u_1 \\ \vdots \\ u_n \end{array} \\ \begin{array}{ccccccc} t_1 & \dots & t_g & t_{g+1} & \dots & t_\ell \end{array} \end{array}, \tag{3}$$

represents the fraction $q = \dfrac{\sum_{i=1}^{g} t_i m_i}{\sum_{i=g+1}^{\ell} t_i m_i}$ where $m_i = u_1^{\alpha_{1,i}} u_2^{\alpha_{2,i}} \cdots u_n^{\alpha_{n,i}}$. The triple (N, T, g) (or simply N) is called a matrix representation of q.

Example 1. Taking $g = 2$, $U = \{a, b\}$, and $\mathbb{K} = \mathbb{Q}(x, y)$, the triple

$$N = \begin{array}{c} \left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \end{array}\right) \begin{array}{c} a \\ b \end{array} \\ \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \end{array} \tag{4}$$

with $t_1 = 2, t_2 = 3xy, t_3 = y^2, t_4 = 5x$ represents the fraction

$$q_1 = \frac{2a + 3b^2xy}{ay^2 + 5bx} \in \mathbb{K}(U). \tag{5}$$

Proposition 1. *Consider a triple (N, T, g) representing a fraction q. Then, for any column vector $v \in \mathbb{Z}^n$, the triple $(N + v\mathbb{1}, T, g)$ also represents the fraction q.*

Proof. For any integer $\delta \in \mathbb{Z}$, adding $\delta\mathbb{1}$ to the k-th row of N amounts to multiply both numerator and denominator of q by the same monomial u_k^δ.

Example 2. By Proposition 1 with $v = (-1, 2)$, the two triples

$$\begin{array}{c} \left(\begin{array}{cc|cc} 1 & 2 & 1 & 1 \\ -2 & -2 & -2 & -1 \end{array}\right) \begin{array}{c} \bar{a} \\ \bar{b} \end{array} \\ \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \end{array} \tag{6} \qquad \begin{array}{c} \left(\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \begin{array}{c} \bar{a} \\ \bar{b} \end{array} \\ \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \end{array} \tag{7}$$

represent the same fraction $\bar{q}_1$ of $\mathbb{K}(\bar{a}, \bar{b})$ written in two different ways:

$$\bar{q}_1 = \frac{2\frac{\bar{a}}{\bar{b}^2} + 3\frac{\bar{a}^2}{\bar{b}^2}xy}{\frac{\bar{a}}{\bar{b}^2}y^2 + 5\frac{\bar{a}}{\bar{b}}x} \tag{8} \qquad \bar{q}_1 = \frac{2 + 3\bar{a}xy}{y^2 + 5\bar{b}x}. \tag{9}$$

2.2 Monomial Map

In the following definition, the ring $\mathbb{K}(\bar{U}^{1/p})$, where $\bar{U} = \{\bar{u}_1, \dots, \bar{u}_n\}$, denotes the ring of fractions $\mathbb{K}\left(\bar{u}_1^{1/p}, \dots, \bar{u}_n^{1/p}\right)$.

Definition 2. *Consider an invertible matrix $C \in \mathbb{Q}^{n\times n}$, and two sets of variables $U = \{u_1, u_2, \ldots, u_n\}$ and $\bar{U} = \{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_n\}$. The matrix C defines the ring homomorphism ϕ^C from $\mathbb{K}(U)$ to $\mathbb{K}(\bar{U}^{1/p})$, where p is the* lcm *of all denominators of the elements of C, in the following way: $\phi^C(u_k) = \prod_{i=1}^{n} \bar{u}_i^{c_{i,k}}$ for $1 \leq k \leq n$. The map ϕ^C is called a monomial map. One simply denotes ϕ^C by ϕ when no confusion is possible.*

In this article, we will consider special monomials maps ϕ and fractions q such that $\phi(q)$ is also a rational fraction of $\mathbb{K}(\bar{U})$.

2.3 Action of a Monomial Map

Proposition 2. *Consider a triple (N, T, g) representing a fraction q in $\mathbb{K}(U)$, and an invertible matrix C in $\mathbb{Q}^{n\times n}$. If CN only contains elements of $\mathbb{Z}$, then the triple (CN, T, g) is a matrix representation of the fraction $\phi^C(q)$ of $\mathbb{K}(\bar{U})$.*

Proof. Immediate.

Corollary 1. *Consider a triple (N, T, g) representing a fraction q, an invertible matrix C in $\mathbb{Q}^{n\times n}$, and a column vector v in $\mathbb{Q}^n$. If $\bar{N} = CN + v\mathbb{1}$ only contains elements of $\mathbb{Z}$, then the triple $(\bar{N}, T, g)$ is a matrix representation of $\phi^C(q)$.*

Proof. Direct consequence of Propositions 1 and 2.

Example 3. Let us respectively denote by N and $\bar{N}$ the matrices from Equations (4) and (7), and consider the invertible matrix C

$$\begin{array}{cc} \begin{pmatrix} 1 & 1 \\ -2 & -1 \end{pmatrix} & \begin{array}{c} \bar{a} \\ \bar{b} \end{array} \\ \begin{array}{cc} a & b \end{array} & \end{array}. \tag{10}$$

Recall the fractions q_1 and $\bar{q}_1$ from Examples 1 and 2. One easily checks that $\bar{N} = CN + v\mathbb{1}$, where $v = (-1, 2)$. Consequently Corollary 1 implies that $\bar{q}_1$ is indeed equal to $\phi^C(q_1)$.

3 Sparsifying a Fraction

Consider a triple (N, T, g) representing a fraction $q \in \mathbb{K}(U)$ with the notations of Definition 1. A sparse matrix N means that many monomials m_i involve a few u_j, implying that the fraction q is sparse w.r.t. to the u_j.

In this article, we have chosen to make the matrix N sparsest in order to simplify the corresponding fraction q. To do so, we allow ourselves monomial changes of variables on U. More precisely, by relying on Corollary 1, we look for an invertible matrix C in $\mathbb{Q}^{n\times n}$ and a column vector v in $\mathbb{Q}^n$, such that the matrix $\bar{N} = CN + v\mathbb{1}$ only has integer values and is sparsest. Anticipating on the algorithms, the matrix C and the vector v given in Example 3 yield a sparsest possible matrix $\bar{N}$.

[8, Algorithm CSB (Compute Sparsest Basis)] solves the problem above in the particular case where v is the zero vector. Indeed, [8, Algorithm CSB] takes as input a full row rank matrix M and returns a sparsest (i.e. with the least number of nonzeros) matrix $\bar{M}$ with entries in $\mathbb{Z}$, which is *row-equivalent* to M (recall two matrices A and B of the same dimension are called row-equivalent if $A = PB$ for some invertible matrix P).

As a consequence, Algorithm CSB needs to be generalized to compute a sparsest basis of the rows of N "modulo" some other basis ; this is what Algorithm CSBmodulo does.

3.1 Algorithm CSBmodulo

Algorithm CSBmodulo below takes as input two matrices N and P such that $\begin{pmatrix} N \\ P \end{pmatrix}$ has a full row rank. It returns a matrix $\bar{N}$ and an invertible matrix $C \in \mathbb{Q}^{n \times n}$ such that $\bar{N} = CN + VP$ for some matrix V in $\mathbb{Q}^{n \times s}$. Moreover, $\bar{N}$ is sparsest and only has entries in $\mathbb{Z}$. The proof of Algorithm CSBmodulo is quite technical, especially proving that the computed $\bar{N}$ is sparsest. Since the proof is not necessary to understand the rest of the article, it has been placed in the appendix.

The idea of Algorithm CSBmodulo is the following. One first considers N and P in a symmetric way by building a sparsest basis of $M = \begin{pmatrix} N \\ P \end{pmatrix}$ at Line 2, thus obtaining a matrix $\bar{M}$, whose rows are then sorted by increasing number of nonzeros at Line 4. As a consequence, $\bar{M} = DM$ for some invertible matrix D. Then the matrix $\bar{N}$ is obtained by choosing n rows $\bar{M}_{r_1}, \ldots, \bar{M}_{r_n}$ of $\bar{M}$. Denoting by E the matrix composed of the n first columns of D, for any choice of rows in $\bar{M}$, one gets $\bar{N} = CN + VP$ for some matrix V, and where C is composed by the rows $E_{r_1}, \ldots, E_{r_n}$. The choice of rows has to be done carefully. First, the matrix C should be invertible. This condition is ensured by Line 9. Second, the matrix $\bar{N}$ should be sparsest. This condition will be ensured by first selecting the sparsest rows (i.e. the first rows of $\bar{M}$ since its rows are sorted).

Example 4. Take the following full row rank matrices

$$N = \begin{pmatrix} 1&1&1&0&1&0&1&1&1 \\ 0&1&0&1&1&0&1&0&0 \\ 0&0&0&0&1&0&1&0&0 \end{pmatrix} \text{ and } P = \begin{pmatrix} 1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&1&1&1 \end{pmatrix}$$

considered later in Example 12. The matrices D and $\bar{M}$ computed by Lines 2–5 of Algorithm CSBmodulo are (using our implementation of CSB)

$$\bar{M} = \begin{pmatrix} 0&1&0&0&1&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0 \\ 1&0&1&0&0&0&0&0&0 \\ 0&1&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0 \\ 0&0&0&0&0&0&1&1&1 \end{pmatrix} \text{ and } D = \begin{pmatrix} 1&1&-1&-1&0&-1 \\ 0&0&0&0&1&0 \\ 0&-1&1&1&0&0 \\ 0&1&-1&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&0&1 \end{pmatrix}.$$

Algorithm 1: $\mathsf{CSBmodulo}(N, P)$

Input: Two matrices $N \in \mathbb{Z}^{n\times\ell}$ and $P \in \mathbb{Z}^{s\times\ell}$ such that $\begin{pmatrix} N \\ P \end{pmatrix}$ has full row rank.

Output: A matrix $\bar{N} \in \mathbb{Z}^{n\times\ell}$ and an invertible matrix $C \in \mathbb{Q}^{n\times n}$ such that $\bar{N} = CN + VP$ for some matrix V in $\mathbb{Q}^{n\times s}$. Moreover, $\bar{N}$ is sparsest.

1 **begin**
2 $\quad M \leftarrow \begin{pmatrix} N \\ P \end{pmatrix}$;
3 $\quad \bar{M} \leftarrow \mathsf{CSB}(M)$;
4 $\quad$ Sort the rows of $\bar{M}$ by increasing number of nonzeros (from top to bottom) ;
5 $\quad$ Compute the invertible matrix $D \in \mathbb{Q}^{(n+s)\times(n+s)}$ such that $\bar{M} = DM$;
6 $\quad$ Denote by $E \in \mathbb{Q}^{(n+s)\times n}$ the n first columns of D ;
7 $\quad$ Consider empty matrices $C \in \mathbb{Q}^{0\times n}$ and $\bar{N} \in \mathbb{Z}^{0\times\ell}$;
8 $\quad$ **for** i *from* 1 *to* $n+s$ **do**
9 $\quad\quad$ **if** $\mathrm{Rank}\begin{pmatrix} C \\ E_i \end{pmatrix} > \mathrm{Rank}(C)$ **then**
10 $\quad\quad\quad C \leftarrow \begin{pmatrix} C \\ E_i \end{pmatrix}$; $\bar{N} \leftarrow \begin{pmatrix} \bar{N} \\ \bar{M}_i \end{pmatrix}$;
11 $\quad$ **return** $\bar{N}$, C ;

The extraction from $\bar{M}$ of the matrix $\bar{N}$ by Lines 6–10 will consider the three first columns of D and ignore rows 2, 4 and 6 of $\bar{M}$, yielding

$$\bar{N} = \begin{pmatrix} 0\,1\,0\,0\,1\,0\,0\,0\,0 \\ 1\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,1\,0\,0 \end{pmatrix}.$$

Remark 1. Algorithm $\mathsf{CSBmodulo}$ extracts the matrix C from the n first columns of D. This extraction could be made by computing the row rank profile [9]: the *row rank profile* of an $m \times n$ matrix A of rank r is the lexicographically smallest sequence of r indices of linearly independent rows of A. An $PLUQ$ decomposition algorithm using Gaussian elimination is proposed in [9] to compute such a set.

3.2 Algorithm getSparsestFraction

This section presents Algorithm $\mathsf{getSparsestFraction}$ which computes a sparsest representation of a fraction q. It relies on Algorithm $\mathsf{CSBmodulo}$. We first present Propositions 3 and 4 which are needed when the number of variables in U can be decreased by a monomial map. Propositions 3 and 4 are in fact a particular case of a more general treatment based on scaling type symmetries.

Proposition 3. *Consider a triple (N, T, g) representing a fraction $q \in \mathbb{K}(U)$. If N has not full row rank, then there exist a monomial map ϕ^C and a full row rank matrix N' such that (N', T, g) represents the fraction $\bar{q} = \phi^C(q)$.*

Proof. If $N \in \mathbb{Z}^{n\times\ell}$ with $\mathrm{Rank}(N) = p < n$, then there exists a full row rank matrix $N' \in \mathbb{Z}^{p\times\ell}$ and an invertible matrix $C \in \mathbb{Q}^{n\times n}$ such that $CN = \begin{pmatrix} N' \\ 0 \end{pmatrix}$. By Proposition 2, the fraction $\bar{q}$ represented by the triple (CN, T, g) is equal to $\phi^C(q)$. Because the matrix CN has $n-p$ zero rows, one can discard the $n-p$ last variables and the triple (N', T, g) still represents the fraction $\bar{q}$.

Example 5. Take $q_2 = \frac{2+3abxy}{y^2+5abx} \in \mathbb{K}(U)$ with $\mathbb{K} = \mathbb{Q}(x, y)$ and $U = \{a, b\}$. A matrix representation of q_2 is $N = \left(\begin{array}{cc|cc} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{array}\right) \begin{matrix} a \\ b \end{matrix}$ with $t_1 = 2, t_2 = 3xy, t_3 = y^2, t_4 = 5x$. By taking $C = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$ and $N' = \begin{pmatrix} 0 & 1 & 0 & 1 \end{pmatrix}$, one has $CN = \begin{pmatrix} N' \\ 0 \end{pmatrix}$. Consequently, the monomial map ϕ^C satisfies $\phi^C(a) = \bar{a}/\bar{b}$ and $\phi^C(b) = \bar{b}$, thus $\phi^C(q_2) = \frac{2+3\bar{a}xy}{y^2+5\bar{a}x} \in \mathbb{K}(\bar{U})$.

Proposition 4. *Consider a triple (N, T, g) representing a fraction $q \in \mathbb{K}(U)$, such that N has full row rank. If $\begin{pmatrix} N \\ \mathbb{1} \end{pmatrix}$ has not full row rank, then there exist a monomial map ϕ^C and a matrix N' such that (N', T, g) represents the fraction $\bar{q} = \phi^C(q)$, where $\begin{pmatrix} N' \\ \mathbb{1} \end{pmatrix}$ has full row rank.*

Proof. Since N has full row rank and $\begin{pmatrix} N \\ \mathbb{1} \end{pmatrix}$ has not full row rank, one has $\mathbb{1} = \sum_{i=1}^{n} \beta_i N_i$ for some $\beta_i \in \mathbb{Q}$. Without loss of generality, one assumes that $\beta_n \neq 0$ (by exchanging some variables in the set U). Using Corollary 1 and the relation $CN - v\mathbb{1} = \begin{pmatrix} N' \\ 0 \end{pmatrix}$ where $C = \begin{pmatrix} & & 0 \\ & I & \vdots \\ & & 0 \\ \beta_1 & \cdots & \beta_n \end{pmatrix}$, $v = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$ and $N' = \begin{pmatrix} N_1 \\ \vdots \\ N_{n-1} \end{pmatrix}$, the fraction $\bar{q}$ represented by the (CN, T, g) is equal to $\phi^C(q)$. Because the last row of CN is zero, one can discard the last variable of U, and the triple (N', T, g) still represents the fraction $\bar{q}$. Moreover, the matrix $\begin{pmatrix} N' \\ \mathbb{1} \end{pmatrix}$ has full row rank.

Example 6. Take $q_3 = \frac{2a+3bxy}{ay^2+5bx} \in \mathbb{K}(U)$ with $\mathbb{K} = \mathbb{Q}(x, y)$ and $U = \{a, b\}$. A matrix representation of q_3 is $N = \left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right) \begin{matrix} a \\ b \end{matrix}$ with $t_1 = 2, t_2 = 3xy, t_3 = y^2, t_4 = 5x$. One has $CN - v\mathbb{1} = \begin{pmatrix} N' \\ 0 \end{pmatrix}$ where $C = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $v = (0, 1)$ and $N' = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix}$. Consequently, $\bar{q}_3$ equals $\phi^C(q_3) = \frac{2\bar{a}\bar{b}+3\bar{b}xy}{\bar{a}\bar{b}y^2+5\bar{b}x} = \frac{2\bar{a}+3xy}{\bar{a}y^2+5x} \in \mathbb{K}(\bar{a})$. Furthermore, $\bar{q}_3$ can be represented by $N' = \left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \end{array}\right) \bar{a}$, where $\begin{pmatrix} N' \\ \mathbb{1} \end{pmatrix}$ has full row rank.

Algorithm 2: getSparsestFraction(N, T, g)

Input: A triple (N, T, g) representing a fraction $q \in \mathbb{K}(U)$

Output: A triple $(\bar{N}, T, g)$ and an invertible matrix C in $\mathbb{Q}^{n \times n}$, such that $(\bar{N}, T, g)$ represents $\phi^C(q)$. Moreover, $\bar{N}$ is sparsest and only has entries in $\mathbb{Z}$.

```
begin
    Apply Propositions 3 and 4 for ensuring that (N; 𝟙) has full row rank ;
    N̄, C ← CSBmodulo(N, 𝟙) ;
    return (N̄, T, g), C ;
```

Example 7. Let us apply Algorithm getSparsestFraction on the triple (N, T, g) of (4), representing the rational fraction q_1 of (5). The matrix $\begin{pmatrix} N \\ \mathbb{1} \end{pmatrix}$ has full row rank (which is 3) so getSparsestFraction(N, T, g) calls CSBmodulo$(N, \mathbb{1})$. During the CSBmodulo$(N, \mathbb{1})$ call, the matrix $\bar{M}$ computed at Line 4 and the invertible matrix D computed at Line 5 are:

$$\bar{M} = \left(\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{array}\right) \text{ and } D = \begin{pmatrix} 1 & 1 & -1 \\ -2 & -1 & 2 \\ 1 & 0 & 0 \end{pmatrix}.$$

Moreover, the matrix C computed by Lines 6–10 is simply the upper-left two by two submatrix of D, which is exactly the matrix (10). Finally the matrix $\bar{N}$ computed by Lines 6–10 is obtained by selecting the two first rows of $\bar{M}$, yielding the matrix (7) and its corresponding fraction (9).

Remark that if one simply computes CN, one gets the matrix representation (6) corresponding to the fraction (8), which is not as nice as the fraction (9) represented by $\bar{N}$.

The following example shows that the monomial map ϕ^C computed by Algorithm getSparsestFraction can involve fractional exponents.

Example 8. Take the matrix representation

$$N = \begin{array}{c} \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \\ \begin{array}{cccc} x & y & x & y \end{array} \end{array} \begin{array}{c} a \\ b \\ c \\ \end{array}$$

representing the fraction $q = \frac{ax+b^3y}{bx+cy} \in \mathbb{K}(a, b, c)$ where $\mathbb{K} = \mathbb{Q}(x, y, z)$. Then getSparsestFraction(N, T, g) returns

$$\bar{N} = \begin{array}{c} \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right) \\ \begin{array}{cccc} x & y & x & y \end{array} \end{array} \begin{array}{c} \bar{a} \\ \bar{b} \\ \bar{c} \\ \end{array} \quad \text{and } C = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 1/2 \\ -3/2 & -1/2 & -3/2 \end{pmatrix}.$$

The following example shows that negative exponents for the parameters are sometimes needed.

Example 9. Take the matrix representation

$$N = \begin{array}{c} \left(\begin{array}{ccc|c} 0 & 1 & -1 & 0 \\ 2 & x & y & z+1 \end{array} \right) \end{array} a$$

representing the fraction $q = \frac{2+ax+\frac{y}{a}}{z+1} \in \mathbb{K}(a)$ where $\mathbb{K} = \mathbb{Q}(x, y, z)$. Then $\mathsf{getSparsestFraction}(N, T, g)$ returns N, showing that the representation above, which contains a negative exponent, is already sparsest. Moreover, N is the unique sparsest representation (in the sense of this article), since any addition of a multiple of $\mathbb{1}$ to N will produce at least three nonzeros.

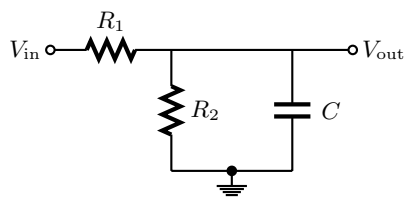

Fig. 1. An electric circuit with two resistors R_1 and R_2 and a capacitor C.

Example 10. The transfer function $H = V_{\text{out}}/V_{\text{in}}$ of the circuit given in Figure 1 is $H = \dfrac{R_2}{R_1 + R_2 + iR_1R_2Cw}$, where $i^2 = -1$ and w is the frequency. We consider $H \in \mathbb{K}(R_1, R_2, C)$ with $\mathbb{K} = \mathbb{C}(w)$ i.e. we consider R_1, R_2 and C as parameters. It can be shown that H admits a scaling symmetry acting on R_1, R_2, and C. As a consequence, using symmetry based techniques, one can discard one parameter by a suitable (nonunique) change of variables. For example, H can be rewritten as

$$H = \frac{x}{1 + x + ix\tau_1 w} \tag{11}$$

by taking $x = R_2/R_1$ and $\tau_1 = R_1C$. Also, H can be rewritten as

$$H = \frac{1}{y + 1 + iy\tau_2 w} \tag{12}$$

by taking $y = R_1/R_2$ and $\tau_2 = R_2C$. Other changes of variables would also be possible but are not given here.

The relations (11) and (12) are not sparsest. Relation (12) can easily be made sparsest using the change of variables $a = y\tau_2$ yielding $H = \frac{1}{y+1+iaw}\cdot$ However, Relation (11) requires a slightly more subtle treatment, by first dividing both numerators and denominators by x thus writing $H = \frac{1}{\frac{1}{x}+1+i\tau_1 w}$ and then taking

$y = 1/x$ (please note that $a = \tau_1$ since $a = y\tau_2 = (R_1/R_2)R_2C = R_1C = \tau_1$). The division by x will be automatically discovered by getSparsestFraction applied to the triple

$$N = \begin{array}{c} \left(\begin{array}{c|ccc} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right) \\ \begin{array}{cccc} 1 & 1 & 1 & iw \end{array} \end{array} \begin{array}{c} x \\ \tau_1 \\ \end{array}$$

thanks to the line $\mathbb{1}$ added to N during the call of CSB, allowing to replace the first line N_1 of N by $N_1 - \mathbb{1} = \big(0 \mid -1\ 0\ 0\big)$. Moreover, our implementation of CSB will negate the row $\big(0 \mid -1\ 0\ 0\big)$ since it only contains nonpositive entries.

3.3 Complexity of getSparsestFraction

Except the CSBmodulo call, the instructions of Algorithm getSparsestFraction are done at most in $\mathrm{O}(n^3)$. In fact, the complexity of getSparsestFraction is dominated by the one of CSBmodulo, which can be exponential in n in the worst case [8].

4 Sparsifying a Sum of Fractions

We now consider a sum of rational fractions in $\mathbb{K}(U)$. It could be rewritten as a single fraction by reducing the fractions to the same denominator. However, one will avoid such a manipulation for two reasons. First, this can increase the sizes of the numerator and denominator. Second, a practitioner might want to keep an expression as a sum of fractions. This last point occurs for example in the Biology context with expressions of the shape $p + \sum \frac{V_i x_i}{x_i + k_i}$ where the x_i are concentrations, p is a polynomial in the x_i and some other parameters, and the V_i and k_i are the constants of some Michaelis-Menten terms [10,11].

4.1 Matrix Representation of a Sum of Fractions

Definition 3. *Consider s fractions $q_i \in \mathbb{K}(U)$, where each fraction q_i is represented by a triple (N^i, T^i, g^i). Then, the set of the triples $H = \{(N^i, T^i, g^i)_{1 \le i \le s}\}$ is called the* matrix representation of the sum $S = \sum_{i=1}^{s} q_i$. *The set H of triples can be written as the following matrix, where the double bar separates two different fractions:*

$$N = \big(N^1 \| N^2 \| \ldots \| N^s \big) . \tag{13}$$

Example 11. Consider the sum of rational fractions S_1 defined by

$$S_1 = \frac{2a + abxy}{ay^2 + 7bx} + abcy + \frac{abcy^2 + 3ay}{ax} \cdot \tag{14}$$

A possible matrix representation of S_1 is

$$\begin{array}{c} \left(\begin{array}{cc|cc||c|c||cc|c} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right) \\ \begin{array}{ccccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 \end{array} \end{array} \begin{array}{c} a \\ b \\ c \\ \end{array} \tag{15}$$

with $t_1 = 2, t_2 = xy, t_3 = y^2, t_4 = 7x, t_5 = y, t_6 = 1, t_7 = y^2, t_8 = 3y, t_9 = x$.

Consider a matrix representation $(N^i, T^i, g^i)_{1 \leq i \leq s}$ of a sum $S = \sum q_i$. Because Proposition 1 can be applied independently on each fraction q_i, one introduces Proposition 5 which generalizes Proposition 1, after introducing the matrix $\mathbb{1}^S$ of size $s \times (\sum_{i=1}^{s} \mathrm{card}(T^i))$ defined as

$$\mathbb{1}^S = \begin{pmatrix} \mathbb{1}_{\mathrm{card}(T^1)} & & & \\ & \mathbb{1}_{\mathrm{card}(T^2)} & & \\ & & \ddots & \\ & & & \mathbb{1}_{\mathrm{card}(T^s)} \end{pmatrix}. \tag{16}$$

Proposition 5. *Consider a set of triples (N^i, T^i, g^i) representing a sum of rational fractions $S = \sum_{i=1}^{s} q_i$ as written in Definition 3. For any $V \in \mathbb{Z}^{n \times s}$, $\bar{N} = N + V\mathbb{1}^S$ is a matrix representation of S.*

Proof. Apply Proposition 1 on each fraction q_i.

4.2 Action of a Monomial Map

The following Proposition 6 and Corollary 2 are respectively the generalizations of Proposition 2 and Corollary 1 for sum of fractions.

Proposition 6. *Consider a set of triples (N^i, T^i, g^i) representing a sum of rational fractions $S = \sum_{i=1}^{s} q_i$ as written in Definition 3. Consider an invertible matrix C in $\mathbb{Q}^{n \times n}$. If CN only have entries in $\mathbb{Z}$, then CN is a matrix representation of $\bar{S} = \phi^C(S)$.*

Corollary 2. *Consider a set of triples (N^i, T^i, g^i) representing a sum of rational fractions $S = \sum_{i=1}^{s} q_i$ as written in Definition 3. Consider an invertible matrix C in $\mathbb{Q}^{n \times n}$ and a vector V in $\mathbb{Q}^{n \times s}$. If $CN + V\mathbb{1}^S$ only have integer entries, then $CN + V\mathbb{1}^S$ is a matrix representation of $\bar{S} = \phi^C(S)$.*

Example 12. Take the map $\phi(a) = \bar{a}$, $\phi(b) = \frac{\bar{a}}{\bar{b}}$, $\phi(c) = \frac{\bar{b}\bar{c}}{\bar{a}}$ defined by the following invertible matrix C:

$$\begin{array}{cc} \begin{pmatrix} 1 & 1 & -1 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{pmatrix} & \begin{matrix} \bar{a} \\ \bar{b} \\ \bar{c} \end{matrix} \\ \begin{matrix} a & \quad b & \quad c \end{matrix} & \end{array}. \tag{17}$$

With this change of variables, the fraction S_1 of Example 11 becomes

$$\bar{S}_1 = \frac{2\bar{a} + \frac{\bar{a}^2}{\bar{b}} xy}{\bar{a}y^2 + 7\frac{\bar{a}}{\bar{b}}x} + \bar{a}\bar{c}y + \frac{\bar{a}\bar{c}y^2 + 3\bar{a}y}{\bar{a}x}. \tag{18}$$

Taking the matrix representation N of S_1 as written in (15), CN is a matrix representation of $\bar{S}_1 = \phi^C(S_1)$. The matrix CN can however be made sparser by considering

$$V = \begin{pmatrix} -1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{19}$$

and by applying Corollary 2. Indeed the following matrix $\bar{N} = CN + V\mathbb{1}^{S_1}$ is also a matrix representation of $\bar{S}_1$

$$\left(\begin{array}{cc|cc||c|c||cc|c} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right) \begin{array}{c} \bar{a} \\ \bar{b} \\ \bar{c} \end{array}, \tag{20}$$

and represents

$$\bar{S}_1 = \frac{2\bar{b} + \bar{a}xy}{\bar{b}y^2 + 7x} + \bar{a}\bar{c}y + \frac{\bar{c}y^2 + 3y}{x}\cdot \tag{21}$$

Remark that adding $V\mathbb{1}^{S_1}$ to CN corresponds to multiplying by $\frac{\bar{b}}{\bar{a}}$ the numerator and the denominator of the first fraction, and dividing by $\bar{a}$ the numerator and the denominator of the third fraction in (18).

4.3 Algorithm getSparsestSumOfFractions

Algorithm getSparsestSumOfFractions relies on the same ideas as Section 3: given a set of triples (N^i, T^i, g^i) representing a sum of rational fractions $S = \sum_{i=1}^{s} q_i$ as written in Definition 3, one looks for an invertible matrix C and a matrix V such that $CN + V\mathbb{1}^S$ is sparsest.

We first present Proposition 7, which is a slight generalization of Proposition 4. As in Section 3, Proposition 7 is needed when the sum of fractions admits scaling type symmetries in the U variables.

Proposition 7. *Consider a set of triples (N^i, T^i, g^i) representing a sum of rational fractions $S = \sum_{i=1}^{s} q_i$ as written in Definition 3. Assume that N has full row rank. If $\begin{pmatrix} N \\ \mathbb{1}^S \end{pmatrix}$ has not full row rank, then there exists a monomial map ϕ^C and matrices N'^i such that the set of triples (N'^i, T^i, g^i) represents the fraction $\bar{S} = \phi^C(S)$, where $\begin{pmatrix} N' \\ \mathbb{1}^S \end{pmatrix}$ has full row rank.*

Proof. The matrix $M = \begin{pmatrix} N \\ \mathbb{1}^S \end{pmatrix}$ has not full row rank, so there exists a non trivial linear dependency between the rows of M. Since both matrices N and $\mathbb{1}^S$ have full row rank, the linear dependency necessarily involves a row of N with a nonzero coefficient. The end of the proof is similar to the one of Proposition 4.

Algorithm 3: getSparsestSumOfFractions(H)

Input: A set $H = \{(N^i, T^i, g^i)_{1\leq i\leq s}\}$ representing a sum of rational fractions $S = \sum_{i=1}^{s} q_i \in \mathbb{K}(U)$ as in Definition 3.

Output: A set $\bar{H} = \{(\bar{N}^i, T^i, g^i)_{1\leq i\leq s}\}$ and an invertible matrix C in $\mathbb{Q}^{n\times n}$, such that $\bar{H}$ represents $\phi^C(q)$. Moreover $\bar{N} = \left(\bar{N}^1 || \cdots || \bar{N}^s\right)$ is sparsest and only has entries in $\mathbb{Z}$.

```
begin
    Apply Propositions 3 and 7 for ensuring that (N ; 1^S) has full row rank ;
    N̄, C ← CSBmodulo(N, 1^S) ;
    Write N̄ as (N̄^1 || ··· || N̄^s) ;
    return {(N̄^i, T^i, g^i)_{1≤i≤s}}, C ;
```

Example 13. Recall the fraction S_1 from Example 11 and its matrix representation N of Equation (15). One checks that the matrix $\begin{pmatrix} N \\ \mathbb{1}^S \end{pmatrix}$ has full row rank. Consequently getSparsestSumOfFractions(N) calls CSBmodulo$(N, \mathbb{1}^S)$. Our implementation of CSBmodulo returns the matrix $\bar{N}$ of Equation (20) and the matrix C of Equation (17). Consequently, getSparsestSumOfFractions computes the sparsest sum of fractions $\bar{S}_1 = \phi^C(S_1)$ of Equation (21).

4.4 Application to systems of ODEs

The techniques presented for the sum of fractions can be adapted for systems of differential equations of the form $X'(t) = F(\Theta, X(t))$ (where $X(t)$ is a vector of functions, $F(\Theta, X(t))$ is a vector of fractions and the Θ are parameters), such as Equation (1). Indeed, one can consider the sum of fractions $\sum_{i=1}^{s} F_i(\Theta, X(t))$, where the $F_i(\Theta, X(t))$ denotes the components of the vector $F(\Theta, X(t))$, and apply Algorithm getSparsestSumOfFractions.

Example 14. Consider the sum S of the three right-hand sides of Equation (1) seen as a fraction of $\mathbb{K}(k_1, k_2, k_3)$ with $\mathbb{K} = \mathbb{Q}(\theta, \alpha, \rho_b, \rho_f, \delta_M, \delta_P, \beta)$. It can represented by

$$N = \begin{array}{c} \left(\begin{array}{cc|c||c|c||cc|cccc} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}\right) \begin{array}{l} k_1 \\ k_2 \\ k_3 \end{array} \\ \begin{array}{ccccccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} \end{array} \end{array}.$$

with $t_1 = \theta(1-G)$, $t_2 = -\alpha P^4 G$, $t_3 = 1$, $t_4 = \rho_b(1-G) + \rho_f G - \delta_M M$, $t_5 = 1$, $t_6 = 4\theta(1-G) - \delta_P P + \beta M$, $t_7 = -4\alpha P^4 G$, $t_8 = 16P^3$, $t_9 = 9P^2$, $t_{10} = 4P$, $t_{11} = 1$.

Algorithm getSparsestSumOfFractions(S) yields the sparsest Equation (2) and the monomial map $\phi(k_1) = \bar{k}_1, \phi(k_2) = \bar{k}_2/\bar{k}_1, \phi(k_3) = \bar{k}_3/\bar{k}_2$ encoded by the

matrix $C = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$. Please note that the matrix $\mathbb{1}^S$ was not useful in the computations, since $\bar{N} = CN$.

5 Proof of Algorithm CSBmodulo

This last section is the most technical part of the article. We first present some definitions and intermediate results. We then present a new corollary showing that all the sparsest row bases of the same matrix (see Definition 4) share some common structure (see Corollary 4). Finally the proof of Algorithm CSBmodulo is presented.

Let M (resp. v) be a matrix (resp. vector). We denote by $\mathcal{N}(M)$ (resp. $\mathcal{N}(v)$) the number of nonzero coefficients of M (resp. v).

Definition 4. *Let $M \in \mathbb{Q}^{n\times\ell}$ a matrix with full row rank. One calls* sparsest row basis of M *any matrix $\bar{M}$ which is sparsest and row-equivalent to M.*

Definition 5. *Let $N \in \mathbb{Q}^{n\times\ell}$ and $P \in \mathbb{Q}^{n\times s}$ two matrices with full row rank. Assume that $\begin{pmatrix} N \\ P \end{pmatrix}$ has full row rank. One calls* sparsest row basis of N modulo P *any matrix N' which is sparsest and satisfies $N' = CN + VP$ for some invertible matrix C and some matrix V.*

Following Lemma 1 is a rephrasing of [8, Theorem 1]. It is the key ingredient ensuring the greedy approach chosen in [8], which consists in repeatedly reducing the number of nonzeros of some row of M, until it is not possible anymore.

Lemma 1. *Take a full row rank matrix M. The matrix M is not a sparsest row basis of M* iff *there exists an index i and a row vector v such that $v_i \neq 0$ and $\mathcal{N}(vM) < \mathcal{N}(M_i)$.*

Corollary 3. *Take a full row rank matrix M. If the matrix M is not a sparsest row basis of M, Lemma 1 applies. Moreover, replacing the row M_i by vM yields a sparser row-equivalent matrix. See [8, Algorithm EnhanceBasis] for an implementation of Corollary 3.*

Proposition 8. *Take a sparsest basis N' of N modulo P with the same assumptions as in Definition 5. Then there exist matrices P', C', V', G', W' such that $\begin{pmatrix} N' \\ P' \end{pmatrix}$ is a sparsest basis of the matrix $\begin{pmatrix} N \\ P \end{pmatrix}$, with $\begin{pmatrix} N' \\ P' \end{pmatrix} = \begin{pmatrix} C' & V' \\ G' & W' \end{pmatrix} \begin{pmatrix} N \\ P \end{pmatrix}$ where C' is invertible.*

Proof. The existence of C' and V' is given by Definition 5. Consider the matrix $M' = \begin{pmatrix} N' \\ P \end{pmatrix}$. If M' is not a sparsest row basis of $\begin{pmatrix} N \\ P \end{pmatrix}$, it can be made sparser using Corollary 3. Moreover, the row to improve is necessarily not a row of N',

since N' is a sparsest row basis of N modulo P. After applying Corollary 3 a certain number of times, one gets a sparsest row basis $\begin{pmatrix} N' \\ P' \end{pmatrix}$ of $\begin{pmatrix} N \\ P \end{pmatrix}$ which proves the existence of P', G' and W'.

Proposition 9. *Take a sparsest basis $\bar{M}$ of a full row rank matrix M. Assume that the rows of $\bar{M}$ and M are sorted by increasing number of nonzeros. Let D be the matrix defined by $D\bar{M} = M$. Then $\mathcal{N}(\bar{M}_i) \leq \mathcal{N}(M_i)$ for any $1 \leq i \leq n$. Moreover, for any $1 \leq i, j \leq n$, if $\mathcal{N}(M_i) < \mathcal{N}(\bar{M}_j)$, then $D_{ij} = 0$.*

Proof. Let us prove the first point and assume $\mathcal{N}(M_i) < \mathcal{N}(\bar{M}_j)$ for some i and j with $D_{ij} \neq 0$. Following ideas from Corollary 3, $\bar{M}$ is not sparsest since $\bar{M}_j$ could be replaced by the sparser row M_i since $M_i = \sum_j D_{ij}\bar{M}_j$ and $D_{ij} \neq 0$.

Let us now prove that $\mathcal{N}(\bar{M}_i) \leq \mathcal{N}(M_i)$ for any $1 \leq i \leq n$. By contradiction, assume that there exists a k such that $\mathcal{N}(\bar{M}_k) > \mathcal{N}(M_k)$ and $\mathcal{N}(\bar{M}_i) = \mathcal{N}(M_i)$ for $i \leq k-1$. Each row M_i is a linear combination of rows of $\bar{M}$. If all k first rows of M_i were linear combinations of the $k-1$ rows of $\bar{M}$, then the k first rows would not be linear independent, and M could not have full row rank. Consequently, there exist two indices i, l and a row vector v such that $i \leq k \leq l$, $\bar{M}_i = v\bar{M}$ with $v_l \neq 0$. Since $\mathcal{N}(\bar{M}_l) \geq \mathcal{N}(\bar{M}_k) > \mathcal{N}(M_k) \geq \mathcal{N}(M_i)$, the row $\bar{M}_l$ can be made sparser by replacing it by the sparser row M_i using Lemma 1. This leads to a contradiction since $\bar{M}$ is sparsest.

The new following corollary proves that all sparsest row bases of some fixed matrix share some common structure.

Corollary 4. *Take two sparsest basis $\bar{M}$ and M' of the same matrix M. Assume that the rows of $\bar{M}$ and M' are sorted by increasing number of nonzeros. Let T the matrix defined by $\bar{M} = TM'$. Then for any $1 \leq i \leq n$, one has $\mathcal{N}(\bar{M}_i) = \mathcal{N}(M'_i)$. Moreover, T is a lower block triangular matrix, where the widths of blocks correspond to the width of the blocks of rows of $\bar{M}$ which have the same number of nonzeros.*

Proof. It is a direct consequence of Proposition 9 applied twice: the first time by considering that $\bar{M}$ is a sparsest row basis of M', the second time by considering that M' is a sparsest row basis of $\bar{M}$.

Lemma 2. *Let $H \in \mathbb{Q}^{m \times \ell}$ and $U' = \mathbb{Q}^{t \times m}$. If $U'H = 0$ then* $\mathrm{Rank}(U') + \mathrm{Rank}(H) \leq m$.

Proof. Consequence of the Rank-nullity theorem applied on the transpose of H

Proof (CSBmodulo is correct). The fact that the computed matrices $\bar{N}, C$ satisfies $\bar{N} = CN + VP$ for some V is a consequence of the selection strategy in the loop. It is left to the reader. Moreover, the matrix $\bar{N}$ has entries in $\mathbb{Z}$ since it is a submatrix of $\bar{M}$ which also have entries in $\mathbb{Z}$ because it was computed by Algorithm CSB.

The difficult point is to show that $\bar{N}$ is sparsest. To prove that point, we assume that $\mathcal{N}(\bar{N}) > \mathcal{N}(N')$ for some sparsest row basis N' of N modulo P, and show a contradiction. By Proposition 8, there exists a matrix P' such that $\begin{pmatrix} N' \\ P' \end{pmatrix}$ is a sparsest basis of the matrix $\begin{pmatrix} N \\ P \end{pmatrix}$. Let us denote M' the matrix obtained by sorting the rows of $\begin{pmatrix} N' \\ P' \end{pmatrix}$ by increasing number of nonzeros. Let us introduce the indices $s_1 < \ldots < s_n$ such that $M'_{s_i} = N'_i$. By Corollary 4, $\mathcal{N}(\bar{M}_i) = \mathcal{N}(M'_i)$ for any $1 \leq i \leq n$, and there exists an invertible lower block triangular matrix T such that $\bar{M} = TM'$.

For sake of simplicity, one assumes that the number of nonzeros in the rows of $\bar{M}$ are strictly increasing, hence the matrix T is lower triangular with nonzero diagonal elements. Denote by $r_1, \ldots, r_n$ the indices of the rows of $\bar{M}$ which are selected by the loop in Algorithm CSBmodulo to produce the matrix $\bar{N}$ (i.e. $\bar{N}_i = \bar{M}_{r_i}$ for $1 \leq i \leq n$). Since we assumed $\mathcal{N}(\bar{N}) > \mathcal{N}(N')$, then $\mathcal{N}(\bar{N}_k) > \mathcal{N}(N'_k)$ for some k. By taking k minimal, one has $\mathcal{N}(\bar{N}_1) \leq \mathcal{N}(N'_1)$, $\ldots$, $\mathcal{N}(\bar{N}_{k-1}) \leq \mathcal{N}(N'_{k-1})$ and $\mathcal{N}(\bar{N}_k) > \mathcal{N}(N'_k)$. From the inequalities above, one has $r_1 \leq s_1$, $r_2 \leq s_2$, $\ldots$, $r_{k-1} \leq s_{k-1}$ and $r_k > s_k$, which we summarize here:

$$\begin{array}{ccc} \bar{M} & T & M' \\ \begin{pmatrix} \vdots \\ \bar{M}_{r_1} \\ \vdots \\ \bar{M}_{r_{k-1}} \\ \vdots \\ \vdots \\ \vdots \\ \bar{M}_{r_k} \\ \vdots \end{pmatrix} & = \begin{pmatrix} T_{1,1} & & \\ \vdots & \ddots & \\ T_{n,1} & \cdots & T_{n,n} \end{pmatrix} & \begin{pmatrix} \vdots \\ \vdots \\ M'_{s_1} \\ \vdots \\ M'_{s_{k-1}} \\ \vdots \\ M'_{s_k} \\ \vdots \\ \vdots \end{pmatrix} \end{array}.$$

Among the first s_k rows of $\bar{M}$, there are $k-1$ rows which were selected by the algorithm. As a consequence, $s_k - k + 1$ rows were not selected, implying that each unselected row E_i with $i \leq s_k$ is a linear combination of the previous rows E_j with $j < i$. By storing row-wise those linear combinations above in a matrix U, one gets a $(s_k - k + 1) \times n$ matrix, where columns from indices $s_k + 1$ to n are zero. Moreover, the matrix U has full row rank since U has a echelon form.

Let us write $D = \begin{pmatrix} E & F \end{pmatrix}$. By definition of U, one has $UD = \begin{pmatrix} UE & UF \end{pmatrix} = \begin{pmatrix} 0 & UF \end{pmatrix}$. Since $\bar{M} = DM$, then $U\bar{M} = UDM = \begin{pmatrix} 0 & UF \end{pmatrix} \begin{pmatrix} N \\ P \end{pmatrix} = UFP$.

On the other side, since $\bar{M} = TM'$, then $U\bar{M} = UTM'$. Since the columns from indices $s_k + 1$ to n of U are zero, the matrix UT also have columns from indices $s_k + 1$ to n which are zero. Moreover, UT has also full rank.

With notations of Proposition 8 and some easy computations, the product UTM' can be written as $UTHN + JP$ where H has s_k rows including the rows $1, 2, \ldots, k$ of C' and some rows of G', and some matrix J.

Consequently, $U\bar{M} = UFP = UTHN + JP$ which implies $(UF - J)P = (UTH)N$. Since $\begin{pmatrix} N \\ P \end{pmatrix}$ has full row rank, UTH (and also $(UF - J)$) is necessarily the zero matrix.

By Lemma 2, $\mathrm{Rank}(UT) + \mathrm{Rank}(H) \leq s_k$. However, $\mathrm{Rank}(UT) = s_k - k + 1$, and $\mathrm{Rank}(H) \geq k$ since the k rows $C'_1, \ldots, C'_k$ are taken from the invertible matrix C'. Thus $\mathrm{Rank}(UT) + \mathrm{Rank}(H) \geq s_k + 1$ which contradicts $\mathrm{Rank}(UT) + \mathrm{Rank}(H) \leq s_k$. As a consequence, the assumption $\mathcal{N}(\bar{N}) > \mathcal{N}(N')$ leads to a contradiction, so $\mathcal{N}(\bar{N}) \leq \mathcal{N}(N')$ and $\bar{N}$ is indeed sparsest.

References

1. Basu, S., Pollack, R., Roy, M.F.: Algorithms in Real Algebraic Geometry. 2nd edn. Volume 10 of Algorithms and Computation in Mathematics. Springer-Verlag Berlin Heidelberg (2006)
2. Lemaire, F., Ürgüplü, A.: A Method for Semi-rectifying Algebraic and Differential Systems Using Scaling Type Lie Point Symmetries with Linear Algebra. In: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation. ISSAC '10, New York, NY, USA, ACM (2010) 85–92
3. Sedoglavic, A.: Reduction of Algebraic Parametric Systems by Rectification of Their Affine Expanded Lie Symmetries. In Anai, H., Horimoto, K., Kutsia, T., eds.: Proceedings of the 2007 Algebraic Biology. AB 2007, Springer Berlin Heidelberg (2007) 277–291
4. Hubert, E., Labahn, G.: Scaling Invariants and Symmetry Reduction of Dynamical Systems. Foundations of Computational Mathematics **13**(4) (2013) 479–516
5. Fels, M., Olver, P.J.: Moving Coframes: II. Regularization and Theoretical Foundations. Acta Applicandae Mathematica **55**(2) (1999) 127–208
6. Olver, P.J.: Applications of Lie groups to differential equations. 2nd edn. Volume 107 of Graduate Texts in Mathematics. Springer Verlag New York (1993)
7. Boulier, F., Lemaire, F., Sedoglavic, A., Ürgüplü, A.: Towards an Automated Reduction Method for Polynomial ODE Models of Biochemical Reaction Systems. Mathematics in Computer Science **2**(3) (2009) 443–464
8. Lemaire, F., Temperville, A.: On Defining and Computing "Good" Conservation Laws. In Mendes, P., Dada, J.O., Smallbone, K., eds.: Proceedings of the 2014 Computational Methods in Systems Biology. CMSB 2014, Springer International Publishing Switzerland (2014) 1–19
9. Dumas, J.G., Pernet, C., Sultan, Z.: Computing the Rank Profile Matrix. In: Proceedings of the 2015 International Symposium on Symbolic and Algebraic Computation. ISSAC '15, New York, NY, USA, ACM (2015) 149–156
10. Henri, V.: Lois générales de l'action des diastases. Librairie Scientifique A. Hermann, Paris (1903)
11. Michaelis, L., Menten, M.L.: Die Kinetik der Invertinwirkung. Biochemische Zeitschrift **49** (1913) 333–369