Annales Mathematicae Silesianae **28** (2014), 7–34
Prace Naukowe Uniwersytetu Śląskiego nr 3240, Katowice

SOLUTIONS TO SYSTEMS OF BINOMIAL EQUATIONS

Tianran Chen, Tien-Yien Li

Annual Lecture dedicated to the memory of Professor Andrzej Lasota

1. Introduction

The problem of solving systems of polynomial equations has been, and will continue to be, one of the most important subjects in both pure and applied mathematics. The need to solve systems of polynomial equations occurs frequently in various fields of science and engineering, such as formulae construction, geometric intersection, inverse kinematics, robotics, computer vision, and the computation of equilibrium states of chemical reaction equations.

In this article, we will focus on solving **binomial polynomial systems**: Systems of polynomial equations in which each equation contains exactly two terms. Binomial polynomial systems (or simply **binomial systems**) represents an important class of polynomial systems. While the solution structure of binomial systems is interesting in its own right, it actually plays a crucially important role in solving general polynomial systems numerically. (We will elaborate this in details in Section 5.1.) Moreover, binomial systems have direct connections to *lattice ideals* [34] and *toric varieties* [9, 15], which admit vast applications.

Received: 22.03.2014. Revised: 6.08.2014.
(2010) Mathematics Subject Classification: 14Q99,14M25.
Key words and phrases: binomial system, BKK bound, master space, polyhedral homotopy.
Research supported in part by NSF under Grant DMS 11-15587 and by NSFC under Grant 11171052.

We will take a computational point of view in this article to outline the different ingredients needed to solve binomial polynomial systems exactly and numerically. To facilitate the discussion, Section 2 introduces notations and conventions to be used. Section 3 discusses the structure of solution sets of binomial systems and their important properties. Section 4 investigates different aspects in solving binomial systems from a computational point of view. In particular, we will discuss the problems in computation when the coefficients of the binomial systems are not provided exactly. Among various kinds of applications for binomial systems, we present, in Section 5, two important and interesting applications. First one illustrates the heavy reliance on solving binomial systems when solving general polynomial systems by the efficient *polyhedral* homotopies. Second example highlights the needs and difficulties in solving binomial systems that arose in *supersymmetric gauge theory* in theoretical physics. Finally, for solving larger binomial systems from applied sciences, parallel computation becomes inevitable. Section 6 showcases some recent developments in the parallel implementation of solvers for binomial systems.

2. Notations and concepts

Throughout this article, vectors are denoted by boldface letters while matrices are denoted by capital letters, and $M_{n\times m}(\mathbb{Z})$ denotes the set of all $n\times m$ matrices of integer entries. A square integer matrix is said to be **unimodular** if its determinant is ± 1; subsequently, a unimodular matrix in $M_{n\times n}(\mathbb{Z})$ must have an inverse that is also inside $M_{n\times n}(\mathbb{Z})$ by *Cramer's rule*.

Though the focus of this article is the binomial systems, it is convenient, and almost necessary, to extend the scope of our discussion to more general *Laurent binomial systems* given as follows. For $\boldsymbol{x} = (x_1, \ldots, x_n)$ and an integer column vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{Z}^n$, the "vector exponent" notation is commonly used for the **Laurent monomial**:

$$\boldsymbol{x}^{\boldsymbol{\alpha}} = (x_1, \ldots, x_n)\begin{pmatrix}\alpha_1\\ \vdots \\ \alpha_n\end{pmatrix} = x_1^{\alpha_1}\cdots x_n^{\alpha_n}.$$

Similarly, for an integer matrix $A \in M_{n\times m}(\mathbb{Z})$ with columns $\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(m)} \in \mathbb{Z}^n$, the "matrix exponent" notation is used for a system of Laurent monomials:

$$\boldsymbol{x}^A = \boldsymbol{x}^{\begin{pmatrix}\boldsymbol{\alpha}^{(1)} & \cdots & \boldsymbol{\alpha}^{(m)}\end{pmatrix}} := (\boldsymbol{x}^{\boldsymbol{\alpha}^{(1)}}, \ldots, \boldsymbol{x}^{\boldsymbol{\alpha}^{(m)}}). \tag{1}$$

Since the exponents here may be negative, it is natural to require each x_i to be nonzero, i.e., $x_i \in \mathbb{C}^* = \mathbb{C} \setminus \{0\}$ for each $i = 1, \ldots, n$. So the variables take values in $(\mathbb{C}^*)^n$ which has a natural group structure given by the componentwise multiplication

$$(x_1, \ldots, x_n) \bullet (y_1, \ldots, y_n) := (x_1 \cdot y_1, \ldots, x_n \cdot y_n).$$

Those notations are particularly convenient, since the familiar identities such as

$$(\boldsymbol{x} \bullet \boldsymbol{y})^A = \boldsymbol{x}^A \bullet \boldsymbol{y}^A \quad \text{and} \quad (\boldsymbol{x}^A)^B = \boldsymbol{x}^{AB} \tag{2}$$

still hold. A **Laurent polynomial** is a linear combination of Laurent monomials, i.e., an expression of the form $\sum_{k=1}^{m} c_k \boldsymbol{x}^{\boldsymbol{\alpha}^{(k)}}$ where each $c_k \in \mathbb{C}$ and $\boldsymbol{\alpha}^{(k)} \in \mathbb{Z}^n$. The set of all such Laurent polynomials naturally forms a ring under the usual polynomial addition and multiplication. A **Laurent binomial** is a Laurent polynomial having exactly two terms with nonzero coefficients[1], i.e., it is an expression of the form $c_1 \boldsymbol{x}^{\boldsymbol{\alpha}} + c_2 \boldsymbol{x}^{\boldsymbol{\beta}}$ for some $c_1, c_2 \in \mathbb{C}^*$ and $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{Z}^n$. Even though much of the established theory is widely applicable to Laurent binomials over arbitrary algebraically closed fields, in this article, however, our attention is restricted to those with complex coefficients. The focus here is solving systems of Laurent binomials equations, or simply **Laurent binomial systems**, over $(\mathbb{C}^*)^n$. More formally, given exponent vectors $\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(m)}, \boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(m)} \in \mathbb{Z}^n$ and the coefficients $c_{i,j} \in \mathbb{C}^*$, the goal is to describe the set of all $\boldsymbol{x} \in (\mathbb{C}^*)^n$ that satisfies the system of equations

$$\begin{cases} c_{1,1}\boldsymbol{x}^{\boldsymbol{\alpha}^{(1)}} + c_{1,2}\boldsymbol{x}^{\boldsymbol{\beta}^{(1)}} & = 0 \\ & \vdots \\ c_{m,1}\boldsymbol{x}^{\boldsymbol{\alpha}^{(m)}} + c_{m,2}\boldsymbol{x}^{\boldsymbol{\beta}^{(m)}} & = 0. \end{cases}$$

Concerning the solution set in $(\mathbb{C}^*)^n$, this system is clearly equivalent to

$$(\boldsymbol{x}^{\boldsymbol{\alpha}^{(1)}-\boldsymbol{\beta}^{(1)}}, \ldots, \boldsymbol{x}^{\boldsymbol{\alpha}^{(m)}-\boldsymbol{\beta}^{(m)}}) = (-c_{1,2}/c_{1,1}, \ldots, -c_{m,2}/c_{m,1}).$$

With more compact "matrix exponent" notations in (1), this system can simply be written as

$$\boldsymbol{x}^A = \boldsymbol{b} \tag{3}$$

[1]An alternative definition for a Laurent binomial that is often used is "a Laurent polynomial with *at most* two terms with nonzero coefficients". This definition would include Laurent monomials as a special case. However, since monomial equations are trivial to solve from a purely computational point of view, here we choose to use the more strict definition.

where the integer matrix $A \in M_{n\times m}(\mathbb{Z})$, having columns $\boldsymbol{\alpha}^{(1)}-\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\alpha}^{(m)}-\boldsymbol{\beta}^{(m)}$, represents the exponents appeared in the Laurent monomials and the vector $\boldsymbol{b} = (-c_{1,2}/c_{1,1}, \ldots, -c_{m,2}/c_{m,1}) \in (\mathbb{C}^*)^m$ collects all the coefficients. In this article, we often take (3) to be the standard form of a Laurent binomial system, with which one must bear in mind that for $A \in M_{n\times m}(\mathbb{Z})$, n represents the number of variables and m represents the number of binomial equations in the system.

3. Solution sets of Laurent binomial systems

In this section, we outline the structural theory of the solution set of a Laurent binomial system as well as the means by which one could study the important properties of the solution set with regard to its dimension, number of components, smoothness, degree, and global parametrizations. The details of a more general theory can be found in [9], [12], [15], and [34].

An important tool in understanding the structure of the solution set of a Laurent binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ (in its standard form (3)) is the *Smith Normal Form* of the exponent matrix A: It is known that for integer matrix A, there are unimodular square matrices $P \in M_{n\times n}(\mathbb{Z})$ and $Q \in M_{m\times m}(\mathbb{Z})$ such that

$$P\,A\,Q = \begin{pmatrix} \overbrace{\begin{matrix} d_1 & & & \\ & \ddots & & \\ & & d_r & \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{matrix}}^{r \qquad m-r} \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} r \\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} n-r \end{matrix} \tag{4}$$

with nonzero integers $d_1 \,|\, d_2 \,|\, \cdots \,|\, d_r$ for $r = \operatorname{rank} A$, unique up to the signs. Here, $a \,|\, b$ means a divides b as usual. This decomposition of the matrix A provides important geometric information about the solution set of $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ summarized in the following proposition:

Proposition 1. *If the solution set of $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ is not empty, then it consists of a finite number of connected components. Furthermore,*

1. *the number of solution components is exactly $\left|\prod_{j=1}^{r} d_j\right|$;*
2. *each solution component has codimension equals* $\operatorname{rank} A = r$*; and*
3. *each solution component is smooth.*

We shall not include the proof of this proposition here, as it is subsumed in the more concrete description of the solution set in Proposition 2. (Also, this proposition can be considered as a corollary of [12, Theorem 2.1] when the theorem is applied to Laurent binomial systems defined over $\mathbb{C}$.)

For P and Q in the Smith Normal Form of A in (4), let $P_r \in M_{r\times n}(\mathbb{Z})$ and $P_0 \in M_{(n-r)\times n}(\mathbb{Z})$ be the top r rows and the remaining $n-r$ rows of P respectively. Similarly, let $Q_r \in M_{m\times r}(\mathbb{Z})$ and $Q_0 \in M_{m\times(m-r)}(\mathbb{Z})$ be the left r columns and the remaining $m-r$ columns of Q respectively. With these notations, the Smith Normal Form of A can be written as

$$\begin{pmatrix} P_r \\ P_0 \end{pmatrix} A \begin{pmatrix} Q_r & Q_0 \end{pmatrix} = \begin{pmatrix} D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{5}$$

with $D = \operatorname{diag}(d_1, \ldots, d_r) \in M_{r\times r}(\mathbb{Z})$ and $\mathbf{0}$'s representing zero block matrices of appropriate sizes.

The square matrix Q in (4) induces a map $\boldsymbol{y} \mapsto \boldsymbol{y}^Q$. This map is an automorphism of $(\mathbb{C}^*)^m$, since Q is unimodular (and hence has an inverse in $M_{m\times m}(\mathbb{Z})$). Thus, considering the solution set in $(\mathbb{C}^*)^n$, the original Laurent binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ is equivalent to

$$(\boldsymbol{x}^A)^Q = \boldsymbol{x}^{AQ} = \boldsymbol{b}^Q.$$

Similarly, since P in (4) is a unimodular $n \times n$ matrix, the map $\boldsymbol{z} \mapsto \boldsymbol{z}^P$ is also an automorphism of $(\mathbb{C}^*)^n$. So the solution sets remain equivalent after the change of variables $\boldsymbol{x} = \boldsymbol{z}^P$, and the Laurent binomial system becomes

$$(\boldsymbol{z}^P)^{AQ} = \boldsymbol{z}^{PAQ} = \boldsymbol{z}^{\left(\begin{smallmatrix} D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix}\right)} = (\boldsymbol{z}^{\left(\begin{smallmatrix} D \\ \mathbf{0} \end{smallmatrix}\right)}, \boldsymbol{z}^{\left(\begin{smallmatrix} \mathbf{0} \\ \mathbf{0} \end{smallmatrix}\right)}) = \boldsymbol{b}^Q = (\boldsymbol{b}^{Q_r}, \boldsymbol{b}^{Q_0}).$$

Since $D = \operatorname{diag}(d_1, \ldots, d_r) \in M_{r\times r}(\mathbb{Z})$, the original system $\boldsymbol{x}^A = \boldsymbol{b}$ can now be decomposed into a combined system

$$(z_1, \ldots, z_r) \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_r \end{pmatrix} = \boldsymbol{b}^{Q_r} \tag{6}$$

$$\mathbf{1} = \boldsymbol{b}^{Q_0} \tag{7}$$

$$z_{r+1}, \ldots, z_n : \quad \text{free} \tag{8}$$

where (7) appears when $r < m$ with $\mathbf{1} = (1, \ldots, 1) \in (\mathbb{C}^*)^{m-r}$, and (8) appears when $r < n$. The word "free" in (8) means the system imposes no constraints on the $n-r$ variables $z_{r+1}, \ldots, z_n$.

Focusing on this decomposed system, it is clear that if $r < m$, then the system is inconsistent unless $\mathbf{1} = \boldsymbol{b}^{Q_0}$. If the system is consistent (namely, (7) holds), then, when $r = n$, the solutions to (6) are exactly

$$(9) \qquad \begin{cases} z_1 = e^{2k_1\pi/d_1}\zeta_1 & \text{for} \quad k_1 = 0, \dots, d_1 - 1 \\ z_2 = e^{2k_2\pi/d_2}\zeta_2 & \text{for} \quad k_2 = 0, \dots, d_2 - 1 \\ \vdots & \vdots \\ z_r = e^{2k_r\pi/d_r}\zeta_r & \text{for} \quad k_r = 0, \dots, d_r - 1 \end{cases}$$

where each ζ_j is a fixed choice of the d_j-th root of j-th coordinate of $\boldsymbol{b}^Q$. Clearly, all of them are isolated and the total number of these solutions is $|\prod_{j=1}^r d_j| = |\det D|$. If $r < n$, then the solution set of the decomposed system in $(\mathbb{C}^*)^n$ breaks into "components" of the form $\{(e^{2k_1\pi/d_1}\zeta_1, \dots, e^{2k_r\pi/d_r}\zeta_r, z_{r+1}, \dots, z_n) : (z_{r+1}, \dots, z_n) \in (\mathbb{C}^*)^{n-r}\}$, and they are in one-to-one correspondence with solutions in (9). Since each component is parametrized by the $n-r$ free variables $z_{r+1}, \dots, z_n$, it is smooth and of dimension $n-r$. Furthermore, they are disjoint, because these components have distinct $z_1, \dots, z_r$ coordinates. Note that the term "components" used here simply means connected components. A deeper meaning of the term is given in Remark 1 below.

To translate the above description of the $(\mathbb{C}^*)^n$-solution set of the decomposed system (in $\boldsymbol{z}$) into a description of the solution set of the original system $\boldsymbol{x}^A = \boldsymbol{b}$ in (3), one may simply apply the change of variables $\boldsymbol{x} = \boldsymbol{z}^P$. Since this map is given by monomials with an inverse $\boldsymbol{z} = \boldsymbol{x}^{P^{-1}}$ that is also given by monomials (hence a *bi-regular* map [21]), it therefore preserves the basic properties of the solution set, such as, the number of solution components as well as their dimension and smoothness.

In summary, the above elaborations assert the following proposition.

PROPOSITION 2. *For the solution set of $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$, let P, Q, Q_0 and D be those matrices appeared in the decompositions of A in (4) and (5):*

If $\mathbf{1} \neq \boldsymbol{b}^{Q_0}$ then the binomial system is inconsistent, and hence its solution set in $(\mathbb{C}^*)^n$ is empty.

If $\mathbf{1} = \boldsymbol{b}^{Q_0}$ then the solution set of $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ consists of $|\prod_{j=1}^r d_j| = |\det D|$ components $V_{k_1,\dots,k_r}$ for $k_1 \in \{0, \dots, d_1 - 1\}, \dots, k_r \in \{0, \dots, d_r - 1\}$. Each component $V_{k_1,\dots,k_r}$ is smooth of dimension $n-r$, and it is parametrized by the smooth global parametrization $\phi_{k_1,\dots,k_r} \colon (\mathbb{C}^*)^{(n-r)} \to V_{k_1,\dots,k_r}$ given by

$$(10) \qquad \phi_{k_1,\dots,k_r}(t_1, \dots, t_{n-r}) = (e^{2k_1\pi/d_1}\zeta_1, \dots, e^{2k_r\pi/d_r}\zeta_r, t_1, \dots, t_{n-r})^P$$

where each ζ_j is a fixed choice of the d_j-th root of the j-th coordinate of $\boldsymbol{b}^Q$.

Note that this proposition includes both cases: the solution set of $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ consists of isolated points or components of positive dimension. When $r = n$, the components are zero-dimensional, i.e., they are isolated, and the "parametrizations" $\phi_{k_1,\ldots,k_r}$ are simply constant maps, each of which describes a single isolated point.

REMARK 1. With the notations introduced in (2), the global parametrization of each component, given in Proposition 2, $\phi_{k_1,\ldots,k_r} : (\mathbb{C}^*)^{n-r} \to V_{k_1,\ldots,k_r}$ can be written as

$$\begin{aligned}\phi_{k_1,\ldots,k_r}(\boldsymbol{t}) &= (e^{2k_1\pi/d_1}\zeta_1, \ldots, e^{2k_r\pi/d_r}\zeta_r, \boldsymbol{t})^P \\ &= \boldsymbol{t}^{P_0} \bullet (e^{2k_1\pi/d_1}\zeta_1, \ldots, e^{2k_r\pi/d_r}\zeta_r)^{P_r}.\end{aligned}$$

This illuminates another geometric feature about the solution set of a Laurent binomial system: components of the solution set with positive dimension can be expressed as conjugate orbits of a $(\mathbb{C}^*)^{n-r}$-action induced by the matrix P_0.

As indicated in Proposition 2, for a consistent Laurent binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ where $A \in M_{n\times m}(\mathbb{Z})$ with $r = \operatorname{rank}(A) < n$, each component of the solution set in $(\mathbb{C}^*)^n$ will be of dimension $n - r > 0$. In this situation, for both theoretical interests and demands in application, one often wishes to identify another important invariant: the *degree* of the components. Degree is a classic concept developed for plane algebraic curves, e.g., the quadratic equation $y - x^2 = 0$ defines a quadratic curve: the parabola. This notion generalizes directly to the case of hyper-surfaces of the complex projective space where the degree is simply the degree of a minimal defining homogeneous polynomial. The more general notion of degree is usually defined in a purely algebraic setting: The degree of a projective algebraic set, i.e., the solution set of a system of homogeneous polynomial equations, is defined to be the leading coefficient of the *Hilbert Polynomial* of its homogeneous coordinate ring multiplied by the factorial of its dimension [21]. For an affine algebraic set, the degree can be defined as the degree of its projective closure: the smallest projective algebraic set that contains it. Fortunately, this abstract definition has a more concrete geometric interpretation: the degree of an algebraic set is also the number of its intersection points with a generic linear space of complementary dimension (a very special case of [21, Theorem 7.7]). The word "generic" here simply means that those linear spaces belong to a fixed set that is open and dense in the space of all linear spaces. It can be shown that the number of intersection points here is a fixed number with regard to different generic linear spaces. This interpretation is indeed the basis of a successful algorithm for computing the degree of a solution component as a by-product of computing "witness sets" [37]. Moreover, with this interpretation

the degree of its solution set of binomial systems can be computed much easily through methods in combinatorial geometry.

PROPOSITION 3. *Let $A \in M_{n\times m}(\mathbb{Z})$, $r = \operatorname{rank} A$, and $P_0 = (\boldsymbol{p}_0^{(1)}, \ldots, \boldsymbol{p}_0^{(n)})$ be the matrix appeared in* (5)*. If $r < n$, then for each solution component of the consistent Laurent binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$, as given in Proposition 2, its degree is*

$$(n-r)! \cdot \mathrm{Vol}_{n-r}(\mathrm{conv}\{\boldsymbol{p}_0^{(1)}, \ldots, \boldsymbol{p}_0^{(n)}, \mathbf{0}\})$$

where $\mathbf{0} = (0, \ldots, 0)^\top \in \mathbb{R}^{n-r}$ and columns $\boldsymbol{p}_0^{(1)}, \ldots, \boldsymbol{p}_0^{(n)}$ of the matrix P_0 are considered as points in $\mathbb{R}^{n-r}$. The notation conv *denotes the operation of taking convex hull, and* Vol_{n-r} *is the volume of a convex body in $\mathbb{R}^{n-r}$ with the $1 \times \cdots \times 1$ hypercube having unit volume.*

While a proof of this proposition, via the route of Hilbert polynomials, can be found in [9] and [15], an alternative proof which suggests our strategy in computing the degree of the solution component is given below. First, we need the following theorem:

THEOREM 1 (Kushnirenko [26]). *Consider the system of k equations*

$$\begin{cases} c_{1,1}\boldsymbol{x}^{\boldsymbol{a}^{(1)}} + c_{1,2}\boldsymbol{x}^{\boldsymbol{a}^{(2)}} + \cdots + c_{1,k}\boldsymbol{x}^{\boldsymbol{a}^{(\ell)}} &= 0 \\ c_{2,1}\boldsymbol{x}^{\boldsymbol{a}^{(1)}} + c_{2,2}\boldsymbol{x}^{\boldsymbol{a}^{(2)}} + \cdots + c_{2,k}\boldsymbol{x}^{\boldsymbol{a}^{(\ell)}} &= 0 \\ & \vdots \\ c_{k,1}\boldsymbol{x}^{\boldsymbol{a}^{(1)}} + c_{k,2}\boldsymbol{x}^{\boldsymbol{a}^{(2)}} + \cdots + c_{k,k}\boldsymbol{x}^{\boldsymbol{a}^{(\ell)}} &= 0 \end{cases}$$

in k variables $\boldsymbol{x} = (x_1, \ldots, x_k)$ in which every equation has the same set of monomials determined by exponent vectors $\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(\ell)} \in \mathbb{Z}^k$. With "generic" coefficients $c_{i,j} \in \mathbb{C}^*$, the solutions of this system in $(\mathbb{C}^*)^k$ are all isolated and nonsingular. The total number of these isolated solutions is

$$k! \cdot \mathrm{Vol}_k(\mathrm{conv}\{\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(\ell)}\}).$$

Here, the term "generic" can be understood as "almost all". Its precise meaning, which is much stronger, can be found in [26]. This theorem is a special case of the more general Bernshtein's theorem [4] which will be provided in Section 5.1. The number of isolated solutions given by both the Kushnirenko's theorem and the Bernshtein's theorem is now commonly known as the *BKK bound* of the system.

Proof of Proposition 3. In the proof, we shall take the degree of an algebraic set in $(\mathbb{C}^*)^n$ to be simply the fixed number of its intersection points with a "generic" affine space of complementary dimension.

Let V be a solution component of $\boldsymbol{x}^A = \boldsymbol{b}$ with dimension $d = n - r > 0$. The focus here is its intersection with a generic affine space of dimension $r = n - d$ represented by the solutions of a system of d linear polynomial equations

$$\begin{cases} c_{1,1}x_1 + c_{1,2}x_2 + \cdots + c_{1,n}x_n + c_{1,0} &= 0 \\ c_{2,1}x_1 + c_{2,2}x_2 + \cdots + c_{2,n}x_n + c_{2,0} &= 0 \\ \vdots & \vdots \\ c_{d,1}x_1 + c_{d,2}x_2 + \cdots + c_{d,n}x_n + c_{d,0} &= 0 \end{cases}$$

having generic complex coefficients $\{c_{j,k}\}$. Considering its geometric interpretation, the degree of V is simply the number of $\boldsymbol{x} \in V$ that satisfies the above equations. Moreover, by Proposition 2 and Remark 1, the component V has a global parametrization given by $\boldsymbol{x} = \phi(\boldsymbol{t}) = \boldsymbol{y} \bullet \boldsymbol{t}^{P_0}$ for some fixed $\boldsymbol{y} = (y_1, \ldots, y_n) \in (\mathbb{C}^*)^n$ as $\boldsymbol{t}$ ranges through $(\mathbb{C}^*)^d$. Since $P_0 = (\boldsymbol{p}_0^{(1)}, \ldots, \boldsymbol{p}_0^{(n)})$, $x_k = y_k \cdot \boldsymbol{t}^{\boldsymbol{p}_0^{(k)}}$ for $k = 1, \ldots, n$. With these, the degree of V is the number of isolated solutions of the system of Laurent polynomial equations

$$\begin{cases} c_{1,1}y_1\boldsymbol{t}^{\boldsymbol{p}_0^{(1)}} + c_{1,2}y_2\boldsymbol{t}^{\boldsymbol{p}_0^{(2)}} + \cdots + c_{1,n}y_n\boldsymbol{t}^{\boldsymbol{p}_0^{(n)}} + c_{1,0} &= 0 \\ c_{2,1}y_1\boldsymbol{t}^{\boldsymbol{p}_0^{(1)}} + c_{2,2}y_2\boldsymbol{t}^{\boldsymbol{p}_0^{(2)}} + \cdots + c_{2,n}y_n\boldsymbol{t}^{\boldsymbol{p}_0^{(n)}} + c_{2,0} &= 0 \\ \vdots & \vdots \\ c_{d,1}y_1\boldsymbol{t}^{\boldsymbol{p}_0^{(1)}} + c_{d,2}y_2\boldsymbol{t}^{\boldsymbol{p}_0^{(2)}} + \cdots + c_{d,n}y_n\boldsymbol{t}^{\boldsymbol{p}_0^{(n)}} + c_{d,0} &= 0 \end{cases} \tag{11}$$

in $(\mathbb{C}^*)^d$. This is a system of d equations in d variables $\boldsymbol{t} = (t_1, \ldots, t_d)$ with generic coefficients. Most importantly, all equation in the system have exactly the same set of monomials $\boldsymbol{t}^{\boldsymbol{p}_0^{(1)}}, \ldots, \boldsymbol{t}^{\boldsymbol{p}_0^{(n)}}$ together with a constant term. By Theorem 1 above, the solutions of this system in $(\mathbb{C}^*)^d$ are all isolated and the number of those isolated solutions is

$$d! \cdot \mathrm{Vol}_d(\mathrm{conv}\{\boldsymbol{p}_0^{(1)}, \ldots, \boldsymbol{p}_0^{(n)}, \boldsymbol{0}\}). \qquad \square$$

From the above proof, the degree of a solution component of a Laurent binomial system can be understood as the number of the $\mathbb{C}^*$-solutions of certain Laurent polynomial system given by its BKK bound.

4. Solving Laurent binomial systems: computational aspects

The main theme of this article is the computation methods for solving a Laurent binomial system of the form $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ with $A \in M_{n\times m}(\mathbb{Z})$. If $\boldsymbol{b}$ is given exactly, then a practical algorithm can be deduced from Proposition 2 to describe the solution of $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ exactly (with no numerical error). Indeed, as long as the Smith Normal Form of A and those matrices P_0, P_r, Q_0, Q_r in (5) are computed, one can explicitly describe the dimension, number of components, and global parametrizations of the solution set in $(\mathbb{C}^*)^n$. Of course, when the solution set consists of isolated points only, the exact solutions can be listed. This algorithm has been implemented in software packages such as: Macaulay2 [19] package Binomials [25][2] developed by Thomas Kahle and the "symbolic" binomial system solver included in Hom4PS-3 [6].

REMARK 2. For a Laurent binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ having only isolated solutions, a potentially more efficient algorithm utilizing the *Hermite Normal Form* (instead of Smith Normal Form) of A is often preferred. Such algorithm is outlined in [30] and has been implemented in software packages such as Hom4PS-2.0 [28], Hom4PS-3 [6], PHCpack [38], and PHoM [20].

When the solution set consists of positive dimensional components, Proposition 3 provides a computationally viable means for computing the degree of each component as the volume of certain convex polytopes. The volume computation of convex polytopes is a well studied subject (see [5] for a broad survey). There are several software packages designed to carry out the volume computation, such as qhull [2], Vinci [14], etc. In the proof of Proposition 3, another strategy for the degree computation was suggested: the degree is also given by the BKK bound of certain Laurent polynomial system. We may therefore find the corresponding Laurent polynomial system first, and compute its BKK bound for the degree. There are many mature software packages specialized for computing the BKK bound, including MixedVolume [13], DEMiCs [35], Gfan [24], MixedVol [16], MixedVol-2.0 [27], and MixedVol-3 [7]. In the computation results presented in the later part of this article, the software package MixedVol-3 is used. Utilizing a robust numerical technique known as "mixed cells enumeration", MixedVol-3 produces outstanding performance and exhibits a great parallel potential in computing the degree as shown in Section 5.2 and Section 6.2.

[2]Actually, Binomials goes much beyond what have been described above. It is capable of finding more general structure of *binomial ideals*, ideals generated by binomial systems, such as performing *primary decomposition* as well as carrying out computation over fields other than $\mathbb{C}$.

All in all, when the coefficients $\boldsymbol{b}$ in the system $\boldsymbol{x}^A = \boldsymbol{b}$ is given in the exact form (with no numerical error), then Propositions 2 and 3 give practical algorithms for computing the dimension, the number of components, the parametrizations (including listing isolated solutions), and the degree of solution components with positive dimension of the solution set of the Laurent binomial system in $(\mathbb{C}^*)^n$.

For problems from applied sciences, it is often the case that the coefficients of the input binomial systems are just given approximately. Such systems, containing errors, can be written as

$$\boldsymbol{x}^A = \boldsymbol{b} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \in \mathbb{C}^n$ with $\|\boldsymbol{\epsilon}\|$ very small. For these systems, accurately approximating the solutions of $\boldsymbol{x}^A = \boldsymbol{b}$ or even determining the consistency of the problem generally yields a tough challenge. The main reason is, the algorithms mentioned above for describing the solution sets can be extremely sensitive to the perturbation in $\boldsymbol{b}$. To illustrate this, consider a simple Laurent binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ with

$$A = \begin{pmatrix} 3 & 2 & 3 & 4 & 2 & 5 \\ 6 & -3 & 8 & -3 & 8 & 7 \\ -2 & -5 & 7 & 3 & 6 & 5 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 2 & 0 & 5 & 0 \\ 1 & 2 & 0 & -3 & 0 & 5 \end{pmatrix}, \tag{12}$$

$\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$, and $\boldsymbol{b} = (b_1, b_2, b_3, b_4, b_5, b_6)$. This system involves only fairly small powers of x_i's. Amazingly, after applying the direct algorithm based on the computation of the Smith Normal Form of matrix A, numerical error in $\boldsymbol{b}$ on the scale of *one in a million* could be magnified to *more than 100%* relative error in the solution! In more details, a direct computation of the Smith Normal Form PAQ of A results in the unimodular matrices

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1500 & 1369 & -482 & -1526 & -74 \\ 2 & -3001 & 2739 & -963 & -3051 & -148 \end{pmatrix}, Q = \begin{pmatrix} -19 & -763 & -499 & -33 & -756 & 2611109 \\ 56 & 2268 & 1484 & 98 & 2248 & -7764251 \\ 82 & 3327 & 2177 & 144 & 3298 & -11390792 \\ -11 & -439 & -287 & -19 & -435 & 1502424 \\ -29 & -1178 & -771 & -51 & -1168 & 4034095 \\ -25 & -1018 & -666 & -44 & -1009 & 3484933 \end{pmatrix}.$$

To proceed, the computation of $\boldsymbol{b}^Q$ required by the direct method would then involve monomials such as

$$b_1^{2,611,109}\, b_2^{-7,764,251}\, b_3^{-11,390,792}\, b_4^{1,502,424}\, b_5^{4,034,095}\, b_6^{3,484,933}$$

which contains exponents on the scale of tens of millions. Consequently, very small numerical error in b_i's could be magnified to an unacceptable level. Table 1 shows the relative error produced after a random perturbation on the scale of one in a million.

Table 1. Relative error in the solutions produced by the direct symbolic algorithm for the Laurent binomial system (12) after a random perturbation of the coefficients on the scale 0.0001% (one millionth). Five (among 6910) solutions that produce biggest relative error are listed along with the average relative error

Solution	#1	#2	#3	#4	#5	Avg.
Rel. error	$> 100\%$	$> 100\%$	$\approx 100\%$	$\approx 100\%$	$\approx 100\%$	54%

Remark 3. For binomial systems whose coefficients are all on the unit circle, i.e., $|b_i| = 1$ for each i, such numerical problems can be greatly alleviated (or even completely eliminated) by simply solving for $\log \boldsymbol{x}$ instead of $\boldsymbol{x}$ itself: With some branch of the (complex) logarithm $\log\colon \mathbb{C}^* \to \mathbb{C}$, write $\log \boldsymbol{x} = (\log x_1, \ldots, \log x_n)$, then it is easy to verify that the solution $\boldsymbol{x}$ of the system $\boldsymbol{x}^A = \boldsymbol{b}$ must satisfy

$$(\log \boldsymbol{x})A = \log \boldsymbol{b}.$$

This technique has been widely used. In particular, almost all software packages implementing "polyhedral homotopy continuation method" (as described in Section 5.1) have employed this technique for solving binomial systems. A detailed analysis of this approach can be found in [39]. Nonetheless, one must notice that this technique requires the computation of $\log \boldsymbol{b}$. Since logarithm becomes infinitely sensitive to changes near 0, this technique may not be as safe when some b_i is close to 0.

For practical needs, we propose, in the following two subsections, alternative numerical methods for approximating solutions and determining the consistency for Laurent binomial systems with coefficients being given approximately.

4.1. A homotopy continuation method for solving binomial systems with approximated coefficients

In this section, a homotopy-based method is proposed which is capable of eliminating numerical difficulties illustrated above. Here let us assume the binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ with $A \in M_{n\times m}(\mathbb{Z})$ of rank r is consistent. The issue of numerically determining the consistency will be discussed in

Section 4.2 instead. Furthermore, let us assume, without loss of generality, $m = r = \operatorname{rank} A$. Recall that when $n > r$, by Remark 1, the components of the solution set of $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ (each of dimension $n-r$) can be expressed as conjugate orbits under a $(\mathbb{C}^*)^{n-r}$ action. It hence suffices to obtain just one point in each orbit. For this purpose, it is evident in the construction of (6) that the original system $\boldsymbol{x}^A = \boldsymbol{b}$ of $m = r$ equations in $n > r$ variables can be replaced by a system of r equations in r variables. More precisely, by Proposition 2, the solutions of the system of r equations

$$(z_1, \ldots, z_r, 1, \ldots, 1)^{PA} = (z_1, \ldots, z_r)^{P_r A} = \boldsymbol{b}$$

in exactly r variables $z_1, \ldots, z_r$ in $(\mathbb{C}^*)^r$ are in one-to-one correspondence with the components of the solution set of the original system $\boldsymbol{x}^A = \boldsymbol{b}$ in $(\mathbb{C}^*)^n$ via the map

$$(z_1, \ldots, z_r) \mapsto \{(z_1, \ldots, z_r, t_1, \ldots, t_{n-r})^P : (t_1, \ldots, t_{n-r}) \in (\mathbb{C}^*)^{n-r}\}.$$

Note that the exponent matrix $P_r A$ of $(z_1, \ldots, z_r)^{P_r A} = \boldsymbol{b}$ is a $r \times r$ matrix of rank r. It is therefore sufficient to restrict our attention to the systems of the form $\boldsymbol{x}^A = \boldsymbol{b}$ where A is an $n \times n$ matrix of full rank.

In our new homotopy-based approach, the target binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ is deformed into a closely related "starting system"

$$\boldsymbol{x}^A = \boldsymbol{\gamma}$$

by the homotopy function

$$H(\boldsymbol{x}, t) = \boldsymbol{x}^A - (1-t)\boldsymbol{\gamma} - t\boldsymbol{b}, \quad t \in [0, 1], \tag{13}$$

where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n) = (\pm 1, \ldots, \pm 1)$ with each γ_i having the same sign as the real part of b_i. Clearly, $H(\boldsymbol{x}, 0) = \boldsymbol{x}^A - \boldsymbol{\gamma}$ is the starting system and $H(\boldsymbol{x}, 1) = \boldsymbol{x}^A - \boldsymbol{b}$ is the target system. In the following proposition, it will be shown that as t varies from 0 to 1, the corresponding solutions of $H(\boldsymbol{x}, t) = \boldsymbol{0}$ also vary smoothly, forming smooth solution paths connecting the solutions of the starting system to the solutions of the target system. The reason to assign ± 1 to γ_i is twofold: First, the expression $\boldsymbol{\gamma}^Q$ (required to turn the system into $\boldsymbol{z}^{PAQ} = \boldsymbol{\gamma}^Q$) can be computed with ease since any power of ± 1 is still ± 1 depending on the parity of the exponent. Second, since all the coefficients are exact (carrying no numerical error), the purely symbolic algorithm can be applied to solve the starting system with no needs to concern the numerical stability.

PROPOSITION 4. *Under the assumption that A is an $n \times n$ matrix with full rank, the solution set of*

$$H(\boldsymbol{x}, t) = \boldsymbol{x}^A - (1-t)\boldsymbol{\gamma} - t\boldsymbol{b} = \boldsymbol{0}$$

in $(\mathbb{C}^*)^n \times [0,1]$ consists of smooth paths parametrized by t.

PROOF. Let h_i be the i-th component of H. Since $h_i(\boldsymbol{x}, t) = \boldsymbol{x}^{\boldsymbol{a}^{(i)}} - (1-t)\gamma_i - tb_i$, it follows that

$$\frac{\partial h_i}{\partial x_j} = a_{ji}\boldsymbol{x}^{\boldsymbol{a}^{(i)}} x_j^{-1}.$$

Therefore the Jacobian matrix $H_{\boldsymbol{x}}$ of $H(\boldsymbol{x}, t)$ with respect to $\boldsymbol{x}$ is

$$\begin{pmatrix} a_{11}\boldsymbol{x}^{\boldsymbol{a}^{(1)}} x_1^{-1} & a_{21}\boldsymbol{x}^{\boldsymbol{a}^{(1)}} x_2^{-1} & \dots & a_{n1}\boldsymbol{x}^{\boldsymbol{a}^{(1)}} x_n^{-1} \\ a_{12}\boldsymbol{x}^{\boldsymbol{a}^{(2)}} x_1^{-1} & a_{22}\boldsymbol{x}^{\boldsymbol{a}^{(2)}} x_2^{-1} & \dots & a_{n2}\boldsymbol{x}^{\boldsymbol{a}^{(2)}} x_n^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n}\boldsymbol{x}^{\boldsymbol{a}^{(n)}} x_1^{-1} & a_{2n}\boldsymbol{x}^{\boldsymbol{a}^{(n)}} x_2^{-1} & \dots & a_{nn}\boldsymbol{x}^{\boldsymbol{a}^{(n)}} x_n^{-1} \end{pmatrix}$$

which can be factorized as

$$\begin{pmatrix} \boldsymbol{x}^{\boldsymbol{a}^{(1)}} & & & \\ & \boldsymbol{x}^{\boldsymbol{a}^{(2)}} & & \\ & & \ddots & \\ & & & \boldsymbol{x}^{\boldsymbol{a}^{(n)}} \end{pmatrix} \begin{pmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1^{-1} & & & \\ & x_2^{-1} & & \\ & & \ddots & \\ & & & x_n^{-1} \end{pmatrix}.$$

The matrix in the middle of this factorization is exactly $A^\top$ which is nonsingular by assumption. Since $\boldsymbol{x} \in (\mathbb{C}^*)^n$, the two other factors are also nonsingular. Therefore the matrix $H_{\boldsymbol{x}}(\boldsymbol{x}, t)$ is nonsingular at any point $(\boldsymbol{x}, t) \in (\mathbb{C}^*)^n \times (0,1)$ that satisfies $H(\boldsymbol{x}, t) = \boldsymbol{0}$. At such a point $(\boldsymbol{x}, t)$, by the *implicit function theorem* the solutions of $H(\boldsymbol{x}, t) = \boldsymbol{0}$ near $(\boldsymbol{x}, t)$ form a smooth path parametrized by t. This is true at any solution point, so such smooth solution path can be extended indefinitely on the t-direction unless it escapes $(\mathbb{C}^*)^n$. But that is impossible, since the sign of each γ_i in $\boldsymbol{\gamma} = (\pm 1, \dots, \pm 1)$ is chosen so that

$$(1-t)\boldsymbol{\gamma} + t\boldsymbol{b} \ \in \ (\mathbb{C}^*)^n \quad \text{for all} \quad t \in [0,1].$$

Hence, for all $t \in [0,1]$, the number of solutions of $H(\boldsymbol{x}, t) = \boldsymbol{0}$ in $(\mathbb{C}^*)^n$ is fixed: it always equals $|\det A|$. Consequently, solution paths cannot escape $(\mathbb{C}^*)^n$. Therefore all solutions path can be extended smoothly to the entire t-interval of $[0,1]$. □

Now, the well developed path tracking algorithms in the numerical continuation methods can be applied to track the smooth solution paths defined by $H(\boldsymbol{x}, t) = \mathbf{0}$ originating at the known solutions of the starting binomial system and ending at the solutions of the target binomial system. Here we refer to [1], [30], and [37] for the basic theory of path tracking algorithms. This homotopy-based algorithm is implemented in Hom4PS-3 [6].

Remarkable improvement of the results by this algorithm is shown in Table 2: For the example in (12), the relative error produced after perturbation is roughly on the same scale as the perturbation itself.

Table 2. Relative error produced by the homotopy based algorithm for the binomial system (12) after a random perturbation of the coefficients on the scale 0.0001% (one millionth). The five solutions listed in this table are the same as those listed in Table 1

Solution	#1	#2	#3	#4	#5	Avg.
Rel. error	$< 10^{-4}\%$	$< 10^{-3}\%$	$< 10^{-3}\%$	$< 10^{-2}\%$	$< 10^{-3}\%$	$< 10^{-3}\%$

4.2. Verifying the consistency numerically

While the binomial system is assumed to be consistent throughout previous sections, in the more general cases, as stated in Section 3, when the number of equations in the given binomial system $\boldsymbol{x}^A = \boldsymbol{b}$ is greater than the rank of the matrix A, the system may become inconsistent. Recall that, with the notations in (5), the system is consistent if and only if

$$\boldsymbol{b}^{Q_0} = \mathbf{1}$$

where $\mathbf{1} = (1, \ldots, 1) \in (\mathbb{C}^*)^{m-r}$. So the question of the consistency of the binomial system can be answered by simply verifying the above equality. This can certainly be achieved quite easily when $\boldsymbol{b}$ is given in exact form. For binomial systems where $\boldsymbol{b}$ is only given approximately, however, checking whether or not $\boldsymbol{b}^{Q_0} = \mathbf{1}$ is an *ill-posed* problem from numerical points of views and should be avoided at all cost. After all, a generic perturbation in $\boldsymbol{b}$, however small in magnitude, will break the above equality.

It is thus necessary to rephrase the question of consistency into a question of closeness: *How close is the binomial system from being consistent?* More precisely, let W be the algebraic set in $(\mathbb{C}^*)^n$ defined by

$$\boldsymbol{b}^{Q_0} = \mathbf{1}$$

then the binomial system is consistent if and only if the given $\boldsymbol{b}$ lies in W. In this context, the distance between $\boldsymbol{b}$ and the smooth part of W may be used

as a reasonable measure of how close the given binomial system is to being consistent. Under this interpretation, this problem becomes *well-posed*, and can be answered via numerical computation.

While the distance between a point and an algebraic set may be generally difficult to compute, the distance in the log-norm space can be obtained quite easily: it is easy to verify that $\boldsymbol{b}^{Q_0} = \mathbf{1}$ is equivalent to

$$\begin{cases} (\mathrm{Re}(\log \boldsymbol{b})) \cdot Q_0 & = \mathbf{0} \\ (\mathrm{Im}(\log \boldsymbol{b})) \cdot Q_0 & = \mathbf{0} \pmod{2\pi} \end{cases}$$

where Re and Im denote the component-wise real and imaginary parts respectively. So the distance, in the log-norm sense, can be computed simply as the distance between $\mathrm{Re}(\log \boldsymbol{b})$ and the kernel of Q_0 and the distance between $\mathrm{Im}(\log \boldsymbol{b})$ and the kernel of Q_0 modulo 2π.

5. Applications

For binomial systems, there is a wide range of application in sciences, engineering, and mathematics. In this section, two particularly interesting and important applications are presented.

5.1. Polyhedral homotopy for solving general polynomial systems

In the 90's, a considerable research effort in Europe had been directed to the problem of solving polynomial systems in two consecutive major projects, PoSSo (Polynomial System Solving) and FRISCO (FRamework for Integrated Symbolic/numerical COmputation), supported by the European Commission. Those research projects focused on the development of the well-established Gröbner basis methods within the framework of computer algebra. Their reliance on symbolic manipulation makes those methods seem somewhat limited to relatively small problems. In 1977, Garcia and Zangwill [17] and Drexler [11] independently discovered that the *homotopy continuation method* could be used to find the full set of isolated solutions to a polynomial system numerically. In the last several decades, the method has been quite well established and proved to be reliable and efficient. Note that *continuation methods* are the method of choice to deal with general nonlinear systems of equations numerically and globally as illustrated by the extensive bibliography listed in [1] where general ideas of the method were discussed.

The state of the art of the homotopy continuation method for solving general polynomial systems is the *polyhedral* homotopy method initiated by B. Huber and B. Sturmfels [23], in which solving binomial systems plays a crucially important role. For an $n \times n$ square polynomial systems

$$P(x_1, \ldots, x_n) = P(\boldsymbol{x}) = \begin{cases} p_1(\boldsymbol{x}) & = \sum_{\boldsymbol{a} \in S_1} c_{1,\boldsymbol{a}}\, \boldsymbol{x}^{\boldsymbol{a}} \\ & \vdots \\ p_n(\boldsymbol{x}) & = \sum_{\boldsymbol{a} \in S_n} c_{n,\boldsymbol{a}}\, \boldsymbol{x}^{\boldsymbol{a}} \end{cases} \tag{14}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{a} = (a_1, \ldots, a_n)^\top \in \mathbb{N}_0^n$, and $\boldsymbol{x}^{\boldsymbol{a}} = x_1^{a_1} \cdots x_n^{a_n}$ as before. Here S_j, a finite subset of $\mathbb{N}_0^n$, is called the **support** of $p_j(\boldsymbol{x})$. For fixed supports $S_1, \ldots, S_n$, it is a basic fact in algebraic geometry that for generic choices of the complex coefficients $c_{j,\boldsymbol{a}} \in \mathbb{C}^*$ the number of isolated solutions of the system $P(\boldsymbol{x}) = \boldsymbol{0}$ in $(\mathbb{C}^*)^n$ is a fixed number. The word "generic" here can be understood as "randomly chosen". Its precise meaning can be found in [4], [23] and [30]. This fixed number also serves as an upper bound on the number of isolated solutions of $P(\boldsymbol{x}) = \boldsymbol{0}$ in $(\mathbb{C}^*)^n$ among all choices of coefficients. In [4], this upper bound, now commonly known as the BKK bound as mentioned before, is formulated in terms of *mixed volume*: For convex polytopes $\mathcal{Q}_1, \ldots, \mathcal{Q}_k \subset \mathbb{R}^k$, let $\lambda_1 \mathcal{Q}_1, \ldots, \lambda_k \mathcal{Q}_k$ represent their scaled version by factors of positive $\lambda_1, \ldots, \lambda_k$ respectively. Then the *Minkowski sum* $\lambda_1 \mathcal{Q}_1 + \cdots + \lambda_k \mathcal{Q}_k$ is also a convex polytope. It can be shown that the volume $\mathrm{Vol}_k(\lambda_1 \mathcal{Q}_1 + \cdots + \lambda_k \mathcal{Q}_k)$ in $\mathbb{R}^k$ is a homogeneous polynomial in $\lambda_1, \ldots, \lambda_k$. The **mixed volume**, denoted by $\mathrm{MVol}(\mathcal{Q}_1, \ldots, \mathcal{Q}_k)$, is defined to be the coefficient of $\lambda_1 \times \lambda_2 \times \cdots \times \lambda_k$ in this polynomial. The theory of BKK bound [4] states that the number of isolated solutions of the system $P(\boldsymbol{x}) = \boldsymbol{0}$ in $(\mathbb{C}^*)^n$ for generic choices of the coefficients is the mixed volume of the convex hull of the supports of $p_1, \ldots, p_n$, i.e.

$$\mathrm{MVol}(\mathrm{conv}\, S_1, \ldots, \mathrm{conv}\, S_n).$$

The Kushnirenko's theorem (Theorem 1) can be considered as a special case of this statement with all the supports $S_1, \ldots, S_n$ being the same.

We shall restrict our focus on solving a polynomial system $P(\boldsymbol{x}) = \boldsymbol{0}$ in (14) with "generic" (nonzero) complex coefficients $c_{j,\boldsymbol{a}} \in \mathbb{C}^*$. When the system with generic coefficients is solved, one can always use it to solve the system with specifically given coefficients with the same supports by the *Cheater's* homotopy [31] (or [36]).

To solve $P(\boldsymbol{x}) = \mathbf{0}$ in (14), consider, with a new variable t, the homotopy

$$H(x_1,\ldots,x_n,t) = H(\boldsymbol{x},t) = \begin{cases} h_1(\boldsymbol{x},t) &= \sum_{\boldsymbol{a}\in S_1} c_{1,\boldsymbol{a}}\, \boldsymbol{x}^{\boldsymbol{a}} t^{\omega_1(\boldsymbol{a})} \\ &\vdots \\ h_n(\boldsymbol{x},t) &= \sum_{\boldsymbol{a}\in S_n} c_{n,\boldsymbol{a}}\, \boldsymbol{x}^{\boldsymbol{a}} t^{\omega_n(\boldsymbol{a})} \end{cases} \tag{15}$$

with "lifting" functions $\omega_1,\ldots,\omega_n$, where each $\omega_k\colon S_k \to \mathbb{Q}$ has randomly chosen images. Note that when $t = 1$, $H(\boldsymbol{x},1) = P(\boldsymbol{x})$. For $\boldsymbol{a} \in S_k$, write $\hat{\boldsymbol{a}} = (\boldsymbol{a}, \omega_k(\boldsymbol{a}))$. In [23], it was shown that if the system $P(\boldsymbol{x}) = \mathbf{0}$ has isolated solutions in $(\mathbb{C}^*)^n$, then there exists $\hat{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}, 1) \in \mathbb{R}^{n+1}$ with $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n)$ and a corresponding collection of pairs $\{\boldsymbol{a}_1, \boldsymbol{a}_1'\} \subset S_1, \ldots, \{\boldsymbol{a}_n, \boldsymbol{a}_n'\} \subset S_n$ such that for each $k = 1,\ldots,n$

$$\langle \hat{\boldsymbol{a}}_k, \hat{\boldsymbol{\alpha}}\rangle = \langle \hat{\boldsymbol{a}}_k', \hat{\boldsymbol{\alpha}}\rangle < \langle \hat{\boldsymbol{a}}, \hat{\boldsymbol{\alpha}}\rangle \quad \text{for all } \boldsymbol{a} \in S_k \setminus \{\boldsymbol{a}_k, \boldsymbol{a}_k'\} \tag{16}$$

and

$$\kappa_{\boldsymbol{\alpha}} := \left|\det\begin{pmatrix} \boldsymbol{a}_1 - \boldsymbol{a}_1' & \ldots & \boldsymbol{a}_n - \boldsymbol{a}_n' \end{pmatrix}\right| > 0.$$

Here $\langle\ ,\ \rangle$ stands for the standard inner product in Euclidean space. Let $\mathcal{T}$ be the collection of all such $\boldsymbol{\alpha}$'s, then

$$\sum_{\boldsymbol{\alpha}\in\mathcal{T}} \kappa_{\boldsymbol{\alpha}}$$

is independent of the choice of the lifting functions $\omega_1,\ldots,\omega_k$. In fact, this number agrees with the number of isolated solutions of the system $P(\boldsymbol{x}) = \mathbf{0}$ in $(\mathbb{C}^*)^n$ known as the BKK bound mentioned before.

Now, for a fixed $\boldsymbol{\alpha}$ in $\mathcal{T}$ along with its corresponding set of pairs $\{\boldsymbol{a}_1, \boldsymbol{a}_1'\} \subset S_1, \ldots, \{\boldsymbol{a}_n, \boldsymbol{a}_n'\} \subset S_n$, let $\beta_k = \langle \hat{\boldsymbol{a}}_k, \hat{\boldsymbol{\alpha}}\rangle = \langle \hat{\boldsymbol{a}}_k', \hat{\boldsymbol{\alpha}}\rangle = \langle \boldsymbol{a}_k', \boldsymbol{\alpha}\rangle + \omega_k(\boldsymbol{a}_k')$ for $k = 1,\ldots,n$. Then by (16), for each $k = 1,\ldots,n$,

$$\beta_k < \langle \hat{\boldsymbol{a}}, \boldsymbol{\alpha}\rangle \quad \text{for all } \boldsymbol{a} \in S_k \setminus \{\boldsymbol{a}_k, \boldsymbol{a}_k'\}. \tag{17}$$

By the change of variables $\boldsymbol{x} = t^{\boldsymbol{\alpha}} \bullet \boldsymbol{y}$ for $\boldsymbol{y} = (y_1,\ldots,y_n)$; that is,

$$\begin{cases} x_1 &= t^{\alpha_1} y_1, \\ &\vdots \\ x_n &= t^{\alpha_n} y_n, \end{cases} \tag{18}$$

we have, for $\boldsymbol{a} = (a_1, \ldots, a_n) \in S_k$ and $\hat{\boldsymbol{a}} = (\boldsymbol{a}, \omega_k(\boldsymbol{a}))$

$$\begin{aligned}
\boldsymbol{x}^{\boldsymbol{a}} t^{\omega_k(\boldsymbol{a})} &= x_1^{a_1} \ \ldots \ x_n^{a_n} \, t^{\omega_k(\boldsymbol{a})} \\
&= (t^{\alpha_1} y_1)^{a_1} \ \ldots \ (t^{\alpha_n} y_n)^{a_n} \, t^{\omega_k(\boldsymbol{a})} \\
&= y_1^{a_1} \ \ldots \ y_n^{a_n} \, t^{a_1\alpha_1 + \cdots + a_n\alpha_n + \omega_k(\boldsymbol{a})} \\
&= \boldsymbol{y}^{\boldsymbol{a}} \, t^{\langle (\boldsymbol{a}, \omega_k(\boldsymbol{a})), (\boldsymbol{\alpha}, 1) \rangle} \\
&= \boldsymbol{y}^{\boldsymbol{a}} \, t^{\langle \hat{\boldsymbol{a}}, \hat{\boldsymbol{\alpha}} \rangle}
\end{aligned}$$

with $\hat{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}, 1)$. Substituting the result into $H(\boldsymbol{x}, t)$ in (15), it follows that

$$\bar{H}^{\boldsymbol{\alpha}}(\boldsymbol{y}, t) := H(t^{\boldsymbol{\alpha}} \bullet \boldsymbol{y}, t) = \begin{cases} \bar{h}_1^{\boldsymbol{\alpha}}(\boldsymbol{y}, t) := h_1(t^{\boldsymbol{\alpha}} \bullet \boldsymbol{y}, t) = \sum_{\boldsymbol{a} \in S_1} c_{1,\boldsymbol{a}} \, \boldsymbol{y}^{\boldsymbol{a}} \, t^{\langle \hat{\boldsymbol{a}}, \hat{\boldsymbol{\alpha}} \rangle} \\ \vdots \\ \bar{h}_n^{\boldsymbol{\alpha}}(\boldsymbol{y}, t) := h_n(t^{\boldsymbol{\alpha}} \bullet \boldsymbol{y}, t) = \sum_{\boldsymbol{a} \in S_n} c_{n,\boldsymbol{a}} \, \boldsymbol{y}^{\boldsymbol{a}} \, t^{\langle \hat{\boldsymbol{a}}, \hat{\boldsymbol{\alpha}} \rangle}. \end{cases}$$

Though the above expression may contain positive or negative powers of t, the minimum exponents of t in each $\bar{h}_k^{\boldsymbol{\alpha}}$ is actually given by β_k. Therefore, if we define

$$H^{\boldsymbol{\alpha}}(\boldsymbol{y}, t) := \begin{cases} t^{-\beta_1} \bar{h}_1(\boldsymbol{y}, t) & = \sum_{\boldsymbol{a} \in S_1} c_{1,\boldsymbol{a}} \, \boldsymbol{y}^{\boldsymbol{a}} \, t^{\langle \hat{\boldsymbol{a}}, \hat{\boldsymbol{\alpha}} \rangle - \beta_1} \\ & \vdots \\ t^{-\beta_n} \bar{h}_n(\boldsymbol{y}, t) & = \sum_{\boldsymbol{a} \in S_n} c_{n,\boldsymbol{a}} \, \boldsymbol{y}^{\boldsymbol{a}} \, t^{\langle \hat{\boldsymbol{a}}, \hat{\boldsymbol{\alpha}} \rangle - \beta_n}, \end{cases} \tag{19}$$

then, by (17), each component of $H^{\boldsymbol{\alpha}}$ has exactly two terms having no powers of t while all other terms have positive powers of t. Hence, when $t = 0$, $H^{\boldsymbol{\alpha}}(\boldsymbol{y}, 0) = \boldsymbol{0}$ is a binomial system of equations

$$\begin{cases} c_{1,\boldsymbol{a}_1} \boldsymbol{y}^{\boldsymbol{a}_1} + c_{1,\boldsymbol{a}_1'} \boldsymbol{y}^{\boldsymbol{a}_1'} & = 0 \\ & \vdots \\ c_{n,\boldsymbol{a}_n} \boldsymbol{y}^{\boldsymbol{a}_n} + c_{n,\boldsymbol{a}_n'} \boldsymbol{y}^{\boldsymbol{a}_n'} & = 0 \end{cases} \tag{20}$$

with $\kappa_{\boldsymbol{\alpha}}$ nonsingular isolated solutions in $(\mathbb{C}^*)^n$. After (20) is solved, these nonsingular solutions obtained can be used as the starting points for following the homotopy paths $\boldsymbol{y}(t)$ of $H^{\boldsymbol{\alpha}}(\boldsymbol{y}, t) = \boldsymbol{0}$ from $t = 0$ to $t = 1$. Note that the change of variables $\boldsymbol{x} = t^{\boldsymbol{\alpha}} \bullet \boldsymbol{y}$ in (18) yields $\boldsymbol{x} \equiv \boldsymbol{y}$ at $t = 1$. Therefore, each end point $\boldsymbol{y}(1)$ at $t = 1$ of the homotopy path $\boldsymbol{y}(t)$ of $H^{\boldsymbol{\alpha}}(\boldsymbol{y}, t) = \boldsymbol{0}$ is also an end point $\boldsymbol{x}(1)$ of the homotopy path $\boldsymbol{x}(t)$ of the homotopy $H(\boldsymbol{x}, t) = \boldsymbol{0}$ given in (15) which, in turn, provides a solution of the target system $P(\boldsymbol{x}) = \boldsymbol{0}$ in (14). Altogether it yields $\kappa_{\boldsymbol{\alpha}}$ of the isolated solutions of $P(\boldsymbol{x}) = \boldsymbol{0}$ in $(\mathbb{C}^*)^n$ along this route. In [23], it was shown that as one follows the homotopy paths

defined by $H^{\boldsymbol{\alpha}}(\boldsymbol{y}, t) = \mathbf{0}$ for all individual $\boldsymbol{\alpha} \in \mathcal{T}$, one obtains all (isolated) solutions of $P(\boldsymbol{x}) = \mathbf{0}$ in $(\mathbb{C}^*)^n$, justifying, in fact, the BKK bound agrees with $\sum_{\boldsymbol{\alpha}\in\mathcal{T}} \kappa_{\boldsymbol{\alpha}}$.

Since its inception, this general method has achieved a great success. It is widely considered to be one of the most efficient, robust and reliable numerical methods for solving systems of polynomial equations. The method is implemented in many mature software packages, such as Hom4PS-2.0 [28], Hom4PS-2.0para [29], Hom4PS-3 [6], PHCpack [38], and PHoM [20].

Remark 4.

1. The purpose of restricting the target polynomial system $P(\boldsymbol{x}) = \mathbf{0}$ in (14) to those having generic complex coefficients is to ensure the smoothness and the finiteness of the homotopy paths of the homotopy $H(\boldsymbol{x}, t) = \mathbf{0}$ in (15). More precisely, with a randomly chosen set of complex coefficients for $P(\boldsymbol{x})$, the solution set in $(\mathbb{C}^*)^n \times [0, 1]$ defined by (19) consists of smooth paths that do not diverge to infinity with *probability one*.
2. Even though the above procedure only addresses the solution set in $(\mathbb{C}^*)^n$ of the target system $P(\boldsymbol{x}) = \mathbf{0}$ in (14), this method has been extended in [32] so that all isolated zeros of the target system $P(\boldsymbol{x})$ in $\mathbb{C}^n$ can be obtained.

5.2. The Master Space problem in gauge theories

In this subsection, we present an example of application in physics that highlights the importance and difficulties in computing the degree of the solution set of a large scale binomial system. In theoretical physics, *supersymmetric gauge theories* form a class of field theories that has attracted a great attention from physicists and mathematicians. As an important tool for such studies, the concept of the "master space" was developed in [22]. Despite the physical significance, here we simply take the subject as a rich source of challenging problems where the solution to binomial systems plays an important role. Central to this theory is the *Jacobian ideal*, the ideal generated by the partial derivatives of the *superpotentials*. In the first place, in one basic case studied in [22, 33], for fixed $m, k \in \mathbb{N}$ the superpotential is given by

$$W = \sum_{i\in\mathbb{Z}_k} \sum_{j\in\mathbb{Z}_m} x_{i,j}y_{i+1,j}z_{i+1,j+1} - y_{i,j}x_{i,j+1}z_{i+1,j+1} \tag{21}$$

where $\mathbb{Z}_k$ and $\mathbb{Z}_m$ are $\{0, 1, \ldots, k-1\}$ and $\{0, 1, \ldots, m-1\}$ respectively, both with modular arithmetic (e.g. $x_{i,m} = x_{i,0}$ for any i and $x_{k,j} = x_{0,j}$ for any j). This is a polynomial in $3mk$ variables; they are $x_{i,j}, y_{i,j}, z_{i,j}$ for

$i \in \mathbb{Z}_k = \{0, 1, \ldots, k-1\}$ and $j \in \mathbb{Z}_m = \{0, 1, \ldots, m-1\}$. For example, when $m = k = 2$, the superpotential is

$$\begin{aligned} W = {} & x_{0,0}y_{1,0}z_{1,1} - y_{0,0}x_{0,1}z_{1,1} + x_{0,1}y_{1,1}z_{1,0} - y_{0,1}x_{0,0}z_{1,0} \\ & + x_{1,0}y_{0,0}z_{0,1} - y_{1,0}x_{1,1}z_{0,1} + x_{1,1}y_{0,1}z_{0,0} - y_{1,1}x_{1,0}z_{0,0}, \end{aligned}$$

a polynomial in 12 variables $x_{0,0}$, $x_{0,1}$, $x_{1,0}$, $x_{1,1}$, $y_{0,0}$, $y_{0,1}$, $y_{1,0}$, $y_{1,1}$, $z_{0,0}$, $z_{0,1}$, $z_{1,0}$, $z_{1,1}$.

Secondly, the *Jacobian ideal* is the ideal generated by the partial derivatives of the superpotential W with respect to variables $x_{i,j}, y_{i,j}, z_{i,j}$,

$$\left\langle \frac{\partial W}{\partial x_{i,j}}, \frac{\partial W}{\partial y_{i,j}}, \frac{\partial W}{\partial z_{i,j}} \right\rangle_{i \in \mathbb{Z}_k, j \in \mathbb{Z}_m}$$

in $\mathbb{C}[x_{i,j}^{\pm}, y_{i,j}^{\pm}, z_{i,j}^{\pm}]$ for $i \in \mathbb{Z}_k, j \in \mathbb{Z}_m$. Notice that in W, each variable appears in exactly two distinct terms. Consequently, the partial derivative of W with respect to each variable consists of exactly two terms, hence it forms a binomial polynomial. For instance,

$$\frac{\partial W}{\partial x_{0,0}} = y_{1,0}z_{1,1} - y_{0,1}z_{1,0}, \quad \frac{\partial W}{\partial x_{0,1}} = -y_{0,0}z_{1,1} + y_{1,1}z_{1,0}.$$

The Jacobian ideal is thus generated by a binomial system.

In physics, the corresponding common zero set of the Jacobian ideal can provide insight to the physical theory behind it. It is therefore important to describe the solution set of the binomial system

$$\begin{cases} \frac{\partial W}{\partial x_{i,j}} & = 0 \\ \frac{\partial W}{\partial y_{i,j}} & = 0 \\ \frac{\partial W}{\partial z_{i,j}} & = 0 \end{cases} \tag{22}$$

for all $i \in \mathbb{Z}_k$ and $j \in \mathbb{Z}_m$. For the case $k = m = 2$ mentioned above, (22) consists of a binomial system of 12 equations in 12 variables. In general, (22) consists of a binomial system of $3mk$ equations in $3mk$ variables.

In [22, 33], many computational tools, from purely symbolic methods (via Macaulay2 [19]) to numerical homotopy continuation methods (using Bertini [3] and Hom4PS-2.0 [28]), have been applied to find the structure of the zero set of the Jacobian ideals. Upon physical needs, the focus of the computation is often concentrated on the dimension of the solution set in $(\mathbb{C}^*)^n$, the number of components, and the degree of each component. From mathematical points of view, the global parametrization of the solution sets also provides deeper insight as well.

Notice that the power of each variable is one in each term of the superpotential W. This special structure makes the Smith Normal Form of the matrices involved fairly easy to compute in the process of solving the binomial systems. Indeed, for any pair (m, k) with $m, k = 1, \ldots, 8$ (except for $(m, k) = (1, 1)$ for which the corresponding system is trivial), the Smith Normal Forms of the matrices involved can be computed within a range from fractions of a second to one minute on a modern workstation. The computation of the degree of a solution component, however, is the most difficult part. As mentioned in Section 4, our new strategy for the degree computation is to calculate the degree via the computation of the BKK bound of the corresponding Laurent polynomial system, for which an efficient software package MixedVol-3 (developed by T. Chen, T.L. Lee and T.Y. Li) is used. As shown in Table, 3 MixedVol-3 is capable of computing the degree of the solution set of (22) for many pairs of $m, k \in \mathbb{N}$ within a reasonable amount of time: 5 hours. The table shows the degree information in the first place. For example, the number 92 on the third column of fourth row in the table indicates that the corresponding solution set (for $k = 2$ and $m = 3$) is of degree 92. Here the time limit of 5 hours is chosen because some of the more challenging cases computed in [22, 33] would consume at least 5 hours using the similar hardware. Secondly the blank entries in the table corresponds to the cases whose run time exceeds 5 hours on the same hardware platform. This run time, of course, depends on the hardware: with more powerful processor (and more processor cores), more entries would be computed within 5 hours. By using a computer cluster in the computational experiment as shown in Section 6.2 (Figure 1), the case with $m = k = 8$ can be computed well within 5 hours. The results in Table 3 exhibits a substantial expansion of the cases computed in [22, 33] in the sense that much more cases are now computable within the same amount of time restrictions on a single computer. Section 6 will discuss the parallel potential of degree computation via this approach in more detail.

Table 3. The degree of the solution set defined by the system (21) in $(\mathbb{C}^*)^n$ for given m and k values. This table lists the results that can be computed within 5 hours on a workstation with quad-core Intel Xeon 2.4Ghz CPU

m/k	1	2	3	4	5	6	7	8
1	N/A	2	4	8	16	32	64	128
2	2	14	92	584	3,632	22,304	135,872	823,424
3	4	92	1,620	26,762	437,038	7,029,180		
4	8	584	26,762	1,169,876	50,467,100			
5	16	3,632	437,038	50,467,100				
6	32	22,304	7,029,180					
7	64	135,872						
8	128	823,424						

6. Parallel implementation of the binomial system solver

From applied sciences, there is no shortage of the demand of solving larger and larger binomial systems. For instance, electing $k = 10$ and $m = 10$ in (21) already results in a binomial system of 300 equations in 300 variables. To solve very large binomial systems, the parallelization of the algorithms described in previous sections becomes inevitably essential. It turns out that each individual part of the computation in those algorithms can all be carried out in parallel efficiently.

Among those algorithms, there are three sets of computations that are most time consuming:

1. The computation of the Smith Normal Form.
2. The computation of the degree.
3. The numerical path tracking algorithms for solving binomial systems with approximated coefficients via homotopy continuation method.

We shall discuss the parallelization of those computations in this section.

6.1. Computing Smith Normal Form using GPU

For the first set, the basic algorithm for computing the Smith Normal Form uses successive row and column reductions of the input matrix (as listed in [8, Algorithm 2.4.14] and [18, Section 8.5.1]). While the computation uses only integers, rather than floating point numbers, in terms of organization of the algorithm it is similar to many matrix reduction algorithms appeared in numerical linear algebra. It is quite well known in the community that such algorithms can be modified, with relative ease, to take advantage of *multi-core* processors or any shared-memory parallel computation architectures [10]. There are, of course, many other parallel computation architectures. In particular, an exciting trend in computing is the use of the powerful GPUs (Graphic Processing Units) for general purpose parallel computation. Fortunately, parallel computing with GPU fits our computation of the Smith Normal Form nicely. The latest version of Hom4PS-3[6] contains a preliminary GPU-based implementation for computing the Smith Normal Form, as a part of its binomial system solver, to take advantage of the immense floating point computation power of modern GPUs. In this implementation each row or column reduction is carried out in parallel following the "single-instruction-multiple-data" paradigm that is based on CUDA, a mature GPU computation framework developed by NVidia. Without elaborating the details of the implementation, we simply note here that the memory organization and *warp*

structure of threads are crucial to a successful implementation. In this regard, more details can be found in the documentation for Hom4PS-3 [6].

As shown in Table 4, for sufficiently large binomial systems, the GPU-based implementation can be substantially faster than the implementation that uses CPU only. For example, when the number of variables reaches 2700, the use of GPU delivers more than 20 times the performance of a CPU-only approach!

Table 4. The number of seconds it takes for each of the two implementations to solve the (22) system as the number of variables grows: The second column shows the results using CPU only and the third column shows the results when GPU carries out most of computation intensive tasks via CUDA. The speedup ratio is computed as the quotient between the time consumption of the CUDA-based (GPU) implementation and the CPU-based implementation

Number of variables	CPU only[1]	With GPU[2]	Speedup ratio (GPU over CPU)
75	0.09s	1.04s[3]	0.09
300	0.12s	1.06s	0.11
675	0.58s	1.09s	0.53
1200	7.09s	1.29s	5.50
2700	71.08s	3.19s[4]	22.28
4800	440.22s	21.58s	20.40
7500	7702.39s	259.59s[5]	29.67

[1] In the CPU only computation, a single Intel Xeon 2.4GHz processor is used.

[2] In the GPU-assisted computation, most of the computationally intensive tasks are carried out by a single NVidia GTX 780 graphic card.

[3] For such a small system most of the time are spent on initializing the CUDA runtime library (the library responsible for setting up the GPU computation environment), transferring data between main memory (used by the CPU) and "device memory" (accessible by the GPU), and resource management.

[4] For cases with up to 2700 variables, there are only relatively small increase in the running time. This phenomenon really showcases the great potential for "data-parallelism" on GPU. The GPU has a large number of processor cores, called CUDA cores, that are capable of performing computation simultaneously on a large amount of data. As a result, the increase in the amount of input data simply puts more CUDA cores to work without consuming more time.

[5] When the number of variables reaches 7500, the work load must be divided into multiple stages due to internal restriction of the particular hardware used. However, this work load division itself consume minimal extra time.

6.2. Parallel degree computation

The second set of expensive computations in computing the degree of solution components of a given binomial system in $(\mathbb{C}^*)^n$ can also be executed in parallel efficiently. Recall that our strategy for the degree computation is to compute BKK bound of corresponding Laurent polynomial system instead, and distinguished numerical results were obtained by employing our

software package MixedVol-3 for this purpose. Actually, the software package MixedVol-3 is capable of carrying out the computation on a wide range of parallel architectures including *multi-core* architecture, computer clusters, and distributed environments with great efficiency and scalability [7]. As demonstrated by Table 3 in Section 5.2, the parallel version of MixedVol-3, when used on a multi-core system, was able to greatly expand the range of cases computable within reasonable time restriction. When used on computer cluster, though with a higher cost for communication between nodes over network, the approach taken by MixedVol-3 also shows promising results. Figure 1 presents the (absolute) speedup ratio for computing the degree of the solution set of (22) in $(\mathbb{C}^*)^n$ for the case $m = k = 8$ (a binomial system of 192 equations in 192 variables) as a growing number of nodes were employed in a computer cluster.

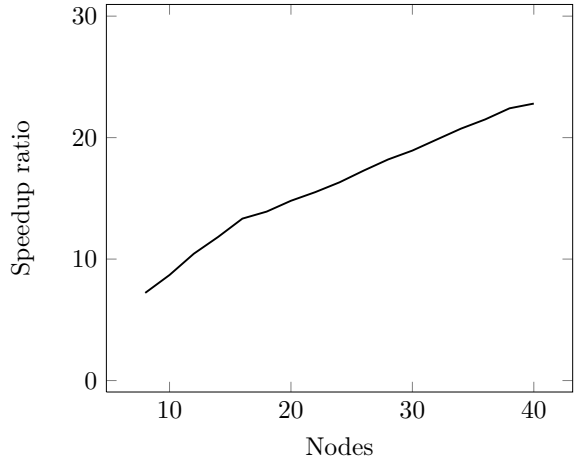

Figure 1. The speedup ratio achieved when MixedVol-3 is used to compute the degree of the solution set of (22) (in $(\mathbb{C}^*)^n$) for the case $m = k = 8$ (a binomial system of 192 equations in 192 variables) as a growing number of nodes were employed in a computer cluster. The speedup ratio for n nodes is computed as the quotient between the time consumed by MixedVol-3 when only one processor is used (serial computation) and the time consumed when n nodes are used

6.3. Parallel path tracking for homotopy continuation method

For the last set of expensive computations, one tracks solution paths defined by $H(\boldsymbol{x}, t) = \mathbf{0}$ as given in (13) from $t = 0$ to $t = 1$, converging to the desired solutions of a given binomial system with approximated coefficients. One great benefit of the homotopy continuation method is that it is *pleasantly*

parallel in the sense that each solution path can be tracked independently from one another. So a serial path tracking algorithm can be easily extended to exploit such "path-level" parallelism. The software package Hom4PS-3 implements this parallel algorithm specialized for solving binomial systems. Here we simply refer to the documentation of Hom4PS-3 for the implementation detail.

To illustrate the parallel potential of this approach with an example, consider the example shown in (12), a system of 6 binomial equations in 6 variables. In total 6910 solution paths (defined by (13)) are to be tracked which will produce the solutions of the system as end points. While a single processor is still capable of computing all the 6910 solutions, utilizing parallel computing hardware, the computation can be completed substantially faster. Table 5 shows speedup ratio achieved when Hom4PS-3 is used on a multi-core workstation with 12 processor cores to solve the system (12) in $(\mathbb{C}^*)^6$. Clearly shown in the table is the nearly linear speedup ratio, i.e., using n processor cores makes the computation nearly n times as fast.

Table 5. The speedup ratio achieved when Hom4PS-3 is used on a computer with 24 processor cores (AMD Opteron 6176 2.3GHz) to solve the system (12). The speedup ratio for n cores is computed as the quotient between the time consumed by the solver when only one processor core is used (serial computation) and the time consumed by the solver when n cores are used

Cores	4	5	6	7	8	10	12	16	20	24
Speedup ratio	3.9	4.9	5.8	6.8	7.7	9.5	11.5	15.0	18.0	22.2

References

[1] Allgower E., Georg K., *Introduction to numerical continuation methods*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[2] Barber C.B., Dobkin D.P., Huhdanpaa H., *The quickhull algorithm for convex hulls*, ACM Trans. Math. Software **22** (1996), 469–483.

[3] Bates D.J., Hauenstein J.D., Sommese A.J., Wampler C.W., *Numerically Solving Polynomial Systems with Bertini*, Society for Industrial and Applied Mathematics, Philadelphia, 2013.

[4] Bernshtein D.N., *The number of roots of a system of equations*, Funct. Anal. Appl. **9** (1975), 183–185.

[5] Büeler B., Enge A., Fukuda K., *Exact volume computation for polytopes: A practical study*, 1998.

[6] Chen T.R., Lee T.L., Li T.Y., *Hom4PS-3: an numerical solver for polynomial systems using homotopy continuation methods.* Available at `http://www.hom4ps3.org`.

[7] Chen T.R., Lee T.L., Li T.Y., *Mixed volume computation in parallel*, Taiwanese J. Math. **18** (2014), 93–114.

[8] Cohen H., *A course in computational algebraic number theory*, Springer-Verlag, Berlin, 1993.

[9] Cox D.A., Little J.B., Schenck H.K., *Toric varieties*, American Mathematical Society, Providence, 2011.
[10] Donfack S., Dongarra J., Faverge M., Gates M., Kurzak J., Luszczek P., Yamazaki I., *On algorithmic variants of parallel Gaussian elimination: Comparison of implementations in terms of performance and numerical properties*, Tech. Rep. 280, LAPACK Working Note (2013). Available at `http://www.netlib.org/lapack/lawnspdf/lawn280.pdf`.
[11] Drexler F.J., *Eine methode zur berechnung sämtlicher lösungen von polynomgleichungssystemen*, Numer. Math. **29** (1977), 45–58.
[12] Eisenbud D., Sturmfels B., *Binomial ideals*, Duke Math. J. **84** (1996), 1–46.
[13] Emiris I.Z., Canny J.F., *Efficient incremental algorithms for the sparse resultant and the mixed volume*, J. Symbolic Comput. **20** (1995), 117–149.
[14] Enge A., Büeler B., Fukuda K., *VINCI: an easy to install C package that implements the state of the art algorithms for volume computation*. Available at `http://www.math.u-bordeaux1.fr/~aenge/?category=software&page=vinci`.
[15] Fulton W., *Introduction to toric varieties*, Princeton University Press, Princeton, 1993.
[16] Gao T., Li T.Y., Wu M., *Algorithm 846: MixedVol: a software package for mixed-volume computation*, ACM Trans. Math. Software **31** (2005), 555–560.
[17] Garcia C.B., Zangwill W.I., *Finding all solutions to polynomial systems and other systems of equations*, Math. Program. **16** (1979), 159–176.
[18] Gockenbach M.S., *Finite-dimensional linear algebra*, CRC Press, Boca Raton, 2011.
[19] Grayson D.R., Stillman M.E., *Macaulay2, a software system for research in algebraic geometry*. Available at `http://www.math.uiuc.edu/Macaulay2/`.
[20] Gunji T., Kim S., Kojima M., Takeda A., Fujisawa K., Mizutani T., *PHoM–a polyhedral homotopy continuation method for polynomial systems*, Computing **73** (2004), 57–77.
[21] Hartshorne R., *Algebraic geometry*, Springer-Verlag, Berlin, 1977.
[22] He Y.H., Candelas P., Hanany A., Lukas A., Ovrut B., *Computational algebraic geometry in string and gauge theory*, Adv. High Energy Phys. 2012, Art. ID 431898, 4 pp.
[23] Huber B., Sturmfels B., *A polyhedral method for solving sparse polynomial systems*, Math. Comp. **64** (1995), 1541–1555.
[24] Jensen A.N., *Gfan, a software system for Gröbner fans and tropical varieties*. Available at `http://home.imf.au.dk/jensen/software/gfan/gfan.html`.
[25] Kahle T., *Decompositions of binomial ideals*, Ann. Inst. Statist. Math. **62** (2010), 727–745.
[26] Kushnirenko A.G., *A Newton polyhedron and the number of solutions of a system of k equations in k unknowns*, Usp. Math. Nauk **30** (1975), 266–267.
[27] Lee T.L., Li T.Y., *Mixed volume computation in solving polynomial systems*, Contemp. Math **556** (2011), 97–112.
[28] Lee T.L., Li T.Y., Tsai C.H., *HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method*, Computing **83** (2008), 109–133.
[29] Lee T.L., Li T.Y., Tsai C.H., *HOM4PS-2.0 para: Parallelization of HOM4PS-2.0 for solving polynomial systems*, Parallel Comput. **35** (2009), 226–238.
[30] Li T.Y., *Numerical solution of polynomial systems by homotopy continuation methods*, in: *Handbook of Numerical Analysis*, ed. P.G. Ciarlet, vol. 11, North-Holland, Amsterdam, 2003, pp. 209–304.
[31] Li T.Y., Sauer T., Yorke J., *The cheater's homotopy: an efficient procedure for solving systems of polynomial equations*, SIAM J. Numer. Anal. (1989), 1241–1251.
[32] Li T.Y., Wang X., *The BKK root count in $\mathbb{C}^n$*, Math. Comput. **65** (1996), 1477–1484.
[33] Mehta D., He Y.H., Hauenstein J., *Numerical algebraic geometry: a new perspective on gauge and string theories*, J. High Energy Phys. **18** (2012), 1–32.
[34] Miller E., Sturmfels B., *Combinatorial commutative algebra*, Springer-Verlag, New York, 2005.

[35] Mizutani T., Takeda A., *DEMiCs: A software package for computing the mixed volume via dynamic enumeration of all mixed cells*, in: *Software for Algebraic Geometry*, Springer, New York, 2008, pp. 59–79.
[36] Morgan A.P., Sommese A.J., *Coefficient-parameter polynomial continuation*, Appl. Math. Comput. **29** (1989), 123–160.
[37] Sommese A.J., Wampler C.W., *The numerical solution of systems of polynomials arising in engineering and science*, World Scientific Press, Singapore, 2005.
[38] Verschelde J., *Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Software **25** (1999), 251–276.
[39] Zhuang Y., *Parallel Implementation of Polyhedral Homotopy Methods*, Ph.D. thesis, University of Illinois at Chicago 2007.

Department of Mathematics
Michigan State University
East Lansing, MI 48823
e-mail: chentia1@msu.edu
e-mail: li@math.msu.edu