Review Article

Numerical Polynomial Homotopy Continuation Method and String Vacua

Dhagash Mehta

Physics Department, Syracuse University, Syracuse, NY 13244, USA

Correspondence should be addressed to Dhagash Mehta, dbmehta@syr.edu

Finding vacua for the four-dimensional effective theories for supergravity which descend from flux compactifications and analyzing them according to their stability is one of the central problems in string phenomenology. Except for some simple toy models, it is, however, difficult to find all the vacua analytically. Recently developed algorithmic methods based on symbolic computer algebra can be of great help in the more realistic models. However, they suffer from serious algorithmic complexities and are limited to small system sizes. In this paper, we review a numerical method called the numerical polynomial homotopy continuation (NPHC) method, first used in the areas of lattice field theories, which by construction finds *all* of the vacua of a given potential that is known to have only isolated solutions. The NPHC method is known to suffer from no major algorithmic complexities and is *embarrassingly parallelizable*, and hence its applicability goes way beyond the existing symbolic methods. We first solve a simple toy model as a warm-up example to demonstrate the NPHC method at work. We then show that all the vacua of a more complicated model of a compactified M theory model, which has an $SU(3)$ structure, can be obtained by using a desktop machine in just about an hour, a feat which was reported to be prohibitively difficult by the existing symbolic methods. Finally, we compare the various technicalities between the two methods.

1. Introduction

A lot of current research in string phenomenology is focused on developing methods to find and analyze vacua of four-dimensional effective theories for supergravity descended from flux compactifications. Stated in explicit terms, one is interested in finding all the vacua (usually, isolated stationary points) of the scalar potential V of such a theory. In particular, given a Kähler potential K and a superpotential W, for uncharged moduli fields, the scalar potential is given by

$$V = e^K \left[K^{A\overline{B}} D_A W D_{\overline{B}} \overline{W} - 3|W|^2 \right], \tag{1.1}$$

where D_A is the Kähler derivative $\partial_A + \partial_A K$ and $K^{A\overline{B}}$ is the inverse of $K_{A\overline{B}} = \partial_A \partial_{\overline{B}} K$. Once the vacua are found, one can then classify them by either using the eigenvalues of the Hessian matrix of V or by introducing further constraints such as $W = 0$.

Finding all the stationary points of a given potential V, amounts to solving the stationary equations, that is, solving the system of equations consisting of the first derivatives of V, with respect to all the fields, equated to zero. The stationary equations for V arising in the string phenomenological models are usually nonlinear. In the perturbative limit, W usually has a polynomial form. This is an important observation since we can then use the algebraic geometry concepts and methods to extract a lot of information about V. Solving systems of nonlinear equations is usually a highly nontrivial task. However, if the system of stationary equations has polynomial-like nonlinearity, then the symbolic methods based on the Gröbner basis technique can be used to solve the system [1]. These symbolic methods ensure that all the stationary points are obtained when the computation finishes. Roughly speaking, for a given system of multivariate polynomial equations, a set of which is called an ideal, the so-called Buchberger Algorithm (BA) or its refined variants can compute a new system of equations, called a Gröbner basis [2]. For the systems known to have only isolated solutions, called 0-dimensional ideals, a Gröbner basis always has at least one univariate equation and the subsequent equations consist of increasing number of variables, that is, it is in a *triangular form* (Note that this is only true for a few specific types of monomial orderings. For other monomial ordering, the new system of equations may not have a triangular form.) The solutions of a Gröbner basis are always the same as the original system, but the former is easier to solve due to its triangular form as the univariate equation can be solved either analytically or numerically quite straightforwardly. Then by back-substituting the solutions in the subsequent equations and continually solving them we can find all the solutions of the system. (Using the Gröbner basis methods, one can also deal with systems which have at least one free variable, called positive dimensional ideals. However, in this paper we only focus on the 0-dimensional ideals.) It should be noted that the BA reduces to Gaussian elimination in the case of linear equations, that is, it is a generalization of the latter. Similarly it is also a generalization of the Euclidean algorithm for the computation of the Greatest Common Divisors of a univariate polynomial. Recently, more efficient variants of the BA have been developed to obtain a Gröbner basis, for example, F4 [3], F5 [4], and Involution Algorithms [5]. Symbolic computation packages such as Mathematica, Maple, and Reduce, have built-in commands to calculate a Gröbner basis. Singular [6], COCOA [7], and MacCaulay2 [8] are specialized packages for Gröbner basis and Computational Algebraic Geometry, available as freeware. MAGMA [9] is also such a specialized package available commercially.

In [10–12], it was shown that one does not need to solve the system using the Gröbner basis techniques, in the usual sense, in order to extract some of the important information such as the dimensionality of the ideal and the number of real roots in the system. But one can indirectly obtain this information by computing the so-called primary decomposition of the ideal (still using the Gröbner basis technique internally). This was a remarkable success as it allowed one to work on nontrivial models and extract a lot of information using a regular desktop machine only. The authors of these papers also made a very helpful computational package, called Stringvacua [11], publicly available. Stringvacua is a Mathematica interface to Singular and has string phenomenology-specific utilities which makes the package quite useful to the users.

However, even with such tricks, there are a few problems with the symbolic methods: the BA is known to suffer from exponential space complexity, that is, the memory (Random Access Memory) required by the machine blows up exponentially with the number of

variables, equations, terms in each polynomial, and so forth. So even for small sized systems, one may not be able to compute a Gröbner basis, nor the related objects such as primary decomposition of the ideal. It is also usually less efficient for systems with irrational coefficients. Another drawback is that the BA is highly sequential, that is, very difficult to efficiently parallelize.

Below we explain a novel numerical method, called the numerical polynomial homotopy continuation (NPHC) method, which overcomes all the shortcomings of the Gröbner basis methods. The method was first introduced in particle physics and condensed matter theory areas in [13–15], where all the stationary points of a multivariate function called the lattice Landau gauge fixing functional [16–19] were found using the NPHC method. Below, we begin by describing the NPHC method for the univariate case before generalizing it to the multivariate case. We then consider a toy model that is used in the Stringvacua manual and also a compactified M theory model. Finding all the vacua using the symbolic methods for both these models is already known to be prohibitively difficult. We briefly describe the models and explain how the corresponding stationary equations can be viewed as having polynomial form. With the help of the NPHC method, we find all the isolated vacua for the model and give a technical comparison between both the symbolic and numerical methods. After mentioning a few other important aspects of the NPHC method in the Frequently Asked Questions section, we conclude the paper.

2. The Numerical Polynomial Homotopy Continuation Method

Here, we explain the numerical polynomial homotopy continuation method. Let us begin by exemplifying the method for the univariate case.

Firstly, we know that for a single-variable equation, $f(x) = \sum_{i=0}^{k} a_i x^i$, with coefficients a_i and the variable x both defined over $\mathbb{C}$, the number of solutions is exactly k if $a_k \neq 0$, counting multiplicities. This powerful result comes from the Fundamental Theorem of Algebra. To get all roots of such single-variable polynomials, there exist many numerical methods such as the *companion matrix* trick for low-degree polynomials and the divide-and-conquer techniques for high-degree polynomials. Here we present the Numerical Polynomial Homotopy Continuation (NPHC) by first describing it for the univariate case which can then be extended to the multivariate case in a straightforward manner. We follow [20, 21] throughout this section unless specified otherwise.

The strategy behind the NPHC method is as follows: first write down the equation or system of equations to be solved in a more general parametric form, solve this system at a point in parameter space where its solutions can be easily found, and finally track these solutions from this point in parameter space to the point in parameter space corresponding to the original system/problem. This approach can be applied to many types of equations (e.g., nonalgebraic equations) which exhibit a continuous dependence of the solutions on the parameters, but there exist many difficulties in making this method a primary candidate method to solve a set of nonalgebraic equations. However, for reasons that will be clear below, this method works exceptionally well for polynomial equations.

To clarify how the method works, we first take a univariate polynomial, say $z^2 - 5 = 0$, pretending that we do not know its solutions (i.e., $z = \pm\sqrt{5}$). We then begin by defining the more general parametric family

$$H(z,t) = (1-t)\left(z^2 - 1\right) + t\left(z^2 - 5\right) = z^2 - (1 + 4t) = 0, \tag{2.1}$$

where $t \in [0, 1]$ is a parameter. For $t = 0$, we have $z^2 - 1 = 0$ and at $t = 1$ we recover our original problem. The problem of getting all solutions of the original problem now reduces to tracking solutions of $H(z, t) = 0$ from $t = 0$ where we know the solutions, that is, $z = \pm 1$, to $t = 1$. The choice of $z^2 - 1$ in (2.1), called the *start system*, should be clear now: this system has the same number of solutions as the original problem and is easy to solve. For multivariate systems, a clever choice of a start system is essential in reducing the computation, and the discussion about this issue will follow soon. Here, we briefly mention the numerical methods used in path-tracking from $t = 0$ to $t = 1$. One of the ways to track the paths is to solve the differential equation that is satisfied along all solution paths, say $z_i^*(t)$ for the ith solution path,

$$\frac{dH(z_i^*(t), t)}{dt} = \frac{\partial H(z_i^*(t), t)}{\partial z} \frac{dz_i^*(t)}{dt} + \frac{\partial H(z_i^*(t), t)}{\partial t} = 0. \tag{2.2}$$

This equation is called the Davidenko differential equation. Inserting (2.1) in this equation, we have

$$\frac{dz_i^*(t)}{dt} = -\frac{2}{z_i^*(t)}. \tag{2.3}$$

We can solve this initial value problem numerically (again, pretending that an exact solution is hitherto unknown) with the initial conditions as $z_1^*(0) = 1$ and $z_2^*(0) = -1$. The other approach is to use Euler's predictor and Newton's corrector methods. This approach works well too. We do not intend to discuss the actual path tracker algorithm used in practice, but it is important to mention that in these path tracker algorithms, almost all apparent difficulties have been resolved, such as tracking singular solutions, multiple roots, solutions at infinity, and so forth. It is also important to mention here that in the actual path tracker algorithms the homotopy is randomly complexified to avoid singularities, that is, taking

$$H(z, t) = \gamma(1 - t)\left(z^2 - 1\right) + t\left(z^2 - 5\right) = 0, \tag{2.4}$$

where $\gamma = e^{i\theta}$ with $\theta \in \mathbb{R}$ chosen randomly.

It is shown that for a generic value of the complex γ the paths are well behaved for $t \in [0, 1)$, that is, for the whole path except the endpoint. This makes sure that there is no singularity or bifurcation along the paths. This is a remarkable trick, called the γ-trick, since this is the reason why we can claim that the NPHC method is guaranteed to find *all* solutions. Note that $\gamma = 1$, for example, is not a generic value.

There are several sophisticated numerical packages well equipped with path trackers such as Bertini [22], PHCpack [23], PHoM [24], HOMPACK [25], and HOM4PS2 [26, 27]. They all are available freely from their respective research groups.

In the above example, the PHCpack with its default settings gives the solutions

$$z = \pm 2.23606797749979 \pm i\, 0.00000000000000. \tag{2.5}$$

Thus, it gives the expected two solutions of the system with a very high numerical precision.

2.1. Multivariate Polynomial Homotopy Continuation

We can now generalize the NPHC method to find all the solutions of a system of multivariate polynomial equations, say $P(x) = 0$, where $P(x) = (p_1(x), \ldots, p_m(x))$ and $x = (x_1, \ldots, x_m)$, that is *known to have isolated solutions* (i.e., a 0-dimensional ideal). To do so, we first need to have some knowledge about the expected number of solutions of the system. There is a classical result, called the *Classical Bèzout Theorem*, that asserts that for a system of m polynomial equations in m variables the maximum number of solutions in $\mathbb{C}^m$ is $\prod_{i=1}^{m} d_i$, where d_i is the degree of the ith polynomial. This bound, called the Classical Bèzout Bound (CBB), is exact for generic values (i.e., roughly speaking, nonzero random values) of coefficients. The *genericity* is well defined, and the interested reader is referred to [21] for details.

Based on the CBB, we can construct a *homotopy*, or a set of problems, similar to the aforementioned one-dimensional case, as

$$H(x,t) = \gamma(1-t)Q(x) + tP(x) = 0, \tag{2.6}$$

where $Q(x)$ is a system of polynomial equations, $Q(x) = (q_1(x), \ldots, q_m(x))$ with the following properties

(1) The solutions of $Q(x) = H(x,0) = 0$ are known or can be easily obtained. $Q(x)$ is called the *start system,* and the solutions are called the *start solutions*.

(2) The number of solutions of $Q(x) = H(x,0) = 0$ is equal to the CBB for $P(x) = 0$.

(3) The solution set of $H(x,t) = 0$ for $0 \leq t \leq 1$ consists of a finite number of smooth paths, called homotopy paths, each parametrized by $t \in [0,1)$.

(4) Every isolated solution of $H(x,1) = P(x) = 0$ can be reached by some path originating at a solution of $H(x,0) = Q(x) = 0$.

We can then track all of the paths corresponding to each solution of $Q(x) = 0$ from $t = 0$ to $t = 1$ and reach $P(x) = 0 = H(x,1)$. By implementing an efficient path tracker algorithm, we can get all the isolated solutions of a system of multivariate polynomials just as in the univariate case.

The homotopy constructed using the CBB is called the *Total Degree Homotopy*. The start system $Q(x) = 0$ can be taken, for example, as

$$Q(x) = \begin{pmatrix} x_1^{d_1} - 1 \\ x_2^{d_2} - 1 \\ \vdots \\ x_m^{d_m} - 1 \end{pmatrix} = 0, \tag{2.7}$$

where d_i is the degree of the ith polynomial of the original system $P(x) = 0$. Equation (2.7) can be easily solved and its total number of solutions (the start solutions) is $\prod_{i=1}^{m} d_i$, all of which are nonsingular. The Total Degree Homotopy is a very effective and popular homotopy whose variants are used in the actual path trackers.

For the multivariate case, a solution is a set of numerical values of the variables which satisfies each of the equations within a given tolerance, Δ_{sol} (~10^{-10} in our set up). Since the variables are allowed to take complex values, all the solutions come with real and imaginary parts. A solution is a real solution if the imaginary part of each of the variables is less than or equal to a given tolerance, $\Delta_{\mathbb{R}}$ (~10^{-7} is a suitable choice for the equations we will be dealing with in the next section, below which the number of real solutions does not change). All of these solutions can be further refined to an *arbitrary precision* limited by the machine precision.

The obvious question at this stage would be if the number of real solutions depends on $\Delta_{\mathbb{R}}$. To resolve this issue, we use a recently developed algorithm called alphaCertified which is based on the so-called Smale's α-theory [28]. This algorithm certifies the real nonsingular solutions of polynomial systems using both exact rational arithmetic and arbitrary precision floating point arithmetic. This is a remarkable step, because using alphaCertified we can prove that a solution classified as a real solution is actually a real solution independent of $\Delta_{\mathbb{R}}$, and hence these solutions are as good as the *exact solutions*.

3. A Toy Model

Here, we apply the NPHC method to a toy model from the examples given in the Stringvacua package. The Kähler potential for this model is given as

$$K = -3\log\left(T + \overline{T}\right), \tag{3.1}$$

and the superpotential is given as

$$W = a + bT^8. \tag{3.2}$$

Here, a and b are parameters. Note that the field T comes along with its complex conjugate. So even though they can be treated as different variables by merely relabeling them, they are not actually independent variables. To avoid this problem, we can write them in terms of real and imaginary parts, that is, $T = t + i\tau$ with τ, and t are real. Finally, we get the potential as

$$V = \frac{1}{3t}\left(4b\left(5b\left(t^2 + \tau^2\right)^7 - 3a\left(t^6 - 21t^4\tau^2 + 35t^2\tau^4 - 7\tau^6\right)\right)\right), \tag{3.3}$$

which has 2 variables. To find the stationary points of V, we need to solve the system of equations consisting of the first-order derivatives of V, with respect to both variables t and τ, equated to zero, that is,

$$\frac{\partial V}{\partial t} = \frac{1}{3t^2}\left(4b\left(5b\left(13t^2 - \tau^2\right)\left(t^2 + \tau^2\right)^6 - 3a\left(5t^6 - 63t^4\tau^2 + 35t^2\tau^4 + 7\tau^6\right)\right)\right) = 0,$$

$$\frac{\partial V}{\partial \tau} = \frac{1}{3t}\left(56b\tau\left(5b\left(t^2 + \tau^2\right)^6 + a\left(9t^4 - 30t^2\tau^2 + 9\tau^4\right)\right)\right) = 0. \tag{3.4}$$

We also note that the stationary equations in this example involve denominators. Since we are not interested in the solutions for which the denominators are zero, we clear them out by multiplying them with the numerators appropriately.

Using the symbolic methods, this task is known to be difficult for general numerical (i.e., floating points) values of parameters a and b, with the computation continuing indefinitely [1, 12].

Firstly, we used the Stringvacua package to compute the dimension of the ideal which turned out to be 0 for generic values of a and b, that is, the system of equations has only isolated solutions. Note that to actually find the solutions of the system, we have to put some numerical values for a and b. The Gröbner basis techniques, as mentioned above, work much better for the cases where parameters are rational. We first use the same values, $a = 1$ and $b = 1$, as used in the Stringvacua manual. Then, we use the command "NumRoots" which computes the number of real roots of the system, that is, 7 in this case, in less than a minute on a desktop machine.

Let us now turn our attention to solving this system using the NPHC method. Firstly, the CBB for this system is 182. We used both Bertini and HOM4PS2 to track all these paths. Both took around one minute to solve this system: there are 86 complex (including real) finite solutions, out of which 36 solutions are real. Out of the 36 real solutions, six of them are distinct solutions (multiplicity one) and the only other distinct solution $(t, \tau) = (0, 0)$ which comes with multiplicity 30. Thus, there are 7 distinct solutions as expected from the Stringvacua's "NumRoots" command. However, we should mention that the Stringvacua package does not give any information about the multiplicity of the solutions, as seen in this example, whereas the NPHC method gives all the solutions with its multiplicities making the method already useful for this simple example. Not only that, but the NPHC also gives the infinite solutions (which are the solutions on the projective space but not on the affine space): the running example has 2 infinite solutions both coming with multiplicity 48. Thus, the total number of solutions in this case, $50 + 6 + (1 \times 30) + (2 \times 48)$, is indeed the same as the CBB.

Note that in these equations all the denominators were multiples of t. The condition that none of the denominators is zero can be imposed algebraically by adding a constraint equation as $1 - zt = 0$ with z being an additional variable. Thus there are now 3 equations in 3 variables. Note that in the Stringvacua package the denominators are thrown away by multiplying each equation appropriately, but the additional equation is not included in the final ideal. In the package, one can of course use the "Saturation" command in order to ensure that this equation is properly taken into account.

We can again solve the above system 3 equations in 3 variables using the Bertini and HOM4PS2. The CBB of this new system is 364. In the end, there are 56 finite complex solutions out of which there are six real solutions, all with multiplicity 1. There are no infinite solutions in this case. This should be expected since the only multiple real solution in the previous system was when the denominator was zero. After adding the constraint equation, we have got rid of this solution and hence left with the rest of the six distinct solutions. Finally, the real solutions (throwing the very small imaginary parts out) are

$$\{t, \tau\} = \{\{-0.5204819146691344, 0.7148265478403096\},$$

$$\{0.5204819146691421, -0.7148265478403003\},$$

$$\{-0.5204819146691322, -0.7148265478403191\},$$

$$\{0.520481914669129, 0.7148265478403104\},$$

$$\{0.8325249117100803, 0\}, \{-0.8325249117100793, 0\}\}.$$

(3.5)

Since we have all the real solutions, we can now compute the Hessian of V at these solutions and separate out the physically interesting vacua. Since the purpose of this paper is to introduce the NPHC method only, we refrain from discussing the interesting physics of these solutions here. A detailed analysis of these solutions and the solutions of other systems will be published elsewhere. For now we discuss how the two methods, the symbolic algebra methods and the NPHC, compare with each other.

4. A Model of Compactified M Theory

Here, we take an example of M theory compactified on the coset $(SU(3) \times U(1))/(U(1) \times U(1))$ from [29] which is also considered in [12]. The coset has $SU(3)$ structure. The corresponding Kähler and superpotential are

$$K = -4 \log\left(-i\left(U - \overline{U}\right)\right) - \log\left(-i\left(T_1 - \overline{T}_1\right)\left(T_2 - \overline{T}_2\right)\left(T_3 - \overline{T}_3\right)\right),$$

$$W = \frac{1}{\sqrt{8}}(4U(T_1 + T_2 + T_3) + 2T_2 T_3 - T_1 T_3 - T_1 T_2 + 200).$$

(4.1)

Here, we use $T_i = -it_i + \tau_i$, for $i = 1, 2, 3$, and $U = -ix + y$. Then the potential is

$$
\begin{aligned}
V = \frac{1}{256 t_1 t_2 t_3 x^4} &\left(40000 + t_3^2 \tau_1^2 - 400\tau_1\tau_2 - 4t_3^2\tau_1\tau_2 + 4t_3^2\tau_2^2 + \tau_1^2\tau_2^2 - 400\tau_1\tau_3 \right. \\
&+ 800\tau_2\tau_3 + 2\tau_1^2\tau_2\tau_3 - 4\tau_1\tau_2^2\tau_3 + \tau_1^2\tau_3^2 - 4\tau_1\tau_2\tau_3^2 + 4\tau_2^2\tau_3^2 - 24t_2 t_3 x^2 \\
&+ 4t_3^2 x^2 - 24t_1(t_2 + t_3)x^2 + 4\tau_1^2 x^2 + 8\tau_1\tau_2 x^2 + 4\tau_2^2 x^2 + 8\tau_1\tau_3 x^2 + 8\tau_2\tau_3 x^2 \\
&+ 4\tau_3^2 x^2 + 1600\tau_1 y - 8t_3^2\tau_1 y + 1600\tau_2 y + 16t_3^2\tau_2 y - 8\tau_1^2\tau_2 y - 8\tau_1\tau_2^2 y \\
&+ 1600\tau_3 y - 8\tau_1^2\tau_3 y + 16\tau_2^2\tau_3 y - 8\tau_1\tau_3^2 y + 16\tau_2\tau_3^2 y + 16t_3^2 y^2 + 16\tau_1^2 y^2 \\
&+ 32\tau_1\tau_2 y^2 + 16\tau_2^2 y^2 + 32\tau_1\tau_3 y^2 + 32\tau_2\tau_3 y^2 + 16\tau_3^2 y^2 \\
&+ t_1^2\left(t_2^2 + t_3^2 + \tau_2^2 + 2\tau_2\tau_3 + \tau_3^2 + 4x^2 - 8\tau_2 y - 8\tau_3 y + 16y^2\right) \\
&\left.+ t_2^2\left(4t_3^2 + \tau_1^2 - 4\tau_1(\tau_3 + 2y) + 4\left(\tau_3^2 + x^2 + 4\tau_3 y + 4y^2\right)\right)\right).
\end{aligned}
$$

(4.2)

We need to solve the stationary equations, that is, the derivatives of V with respect to $t_1, t_2, t_3, \tau_1, \tau_2, \tau_3, x$, and y equated to zero. We also need to add an additional equation $1 - z(t_1 t_2 t_3 x) = 0$ to ensure that none of the denominators of the stationary equations are zero.

Thus, in total there are 9 equations in 9 variables. This system only has isolated solutions, (In [12], this system is reported to have positive dimensional components in its solution space. However, the denominator equation was not included in the analysis there. Once we include the denominator equation in the system, the combined system has no positive dimensional components. Hence, there is no discrepancy here.) The equations are quite complicated, and we avoid writing all of them down here. This system of equations is not only *prohibitively difficult* to be solved completely but also not tractable even using the primary decomposition techniques (except that some information about the solutions may be obtained if one further restricts the system such as taking $y = 0$) [1, 12]. In short, it is not possible to handle this system in its full glory using the available symbolic methods.

Now, let us move to the NPHC method. Firstly, since there are four equations of degree 3, another four equations of degree 4, and one equation of degree 5, the CBB is 103680. This system is actually quite straightforward to solve using the NPHC method. The HOM4PS2 package, for example, solves the full system in around 1 hour on a regular desktop machine: there are 516 total solutions for this system, out of which there are only 12 real solutions. The solutions in the order $\{y, \tau_1, \tau_2, \tau_3, t_2, t_3, x, t_1\}$ are

$$\{\{-3.3333333333335, 1.3333333333308, 3.333333333331, 3.333333333336,$$
$$-6.6666666666666705, -6.666666666666667, -6.66666666666667, -2.6666666666603\},$$
$$\{-3.333333333334, 1.3333333333288, 3.333333333337, 3.3333333333335,$$
$$6.66666666667, 6.666666666671, -6.666666666669, 2.6666666666594\},$$
$$\{-3.3333333333344, 1.3333333333324, 3.33333333333, 3.3333333333375,$$
$$-6.666666666668, -6.666666666666, 6.666666666667, -2.6666666666634\},$$
$$\{-3.3333333333264, 1.3333333333488, 3.3333333333286, 3.333333333322,$$
$$6.666666666659, 6.66666666666, 6.666666666663, 2.6666666666807\},$$
$$\{3.333333333338, -1.3333333333228, -3.333333333334, -3.333333333342,$$
$$6.666666666674, 6.666666666669, 6.6666666666705, 2.6666666666554\},$$
$$\{3.3333333333286, -1.3333333333406, -3.333333333338, -3.3333333333215,$$
$$6.666666666663, 6.666666666668, -6.666666666665, 2.6666666666714\},$$
$$\{3.3333333333313, -1.3333333333337, -3.3333333333366, -3.3333333333277,$$
$$-6.666666666667, -6.6666666666705, 6.666666666668, -2.666666666663\},$$
$$\{3.3333333333313, -1.3333333333341, -3.3333333333326, -3.333333333335,$$
$$-6.6666666666705, -6.666666666671, -6.666666666666, -2.6666666666634\},$$
$$\{0., 0., 0., 0., 7.453559924993, 7.4535599249993, -7.453559925, 2.9814239699997\},$$
$$\{0., 0., 0., 0., 7.453559924999, 7.4535599249993, 7.453559925, 2.9814239699997\},$$

$$\{0., 0., 0., 0., -7.4535599249992, -7.453559925, -7.4535599249992, -2.9814239699997\},$$

$$\{0., 0., 0., 0., -7.4535599249993, -7.4535599249992, 7.4535599249, -2.9814239699\}\}.$$

$$(4.3)$$

It is easy to recognize that some of the numbers in the above list of solutions are rational numbers, for example, $3.33333 \simeq 10/3$. We can now easily compute the eigenvalues of the Hessian of the potential and other related quantities of all these solutions and hence classify the vacua in terms of physics. However, again we refrain from discussing the interesting physics of these solutions here. The full analysis will be published elsewhere.

5. Comparison between Gröbner Basis Techniques and the NPHC Method

Here, we compare the two different methods. Firstly, the Gröbner basis techniques solve the system symbolically. This is immensely significant since one then has a *proof* for the results and/or the results in closed form. There is caveat here however: if the univariate equation in a Gröbner basis is of degree 5 or higher, then the Abel-Ruffini theorem prevents us from solving it exactly, in general, at least in terms of the radicals of its coefficients (this does not mean that the univariate equation cannot be solved exactly at all). In such a situation, one may end up solving this equation numerically, and hence the above-mentioned feature of the symbolic method no longer applies. The NPHC method is a numerical method. That said, the method by construction gives *all* of the isolated solutions for the system known to have only isolated solutions, up to a numerical precision. The solutions then can be refined to within an arbitrary precision up to the machine precision by the Newton's corrector method or otherwise. Moreover, using the alphaCertified method, we can certify if the real nonsingular solutions obtained by the above packages are actually the real nonsingular solutions of the system independent of the numerical precision used during the computation. Hence, though the solutions cannot be obtained in a closed form using the NPHC method, the solutions are as good as exact solutions for all the practical purposes.

We should emphasize here that using the methods presented in [10–12] one can learn quite a lot about a system without having to necessarily obtain its solutions. In particular, one can use the so-called primary decomposition of the ideal (though making use of the Gröbner basis technique only) to obtain information such as the dimensionality of the solution space and number of isolated real roots. This is indeed a clever way to resolve the above-mentioned issue up to a certain level. However, here, the next difficulty comes in the form of algorithmic complexity. The BA is known to suffer from exponential space complexity, which roughly means that the memory (Random Access Memory) required by the machine blows up exponentially with increasing number of polynomials, variables, monomials, and/or degree of the polynomials involved in the system. Hence, even the computation for the primary decomposition may not finish for large sized systems, whereas the NPHC method is strikingly different from the Gröbner basis techniques in that the algorithm for the former suffers from no known major complexities. Hence one can in principle find all solutions of bigger systems.

The BA is a highly sequential algorithm, that is, each step in the algorithm requires knowledge of the previous one. Thus, although recently there are certain parts of the BA which have been parallelized, in general, it is extremely difficult to parallelize the algorithm.

On the other hand, in the NPHC method, the path tracking is *embarrassingly parallelizable*, because each start solution can be tracked completely independently of the others. This feature along with the rapid progress towards the improvements of the algorithms makes the NPHC well suited for a large class of physical problems arising not only in string phenomenology but in condensed matter theory, lattice QCD, and so forth.

The BA is mainly defined for systems with rational coefficients, while in real-life applications, the systems may have real coefficients. The NPHC method being a purely numerical method by default incorporates floating point coefficients as well.

In conclusion, both the Gröbner basis techniques and the NPHC have advantages and disadvantages. However, for practical purposes, the NPHC method is a far more efficient and promising method for realistic systems.

6. Frequently Asked Questions

In this section, we collect the frequently asked questions and their answers.

(1) What does the NPHC method tell us about systems which do not have any solutions?

As mentioned above, the NPHC method by construction (in conjunction with the γ-trick) gives all real and complex solutions of a system of multivariate polynomial equations that is known to have only isolated solutions. Hence, we are always sure that we have got all the solutions numerically. This statement is true for all cases such as when the system has no complex solutions and/or no real solutions or no solution at all. One possible issue, as mentioned above, regards the classification of the real solutions independent of the tolerance used. This can be resolved by using, for example, the alphaCertified algorithm which certifies when a solution is a real.

(2) For many practical problems, only real solutions are required. Thus, when implementing the NPHC, a huge amount of computational effort is wasted in getting the other types of solutions. Would it not be helpful to track only the real solutions?

It would be much more useful if there was a way of getting only real solutions. However, for a number of technical reasons nicely discussed in [21], a path tracker does not know in advance if a given start solution will end up being a real solution of the original system. Moreover, one can wonder if a root count exists only for the real solutions of a system. This would involve obtaining a corresponding fundamental theorem of algebra on the real space for the multivariate case. This, however, has yet to be achieved. Hence, the best way for now is to track all complex (including real paths) solutions and then filter out the real solutions.

(3) Can the NPHC method be used as a global or local minimization method?

Absolutely, most of the conventional methods used to minimize a function are based on the Newton-Raphson method, where a start solution is guessed and is then refined by successive iterations in the direction of the minima. By performing this algorithm several times on the functions, one can obtain many minima of the given potential. Recently, more efficient methods such as the basin-hopping method are available for local minimization [30]. However, we are never sure if we have got all the minima from any of these methods. For the global minimization, we may

use more efficient methods such as Simulated Annealing and Genetic Algorithm. However, these methods are known to fail for larger systems since it can easily get trapped at a local minimum when trying to find the global minimum. Thus, in addition to the usual error from the numerical precision of the machine, there can be an error of an *unknown* order (i.e., we do not know if the found one is the global minimum!). But if the function has a polynomial-like nonlinearity, in theory the NPHC method can give all the minima since it obtains all the stationary points. So it solves the local minimization problem. Moreover, it is then easy to identify *the* global minimum out of the minima, and hence we are sure that the found one is actually the global minimum.

(4) As in the example system in this paper, for many systems the number of actual solutions may be well below the CBB. Is there any remedy for this issue?

The main reason why the number of actual solutions is less than the CBB for many systems is that the CBB does not take the sparsity (i.e., very few monomials in each polynomial in the system) of the system into account. There is indeed a tighter upper bound on the number of complex solutions, called the Bernstein-Khovanskii-Kushnirenko (BKK) count [20, 21], which takes this sparsity into account and thus in most cases is much lower than the CBB. In many cases, it is in fact equal to the number of solutions. The BKK bound can thus save a lot of computation time since the number of paths to be tracked is less than the CBB. The details on the BKK count relating to string phenomenology problems will be published elsewhere.

(5) Are there any alternative/supplementary numerical methods?

There are not many methods to find the stationary points of a multivariate function around, compared to the number of methods to find minima. One of the methods that can find stationary points is the Gradient-minimization method which finds all the minima of an auxiliary function $E = |\nabla V|^2$ whose minima are the stationary points of V provided we further restrict E to be zero [31]. One can find many minima of E using some conventional minimization method such as the Conjugate Gradient method or the Simulated Annealing method. However, it is known that as the system size increases, the number of minima of E that are not the minima of V, that is, $E > 0$, increases rapidly, making the method inefficient [32]. Another method is the Newton-Raphson method (and its sophisticated variants) [30, 32–34]. There, an initial guess is refined iteratively to a given precision. It should be emphasized, however, that no matter how many different random initial guesses are fed into the algorithm, we can never be sure to get all the solutions in the end, unlike the NPHC method. However, these two methods can be supplementary methods for bigger systems to get an idea on what to expect there.

(6) This paper mainly deals with the potentials having polynomial-like nonlinearity which may be usual in the perturbation limit. What about the fully nonperturbative potentials?

The most interesting application for this method would be in the nonperturbative regime, certainly. This question can be stated in different words: is it possible to translate the stationary equations for the nonperturbative potential (i.e., the potentials which have logarithm and exponential terms), and if so, how? Once we can translate the equations in the polynomial form, we can again use the NPHC method as before. The answer is already available in [10]. In this work, Gray et al.

have already prescribed how to translate the corresponding equations arising in the nonperturbative regime which usually involve logarithms and/or exponentials, by using dummy variables. After that, we can solve the system using algebraic geometry methods, such as the Gröbner basis, or for more complicated cases, the NPHC method presented in this paper. Once we have all the solutions, we can extract the solutions in terms of the original variables which were logarithms and/or exponentials of the fields. This trick makes all the algebraic geometry methods, not only the NPHC method, applicable to finding the vacua of the potentials in the nonperturbative regime.

(7) This method assumes that one knows that the system under consideration has only isolated solutions. But, in general, one may not know if a given system has only isolated solutions or it contains some positive dimensional components. In that case, do not we need to rely on the Gröbner basis techniques only, at least to check the dimension of the system?

Firstly, in many systems, once we add the constraint equation (i.e., the denominators are never zero), they usually turn out to have only isolated solutions. Thus, there are way too many interesting systems in string phenomenology which only have isolated solutions. Of course, there may be many more systems which would have positive dimensional solution components. To solve such systems, there is a recently developed generalization of the numerical homotopy continuation method, called the Numerical Algebraic Geometry method. This method finds out each of the positive dimensional solution components with its dimensionality. This method is also *embarrassingly parallelizable* and hence goes far beyond the reach of the Gröbner basis methods. The details of this method are much more involved and beyond the scope of the present paper. But, in short, to find out the dimensionality of the system we do not necessarily need to rely on the Gröbner basis methods. The details of this method with applications will be published elsewhere.

7. Summary

In this paper, we have reviewed a novel method, called the numerical polynomial homotopy continuation (NPHC) method, which can find all the string vacua of a given potential. It does not suffer from any major algorithmic complexities compared to the existing symbolic algebra methods based on the Gröbner basis techniques, which are known to suffer from exponential space complexity. Moreover, the NPHC method is *embarrassingly parallelizable,* making it a very efficient alternative to the existing symbolic algebra methods. As an example, we studied a toy model and, using the NPHC method, found all the vacua within less than a minute using a regular desktop machine. Note that this system with the irrational coefficients is already a difficult task using the Gröbner basis techniques. In addition to that, using the NPHC method, with just about an hour of computation on a regular desktop machine, we found all vacua of an M theory model compactified on the coset $(SU(3) \times U(1))/(U(1) \times U(1))$, which has an $SU(3)$ structure. This system was reported to be a prohibitively difficult problem using the symbolic method. Thus, we have already shown how efficiently the NPHC method can solve the problems that are yet far beyond the reach of the traditional symbolic methods. We also emphasize that using the procedure prescribed in [10] to translate the stationary equations arising in the nonperturbative regime, by replacing logarithm and exponential terms of the field variables by dummy variables, into the polynomial form, we can use the NPHC method

to find the vacua for the nonperturbative potentials as well. It is this application of the method which makes it quite promising. With the help of the NPHC method it is thus hoped that we can go far beyond the reach of the existing methods and study realistic models very efficiently.
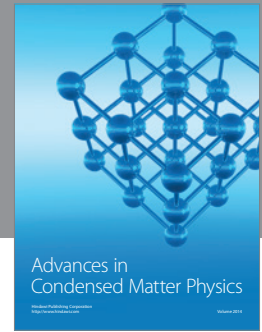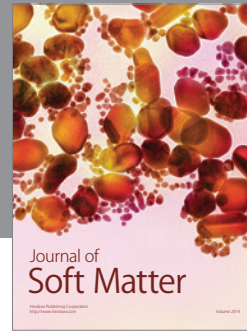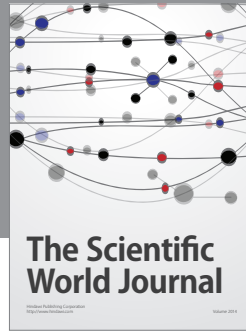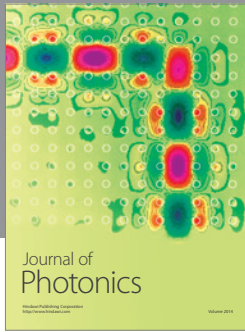
Acknowledgments

References

[1] J. Gray, "A simple introduction to Gröbner basis methods in string phenomenology," *Advances in High Energy Physics*, vol. 2011, Article ID 217035, 12 pages, 2011.

[2] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e*, Undergraduate Texts in Mathematics, Springer, Secaucus, NJ, USA, 3rd edition, 2007.

[3] J. C. Faugère, "A new efficient algorithm for computing groebner bases (f4)," *Journal of Pure and Applied Algebra*, pp. 75–83, 1999.

[4] J. C. Faugère, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)," in *Proceedings of the International Symposium on Symbolic and Algebraic Computation, (ISSAC '02)*, pp. 75–83, New York, NY, USA, July 2002.

[5] V. P. Gerdt, "Involutive algorithms for computinggroebner bases," http://arxiv.org/abs/math/0501111.

[6] G.-M Pfister, G. S. H. Decker, and W. Greuel, "Singular 3-1-3—a computer algebra system for polynomial computations," 2011, http://www.singular.uni-kl.de/.

[7] CoCoA Team, "CoCoA: a system for doing computations in commutative algebra," http://cocoa.dima.unige.it/.

[8] D. R. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry," http://www.math.uiuc.edu/Macaulay2/.

[9] W. Bosma, J. Cannon, and C. Playoust, "The magma algebra system. I. The user language," *Journal of Symbolic Computation*, vol. 24, no. 3-4, pp. 235–265, 1997.

[10] J. Gray, Y.-H. He, A. Ilderton, and A. Lukas, "A new method for finding vacua in string phenomenology," *Journal of High Energy Physics*, no. 7, p. 23, 2007.

[11] J. Gray, Y.-H. He, A. Ilderton, and A. Lukas, "STRINGVACUA. Mathematica package for studying vacuum configurations in string phenomenology," *Computer Physics Communications*, vol. 180, no. 1, pp. 107–119, 2009.

[12] J. Gray, Y.-H. He, and A. Lukas, "Algorithmic algebraic geometry and flux vacua," *Journal of High Energy Physics*, no. 9, p. 31, 2006.

[13] D. B. Mehta, A. Sternbeck, L. von Smekal, and A. G. Williams, "Lattice landau gauge and algebraic geometry," in *Proceedings of the International Workshop on QCD Green's Functions, Confinement, and Phenomenology, (QCD-TNT '09)*, vol. 25, ECT, Trento, Italy, 2009.

[14] D. Mehta, *Lattice vs. Continuum: landau gauge fixing and 't Hooft-Polyakov monopoles*, Ph.D. thesis, The University of Adelaide, Australasian Digital Theses Program, Adelaide, Australia, 2009.

[15] D. Mehta, "Finding all the stationary points of a potential-energy landscape via numerical polynomial-homotopy-continuation method," *Physical Review E*, vol. 84, no. 2, Article ID 025702, 4 pages, 2011.

[16] M. Kastner and D. Mehta, "Phase Transitions Detached from Stationary Points of the Energy Landscape," *Physical Review Letters*, vol. 107, no. 16, Article ID 160602, 5 pages, 2011.

[17] D. Mehta and M. Kastner, "Stationary point analysis of the one-dimensional lattice Landau gauge fixing functional, aka random phase XY Hamiltonian," *Annals of Physics*, vol. 326, no. 6, pp. 1425–1440, 2011.

[18] L. von Smekal, A. Jorkowski, D. Mehta, and A. Sternbeck, "Lattice Landau gauge via stereographic projection," in *Proceedings of the 8th Conference Quark Confinement and the Hadron Spectrum, (CONFINEMENT '08)*, vol. 8, Mainz, Germany, 2008.

[19] L. von Smekal, D. Mehta, A. Sternbeck, and A. G. Williams, "Modified lattice landau gauge," *PoS, LAT*, vol. 382, 2007.

[20] T. Y. Li, "Numerical Solution of Polynomial Systems by Homotopy Continuation Methods," *Handbook of Numerical Analysis*, vol. 11, pp. 209–304, 2003.

[21] A. J. Sommese and C. W. Wampler, II, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*, World Scientific Publishing , Hackensack, NJ, USA, 2005.

[22] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, http://www.nd.edu/~sommese/bertini.

[23] J. Verschelde, "Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation," *ACM Transactions on Mathematical Software*, vol. 25, no. 2, pp. 251–276, 1999.

[24] T. Gunji, S. Kim, M. Kojima, A. Takeda, K. Fujisawa, and T. Mizutani, "PHoM—a polyhedral homotopy continuation method for polynomial systems," *Computing. Archives for Scientific Computing*, vol. 73, no. 1, pp. 57–77, 2004.

[25] A. P. Morgan, A. J. Sommese, and L. T. Watson, "Finding all isolated solutions to polynomial systems using HOMPACK," *Association for Computing Machinery. Transactions on Mathematical Software*, vol. 15, no. 2, pp. 93–122, 1989.

[26] T. Gao, T. Y. Li, and M. Wu, "Algorithm 846: mixed vol: a software package for mixed-volume computation," *Association for Computing Machinery. Transactions on Mathematical Software*, vol. 31, no. 4, pp. 555–560, 2005.

[27] T. L. Lee, T. Y. Li, and C. H. Tsai, "HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method," *Computing. Archives for Scientific Computing*, vol. 83, no. 2-3, pp. 109–133, 2008.

[28] J. D. Hauenstein and F. Sottile, "alphaCertified: certifying solutions to polynomialsystems," http://arxiv.org/abs/1011.1091.

[29] A. Micu, E. Palti, and P. M. Saffin, "M-theory on seven-dimensional manifolds with SU(3) structure," *Journal of High Energy Physics*, no. 5, p. 48, 2006.

[30] D. Wales, *Energy Landscapes : Applications to Clusters, Biomolecules and Glasses (Cambridge Molecular Science)*, Cambridge University Press, New York, NY, USA, 2004.

[31] K. Broderix, K. K. Bhattacharya, A. Cavagna, A. Zippelius, and I. Giardina, "Energy landscape of a Lennard-Jones liquid: statistics of stationary points," *Physical Review Letters*, vol. 85, no. 25, pp. 5360–5363, 2000.

[32] J. P. K. Doye and D. J. Wales, "Saddle points and dynamics of Lennard-Jones clusters, solids, and supercooled liquids," *Journal of Chemical Physics*, vol. 116, no. 9, pp. 3777–3788, 2002.

[33] T. S. Grigera, A. Cavagna, I. Giardina, and G. Parisi, "Geometric approach to the dynamic glass transition," *Physical Review Letters*, vol. 88, no. 5, pp. 555021–555024, 2002.

[34] D. J. Wales and J. P. K. Doye, "Stationary points and dynamics in high-dimensional systems," *Journal of Chemical Physics*, vol. 119, no. 23, pp. 12409–12416, 2003.