A software framework for computing Newton polytopes of resultants and (reduced) discriminants*

Ioannis Z. Emiris** Vissarion Fisikopoulos** Christos Konaxis‡

March 10, 2013

Abstract

We present a new software for computing Newton polytopes of resultant and discriminant polynomials. We illustrate its use with a number of examples.

1 Introduction

We work with Laurent polynomials having fixed support sets $A_i \subset \mathbb{Z}^n$ in n unknowns $x = (x_1, x_2, \dots, x_n)$ over an algebraically closed field K i.e. $f_i(x) = \sum_{a \in A_i} c_{i,a} x^a$, $c_{i,a} \neq 0$. The *Newton polytope* of a polynomial f, denoted by $N(f)$, is the convex hull of its support set.

Sparse resultant. Sparse, or toric, resultants, or simply resultants, study systems such as:

$$f_0(x) = f_1(x) = \cdots = f_n(x) = 0. \tag{1}$$

Let $A_0, \dots, A_n \subset \mathbb{Z}^n$ be the respective supports, assuming they form an essential family [Stu94, Sec.1]. Polynomials $f_0(x) \dots, f_n(x)$ are defined on the A_i's with symbolic coefficients $c_{i,a}, i = 0, \dots, n,\ a \in A_i$. Given $A_0, \dots, A_n$ we define the *sparse resultant* of system (1) to be the unique (up to sign) irreducible integer polynomial $\mathcal{R}_{A_0,\dots,A_n}$ in the $c_{i,a}$, which vanishes iff (1) has a solution in $(K^*)^n$. The sparse resultant has $\sum_{i=0}^n |A_i|$ variables, however the intrinsic dimension of its Newton polytope, called *resultant polytope*, is [GKZ94]: $\dim(N(\mathcal{R})) = \sum_{i=0}^n |A_i| - 2n - 1$.

A-discriminant. Let A be a subset of $\mathbb{Z}^n$ s.t. it generates $\mathbb{Z}^n$ as an affine lattice and $f = \sum_{a \in A} c_a x^a$ be a generic polynomial w.r.t. A, i.e. with generic coefficients $c_a \neq 0$. The A-discriminant is the unique (up to sign) irreducible integer polynomial Δ_A in the unknowns c_a which vanishes iff f has a multiple root in $(K^*)^n$, namely

$$\exists x^* \in (K^*)^n \qquad \text{s.t.} \qquad f(x^*) = \frac{\partial f}{\partial x_1}(x^*) = \cdots = \frac{\partial f}{\partial x_n}(x^*) = 0. \tag{2}$$

Δ_A is homogeneous, and quasi-homogeneous relative to the weight defined by any vector in the rowspan of the $(n+1) \times m, m = |A| > n+1$, integer matrix (also called A by abuse of notation) whose first row consists of ones, and whose columns are $(1, a)$, $a \in A$. The intrinsic dimension of its Newton polytope, called *discriminant polytope*, is therefore $|A| - n - 1$.

*Poster presentation at the 12th International Conference on Effective Methods in Algebraic Geometry (MEGA) 2013, Frankfurt, Germany.

**Dept Informatics & Telecommunications, U. of Athens, Greece. `{emiris,vfisikop}@di.uoa.gr`

‡Archimedes Center for Modeling, Analysis & Computation, U. of Crete, Greece. `ckonaxis@acmac.uoc.gr`

The Cayley trick. Given pointsets $A_0, \dots, A_n \subset \mathbb{Z}^n$, we define the pointset

$$\mathcal{A} := \bigcup_{i=0}^{n} (A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}, \tag{3}$$

where $e_0, \dots, e_n$ form an affine basis of $\mathbb{R}^n$: $e_0 = (0, \dots, 0)$, $e_i = (0, \dots, 0, 1, 0, \dots, 0), i = 1, \dots, n$. The regular tight mixed subdivisions of Minkowski sum $A_0 + \cdots + A_n$ are in bijection with the regular triangulations of $\mathcal{A}$.

2 Algorithms for resultant and (reduced) discriminant polytopes

Our software `ResPol` computes (projections of) resultant polytopes, and (reduced) discriminant polytopes. It is written in C++, uses the CGAL library [1], principally the experimental CGAL package `triangulation`, and is publicly available at `http://respol.sourceforge.net`. It offers binary files for 32 and 64-bit Linux systems. To compile `ResPol`, e.g. on other architectures, one may consult the README file available with our distribution, once CGAL is installed.

Resultant polytope. An output-sensitive algorithm and an implementation for the resultant polytope are presented in [EFKP12]. Its complexity is polynomial in the number of polytope vertices and the number of full-dimensional cells in the triangulation of the polytope constructed by the algorithm. The method defines a *vertex oracle* which, given direction $c \in \mathbb{R}^{|\mathcal{A}|}$, computes vertex $v \in N(\mathcal{R})$ s.t. $c^T v$ is maximized. The oracle is implemented by computing a regular triangulation of the Cayley set $\mathcal{A}$. Then v equals the exponent ρ of the extreme monomial in [Stu94, Thm.2.1]. Using the oracle, the entire polytope can be reconstructed: Initialize with the convex hull of a sufficient number of vertices for the hull to be full-dimensional. Given a convex polytope, c is the outer normal to a facet F. The method either finds a new vertex and removes F (and possibly other facets), or a vertex on the hyperplane of F, which confirms that F is valid, so is never tested again. One reconstruction method is implemented in [Hug06] given a vertex oracle. Our implementation is optimized for vertex oracles that compute triangulations.

Example 1. Let $A_0 = \{0, 1, 3\}$ and $A_1 = \{0, 3, 4\}$ and consider two generic, relative to these supports, univariate $f_0 = a_0 + a_1 x + a_2 x^3$, $f_1 = b_0 + b_1 x^3 + b_2 x^4$. Their resultant is the Sylvester resultant of f_0, f_1. To compute its polytope, prepare text file `file.txt`:

`1`	dimension of the input supports A_0, A_1
`3 3 |`	cardinalities of the A_i's, "|" implies no projection
`[[0],[1],[3],[0],[3],[4]]`	joint list of all support points

The third line contains the points of A_0 followed by those of A_1. Running command `./res_enum_d < file.txt`, the set of the resultant polytope vertices: $(0, 3, 1, 1, 2, 0)$, $(0, 0, 4, 3, 0, 0)$, $(3, 0, 1, 0, 3, 0)$, $(4, 0, 0, 0, 0, 3)$, $(0, 4, 0, 1, 0, 2)$, $(3, 1, 0, 0, 1, 2)$, are written in the standard output. If "`3 3 | 0 3`" was used as second line then the result would be the orthogonal projection of the resultant polytope in the first and fourth coordinate (counting starts at 0).

Discriminant polytope. We extend `ResPol` to compute (reduced) discriminant polytopes following two approaches. The first focuses on reduced discriminants. By employing the Horn-Kapranov parameterization, the problem is reduced to implicitization. The Newton polytope of the implicit equation of the parameterization, or implicit polytope, is computed as the projection of a resultant polytope [EKKB13] and it contains (a translate of) the reduced discriminant polytope. This approach is discussed below.

The second approach defines vertex oracles for the discriminant polytope and uses Beneath-Beyond. There are several procedures to get a vertex oracle. In [Rin13] is given a procedure and

[1] CGAL: Computational Geometry Algorithms Library. http://www.cgal.org.

an implementation (`tropli`) for such an oracle using tropical geometry: `tropli`, given direction $c \in \mathbb{R}^{|A|}$, computes a vertex $v \in N(\Delta_A)$ s.t. $c^T v$ is minimized. `Respol` can use this oracle to reconstruct the discriminant polytope. One can also define a vertex oracle using the η-vectors from [GKZ94, ch.11], Such an oracle involves the computation of (normalized) volumes of lower dimensional simplices, and has not yet been implemented in `ResPol`.

Regarding the first approach, given A, let $B = (b_{ij}) \in \mathbb{Z}^{n\times(m-n-1)}$ be a matrix whose column vectors are a basis of the integer kernel of A. Then B is of full rank. We assume that its maximal minors have unit gcd (i.e. the rows generate $\mathbb{Z}^{m-n-1}$). Since the first row of A equals $(1,\dots,1)$, the columns of B add up to 0. Set $d = m-n-1$. Let $y_1,\dots,y_d$ be homogenous parameters and set $y_1 = 1$ so as to dehomogenize the parameterization. We denote by $l_i, i = 1,\dots,m$ the inner product of the i-th row of B and the parameter vector $(1, y_2,\dots,y_d)$: $l_i := \sum_{j=1}^{d} b_{ij} y_j$. The l_i correspond bijectively to the coefficients $c_a, a \in A$ of f and are thus the discriminant variables. The, so called, Horn-Kapranov parametrization [GKZ94, Kap91], is defined as:

$$x_j = \prod_{i=1}^{m} l_i^{b_{ij}},\ j = 1,2,\dots,d. \tag{4}$$

The implicit equation of (the closure of) its image is a polynomial Δ_B in $x := (x_1,\dots,x_d)$, called the *reduced discriminant*, which is the dehomogenized version of Δ_A; it is obtained from Δ_A by specializing some $n+1$ of its variables so as to remove the $n+1$ quasi-homogeneities. It follows that $N(\Delta_B)$ is the projection of $N(\Delta_A)$ in a space of dimension equal to its intrinsic dimension and retains the combinatorial structure of $N(\Delta_A)$.

Example 2. Let $A = \{0,1,2,3,4\}$ and $f = c_0 + c_1t^1 + c_2t^2 + c_3t^3 + c_4t^4$ be a generic quartic.

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} 3 & 2 & 1 \\ -4 & -3 & -2 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Here $m = 5, n = 1, d = 3$ and $l_1 = 3 + 2y_2 + y_3, l_2 = -4 - 3y_2 - 2y_3, l_3 = y_3, l_4 = y_2, l_5 = 1$, and the Horn-Kapranov parameterization is:

$$x_1 = \frac{(3+2y_2+y_3)^3}{(-4-3y_2-2y_3)^4}, \quad x_2 = \frac{(3+2y_2+y_3)^2 y_2}{(-4-3y_2-2y_3)^3}, \quad x_3 = \frac{(3+2y_2+y_3)y_3}{(-4-3y_2-2y_3)^2}. \tag{5}$$

We prefer to have rational parameterizations with a single monomial in the denominator because this facilitates the computation of the implicit polytope. We introduce a new parameter y_4 expressing the common denominator in (5) and obtain the parameterization

$$x_1 = \frac{(3+2y_2+y_3)^3}{y_4^4},\ x_2 = \frac{(3+2y_2+y_3)^2 y_2}{y_4^3},\ x_3 = \frac{(3+2y_2+y_3)y_3}{y_4^2},\ y_4 = -4-3y_2-2y_3,$$

from which we define the polynomials

$$\begin{aligned} F_0 &: = x_1y_4^4 - (3+2y_2+y_3)^3, \quad F_1 := x_2y_4^3 - (3+2y_2+y_3)^2y_2, \\ F_2 &: = x_3y_4^2 - (3+2y_2+y_3)y_3, \quad F_3 := y_4 + 4 + 3y_2 + 2y_3, \end{aligned}$$

whose supports are given as input to `ResPol`. The above procedure is demonstrated in the `Maple` file `horn_example2.mw` available with our distribution. Then, we prepare the input `file.txt`:

```
3
11 7 4 4  |0 11 18
[[0, 0, 4], [0, 0, 0], [1, 0, 0], [0, 1, 0], [2, 0, 0], [1, 1, 0], [0, 2, 0],
 [3, 0, 0], [2, 1, 0], [1, 2, 0], [0, 3, 0], [0, 0, 3], [1, 0, 0], [2, 0, 0],
 [1, 1, 0], [3, 0, 0], [2, 1, 0], [1, 2, 0], [0, 0, 2], [0, 1, 0], [1, 1, 0],
 [0, 2, 0], [0, 0, 1], [0, 0, 0], [1, 0, 0], [0, 1, 0]]
```

The second line after '|' instructs `ResPol` to project to the space defined by x_1, x_2, x_3. Executing `./res_enum_d < file.txt`, we obtain the vertices $(0,0,12)$, $(0,8,0)$, $(6,0,0)$, $(0,0,0)$ in the standard output. They define a polytope containing a translate of $N(\Delta_B)$.

To compute the discriminant polytope using `tropli` we prepare a textfile `file.txt`:

```
1
5 0 |
[[0], [1], [2], [3], [4]]
```

where the zero after the cardinality 5 of the support in the second line is needed because `ResPol` expects the number of supports to be one more than the dimension. Executing the command `./res_enum_d -d < file.txt`, we obtain the vertices of $N(\Delta_A)$: $(1,0,4,0,1)$, $(0,3,0,3,0)$, $(0,4,0,0,2)$, $(0,2,3,0,1)$, $(0,2,2,2,0)$, $(2,0,0,4,0)$, $(3,0,0,0,3)$, $(1,0,3,2,0)$ in the standard output. The corresponding vertices of $N(\Delta_B)$ may be computed as follows: By renaming the l_i's as c_i's we have from (5) that $x_1 = c_0^3 c_1^{-4} c_4$, $x_2 = c_0^2 c_1^{-3} c_3$, $x_3 = c_0 c_1^{-2} c_2$, which gives the correspondence: $(\kappa, \lambda, \mu) \longmapsto (3\kappa + 2\lambda + \mu, -4\kappa - 3\lambda - 2\mu, \mu, \lambda, \kappa)$, between the vertices of Δ_B and Δ_A. Moreover, this yields the correspondence: $(a_1, a_2, a_3, a_4, a_5) \longmapsto (a_5, a_4, a_3)$ between the vertices of Δ_A and Δ_B. Hence, from the set of vertices of $N(\Delta_A)$ above, we obtain the vertices of $N(\Delta_B)$: $(0,2,3)$, $(0,2,2)$, $(1,0,3)$, $(1,0,4)$, $(0,3,0)$, $(0,4,0)$, $(3,0,0)$, $(2,0,0)$, which are all contained in the polytope defined by the set of vertices predicted by `ResPol`.

Acknowledgment. We thank L. Peñaranda for his contribution in `ResPol`. IE's research has been co-financed by the EU (European Social Fund) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework, Research Funding Program "ARISTEIA", project "ESPRESSO: Exploiting Structure in Polynomial Equation and System Solving in Physical Modeling". VF is partially supported by project "Computational Geometric Learning", which acknowledges the financial support of the Future and Emerging Technologies program within the 7th Framework Program for research of the European Commission, under FET-Open grant number 255827. CK is supported by the FP7-REGPOT-2009-1 project "Archimedes Center for Modeling, Analysis and Computation" (under grant agreement nº 245749).

References

[EFKP12] I.Z. Emiris, V. Fisikopoulos, C. Konaxis, and L. Peñaranda. An output-sensitive algorithm for computing projections of resultant polytopes. In *Proc. ACM Symp. Comp. Geometry*, pages 179–188, 2012.

[EKKB13] I.Z. Emiris, T. Kalinka, C. Konaxis, and T. Luu Ba. Sparse implicitization by interpolation: Characterizing non-exactness and an application to computing discriminants. *Computer-Aided Design*, 45(2):252 – 261, 2013.

[GKZ94] I.M. Gelfand, M.M. Kapranov, and A.V. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants.* Birkhäuser, Boston, 1994.

[Hug06] P. Huggins. iB4e: A software framework for parametrizing specialized LP problems. In A. Iglesias and N. Takayama, editors, *Int. Conf. Math. Software'06*, volume 4151 of *LNCS*, pages 245–247. Springer, 2006.

[Kap91] M. Kapranov. A characterization of A-discriminant hypersurfaces in terms of the Gauss map. *Math. Ann*, 290:277–285, 1991.

[Rin13] F. Rincón. Computing tropical linear spaces. *J. Symbolic Computation*, 51:86–98, 2013. Spec. Issue on Effective Methods in Algebraic Geometry.

[Stu94] B. Sturmfels. On the Newton polytope of the resultant. *J. Algebr. Combin.*, 3:207–236, 1994.