Abstract

In this paper, an original reduction algorithm for solving simultaneous multivariate polynomial equations is presented. The algorithm is exponential in complexity, but the well-known algorithms, such as the extended Euclidean algorithm and Buchberger’s algorithm, are superexponential. The superexponential complexity of the well-known algorithms is due to their not being “minimal” in a certain sense. Buchberger’s algorithm produces a Gröbner basis. The proposed original reduction algorithm achieves the required task via computation of determinants of parametric Sylvester matrices, and produces a Rabin basis, which is shown to be minimal, when two multivariate polynomials are reduced at a time. The minimality of Rabin basis allows us to prove exponential lower bounds for the space complexity of an algebraic proof of certification, for a specific computational problem in the computational complexity class \( \text{PSPACE} \), showing that the complexity classes \( \text{PSPACE} \) and \( \mathcal{P} \) cannot be the same. By the same reasoning, it follows that \( \text{Co-\textit{NP}} \neq \text{\textit{NP}} \) and \( \text{Co-\textit{NP}} \neq \mathcal{P} \), and that the polynomial time hierarchy does not collapse. It is also shown that the class \( \text{BPP} \) of languages decidable by bounded error probabilistic algorithms with (probabilistic) polynomial time proofs for the member-
ship of input words is not the same as any one of the complexity classes \( \mathcal{P} \), \( \mathcal{NP} \) and \( \mathcal{Co-NP} \). By relativization of \( \mathsf{BPP} \) with respect to \( \mathcal{P} \) and \( \mathcal{NP} \), it can be again inferred that \( \mathcal{NP} \neq \mathcal{P} \) and that \( \mathcal{Co-NP} \neq \mathcal{P} \).

**Keywords.** Extended Euclidean algorithm; Buchberger’s algorithm; Gröbner basis; Parametric Sylvester matrices; Parametric resolvent; Rabin basis; Minimality; Algebraic complexity theory.

## 1 Introduction

Two prominent methods for reducing simultaneous multivariate polynomial equations are the extended Euclidean algorithm and Buchberger’s algorithm. The reduction can be performed eliminating one variable at a time. If two multivariate polynomials vanish simultaneously, then so does their parametric \( \gcd \), where the parametric \( \gcd \) is an element in the integral domain or the field of rational functions in the remaining variables. However, the converse may not be true: for two multivariate polynomials, if the parametric \( \gcd \) vanishes, for a particular interpretation of variables from the algebraic closure of the ground field without any free variables, then the same cannot be guaranteed for the two multivariate polynomials, whose \( \gcd \) has vanished under the chosen interpretation. The failure of the converse to hold true in general contributes to the superexponential complexity of these two well-known algorithms (Section 2).

Now, for the two multivariate polynomials in discussion to vanish simultaneously, it is both necessary and sufficient that the determinant of the parametric Sylvester matrix, called their resultant with respect to the variable being eliminated, vanishes, for any or some values in the ground field, in which the zeros are being searched for, assuming that neither of the two multivariate polynomials vanishes identically, under the chosen interpretation nullifying the resultant. The equivalence of the vanishing resultant — except when at least one of the two multivariate polynomials identically vanishes, under the interpretation of values to the variables other than the variable being eliminated in the current reduction step — to the sharing of a common zero in the algebraic closure of the ground field without parameters lends us the minimality criterion, for the reduction of the two multivariate polynomials. The entries of the Sylvester matrix being multivariate polynomials themselves, albeit without the variable being eliminated in the current step, the Gaussian elimination procedure cannot be applied, even though the final resultant, which is the determinant of the Sylvester matrix, is the same. The reason for inapplicability of the Gaussian elimination procedure for the determinant of the parametric matrices is that the computations in the intermediate stages might become excessively large, causing superexponential space and time for the completion of its computations, as can
be achieved by simplification of intermediate results. The overall performance of the Gaussian elimination procedure for computation of the parametric resultant may be comparable to the extended Euclidean algorithm or Buchberger’s algorithm, if not worse than either. A step-by-step simplified computation of the resultant that does not run into the space or time explosion problem, which is experienced with the Gaussian elimination method for the parametric matrices, is also presented. The reduced multivariate polynomial basis obtained by taking the resultant for each reduction step is called a Rabin basis, in honour of Professor Michael Oser Rabin, for his profound contributions to computer science (Section 3).

The minimality criterion allows us to derive exponential lower bounds for space requirement for an algebraic proof of certification, for a specific computational problem in two variables with undetermined coefficients over any finite field. The specific computational problem is shown to exist in \( \text{PSPACE} \), by exhibiting an algorithm for solving it, requiring space bounded by a linear expression in the sum of the degrees of the two independent variables and logarithm of the cardinality of the finite field, excluding the space required for the finite field arithmetic operations. It is customarily acknowledged that, for a computational problem to be in \( \text{P} \), a polynomial time algorithm, together with a proof of certification — the proof being bounded in space by a polynomial in appropriate parameter values for the instance — must exist for its correctness of operation. The class of nondeterministic computational or decision problems, for which it is possible to produce machine checkable deterministic proofs, bounded in space by a polynomial for each such specific computational problem, denoted by \( \text{N\text{PSPACE}}_{\text{with proof in PSPACE}} \), relative to any particular fixed system of deductive or symbolic logic, equipped with rules of inference, that might be extensible, is included in \( \text{NP} \), and hence the two complexity classes \( \text{N\text{PSPACE}}_{\text{with proof in PSPACE}} \text{ NP} \) represent the same complexity class. The machine checkable proofs may include references to external facts, the rules of inference may be specialized to a specific computational or decision problem, and the extensibility is the system’s or users’ ability to add more rules perhaps adaptively and / or interactively. The nonexistence of a polynomial time deterministic verification algorithm for a computational problem can be inferred from the nonexistence of a proof of correctness for any such algorithm, for its solution, that is bounded in space by a polynomial in the acceptable parameter values for its instances. A discernment of Herbrand’s theorem, as applied to multivariate polynomials, shows that there cannot be a shorter form for the algebraic proof of certification, for the specific computational problem in \( \text{PSPACE} \), because an immediate reflection shows that the degree of the resultant for an instance to the computational problem under investigation, even when the number variables is only two, is exponential, which must be combined with the fact that further mod and gcd operations may have to be performed, for the completion of the proof. In particular, the algebraic form of the resultant needs to hold for all undetermined coefficients, degrees and field characteristics, for the application of Herbrand’s theorem. Moreover, it is easily possible to assume multivariate polynomials with undetermined coefficients (with more than just
two independent variables and undetermined coefficients) as instances for the specific computational problem in $\mathcal{PSPACE}$, and again invoking Herbrand’s theorem, recursion can be applied, to produce an algebraic proof of correctness for the specific computational problem in discussion. By restricting interpretation of variables to small dimension extension fields, a deterministic algorithm running in linear space (possibly excluding the space required for the finite field arithmetic operations) can be exhibited, for the specific computational problem with generalization to multiple variables. The occurrence of recursion effectively annihilates any little hope of finding a proof of certification bounded in space by a polynomial, for any deterministic algorithm for the specific computational problem in its most generality, even when the interpretation is restricted to small dimension extension fields. In summary, we have to become contented in accepting that $\mathcal{PSPACE}$ cannot be $\mathcal{P}$. In fact, an almighty can be assumed to be capable of guessing the correct answer to the question posed as part of the specific computational problem, but the impossibility of producing a polynomial space proof of certification shows that $\mathcal{PSPACE}$ cannot be $\mathcal{NP}$, either.

By the same reasoning, it follows that $\text{Co-}\mathcal{NP} \neq \mathcal{P}$ and $\text{Co-}\mathcal{NP} \neq \mathcal{NP}$, and that the polynomial time hierarchy does not collapse. In addition to these results, it is also shown that the class $\mathcal{BPP}$ of languages acceptable by bounded error probabilistic algorithms with probabilistic polynomial time proofs for the membership of an input word is not the same as any one of the complexity classes $\mathcal{P}$, $\mathcal{NP}$ and $\text{Co-}\mathcal{NP}$. From the discussions, it follows again that $\mathcal{NP} \neq \mathcal{P}$ and that $\text{Co-}\mathcal{NP} \neq \mathcal{P}$, by relativization of $\mathcal{BPP}$, with respect to $\mathcal{P}$ and $\mathcal{NP}$ (Section 4).

2 Extended Euclidean Algorithm, Buchberger’s Algorithm and Gröbner Basis

Let $\mathbb{F}$ be a field and $\mathbb{F}[x_1, \ldots, x_n]$, for some positive integer $n \geq 2$, be the integral domain of polynomials in $n$ independent variables $x_1, \ldots, x_n$, with coefficients in $\mathbb{F}$. Let $x = (x_1, \ldots, x_n)$, $\alpha(x) = \sum_{i=0}^{d} a_i(x_1, \ldots, x_{n-1})x_i^n$ and $\beta(x) = \sum_{i=0}^{d} b_i(x_1, \ldots, x_{n-1})x_i^n$ be two polynomials in $\mathbb{F}[x_1, \ldots, x_n]$, both of degree $d \geq 1$. It is further assumed that the polynomials $a_i(x_1, \ldots, x_{n-1})$ and $b_i(x_1, \ldots, x_{n-1})$ in $\mathbb{F}[x_1, \ldots, x_{n-1}]$ are all nonzero, and that each requires at least $L_{\text{min}}$ units of space, for $0 \leq i \leq d$, such that they could include more than $L_{\text{min}} \geq 2$ terms with very diverse exponent vectors, so that their products after expansion may contain only insignificantly small number of collision terms, for applying cancellations or simplification of terms, or they may admit succinct representations requiring at least $L_{\text{min}}$ units of space, when their products are not expanded.

The operation of the extended Euclidean algorithm for computation of the parametric $\text{gcd}$ is explained in the sequel. Since the two input polynomials are of the same degree $d$ in $x_n$, an application of two consecutive steps to eliminate the highest degree term, i.e., $x_n^d$, results in two multivariate polynomials of degree $d-1$ each, such that their coefficients would need $2L_{\text{min}}$ units of space. Now,
by induction, an application of two consecutive steps to eliminate \( x^d - i \), from the two multivariate polynomials obtained as the result of the last consecutive pair of steps by eliminating \( x^d - i + 1 \), for \( i = 1, 2, \ldots, d - 1 \), would result in \( 2^i D_{\text{min}} \) units of space, without expansion. Thus, when the products are not expanded, the overall space requirement for the elimination of \( x_n \) is at least \( O(2^d L_{\text{min}}) \).

One more insight is concerning the final degree of any of the variables \( x_i \), for \( 1 \leq i \leq n - 1 \). For simplicity, let the degree of occurrence of the variable \( x_i \), for some fixed index \( i \), where \( 1 \leq i \leq n - 1 \), be \( \delta_i \geq 2 \), for each term occurring as the coefficient of \( x_n \) in either input polynomial. The elimination procedure produces coefficients as multivariate polynomials in \( \mathbb{F}[x_1, \ldots, x_{n-1}] \). Assuming that the occurrence of cancellations while simplifying the computations is a rare event, the degree of occurrence of the variable \( x_i \) in the parametric \( \text{gcd} \) can be lower bounded by \( O(2^d \delta_i) \), for \( 1 \leq i \leq n - 1 \).

Expansion and simplification of the products formed in the intermediate steps might not produce a lot of cancellations, and would only be expected to further blow up the space requirement. To eliminate \( x_n^d \), by a consecutive pair steps, would need at least \( O(L_{\text{min}}^2) \) space, after expansion, and by induction, to eliminate \( x_n^{d-1} \), by a consecutive pair steps, would need at least \( O(L_{\text{min}}^2) \) space, after expansion, for \( i = 1, 2, \ldots, d \), resulting in the overall space requirement of at least \( O(L_{\text{min}}^2) \) space, after expansion. This is the problem that causes the space explosion when the extended Euclidean algorithm is applied, for eliminating the variable \( x_n \), from the two input multivariate polynomials \( \alpha(x) \) and \( \beta(x) \). Most of the zeros of the parametric \( \text{gcd} \) might not lead to the common zeros of \( \alpha(x) \) and \( \beta(x) \).

Buchberger’s algorithm follows closely the operational principle of the extended Euclidean algorithm and, in effect, emulates the latter by considering the exponent vector as a whole, in the sum of terms form. The multivariate polynomials so produced are collected in the Gröbner basis [1], named after the Ph D advisor of the author, presumably connoting Hilbert’s basis theorem.

## 3 Parametric Sylvester Matrix, Parametric Resultant and Rabin Basis

Let \( \alpha(x) \) and \( \beta(x) \) be multivariate polynomials in \( \mathbb{F}[x_1, \ldots, x_n] \), of degrees \( d_{\alpha} \geq 1 \) and \( d_{\beta} \geq 1 \). The Sylvester matrix corresponding to the polynomials \( \alpha(x) \) and \( \beta(x) \), for elimination of the variable \( x_n \), is a \((D \times D)\) matrix, where \( D = d_{\alpha} + d_{\beta} \), with entries either 0 or any of the multivariate polynomials \( a_i(x_1, \ldots, x_{n-1}) \), for \( 0 \leq i \leq d_{\alpha} \), and \( b_j(x_1, \ldots, x_{n-1}) \), for \( 0 \leq j \leq d_{\beta} \). It is assumed that the number of terms in the sum of terms form of expansion of the entries is at most \( L_{\text{max}} \) each for these polynomials. The resultant, denoted by \( \text{Res}(\alpha(x), \beta(x)) \), with respect to the variable \( x_n \), is the determinant of the \( D \times D \) Sylvester matrix. The expansion of the determinant form as the sum of \( D! \) many product terms shows that the number of terms in the resultant can be at most \( D! L_{\text{max}}^D < (D L_{\text{max}})^D \).
Now, comparing with the lower bounds for the number of terms obtained in the case of the extended Euclidean algorithms as in Section 2, it may be found that 
\[ (DL_{\text{max}})D = \frac{D \log_2 L_{\text{min}} (DL_{\text{max}})}{D \log_2 (DL_{\text{max}})} = L_{\text{min}}^{\frac{D \log_2 (DL_{\text{max}})}{D \log_2 L_{\text{min}}}}. \]
With \( d_s = d \) and \( D = 2d \), if \( \frac{2d \log_2 (2dL_{\text{max}})}{\log_2 L_{\text{min}}} \) is much smaller than \( 2^d \), then, clearly, \( \text{Res}(\alpha(x), \beta(x)) \) is much better. The observation holds, even when \( d_s \neq d \).

On the other hand, if the degree of occurrence of a variable \( x_i \) is at most \( \Delta_i \), for a fixed index \( i \), where \( 1 \leq i \leq n - 1 \), in the multivariate polynomial coefficients of the input polynomials, then the degree of occurrence of the variable \( x_i \) in \( \text{Res}(\alpha(x), \beta(x)) \), with respect to the variable \( x_s \), is at most \( D\Delta_i \). With \( D = 2d \), if \( 2d\Delta_i \) is much smaller than \( 2^d \delta_{x_s} \), as found in the previous section, then, again clearly, computation of the parametric resultant yields superior performance compared to the extended Euclidean algorithm.

However, the straightforward expansion of the determinant form results in exponential time complexity, owing to the \( D! \) many terms in the sum. Similarly, the Gaussian elimination method could deliver a worse performance than the extended Euclidean algorithm, because the intermediate results may not collapse into a small number of terms until the final result.

The following propagation of computations of the determinants of smaller dimension square submatrices to larger square submatrices is useful. For the computation of the determinant of a \( D \times D \) matrix, for a large dimension \( D > 1 \), let the determinant \( \det(S_k) \) and inverse \( S_k^{-1} \) of a \( k \times k \) submatrix, \( S_k \), be found, where \( S_k \) is a submatrix of the \( k \times D \) matrix \( P_k \), obtained by collecting the first \( k \) rows of the \( D \times D \) matrix, inductively, for some \( k \geq 2 \), but \( k \leq D - 1 \). Let \( P_{k+1} \) be the \((k + 1) \times D \) matrix obtained by adjoining the next row in the \( D \times D \) matrix to \( P_k \). Assuming that the determinant of the given \( D \times D \) matrix, which is \( \text{Res}(\alpha(x), \beta(x)) \), for the Sylvester matrix, with respect to the variable \( x_s \), does not identically vanish, as an element in the integral domain \( \mathbb{F}[x_1, \ldots, x_{n-1}] \), the rows of the matrix \( P_{k+1} \) are linearly independent over the field of fractions of the integral domain \( \mathbb{F}[x_1, \ldots, x_{n-1}] \). By the equality of the row rank to the column rank, there are \( k + 1 \) linearly independent columns of \( P_{k+1} \). Now, of these linearly independent columns, \( k \) of the columns can be chosen to be those corresponding to the columns of \( S_k \), for the following reason: the columns corresponding to \( S_k \) are linearly independent, by its invertibility, and if every column of \( P_{k+1} \) were a linear combination of the \( k \) columns, corresponding to those of \( S_k \), then the column rank of \( P_{k+1} \) itself would be \( k \). For the Sylvester matrix with respect to the variable \( x_s \), the linear combination is taken over the field of fractions of the integral domain \( \mathbb{F}[x_1, \ldots, x_{n-1}] \). Thus, at any point, if it is not possible to propagate the computation of the determinant from a \( k \times k \) submatrix to \((k + 1) \times (k + 1) \) submatrix, for the reason that the column rank cannot increase, after adjoining any of the remaining \( D - k \) rows to \( P_k \), then the determinant of the given matrix itself vanishes, and, for the Sylvester matrix, with respect to the variable \( x_s \), \( \text{Res}(\alpha(x), \beta(x)) \) itself identically vanishes. Given \( S_k \), \( \det(S_k) \) and \( S_k^{-1} \), the computations required for identifying an appropriate column in \( P_{k+1} \), in order to form the \((k + 1) \times (k + 1) \) matrix \( S_{k+1} \),
and the determinant and the inverse of \( S_{k+1} \), can be performed using standard formulas from matrix algebra. This method of computation of the parametric resultant, \( \text{Res}(\alpha(x), \beta(x)) \), avoids needless space and time explosion, that can be observed in the Gaussian elimination method.

If \( \text{Res}(\alpha(x), \beta(x)) = 0 \), for some interpretation of the variables \( x_i = \xi_i \) in the algebraic closure of \( F \), for \( 1 \leq i \leq n-1 \), and neither of \( \alpha(x) \) and \( \beta(x) \) vanishes identically as respective single variable polynomials in \( x_n \), for the ground instances of \( x_i = \xi_i \), for \( 1 \leq i \leq n-1 \), then \( \alpha(x) \) and \( \beta(x) \) share a common zero in the algebraic closure of \( F \). This property is called the \textit{minimality} of the reduction step, for the elimination of the variable \( x_n \), from the two participating multivariate polynomials. The collection of multivariate polynomials as obtained by computing the parametric resultant of two multivariate polynomials, at a time, with respect to any of the independent variables, is called a Rabin basis.

4 Proof Showing That \( \mathcal{NPSPACE} \) Can Be Neither \( \mathcal{NP} \) Nor \( \mathcal{P} \)

4.1 Proof Showing that \( \mathcal{NPSPACE}_{\text{with proof in PSPACE}} = \mathcal{NP} \)

For the definitions of the computational complexity classes denoted by \( \mathcal{P} \), \( \mathcal{NP} \), \( \mathcal{NPSPACE} \) and \( \mathcal{PSPACE} \), the readers are referred to [5], where \( \mathcal{NPSPACE} \) appears as \( \mathcal{NPS} \) and \( \mathcal{PSPACE} \) as \( \mathcal{PS} \). As an additional complexity class, let \( \mathcal{NPSPACE}_{\text{with proof in PSPACE}} \) be the collection of languages over an alphabet containing at least two symbols, such that for the acceptability of an input word, for each language independently, a nondeterministic requiring space bounded by a polynomial (specific to the particular language) in the string length of the input word, as with \( \mathcal{NPSPACE} \), exists, but with an additional property that a proof of acceptance can be automatically generated with respect to any particular system of logic with its own rules of inference. Some of the rules of inference may be specialized to the particular language. The proof is assumed to be machine checkable, for its validity, in \( \mathcal{PSPACE} \). The time required to check the validity of each step in the proof is taken to be bounded by a fixed, but sufficiently large, constant. The syntax checking of the proof is also assumed to require time bounded by a polynomial in the size of the proof. More specifically, the decision problem of proof checking is required to belong to the complexity class \( \mathcal{P} \). It is easy to see that \( \mathcal{NP} \subseteq \mathcal{NPSPACE}_{\text{with proof in PSPACE}} \), because of the deterministic polynomial time verification condition for the languages in \( \mathcal{NP} \).

For the converse inclusion, let \( \Sigma \) be an alphabet of at least two distinct symbols, and let \( \mathcal{L} \subseteq \Sigma^* \) be a language in \( \mathcal{NPSPACE}_{\text{with proof in PSPACE}} \). By assumption, there is a polynomial \( p_\mathcal{L}(|\omega|) \), for every word \( \omega \in \Sigma^* \), where \( |\omega| \) denotes the string length of \( \omega \), such that whenever \( \omega \in \mathcal{L} \), there is a proof attesting
to this fact, of at most $p_\omega(|\omega|)$ bits of information, relative to a particular fixed system of logic, together with the rules of inference, perhaps specialized for the the language $L$. Now, for every $\omega \in L$, the proof that $\omega$ indeed belongs to $L$ can be guessed, checked for syntactic correctness of the proof, and finally checked for the validity of the proof itself, in overall time bounded by some polynomial in $|\omega|$. For an external user, $p_\omega(|\omega|)$ may still remain oblivious, as the technicality of asserting the membership of $L$ to $\text{NPH} \text{SPACE}$ with proof in $\text{PH} \text{SPACE}$ assumes only its existence.

4.2 A Specific Computational Problem in $\text{PH} \text{SPACE}$ Belonging to Neither $\text{NP}$ Nor $\text{P}$

Let $p$ be a large prime number, and $Z_p$ be the finite field of integers with arithmetic operations mod $p$. Let $m, n \geq 5$ be positive integers and $f(t, x) = \sum_{i=0}^{n-1} a_i(t)x^i + x^n \in Z_p[t, x]$, where $a_i(t) \in Z_p[t]$, for $0 \leq i \leq n - 1$, are nonzero polynomials, with undermined coefficients for the purpose of description of the computational problem, of degree at most $m$ each. For convenience, it is assumed that there is exactly one polynomial of degree $m$ and that all the remaining $n - 1$ polynomials are of degree at most $m - 1$.

The computational question is, “what is the number of distinct values of $t$ in the algebraic closure of $Z_p$ , for which there can be a solution, for $x \in Z_p$ , such that $f(t, x) = 0$?” This question is akin to the problems studied along the lines of [5]. Since the polynomials $a_i(t)$, for $0 \leq i \leq n - 1$, are all of degree at most $m$, each, an immediate observation is to search for values of $t$ in extension fields of $Z_p$ of degree at most $m$. As a refinement based on this observation, the computational task is to enumerate (produce as output) the number of solutions for $t$ in the extension field of degree $d$ over $Z_p$, for each degree $d$, where $1 \leq d \leq m$, such that $f(t, x) = 0$, for some $x \in Z_p$. Obviously, the computational problem is in $\text{PH} \text{SPACE}$. In [2], the authors discuss another similar computational problem, but do not assert whether the problem they study indeed belongs to $\text{PH} \text{SPACE}$. Instead, their contention is restricted to the hardness of solving simultaneous multivariate polynomial equations, in general.

It is easily observed that $g(t) = \text{Res}(f(t, x), x^p - x)$ is a polynomial of degree exactly $m \cdot p$, and does not depend on $n$, by the convenient assumption made. Also, if $g(t) = 0$, neither of $f(t, x)$ and $(x^p - x)$ vanishes, since the leading coefficients of $x$ in both the polynomials are equal to the constant 1. However, the multiplicity of occurrence of a root of $g(t)$ must be accurately accounted for, as may be inferred by observing that the two polynomials $(\phi(t) + x)$ and $(\phi(t) + (\psi(t))^2 + x)$ share a common zero in the algebraic closure of $Z_p$, exactly when $(\psi(t))^2 = 0$, but each such value of $t$ in the algebraic closure of $Z_p$ must be taken into account as occurring with only multiplicity one. Thus, $h(t) = \gcd (g(t), g'(t))$, where $g'(t)$ is the formal derivative of the polynomial $g(t)$, must be computed, and finally, the degree of $t$ in the polynomial $\frac{g(t)}{h(t)}$ yields the answer to the first question. As to the second question, the degree of $\gcd$ of $\frac{g(t)}{h(t)}$ with $(t^p - t)$ yields the number of values of $t$ in the degree $d$ extension.
of \( Z_p \), such that \( f(t, x) = 0 \), for some \( x \in Z_p \), for \( 1 \leq d \leq m \).

It is quite a simple matter to generalize the problem to higher dimensions. Let \( f(x) \in Z_p [x_1, \ldots, x_n] \), for some integer \( n \geq 3 \), but requiring the number of zeros in the extensions of degree at most \( m_i \), larger than 2, for \( 1 \leq i \leq n \), possibly including zeros in \( \prod_{i=1}^{n} GF(p, d_i) \), for all possible index vectors \( (d_1, \ldots, d_n) \), where \( 1 \leq d_i \leq m_i \) and \( GF(p, d_i) \) is the \( d_i \) degree extension of \( Z_p \). It is worth noting that a comparison of \( m_i \) to the degree of occurrence of \( x_i \) is omitted, for the purpose of stating the problem in its most generality. The enumeration problem can be easily shown to be in \( \mathcal{PSPACE} \), because \( m_i \), for \( 1 \leq i \leq n \), are fixed inputs to the instance. If it is required to consider values for \( x_i \) in degree \( d_i \) extensions of \( Z_p \), that are not in any smaller dimension extension, then the condition as to whether or not \( x_i^{p^j} - x_i \neq 0 \) holds, for \( 1 \leq j \leq d_i - 1 \), must be checked for. Now, in the algebraic proof, if such a condition must be expressed, then the principle of inclusion-and-exclusion must also be applied, in addition to the division by \( \text{gcd} \) with derivatives, as may be required, for example, in the expression for the Möbius inversion formula. An algebraic proof of validity of the output of the \( \mathcal{PSPACE} \) algorithm for this problem — i.e., proof of certification for the validity of its output as a specific indicator of an algebraic expression — would definitely need an enormously large space.

By accommodating more variables and equations and raising the same question concerning the number of distinct solutions to the systems of simultaneous multivariate equations, in general, the contention that \( \mathcal{PSPACE} \neq \mathcal{P} \) can be more aptly testified, because the solution space cannot be bounded by a polynomial, disallowing any claim of producing a direct polynomial time algebraic proof, for the validation of the answer produced by a nondeterministic algorithm.

The philosophic question under investigation is whether there can be a shorter form of solution for the computational problem exhibited to be in \( \mathcal{PSPACE} \). The answer to the question is that the algebraic form, depicted as the solution to the problem, holds for all the prime numbers occurring in place of the field characteristics and for all the other indeterminate parameters as part of the problem instances, and hence there cannot be a shorter form, by Herbrand’s theorem. It may be recalled that, in the first illustration with two variables \( t \) and \( x \), the degree of the resultant of \( f(t, x) \) and \( (x^p - x) \) does not depend on the degree \( n \) of occurrence of the variable \( x \) in the polynomial \( f(t, x) \), but there are implicitly infinitely many possible choices for the coefficients of the polynomials \( a_i(t) \), for \( 0 \leq i \leq n - 1 \), for various choices of \( n \) and \( p \), with the numerical value of \( n \) bounded by some polynomial in \( \log(p) \). This situation should not be confused with the way the determinant is computed. In the case of the determinant, there was an easier way to compute it, and in the context of the specific computational problem in \( \mathcal{PSPACE} \), there is simply no alternative algebraic proof attesting the validity of the solution produced by the algorithm requiring space bounded by a polynomial in appropriate values of the parameters of the problem instances.
4.3 A Specific Computational Problem in Co-NP Belonging to Neither N\!P\! nor P

Let \( p \) be a large prime number, and \( \mathbb{Z}_p \) be the finite field of integers with arithmetic operations \( \mod p \). Let \( m, n \geq 5 \) be positive integers and \( f(t, x) = \sum_{i=0}^{n-1} a_i(t)x^i + x^n \in \mathbb{Z}_p [t, x] \), where \( a_i(t) \in \mathbb{Z}_p [t] \), for \( 0 \leq i \leq n - 1 \), are undetermined coefficient polynomials in \( t \), of degree at most \( m \) each.

The computational decision question is whether or not the following holds:
\[ f(t, x) \neq 0, \quad \text{for all } t, x \in \mathbb{Z}_p? \]
If \( f(t, x) = 0 \), for some \( t, x \in \mathbb{Z}_p \), then the problem is in N\!P\!, and if \( f(t, x) \neq 0 \), for every \( t, x \in \mathbb{Z}_p \), then the problem is in Co-N\!P\!. The condition that \( f(t, x) \neq 0 \), for every \( t, x \in \mathbb{Z}_p \), is equivalent to the condition that \( \gcd \) of the three polynomials \( f(t, x) \) and \((x^p - x)\) and \((t^p - t)\) is 1. A proof of certification could be a derivation that the \( gcd \) is indeed 1.

For a single variable polynomial \( \phi(x) \in \mathbb{Z}_p [x] \), by repeated squaring method, \( x^{2i} \mod \phi(x) \), for \( i = 1, 2, 3, \ldots, \lfloor \log_2(p) \rfloor \), can be computed, from which \((x^p - x)\mod \phi(x)\) can also be computed, and eventually, \( \gcd \) of \((x^p - x)\) and \( \phi(x) \) can be shown in a derivation sequence.

However, the same algorithm doesn’t work with two variables, and the following system of simultaneous multivariate polynomial equations must be considered:
\[
\begin{align*}
f(t, ,x) &= 0, \\
(x^p - x) &= 0, \\
(t^p - t) &= 0
\end{align*}
\]
Thus, the resultant of \( f(t, ,x) \) with respect to any of the remaining two polynomials must be computed, and the \( \gcd \) of the resultant and the other equation left out must be computed. This two-step method is necessitated by the fact that the last two polynomial equations, \( (x^p - x) = 0 \) and \( (t^p - t) = 0 \), have no variable in common. The space requirement for the resultant in the first step cannot be bounded by a polynomial in \( \log(p) \).

Thus, in order to testify the condition that \( f(t, x) \), \( (x^p - x) \) and \( (t^p - t) \) have no zeros in common, simultaneously satisfying them, the size of the proof cannot be bounded in space by some polynomial in \( \log(p) \), and hence \( \text{Co-NP} \neq \text{P} \).

By accommodating more variables and equations and raising the same question concerning the nonexistence of solutions to the systems of simultaneous multivariate equations, in general, the contention that \( \text{Co-NP} \neq \text{P} \) can be more aptly testified, because the solution space cannot be bounded by a polynomial in the acceptable parameter values, and in particular, by a polynomial in \( \log(p) \), in the preceding example, disallowing any claim of producing a direct polynomial time algebraic proof, for the validation of the answer produced by any algorithm.

Now, even if a deterministic algorithm for the problem is equipped with nondeterministic choices of size bounded by some polynomial in \( \log(p) \), the size of the proof remains exponential, and hence \( \text{Co-NP} \neq \text{NP} \). The supporting evidence for the problem instance to be not in \( \text{NP} \) is that there is no deterministic proof of certification, bounded in size by some polynomial in the parameter values, even if an assumed deterministic algorithm for it is equipped with the
capability to make nondeterministic choices of size bounded by some polynomial in the acceptable values of the parameters of the problem instance.

Some of the implications of the assertions that $\mathcal{NP} \neq \mathcal{P}$ and that $\text{Co-}\mathcal{NP} \neq \mathcal{NP}$ are that $\mathcal{PSPACE} \neq \mathcal{P}$, $\mathcal{PSPACE} \neq \mathcal{NP}$ and $\mathcal{PSPACE} \neq \text{Co-}\mathcal{NP}$, and that the polynomial time hierarchy does not collapse.

4.4 Computational Complexity of Univariate Polynomial Modular Arithmetic Operations for Polynomials with Sparse Nonzero Coefficients

Let $p$ be a large prime number, $m, n \geq 5$ be positive integers, $a_{k-1}, i_k, b_{i-1}, j_i \in \mathbb{Z}_p \setminus \{0\}$, for $1 \leq k \leq m$ and $1 \leq l \leq n$. Let $f(x) = a_0 + \sum_{k=1}^{m-1} a_k x^k + x^m$ and $g(x) = b_0 + \sum_{l=1}^{n-1} b_l x^l + x^n$. The computational decision question is whether $(f(x))^{p-1} - 1 \cdot (g(x))^{p-1} - 1 = 0$, for all $x \in \mathbb{Z}_p$. The decision problem can also be posed as the question that $\gcd (f(x), g(x)) \neq 0$, for all $x \in \mathbb{Z}_p$, and belongs to the complexity class $\text{Co-}\mathcal{NP}$.

If $g(x)$ is taken to be the polynomial $(x^{p-1} - 1)$ and $a_0 \neq 0$, then the decision problem being discussed reduces to the question as to whether $f(x)$ is not 0, for all $x \in \mathbb{Z}_p$, and becomes equivalent to the question as to whether $\gcd (f(x), (x^{p-1} - 1)) = 1$.

Since the degree $i_m$ of $f(x)$ may be chosen to be arbitrarily large, the computation of $(x^{p-1} - 1) \mod f(x)$ could require space that is exponential in the string length of $f(x)$, due to its sparseness. In the derivation sequence of $\gcd (f(x), (x^{p-1} - 1))$ the coefficients of intermediate polynomials in the proof script can be interpreted as hash functions in the undetermined coefficients $a_k$, for $0 \leq k \leq m - 1$. If each of the hash functions, treated independently, vanishes with only exponentially insignificant probability of only once in $p$ many chances, for any instantiation of the undetermined coefficients $a_k$, for $0 \leq k \leq m - 1$, then the space requirement for the derivation sequence cannot be upper bounded by any polynomial in the string length of the sparse polynomial $f(x)$. This evidence fortifies the contention that $\text{Co-}\mathcal{NP} \neq \mathcal{P}$.

The descriptions of in the next couple of paragraphs are intended to illustrate how to navigate possible shortcomings. The first example illustrates an attempt to generate an instance of $\text{Co-}\mathcal{NP}$, but ends up with an instance in $\mathcal{NP}$, thereby belittling any attempt to claim that there cannot be a shorter proof. In the second example, a more general problem is considered, but the complementary instance cannot be guaranteed to belong to $\mathcal{NP}$, thereby restricting the application of its scope to the contention that $\mathcal{PSPACE} \neq \mathcal{P}$. The first example is rarefied subsequently, in a third paragraph, illustrating how to indirectly show the nonexistence of a shorter proof.

In order to show the existence of sparse polynomials in $\mathbb{Z}_p [x]$, that do not vanish for any interpretation of $x \in \mathbb{Z}_p$, let $\gamma_i, \delta_i, r_i \in \mathbb{Z}_p \setminus \{0\}$ such that $\gcd (r_i, p - 1) > 1$ and $(\gamma_i^{-1} \delta_i)^{\frac{p-1}{r_i} - 1} \neq 1 \mod p$, for $1 \leq i \leq \kappa$, for some fixed small integer $\kappa$, such as about 20. Some of the integers $r_i$ can be
conveniently chosen to be larger than $\frac{p}{2}$, where $1 \leq i \leq \kappa$. The polynomial $\prod_{i=1}^{\kappa} \left( \gamma_i x^{r_i} - \delta_i \right)$, after expansion and simplification applying the rule $x^{p^{-1}} \equiv 1 \mod \left( x^{p^{-1}} - 1 \right)$, is taken to be a polynomial $f(x)$ in the simplified form. It is convenient to choose $r_i$ such that there are ample number of identities of the form $r_i + r_j = r_k + r_l \mod (p - 1)$, for some distinct indexes $i, j, k, l$, and more generally, the indexes may be carefully selected with an objective of keeping the number of terms in the polynomial $f(x)$ small. However, the instances of $f(x)$ as constructed may also be shown to belong to the complexity class $\mathcal{NP}$, as well, by observing that the hidden parameters may be guessed and substituted in the corresponding expressions, to reconstruct the polynomial $f(x)$. In this case, there is a possibility of claiming the existence of a shorter proof to the effect of showing that $f(x) \neq 0$, for every $x \in \mathbb{Z}_k$, by checking the derivation of $f(x)$ as assumed. The attempts in the preceding subsections are intended to somehow indirectly show that this hope of generating an easy proof does not exist.

One more passing insight is that if the question in this section was to show $\gcd \left( f(x), g(x) \right) = 1$, then it is not easy to show that the problem belongs to $\text{Co-}\mathcal{NP}$, because the complementary problem of checking whether $f(x)$ and $g(x)$ have a nontrivial common factor, or equivalently, whether they share a common zero in the algebraic closure of $\mathbb{Z}_k$, is not guaranteed to be in $\mathcal{NP}$. The irreducible common factors of $f(x)$ and $g(x)$ may be of large degree, such as comparable to some constant fraction of $p$. Thus, in this case, the contention is restricted to $\mathcal{PSPACE} \neq \mathcal{P}$.

The first example is rarefied in this paragraph. Let $f(x)$ be a sparse polynomial of arbitrary large degree, satisfying the Eisenstien’s criterion [6], for irreducibility over the field of rational numbers. Constructively, let $f(x) = a_0 + \sum_{k=1}^{m-1} a_k x^{i_k} + x^{i_m}$, such that $1 \leq i_1 < \ldots < i_m < p - 1$, where the coefficients $a_k$, $0 \leq k \leq m - 1$ are chosen as follows: let $q$ be a large prime integer, different from $p$, $a'_k$ be arbitrary, but large integers, such that $a_k = qa'_k$ and $\gcd \left( a_0, q \right) = \gcd \left( a'_0, q \right) = 1 \mod q$, for $1 \leq k \leq m - 1$. The polynomial $f(x)$ is identified with $f(x) \mod p$, by simplifying its coefficients in $\mathbb{Z}_p$. Now, it is tactically assumed that, with some positive probability, $\gcd \left( f(x), (x^p - x) \right) = 1$. If this condition must be deterministically checked, during the construction phase, then the route followed is as follows: a small degree sparse polynomial $h(x)$ is constructed, such that $\gcd \left( h(x), (x^p - x) \right) = 1$, and $f(x)$ is taken to be the polynomial obtained by substituting $x^t$ for $t$ in $h(t)$, where $r$ is a large positive integer relatively prime to $(p - 1)$. The exponents in $f(x)$ are taken $\mod (p - 1)$. As discussed before, the possibility of deterministic polynomial time proof checking, after allowing for nondeterministic choices of hidden nature, relinquishes any claim that there cannot be short deterministic proof. With the tactical assumption, $f(x)$ is only probabilistically assumed to be relatively prime to $(x^p - x)$, but there is no direct evidence of a short deterministic proof testifying the claim that $\gcd$ of $f(x)$ and $(x^p - x)$ is indeed 1. The tactical assumption is contented with some positive probability, however little and insignificant it may be, as may be seen to be different from
the restriction posed for a decision problem to be in the complexity class $\mathcal{NP}$, viz., that the verification step must comply with deterministic polynomial time proof checking. Even if the conditions of the Eisenstein’s criterion are deterministically shown to be satisfied, it may be observed that, in the construction phase, any attempt to deterministically check that $\gcd(f(x), (x^p - x)) = 1$ was deliberately avoided, for precisely this reason. As a last refinement, it is still possible to pose the question as to whether $\gcd(f(x), (x^\nu - 1)) = 1$, where $\nu$ is a large positive integer dividing $(p - 1)$. The probability of the event $\gcd(f(x), (x^\nu - 1)) = 1$, for a large divisor $\nu$ of $(p - 1)$, which is strictly smaller than $(p - 1)$ itself, is enhanced considerably when compared to the probability of the event $\gcd(f(x), (x^{(p-1)} - 1)) = 1$. This prospect of enhanced probability effectively seals the contention that $\text{Co-NP} \neq \mathcal{P}$.

4.5 Implications of the Fact that $\mathcal{PSPACE} = \mathcal{IP}$

In [7], it is shown that $\mathcal{IP} = \mathcal{PSPACE}$. By Savitch’s theorem [5], the complexity classes $\mathcal{NPSPACE}$ and $\mathcal{PSPACE}$ represent the same complexity class. Thus, $\mathcal{PSPACE}$ is closed under augmentation by the capability to make nondeterministic choices of size bounded by some polynomial in the string length of the input word. Allowing for the nondeterministic choices of space bounded by a polynomial in the string length of the input word, the class of languages that admit bounded error probabilistic polynomial time proofs, as computable by nondeterministic polynomial time algorithms, is exactly $\mathcal{IP}$. Let $\mathcal{BPP}$ be the class of languages acceptable by bounded error probabilistic algorithms with probabilistic polynomial time proofs attesting the membership of an input word for each such language. Tentatively, if it is assumed that $\mathcal{BPP} = \mathcal{P}$, then clearly, it must be the case that $\mathcal{IP} = \mathcal{NPSPACE} = \mathcal{PSPACE} = \mathcal{NP}$. However, the discussion of the previous subsection shows that $\mathcal{PSPACE} \neq \mathcal{NP}$, implying that $\mathcal{BPP} \neq \mathcal{P}$. In fact, by equipping a probabilistic algorithm with the capability to make nondeterministic choices of size bounded by some polynomial in the string length of the input word, an algorithm for a language in the complexity class $\mathcal{NPSPACE}$ can be emulated, but a nondeterministic polynomial time algorithm, equipped with the same additional capability, remains only a nondeterministic polynomial time algorithm. Thus, $\mathcal{BPP} \neq \mathcal{NP}$, and by complementarity, $\mathcal{BPP} \neq \text{Co-NP}$, as well. By the same reasoning, if a probabilistic algorithm for some language in the complexity class $\mathcal{BPP}$ is equipped with an additional capability of a polynomial time algorithm, then it still remains a probabilistic algorithm for the same or some other language in the complexity class $\mathcal{BPP}$. But if the probabilistic algorithm is equipped with the capability of nondeterministic choices bounded by some polynomial in the string length of the input word, then the said algorithm effectively emulates an algorithm for some language in the complexity class $\mathcal{IP}$, and therefore, clearly, $\mathcal{NP} \neq \mathcal{P}$, whereby it follows that $\text{Co-NP} \neq \mathcal{P}$, as well. The informal arguments can be made precise by introducing appropriate relativization of complexity classes.
5 Conclusions

This paper presents an original reduction method for solving simultaneous multivariate polynomial equations, by eliminating one variable, from two equations, taken at a time. The reduction method is shown to satisfy a certain minimality criterion, and hence becomes optimal in respect of the constraints stated. A mathematical problem that is in $\mathcal{PSPACE}$ but that which cannot be in either of $\mathcal{NP}$ and $\mathcal{P}$ is also presented. As an afterthought, the trace of execution of an algorithm, even allowing for nondeterministic choices of sizes bounded by some polynomial in the appropriate values of the input parameters for the instances, must also be bounded in size by some polynomial in those parameter values. The trace may be supplied as the input to a debugger program for validation of its operation. This part may be included in the polynomial time certification for the algorithm, for each given input instance. The contentments assert that $\mathcal{PSPACE} \neq \mathcal{NP}$ and $\mathcal{PSPACE} \neq \mathcal{P}$, affirmatively. For probabilistic algorithms, the class $\mathcal{BPP}$ of languages decidable by bounded error probabilistic algorithms with probabilistic polynomial time proofs for the membership of an input word is not the same as the complexity class $\mathcal{P}$. In fact, by equipping a probabilistic algorithm with the capability to make nondeterministic choices of size bounded by some polynomial in the string length of the input word, an algorithm for some language in $\mathcal{NPSPACE}$ can be emulated, but a nondeterministic polynomial time algorithm, equipped with the same additional capability, remains only a nondeterministic polynomial time algorithm. Thus, $\mathcal{BPP} \neq \mathcal{NP}$, as well. It follows that $\mathcal{NP} \neq \mathcal{P}$ — hence that $\text{Co-}\mathcal{NP} \neq \mathcal{P}$ — by relativization of $\mathcal{BPP}$, with respect to $\mathcal{P}$ and $\mathcal{NP}$.

Acknowledgements

The authors gratefully acknowledge fruitful discussions with Professor Michael Oser Rabin, Professor Ronald Linn Rivest and Professor Adi Shamir. The second author received introductions about machine checkable proofs and polynomial time verification certifications during a stint of postdoctoral position under the supervision of Professor Amir Pnueli, at the Weizmann Institute of Science, during the year 2002.

References

[1] Bruno Buchberger, “An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal”, Ph. D. Thesis, University of Innsbruck (1965), English translation by M. Abramson in Journal of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions, Vol. 41(3), 2006, pp. 475–511
[2] D. Castro, M. Giusti, J. Heintz, G. Matera, and L. M. Pardo, “The Hardness of Polynomial Equation Solving”, Foundations of Computational Mathematics, Vol. 3(4), 2003, pp. 347–420

[3] J.-C. Faugère, “A New Efficient Algorithm for Computing Gröbner Bases (F4)”, Journal of Pure and Applied Algebra, Vol. 139(1), 1999, pp. 61–88

[4] J.-C. Faugère, “A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)”, Proc. International Symposium on Symbolic and Algebraic Computation, ACM Press, 2002, pp. 75–83

[5] J. E. Hopcroft, R. Motwani, and J. D. Ullman, “Introduction to Automata Theory, Languages and Computation”, Pearson Education, 2007

[6] Serge Lang, “Algebra”, Springer-Verlag, 2002 (Previously Published by Addison-Wesley, 1993)

[7] Adi Shamir, “$IP = PSPACE$”, Journal of the ACM, Vol. 39(4), 1992, pp. 869–877

[8] André Weil, “Number of Solutions of Equations in Finite Fields”, Bulletin of the American Mathematical Society, Vol. 55(5), May 1949, pp. 497–508