Journal of Software for Algebra and Geometry

Decomposable sparse polynomial systems

TAYLOR BRYSIEWICZ, JOSE ISRAEL RODRIGUEZ, FRANK SOTTILE AND THOMAS YAHL

vol 11 2021

JSAG **11** (2021), 53–59
https://doi.org/10.2140/jsag.2021.11.53

The Journal of Software for
Algebra and Geometry

Decomposable sparse polynomial systems

TAYLOR BRYSIEWICZ, JOSE ISRAEL RODRIGUEZ, FRANK SOTTILE AND THOMAS YAHL

ABSTRACT: The *Macaulay*2 package `DecomposableSparseSystems` implements methods for studying and numerically solving decomposable sparse polynomial systems. We describe the structure of decomposable sparse systems and explain how the methods in this package may be used to exploit this structure, with examples.

1. INTRODUCTION. Améndola and Rodriguez [2016] gave numerical methods to efficiently solve systems of sparse polynomial equations in a family, when that family is decomposable (Definition 1). A consequence of Esterov's study of Galois groups of systems of sparse polynomial equations [2019] is that for sparse systems, recognizing and computing a decomposition is algorithmic. Solving a decomposable sparse system reduces to solving two smaller sparse polynomial systems. In [Brysiewicz et al. 2021], we presented algorithms to detect and compute such decompositions, and a recursive algorithm exploiting decomposability for solving a decomposable sparse polynomial system using numerical homotopy continuation.

The *Macaulay*2 package `DecomposableSparseSystems` implements methods for decomposable sparse polynomial systems. These include methods to detect decomposability, to compute a decomposition, and a recursive procedure to compute numerical solutions to a given decomposable sparse system. Detection and computation of decompositions uses integer linear algebra, including computing a Smith normal form and the corresponding monomial changes of variables. Numerical homotopy continuation is used to compute solutions. When no further decompositions are possible, the algorithm solves multivariate systems using numerical software chosen by the user (default: `PHCpack` [Verschelde 1999]), and solves univariate polynomials using companion matrices.

Using the methods in `DecomposableSparseSystems` to solve a decomposable system allows for quicker solving and more accurate solution counts than calling other solvers. One reason is that after each decomposition, the child systems always involve either fewer variables, or polynomials of smaller degree. The cost of the methods in `DecomposableSparseSystems` is low as they rely only on linear algebra and numerical homotopy algorithms.

Research of Sottile and Yahl was supported by grant 636314 from the Simons Foundation.
MSC2020: 14M25, 65H10, 65H20.
Keywords: sparse, decomposable, polynomial, solving.
`DecomposableSparseSystems` version 1.0.1

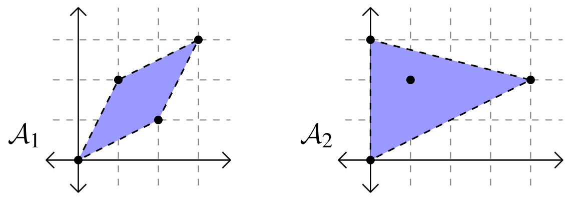

Figure 1. A pair of supports.

2. Decomposable sparse polynomial systems.

A *branched cover* is a dominant map $\pi : X \to Y$ of irreducible varieties X and Y of the same dimension. There is a number d (the *degree* of π) and an open dense subset V of Y such that $\pi^{-1}(v)$ consists of d points for $v \in V$. When $d > 1$, the branched cover is *nontrivial*.

Definition 1. A branched cover $\pi : X \to Y$ is *decomposable* if it is a composition of nontrivial branched covers. That is, if there is a dense open subset $U \subset Y$ and a variety Z such that $\pi^{-1}(U) \to U$ factors as

$$\pi^{-1}(U) \to Z \to U,$$

with each map a nontrivial branched cover.

In general it is not easy to determine if a branched cover is decomposable, or even to compute a decomposition for a decomposable branched cover. (See [Améndola et al. 2016, Section 5.4] and [Brysiewicz et al. 2021, Section 1.2] for examples and a discussion.)

An integer vector $\alpha \in \mathbb{Z}^n$ is the exponent of a (Laurent) monomial $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. A (complex) linear combination of monomials $\sum c_\alpha x^\alpha$ is a (Laurent) polynomial. Monomials are multiplicative maps $(\mathbb{C}^\times)^n \to \mathbb{C}^\times$ and polynomials are maps $(\mathbb{C}^\times)^n \to \mathbb{C}$. For a finite set $\mathcal{A} \subset \mathbb{Z}^n$ of exponents, the set of all polynomials whose monomials have exponents contained in $\mathcal{A}$ (have *support* $\mathcal{A}$) forms the vector space $\mathbb{C}^{\mathcal{A}}$. Given a list $\mathcal{A}_\bullet = (\mathcal{A}_1, \dots, \mathcal{A}_n)$ of finite subsets of $\mathbb{Z}^n$, write $\mathbb{C}^{\mathcal{A}_\bullet}$ for the vector space $\mathbb{C}^{\mathcal{A}_1} \oplus \cdots \oplus \mathbb{C}^{\mathcal{A}_n}$ of lists $F = (f_1, \dots, f_n)$ of polynomials with f_i having support $\mathcal{A}_i$. Such a list $F \in \mathbb{C}^{\mathcal{A}_\bullet}$ is a function $F : (\mathbb{C}^\times)^n \to \mathbb{C}^n$, and $F = 0$ is a system of sparse polynomials with support $\mathcal{A}_\bullet$ whose solutions are $F^{-1}(0)$.

Example 2. Let $\mathcal{A}_\bullet = (\mathcal{A}_1, \mathcal{A}_2)$ be the pair of supports in $\mathbb{Z}^2$ illustrated in Figure 1. The corresponding vector spaces of polynomials are

$$\mathbb{C}^{\mathcal{A}_1} = \{a_1 + a_2xy^2 + a_3x^2y + a_4x^3y^3 \mid a_i \in \mathbb{C}\},$$
$$\mathbb{C}^{\mathcal{A}_2} = \{b_1 + b_2y^3 + b_3xy^2 + b_4x^4y^2 \mid b_j \in \mathbb{C}\},$$

and $\mathbb{C}^{\mathcal{A}_\bullet}$ is the space of systems of the form

$$F = \begin{pmatrix} a_1 + a_2xy^2 + a_3x^2y + a_4x^3y^3 \\ b_1 + b_2y^3 + b_3xy^2 + b_4x^4y^2 \end{pmatrix}, \quad a_i, b_j \in \mathbb{C}.$$

In DecomposableSparseSystems, the family $\mathbb{C}^{\mathcal{A}_\bullet}$ is encoded by a list of matrices whose column vectors are the exponent vectors of each polynomial. Given a system $F \in \mathbb{C}^{\mathcal{A}_\bullet}$, these data can be extracted from a given system via the *Macaulay*2 function `exponents`.

The Bernstein–Kushnirenko theorem [Bernstein 1975] provides a sharp upper bound on the number of solutions to a system of sparse polynomials. Denote the convex hull of a set $\mathcal{A} \subseteq \mathbb{R}^n$ by $\operatorname{conv}(\mathcal{A})$. Given a list of supports $\mathcal{A}_\bullet = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$, let $\mathrm{MV}(\mathcal{A}_\bullet)$ be the mixed volume (see [Ewald 1996, Section IV.3]) of the list $(\operatorname{conv}(\mathcal{A}_1), \ldots, \operatorname{conv}(\mathcal{A}_n))$.

Theorem 3 (Bernstein–Kushnirenko). *Let $\mathcal{A}_\bullet$ be a list of n finite subsets of $\mathbb{Z}^n$. For $F \in \mathbb{C}^{\mathcal{A}_\bullet}$, the number of isolated solutions in $(\mathbb{C}^\times)^n$ to the system $F = 0$ is bounded above by $\mathrm{MV}(\mathcal{A}_\bullet)$ and this bound is achieved for F lying in a dense, open subset of $\mathbb{C}^{\mathcal{A}_\bullet}$.*

Define $X_{\mathcal{A}_\bullet} \subset (\mathbb{C}^\times)^n \times \mathbb{C}^{\mathcal{A}_\bullet}$ to be the set of pairs (x, F) such that $F(x) = 0$. For $F \in \mathbb{C}^{\mathcal{A}_\bullet}$, the fiber $\pi^{-1}(F)$ of the map $\pi : X_{\mathcal{A}_\bullet} \to \mathbb{C}^{\mathcal{A}_\bullet}$ consists of solutions to $F = 0$. By the Bernstein–Kushnirenko theorem, the map π has degree $\mathrm{MV}(\mathcal{A}_\bullet)$. When $\mathrm{MV}(\mathcal{A}_\bullet) \geq 1$, it is a branched cover. When the branched cover $\pi : X_{\mathcal{A}_\bullet} \to \mathbb{C}^{\mathcal{A}_\bullet}$ is decomposable, we say the sparse system $F \in \mathbb{C}^{\mathcal{A}_\bullet}$ is decomposable. Decomposability depends only on the support $\mathcal{A}_\bullet$ of a system.

There are two transparent ways for a sparse system to decompose.

Lacunary. A system $F \in \mathbb{C}^{\mathcal{A}_\bullet}$ is *lacunary* if there is a surjective monomial map $\Phi : (\mathbb{C}^\times)^n \to (\mathbb{C}^\times)^n$ such that $F = G \circ \Phi$ for some sparse polynomial system G. We require that Φ be nontrivial in that its kernel is not the identity subgroup. A lacunary system $F = G \circ \Phi = 0$ can be solved by computing solutions, $z_1, \ldots, z_d$, to the system $G = 0$ and then computing the fibers $\Phi^{-1}(z_1), \ldots, \Phi^{-1}(z_d)$. In appropriate coordinates, Φ is diagonal, and $\Phi^{-1}(z)$ is obtained by extracting roots of the components of z.

Example 4. Consider the following system with support from Example 2:

$$F(x, y) = \begin{pmatrix} 1 - 2xy^2 + 3x^2y - 4x^3y^3 \\ 2 + 3y^3 + 5xy^2 + 7x^4y^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

It is lacunary as it is the composition of the following maps:

$$G(s, t) = \begin{pmatrix} 1 - 2st^2 + 3st - 4s^2t^3 \\ 2 + 3st^3 + 5st^2 + 7s^2t^2 \end{pmatrix}, \quad \Phi(x, y) = (x^3, x^{-1}y).$$

This can be detected via the methods in DecomposableSparseSystems:

```
i1 : R = CC[x,y];

i2 : F = {1-2*x*y^2+3*x^2*y-4*x^3*y^3,2+3*y^3+5*x*y^2+7*x^4*y^2};

i3 : isLacunary F
o3 = true
```

The method `isLacunary` extracts the set of supports of the system and computes the Smith normal form of a matrix associated to these supports to determine whether the system is lacunary.

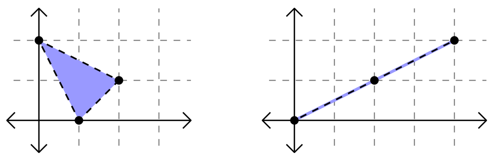

Figure 2. Triangular support.

Triangular. A system $F \in \mathbb{C}^{\mathcal{A}_\bullet}$ is *triangular* if there exists $k < n$ so that after a monomial change of variables, the system F has the form

$$F = (F_1(x_1, \dots, x_k), \dots, F_k(x_1, \dots, x_k), F_{k+1}(x_1, \dots, x_n), \dots, F_n(x_1, \dots, x_n)).$$

Solutions to triangular systems are computed by first computing the solutions $z_1, \dots, z_d$ of the square subsystem $(F_1, \dots, F_k) = 0$. A residual system is obtained by substituting z_1 into the original system for the first k variables, $F_2(z_1, x_{k+1}, \dots, x_n)$. Solutions to the original system are obtained by solving the residual system and then applying a homotopy algorithm as described in [Brysiewicz et al. 2021].

Example 5. Consider the system

$$F(x, y, z) = \begin{pmatrix} y^2 - 2x + 3x^2y \\ 2 + 3x^2y + 5x^4y^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Figure 2 shows the supports. This system is triangular as the second polynomial is quadratic in the monomial x^2y. The method `isTriangular` detects this subsystem.

```
i4 : F = {y^2-2*x+3*x^2*y,2+3*x^2*y+5*x^4*y^2};

i5 : isTriangular F
o5 = true
```

A consequence of Esterov's study of Galois groups of sparse polynomial systems [2019] and Pirola and Schlesinger's result that a branched cover is decomposable if and only if its Galois group is imprimitive [2005] is that a sparse polynomial system is decomposable if and only if it is either lacunary or triangular. In each case, the solutions to the original system are computed via solutions to simpler systems. The methods in `DecomposableSparseSystems` iteratively decompose these sparse polynomial systems to efficiently solve them.

3. Main method: solveDecomposableSystem.

The main method implemented in the package `DecomposableSparseSystems` is named `solveDecomposableSystem` and this implements Algorithm 9 in [Brysiewicz et al. 2021]. It takes as input a sparse polynomial system $F \in \mathbb{C}^{\mathcal{A}_\bullet}$ and outputs all solutions to $F = 0$ in the algebraic torus. It recursively checks whether or not the input sparse polynomial system is decomposable, computes the decomposition, and then calls itself on each portion of the decomposition. When the input is not decomposable it solves multivariate polynomial systems with the numerical solver given by the option `Software` (default: `PHCpack`) and it solves univariate polynomial systems using companion matrices. For complete details, see [Brysiewicz et al. 2021, Section 3.1].

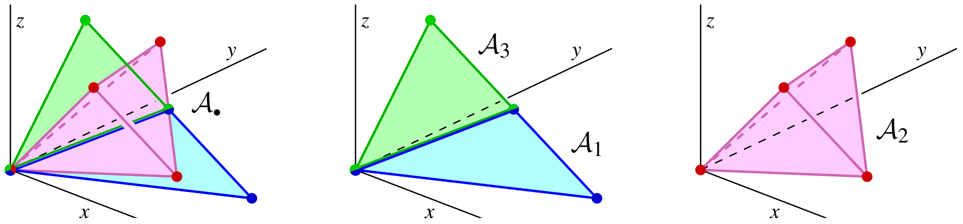

Figure 3. Support of F.

3.1. ***Using the main method.*** Consider the system

$$F = \begin{pmatrix} 2+xyz-x^2y \\ 4-y^2z+2xz^2-3x^2z \\ 1-yz^2-3xyz \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

This system is supported on the triple $\mathcal{A}_\bullet = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ shown in Figure 3.

The method `isDecomposable` determines that this system is decomposable. In particular, it is triangular with a subsystem indexed by the first and third polynomials. This can be observed in the figure as the span of the supports $\mathcal{A}_1$ and $\mathcal{A}_3$ are coplanar. It is also lacunary, as the exponent vectors lie in the sublattice of $\mathbb{Z}^3$ of index 3 generated by the columns of

$$\begin{pmatrix} 1 & 1 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

The solutions to $F = 0$ are found via the main method, `solveDecomposableSystem`.

```
i6 : R = CC[x,y,z];

i7 : F = {2+x*y*z-x^2*y,4-y^2*z+2*x*z^2-3*x^2*z,1-y*z^2-3*x*y*z};

    -- True if and only if the sparse system F is decomposable.

i8 : isDecomposable F
o8 : true

    -- A list of numerical solutions to F=0.

i9 : S = solveDecomposableSystem F;

    -- Evaluates F at the first numerical solution.

i10 : F/(f-> sub(f, matrix {S_0}))
o10 = {1.77636e-15, 4.44089e-16+1.4623e-16*ii, 4.66294e-15}
```

Our main method also accepts a two-argument input `(A,C)` where `A` is a list of matrices whose columns support a system of (Laurent) polynomial equations, and `C` is a list, whose i-th entry is the list of coefficients for the i-th polynomial equation. We demonstrate some of the other types of inputs here, and leave details to the documentation.

```
i11 : (A,C) = (F/exponents/matrix/transpose,
                 F/coefficients/last/entries/flatten);

i12 : S = solveDecomposableSystem (A,C);
```

```
  -- Expected timing for solving a specific system.

i13 : benchmark "solveDecomposableSystem(A,C)";
o13 = .0605920270512821

  -- Expected timing for solving a random system with support A.

i14 : benchmark "solveDecomposableSystem(A, )";
o14 = .0558867168108108
```

3.2. ***Options for the main method.*** Numerical in nature, the function `solveDecomposableSystem` features a variety of options for the user. The option `Software` (default: `PHCpack`) dictates which numerical solver is used to solve multivariate sparse systems which are not decomposable. The method `solveDecomposableSystem` removes solutions having any coordinate which is numerically zero up to `Tolerance` (default: 10^{-5}) throughout the computation. Having this tolerance is necessary, as our methods are for Laurent polynomials with solutions in the complex torus $(\mathbb{C}^\times)^n$, while the solvers we call may return solutions in $\mathbb{C}^n$ that are not in the torus.

Setting the option `Verify` (default: `0`) to have the value `1` significantly increases the probability that `solveDecomposableSystem` computes the correct number of solutions. It does this by checking that the algorithm specified by the `Software` option computes $\mathrm{MV}(\mathcal{A}_\bullet)$ solutions to any system F with support $\mathcal{A}_\bullet$, where $\mathrm{MV}(\mathcal{A}_\bullet)$ is probabilistically determined using `mixedVolume` in the package `Polyhedra` [Birkner 2009]. If the mixed volume according to `Polyhedra` and the number of solutions do not agree, then the missing solutions are searched for using techniques related to those in `MonodromySolver` [Duff et al. 2019]. Lastly, we allow the user to compute the solutions to F by first solving an internally generated random instance and then using that in a parameter homotopy [Li et al. 1989] to solve F by setting `Strategy` to `FromGeneric`. We conclude by using the options `Verify` and `Strategy` on an example with 6000 solutions.

```
i15 : A = <<< omitted, see example from Section 4 in [4] with
              i_1=(2,0,0,2,0)
              i_2=(4,4,2,2,2)
              j_1=(0,2,0,1,3)
              j_2=(0,0,1,0,2)
              >>;

  -- A has five supports, print the first one

i16 : print(length A,  A_0)
(5, | 0 2 4 4 6 |)
    | 0 0 0 4 4 |
    | 0 0 0 2 2 |
    | 0 2 4 2 4 |
    | 0 0 0 2 2 |

i17 : elapsedTime (F,S) = solveDecomposableSystem(A,,Verify=>1);
     -- 8.93938 seconds elapsed

i18 : elapsedTime S' = solveDecomposableSystem(F,Strategy=>FromGeneric);
     -- 29.0802 seconds elapsed

i19 : print(#S,#S')
o19 = (6000, 6000)
```

SUPPLEMENT. The online supplement contains version 1.0.1 of `DecomposableSparseSystems`.

REFERENCES.

[Améndola et al. 2016] C. Améndola, J. Lindberg, and J. I. Rodriguez, “Solving Parameterized Polynomial Systems with Decomposable Projections”, 2016. arXiv

[Bernstein 1975] D. N. Bernstein, “The number of roots of a system of equations”, *Funkcional. Anal. i Priložen.* **9**:3 (1975), 1–4. In Russian; translated in *Functional Analysis and Its Applications*, **9**:3, (1975), 183–185. MR

[Birkner 2009] R. Birkner, “Polyhedra: a package for computations with convex polyhedral objects”, *J. Softw. Algebra Geom.* **1** (2009), 11–15. MR Zbl

[Brysiewicz et al. 2021] T. Brysiewicz, J. I. Rodriguez, F. Sottile, and T. Yahl, “Solving decomposable sparse systems”, *Numerical Algorithms* (2021).

[Duff et al. 2019] T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, and J. Sommars, “Solving polynomial systems via homotopy continuation and monodromy”, *IMA J. Numer. Anal.* **39**:3 (2019), 1421–1446. MR Zbl

[Esterov 2019] A. Esterov, “Galois theory for general systems of polynomial equations”, *Compos. Math.* **155**:2 (2019), 229–245. MR Zbl

[Ewald 1996] G. Ewald, *Combinatorial convexity and algebraic geometry*, Graduate Texts in Mathematics **168**, Springer, 1996. MR Zbl

[Li et al. 1989] T. Y. Li, T. Sauer, and J. A. Yorke, “The cheater’s homotopy: an efficient procedure for solving systems of polynomial equations”, *SIAM J. Numer. Anal.* **26**:5 (1989), 1241–1251. MR Zbl

[Pirola and Schlesinger 2005] G. P. Pirola and E. Schlesinger, “Monodromy of projective curves”, *J. Algebraic Geom.* **14**:4 (2005), 623–642. MR Zbl

[Verschelde 1999] J. Verschelde, “Algorithm 795: PHCpack: A General-Purpose Solver for Polynomial Systems by Homotopy Continuation”, *ACM Trans. Math. Softw.* **25**:2 (1999), 251–276. Version containing reference manual available at http://homepages.math.uic.edu/ jan/PHCpack/phcpack.html. Zbl

RECEIVED: 8 Jun 2020 REVISED: 24 Jan 2021 ACCEPTED: 25 Mar 2021

TAYLOR BRYSIEWICZ:
taylorbrysiewicz@gmail.com
Max-Planck Institut fur Mathematik, Leipzig, Germany

JOSE ISRAEL RODRIGUEZ:
jose@math.wisc.edu
Department of Mathematics, University of Wisconsin, Madison, WI, United States

FRANK SOTTILE:
sottile@math.tamu.edu
Department of Mathematics, Texas A&M University, College Station, TX, United States

THOMAS YAHL:
thomasjyahl@math.tamu.edu
Department of Mathematics, Texas A&M University, College Station, TX, United States

msp

Journal of Software for Algebra and Geometry vol 11, no 1, 2021