Craig Interpolants in Finite Fields using Algebraic Geometry: Theory and Algorithms*

Utkarsh Gupta[1], Irina Ilioaea[2], Priyank Kalla[1], and Florian Enescu[2]
[1]Electrical and Computer Engineering, University of Utah, Salt Lake City UT, USA
[2]Mathematics and Statistics, Georgia State University, Atlanta GA, USA

Abstract. This paper considers Craig interpolation for a mutually inconsistent pair of polynomial constraints over finite fields $\mathbb{F}_q$, for q any prime power. Using techniques from algebraic geometry, we show that Nullstellensatz over finite fields admits Craig interpolation. The constraints are represented as polynomial ideals with inconsistent varieties, and it is shown how various interpolants, including the smallest and the largest one, can be computed using the Gröbner basis (GB) algorithm. The number of all possible interpolants can also be easily identified. We describe techniques to explore and traverse the interpolant lattice: starting with the Gröbner basis of the smallest interpolant, we generate progressively larger ones, terminating in the largest interpolant.

1 Introduction

Craig interpolation is a method to construct and refine abstractions of functions. It finds application in formal verification of hardware designs and software programs, in logic synthesis of Boolean functions, and also as a tool in proof complexity theory. It is a logical tool to extract concise explanations for the infeasibility of a mutually inconsistent set of statements. Craig [1] showed that for a valid implication $A \implies B$, where A, B are first order formulae containing no free variables, there is a formula I such that $A \implies I$, $I \implies B$ and the non-logical symbols of I appear in both A and B. The formula I is called the *Craig interpolant*, or interpolant for short. As propositional logic also admits Craig interpolation, the formal verification community has extensively investigated interpolants and their computation from resolution proofs of CNF-SAT problems. In the propositional logic domain, the concept is stated with a slight modification.

Definition 1.1. Let (A, B) be a pair of CNF formulae (sets of clauses) such that $A \wedge B$ is unsatisfiable. Then there exists a formula I such that: (i) $A \implies I$; (ii) $I \wedge B$ is unsatisfiable; and (iii) I refers only to the common variables of A and B, i.e. $Var(I) \subseteq Var(A) \cap Var(B)$. The formula I is called the **interpolant** of (A, B).

Given the pair (A, B) and their refutation proof, a procedure called the *interpolation system* constructs the interpolant in linear time and space in the size of the proof [2]. As the abilities of SAT solvers for proof refutation have improved, interpolants have been exploited as abstractions in various problems that can be formulated as unsatisfiable

* This research is funded in part by the US National Science Foundation grants CCF-1619370 and CCF-1320385.

instances, e.g. model checking [2], logic synthesis [3], etc. Their use as abstractions have also been replicated in other (combinations of) theories [4] [5] [6] [7], etc.

In this paper, we introduce the notion of *Craig interpolants in polynomial algebra over finite fields* ($\mathbb{F}_q$) of q elements, where $q = p^k$ is a prime power. Given a mutually inconsistent pair of sets of polynomials with coefficients from $\mathbb{F}_q$ that have no common zeros, we show that Nullstellensatz over finite fields admits interpolation. We represent the sets A, B (from Def. 1.1) as varieties of corresponding ideals, and prove the existence of an interpolant for the pair (A, B). In this setting, *interpolants correspond to varieties* – subsets of the n-dimensional affine space $\mathbb{F}_q^n$ – and are represented by polynomial ideals, more precisely, by a *Gröbner basis of corresponding ideals.*

Intuitively, it should be apparent that polynomial algebra over finite fields would admit Craig interpolation (a first order theory over $\mathbb{F}_q$ definitely admits quantifier elimination [8]). However, our literature search for interpolants and their computation with polynomials in arbitrary finite fields did not reveal much prior work in this area. Recent years have witnessed investigations in formal verification, abstraction and synthesis of datapath circuits with k-bit operands, where the problems have been modeled over finite fields ($\mathbb{F}_{2^k}$) [9] [10] or over finite integer rings ($\mathbb{Z}_{2^k}$) [11]. Interpolants can be exploited as abstractions of functions ($f : \mathbb{F}_{2^k} \to \mathbb{F}_{2^k}$) in this domain, and can make these approaches practical. Motivated by the above needs, this paper presents the theory of Craig interpolation in finite fields, and describes algorithms to compute them.

Contributions: Using the extensive machinery of algebraic geometry in finite fields, this paper makes the following contributions: 1) Formally define the notion of interpolants in polynomial algebra over finite fields $\mathbb{F}_q$, and prove their existence in this domain. 2) Derive the relationship of interpolants with elimination ideals, and show how to compute them using Gröbner bases. 3) Compute the *smallest* interpolant, i.e. the one contained in every other interpolant. Analogously, compute the *largest* interpolant, i.e. the one containing all other interpolants. 4) Count the total number of all possible interpolants. 5) We show how all interpolants can be enumerated in $\mathbb{F}_2$. However, as it is impractical to explore all possible interpolants, we present an algorithm to heuristically enumerate a few interpolants (explore the interpolant lattice): beginning with the smallest, progressively visiting larger ones, and terminating at the largest interpolant.

Paper Organization: The following section briefly reviews prior work in Craig interpolation in various theories, and contrasts it against the concepts presented in this paper. Section 3 describes the preliminary concepts of algebraic geometry and Gröbner bases in finite fields. Section 4 describes the theory of interpolation in finite fields and shows how they can be computed using the Gröbner basis algorithm. Section 5 describes techniques and an algorithm to enumerate the interpolants. Section 6 describes some of our experiments with unsat instances to generate the interpolants. Section 7 concludes the paper. Some of the proofs of the theorems and lemmas are omitted from the main body of the manuscript and are included in an appendix.

2 Review of Previous Work

In the past decade or so, there has been an explosion in the study, classification and application of interpolants. In abstraction-based model checking, interpolants are used as over-approximate image operators [2]. In Boolean function decomposition, given a function $F(A,B,C)$ with support variables partitioned into disjoint subsets A,B,C, it is required to decompose $F = G(A,C) \odot H(B,C)$, where $\odot$ denotes the Boolean $\vee, \wedge, \oplus$ operations. The existence of such a decomposition with the given variable partition is formulated as a unsatisfiability checking problem. Craig interpolants can then be used to compute G,H [3] [12]. In proof complexity, interpolants have been used as a tool to derive lower bounds; *e.g.* by reasoning that if $A \implies B$ does not have a simple interpolant, then it cannot have a simple proof [13]. The authors in [14] present an interpolation theorem for Nullstellensatz refutations and the polynomial calculus [15] which can then be used for proving lower bounds.

The use of interpolants as abstractions has also been replicated in other combinations of theories. For example, the theory of linear inequality [4], data-type theories [5], linear arithmetic and difference logic [6], bit-vector SMT theories [7], etc., are just a few of the many instances of the usage of interpolation in various domains outside of purely propositional logic. The aforementioned works derive interpolants from resolution proofs obtained from SAT/SMT-solvers ([6]), or generate them by solving constrains in the theories of linear arithmetic with uninterpreted functions ([16]), or exploit their connection to quantifier elimination ([5]), etc. As an alternative to interpolation, [17] suggests the use of local proofs and symbol eliminating inferences for invariant generation. However, the problem has been insufficiently investigated over polynomial ideals in finite fields from an algebraic geometry perspective.

The works that come closest to ours are by Gao *et al.* [8] and [18]. While they do not address the interpolation problem per se, they do describe important results of Nullstellensatz, projections of varieties and quantifier elimination over finite fields that we extensively utilize in this paper.

The work of [19] classifies (orders) the interpolants according to their logical strength for model checking. They present a labeled interpolation system built on the resolution proof where each vertex of the proof is annotated with partially ordered labels. Interpolants generated from different sets of labels have the same order of strength as the order of the labels. This way a (sub-)lattice of interpolants is generated with the smallest interpolant being the same as obtained from the McMillan's system (L_M) [2] and the largest being the complement of inverse of L_M. In contrast, we present a method for polynomials in $\mathbb{F}_2$ that can generate the complete lattice of interpolants with the absolute smallest and absolute largest interpolants. The labeled interpolation system of [19] is generalized to support propositional hyper-resolution proofs [20]. More recently, [21] presents the notion of interpolation abstraction, and describes a semantic framework for exploring interpolant lattices. In contrast to these works that qualitatively order the interpolants w.r.t. a given application (e.g. model checking), we describe a method to explore interpolants based on the cardinality of the zero-sets of polynomial ideals, which in turn corresponds to the size of the abstraction.

3 Notation and Preliminary Concepts

Let $\mathbb{F}_q$ denote the finite field of q elements where $q = p^k$ is a prime power, $\overline{\mathbb{F}_q}$ be its algebraic closure, and $R = \mathbb{F}_q[x_1,\dots,x_n]$ the polynomial ring in n variables $x_1,\dots,x_n$, with coefficients from $\mathbb{F}_q$. A monomial is a power product of the form $X = x_1^{e_1} \cdot x_2^{e_2} \cdots x_n^{e_n}$, where $e_i \in \mathbb{Z}_{\geq 0}, i \in \{1,\dots,n\}$. A *polynomial* $f \in R$ is written as a finite sum of terms

$f = c_1X_1 + c_2X_2 + \cdots + c_tX_t$, where $c_1, \ldots, c_t$ are coefficients and $X_1, \ldots, X_t$ are monomials. Impose a monomial order $>$ (a term order) on the ring – i.e. a total order and a well-order on all the monomials of R s.t. multiplication with another monomial preserves the order. Then the monomials of all polynomials $f = c_1X_1 + c_2X_2 + \cdots + c_tX_t$ are ordered w.r.t. to $>$, such that $X_1 > X_2 > \cdots > X_t$. Subject to $>$, $lt(f) = c_1X_1$, $lm(f) = X_1$, $lc(f) = c_1$, are the *leading term*, *leading monomial* and *leading coefficient* of f, respectively. In this work, we consider mostly with lexicographic (lex) term orders.

Ideals, Varieties and Gröbner Bases: Given a set of polynomials $F = \{f_1, \ldots, f_s\}$ in R, the *ideal* $J \subseteq R$ generated by them is: $J = \langle f_1, \ldots, f_s \rangle = \{\sum_{i=1}^{s} h_i \cdot f_i : \ h_i \in R\}$. The polynomials $f_1, \ldots, f_s$ form the *basis* or the *generators* of J.

Let $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$ be a point in the affine space, and f a polynomial in R. If $f(\boldsymbol{a}) = 0$, we say that f *vanishes* on $\boldsymbol{a}$. We have to analyze the *set of all common zeros* of the polynomials of F that lie within the field $\mathbb{F}_q$. This zero set is called the *variety*. It depends not just on the given set of polynomials but rather on the ideal generated by them. We denote it by $V_{\mathbb{F}_q}(J) = V_{\mathbb{F}_q}(f_1, \ldots, f_s)$, where:

$$V_{\mathbb{F}_q}(J) = V_{\mathbb{F}_q}(f_1, \ldots, f_s) = \{\boldsymbol{a} \in \mathbb{F}_q^n : \forall f \in J, f(\boldsymbol{a}) = 0\}.$$

Varieties can be different when restricted to the given field $\mathbb{F}_q$ or considered over its algebraic closure $\overline{\mathbb{F}_q}$. We will generally drop the subscript when considering varieties over $\mathbb{F}_q$ and denote $V(J)$ to imply $V_{\mathbb{F}_q}(J)$. The subscripts will be used, however, to avoid any ambiguities, e.g. when comparing $V_{\mathbb{F}_q}(J)$ against the one over the closure $V_{\overline{\mathbb{F}_q}}(J)$.

Given two ideals $J_1 = \langle f_1, \ldots, f_s \rangle, J_2 = \langle h_1, \ldots, h_r \rangle$, the sum $J_1 + J_2 = \langle f_1, \ldots, f_s, h_1 \ldots, h_r \rangle$, and their product $J_1 \cdot J_2 = \langle f_i \cdot h_j : 1 \leq i \leq s, 1 \leq j \leq r \rangle$. Ideals and varieties are dual concepts: $V(J_1 + J_2) = V(J_1) \cap V(J_2)$, and $V(J_1 \cdot J_2) = V(J_1) \cup V(J_2)$. Moreover, if $J_1 \subseteq J_2$ then $V(J_1) \supseteq V(J_2)$.

Gröbner Basis: An ideal may have many different sets of generators: $J = \langle f_1, \ldots, f_s \rangle = \cdots = \langle g_1, \ldots, g_t \rangle$. Given a non-zero ideal J, a *Gröbner basis* (GB) for J is a finite set of polynomials $G = \{g_1, \ldots, g_t\}$ satisfying $\langle \{lm(f) \mid f \in J\} \rangle = \langle lm(g_1), \ldots, lm(g_t) \rangle$. Then $J = \langle G \rangle$ holds and so $G = GB(J)$ forms a basis for J. A GB G possesses important properties that allow to solve many polynomial computation and decision problems. The famous Buchberger's algorithm (see Alg. 1.7.1 in [22]) takes as input the set of polynomials $F = \{f_1, \ldots, f_s\}$ and computes the GB $G = \{g_1, \ldots, g_t\}$. A GB can be *reduced* to eliminate redundant polynomials from the basis. A reduced GB is a canonical representation of the ideal. In this work, the set G will denote a reduced GB, and any reference to computation of an ideal can be construed as constructing its GB.

Varieties over finite fields and the structure of Gröbner bases: When the variety of an ideal is finite, then the ideal is said to be *zero-dimensional*. As $V_{\mathbb{F}_q}(J)$ is a finite set, J is zero-dimensional. A GB for a zero dimensional ideal exhibits a special structure that we exploit in this work.

For all elements $\alpha \in \mathbb{F}_q, \alpha^q = \alpha$. Therefore, the polynomial $x^q - x$ vanishes everywhere in $\mathbb{F}_q$, and is called the vanishing polynomial of the field, sometimes also referred to as the field polynomial. Denote by $J_0 = \langle x_1^q - x_1, \dots, x_n^q - x_n \rangle$ the ideal of all vanishing polynomials in the ring R. Then $V_{\mathbb{F}_q}(J_0) = V_{\overline{\mathbb{F}_q}}(J_0) = \mathbb{F}_q^n$. Therefore, given any ideal J, $V_{\mathbb{F}_q}(J) = V_{\overline{\mathbb{F}_q}}(J) \cap \mathbb{F}_q^n = V_{\overline{\mathbb{F}_q}}(J) \cap V_{\overline{\mathbb{F}_q}}(J_0) = V_{\overline{\mathbb{F}_q}}(J + J_0) = V_{\mathbb{F}_q}(J + J_0)$.

Theorem 3.1 (*The Weak Nullstellensatz over finite fields (from Theorem 3.3 in [18])*)**.** *For a finite field $\mathbb{F}_q$ and the ring $R = \mathbb{F}_q[x_1, \dots, x_n]$, let $J = \langle f_1, \dots, f_s \rangle \subseteq R$, and let $J_0 = \langle x_1^q - x_1, \dots, x_n^q - x_n \rangle$ be the ideal of vanishing polynomials. Then $V_{\mathbb{F}_q}(J) = \emptyset \iff 1 \in J + J_0 \iff G = reducedGB(J + J_0) = \{1\}$.*

To find whether a set of polynomials $f_1, \dots, f_s$ have no common zeros in $\mathbb{F}_q$, we can compute the reduced GB G of $\{f_1, \dots, f_s, x_1^q - x_1, \dots, x_n^q - x_n\}$ and see if $G = \{1\}$. If $G \neq \{1\}$, then $f_1, \dots, f_s$ do have common zeros in $\mathbb{F}_q$, and G consists of the finite set of polynomials $\{g_1, \dots, g_t\}$ with the following properties.

Theorem 3.2 (*Gröbner bases in finite fields (application of Theorem 2.2.7 from [22] over $\mathbb{F}_q$)*)**.** *For $G = GB(J + J_0) = \{g_1, \dots, g_t\}$, the following statements are equivalent:*

1. *The variety $V_{\mathbb{F}_q}(J)$ is finite.*
2. *For each $i = 1, \dots, n$, there exists some $j \in \{1, \dots, t\}$ such that $lm(g_j) = x_i^l$ for some $l \in \mathbb{N}$.*
3. *The quotient ring $\frac{\mathbb{F}_q[x_1 \dots, x_n]}{\langle G \rangle}$ forms a finite dimensional vector space.*

In other words, the ideal $J + J_0$ is zero-dimensional, and for each variable x_i, there exists an element in the GB whose leading term is a pure power of x_i. When that happens, we can also count the number of solutions. For a GB G, let $LM(G)$ denote the set of leading monomials of all elements of G: $LM(G) = \{lm(g_1), \dots, lm(g_t)\}$.

Definition 3.1 (*Standard Monomials*)**.** Let $\boldsymbol{X^e} = x_1^{e_1} \cdots x_n^{e_n}$ denote a monomial. The set of standard monomials of G is defined as $SM(G) = \{\boldsymbol{X^e} : \boldsymbol{X^e} \notin \langle LM(G) \rangle\}$.

Theorem 3.3 (*Counting the number of solutions (Theorem 3.7 in [18])*)**.** *Let $G = GB(J + J_0)$, and $|SM(G)| = m$, then the ideal J vanishes on m distinct points in $\mathbb{F}_q^n$. In other words, $|V(J)| = |SM(G)|$.*

3.1 Radical ideals and the Strong Nullstellensatz

Definition 3.2. Given an ideal $J \subset R$ and $V(J) \subseteq \mathbb{F}_q^n$, the *ideal of polynomials that vanish on* $V(J)$ is $I(V(J)) = \{f \in R : \forall \boldsymbol{a} \in V(J), f(\boldsymbol{a}) = 0\}$.

If $I_1 \subset I_2$ are ideals then $V(I_1) \supset V(I_2)$, and similarly if $V_1 \subset V_2$ are varieties, then $I(V_1) \supset I(V_2)$.

Definition 3.3. For any ideal $J \subset R$, the **radical** of J is defined as $\sqrt{J} = \{f \in R : \exists m \in \mathbb{N} s.t. f^m \in J\}$.

When $J = \sqrt{J}$, J is called a radical ideal. Over algebraically closed fields, the *Strong Nullstellensatz* establishes the correspondence between radical ideals and varieties. Over finite fields, it has a special form.

Lemma 3.1. (From [8]) For an arbitrary ideal $J \subset \mathbb{F}_q[x_1, \dots, x_n]$, and $J_0 = \langle x_1^q - x_1, \dots, x_n^q - x_n \rangle$, the ideal $J + J_0$ is radical; i.e. $\sqrt{J + J_0} = J + J_0$.

Theorem 3.4 (*The Strong Nullstellensatz over finite fields (Theorem 3.2 in [8])*)**.** For any ideal $J \subset \mathbb{F}_q[x_1, \dots, x_n]$, $I(V_{\mathbb{F}_q}(J)) = J + J_0$.

3.2 Projection of varieties and elimination ideals in finite fields

Definition 3.4. Given an ideal $J = \langle f_1, \dots, f_s \rangle \subset R$ and its variety $V(J) \subset \mathbb{F}_q^n$, the l-th projection of $V(J)$ denoted as $Pr_l(V(J))$ is the mapping

$$Pr_l(V(J)) : \mathbb{F}_q^n \to \mathbb{F}_q^{n-l},\ Pr_l(a_1, \dots, a_n) = (a_{l+1}, \dots, a_n)$$

for every $\boldsymbol{a} = (a_1, \dots, a_n) \in V(J)$.

Definition 3.5. Given an ideal $J \subset \mathbb{F}_q[x_1, \dots, x_n]$, the l-th elimination ideal J_l is an ideal in R defined as $J_l = J \cap \mathbb{F}_q[x_{l+1}, \dots, x_n]$.

The next theorem shows how we can obtain the generators of the l-th elimination ideal using Gröbner bases.

Theorem 3.5 (*Elimination Theorem [23]*)**.** Given an ideal $J \subset R$ and its GB G *w.r.t.* the lexicographical (lex) order on the variables where $x_1 > x_2 > \cdots > x_n$, then for every $0 \leq l \leq n$ we denote by G_l the GB of l-th elimination ideal of J and compute it as:

$$G_l = G \cap \mathbb{F}_q[x_{l+1}, \dots, x_n]$$

In a general setting, the projection of a variety is a subset of the variety of an elimination ideal: $Pr_l(V(J)) \subseteq V(J_l)$. However, operating over finite fields, when the ideals contain the vanishing polynomials, then the above set inclusion turns into an equality.

Lemma 3.2 (Lemma 3.4 in [8])**.** Given an ideal $J \subset R$ that contains the vanishing polynomials of the field, then $Pr_l(V(J)) = V(J_l)$, i.e. the l-th projection of the variety of ideal J is equal to the variety of its l-th elimination ideal.

We will utilize all of the above concepts to derive the results in this paper.

4 Theory

We describe the setup for Craig interpolation in the ring $R = \mathbb{F}_q[x_1, \dots, x_n]$. Partition the variables $\{x_1, \dots, x_n\}$ into disjoint subsets A, B, C. We are given two ideals $J_A \subset \mathbb{F}_q[A, C], J_B \subset \mathbb{F}_q[B, C]$ such that the C-variables are common to the generators of both J_A, J_B. *From here on, we will assume that all ideals include the corresponding vanishing polynomials.* For example, generators of J_A include $\boldsymbol{A^q - A}, \boldsymbol{C^q - C}$ where $\boldsymbol{A^q - A} = \{x_i^q - x_i : x_i \in A\}$, and so on. Then these ideals become radicals and we can apply Lemmas 3.1 and 3.2. We use $V_{A,C}(J_A)$ to denote the variety of J_A over the $\mathbb{F}_q$-space spanned by A and C variables, i.e. $V_{A,C}(J_A) \subset \mathbb{F}_q^A \times \mathbb{F}_q^C$. Similarly, $V_{B,C}(J_B) \subset \mathbb{F}_q^B \times \mathbb{F}_q^C$.

Now let $J = J_A + J_B \subseteq \mathbb{F}_q[A, B, C]$, and suppose that it is found by application of the Weak Nullstellensatz (Thm. 3.1) that $V_{A,B,C}(J) = \emptyset$. When we compare the varieties of J_A and J_B, then we can consider the varieties in $\mathbb{F}_q^A \times \mathbb{F}_q^B \times \mathbb{F}_q^C$, as $V_{A,B,C}(J_A) = V_{A,C}(J_A) \times \mathbb{F}_q^B \subset \mathbb{F}_q^A \times \mathbb{F}_q^B \times \mathbb{F}_q^C$. With this setup, we define the interpolants as follows.

Definition 4.1 (*Interpolants in finite fields*)**.** Given two ideals $J_A \subset \mathbb{F}_q[A,C]$ and $J_B \subset \mathbb{F}_q[B,C]$ where A,B,C denote the three disjoint sets of variables such that $V_{A,B,C}(J_A) \cap V_{A,B,C}(J_B) = \emptyset$. Then there exists an ideal J_I satisfying the following properties:

1. $V_{A,B,C}(J_I) \supseteq V_{A,B,C}(J_A)$
2. $V_{A,B,C}(J_I) \cap V_{A,B,C}(J_B) = \emptyset$
3. The generators of J_I contain only the C-variables; or $J_I \subseteq \mathbb{F}_q[C]$.

We call $V_{A,B,C}(J_I)$ the **interpolant** in finite fields of the pair $(V_{A,B,C}(J_A), V_{A,B,C}(J_B))$, and the corresponding ideal J_I is called the **ideal-interpolant**.

As the generators of J_I contain only the C-variables, the interpolant $V_{A,B,C}(J_I)$ is of the form $V_{A,B,C}(J_I) = \mathbb{F}_q^A \times \mathbb{F}_q^B \times V_C(J_I)$.

Example 4.1. *Consider the ring $R = \mathbb{F}_2[a,b,c,d,e]$, partition the variables as $A = \{a\}, B = \{e\}, C = \{b,c,d\}$. Let ideals*

$$\begin{aligned} J_A &= \langle ab, bd, bc+c, cd, bd+b+d+1\rangle + J_{0,A,C} \\ J_B &= \langle b, d, ec+e+c+1, ec\rangle + J_{0,B,C} \end{aligned}$$

where $J_{0,A,C}$ and $J_{0,B,C}$ are the corresponding ideals of vanishing polynomials. Then, we have

$$\begin{aligned} V_{A,B,C}(J_A) &= \mathbb{F}_q^B \times V_{A,C}(J_A) \\ &= (abcde) : \{01000, 00010, 01100, 10010, \\ &\qquad\qquad\quad 01001, 00011, 01101, 10011\} \\ V_{A,B,C}(J_B) &= \mathbb{F}_q^A \times V_{B,C}(J_B) \\ &= (abcde) : \{00001, 00100, 10001, 10100\} \end{aligned}$$

The ideals J_A, J_B have no common zeros as $V_{A,B,C}(J_A) \cap V_{A,B,C}(J_B) = \emptyset$. The pair (J_A, J_B) admits a total of 8 interpolants:

1. $V(J_S) = (bcd) : \{001, 100, 110\}$
   $J_S = \langle cd, b+d+1\rangle$
2. $V(J_1) = (bcd) : \{001, 100, 110, 101\}$
   $J_1 = \langle cd, bd+b+d+1, bc+cd+c\rangle$
3. $V(J_2) = (bcd) : \{001, 100, 110, 011\}$
   $J_2 = \langle b+d+1\rangle$
4. $V(J_3) = (bcd) : \{001, 100, 110, 111\}$
   $J_3 = \langle b+cd+d+1\rangle$
5. $V(J_4) = (bcd) : \{001, 100, 110, 011, 111\}$
   $J_4 = \langle bd+b+d+1, bc+b+cd+c+d+1\rangle$
6. $V(J_5) = (bcd) : \{001, 100, 110, 101, 111\}$
   $J_5 = \langle bc+c, bd+b+d+1\rangle$
7. $V(J_6) = (bcd) : \{001, 100, 110, 101, 011\}$
   $J_6 = \langle bd+b+d+1, bc+cd+c\rangle$
8. $V(J_L) = (bcd) :$
   $\{001, 011, 100, 101, 110, 111\}$
   $J_L = \langle bd+b+d+1\rangle$.

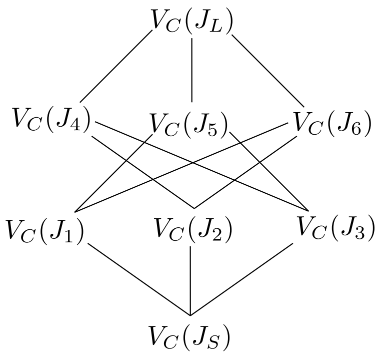

Fig. 1: Interpolant lattice

It is easy to check that all $V(J_I)$ satisfy the 3 conditions of Def. 4.1. Note also that $V(J_S)$ is the smallest interpolant, contained in every other interpolant. Likewise, $V(J_L)$ contains all other interpolants and it is the largest. The other containment relationships are shown in the corresponding interpolant lattice in Fig. 1; i.e. $V_C(J_1) \subset V_C(J_5), V_C(J_1) \subset V_C(J_6)$, and so on.

Theorem 4.1. An ideal-interpolant J_I, and correspondingly the interpolant $V_{A,B,C}(J_I)$, as given in Def. 4.1, always exists.

Proof. Consider the elimination ideal $J_I = J_A \cap \mathbb{F}_q[C]$. We show J_I satisfies the three conditions for the interpolant.
Condition 1: $V_{A,B,C}(J_I) \supseteq V_{A,B,C}(J_A)$. This condition is trivially satisfied due to construction of elimination ideals. As $J_I \subseteq J_A$, $V_{A,B,C}(J_I) \supseteq V_{A,B,C}(J_A)$.
Condition 2: $V_{A,B,C}(J_I) \cap V_{A,B,C}(J_B) = \emptyset$. This condition can be equivalently stated as $V_{B,C}(J_I) \cap V_{B,C}(J_B) = \emptyset$ as neither J_I nor J_B contains any variables from the set A. We prove this condition by contradiction. Let's assume that there exists a common point $(\mathbf{b}, \mathbf{c})$ in $V_{B,C}(J_I)$ and $V_{B,C}(J_B)$. We know that the projection of the variety $Pr_A(V_{A,C}(J_A))$ is equal to the variety of the elimination ideal $V_C(J_I)$, where $J_I = J_A \cap \mathbb{F}_q[C]$, due to Lemma 3.2. Therefore, the point $(\mathbf{c})$ in the variety of J_I can be extended to a point $(\mathbf{a}, \mathbf{c})$ in the variety of J_A. This implies that the ideals J_A and J_B vanish at $(\mathbf{a}, \mathbf{b}, \mathbf{c})$. This is a contradiction to our initial assumption that the intersection of the varieties of J_A and J_B is empty. Thus J_I, J_B have no common zeros.
Condition 3: The generators of J_I contain only the C-variables. This condition is trivially satisfied as J_I is the elimination ideal obtained by eliminating A-variables in J_A.
□

The above theorem not only proves the existence of an interpolant, but also gives a procedure to construct one: $J_I = J_A \cap \mathbb{F}_q[C]$. In other words, compute a reduced Gröbner basis G of J_A w.r.t. an elimination order $A > B > C$ and take $G_I = G \cap \mathbb{F}_q[C]$. Then G_I gives the generators for the ideal-interpolant J_I.

Example 4.2. *The elimination ideal J_I computed for J_A from Example 4.1 is $J_I = J_S = \langle cd, b+d+1 \rangle$ with variety $V_C(J_I) = (bcd) : \{001, 100, 110\}$. This variety over the variable set A and C is $V_{A,C}(J_I) = (abcd) : \{0001, 0100, 0110, 1001, 1100, 1110\}$, and it contains $V_{A,C}(J_A)$. Moreover, $V_{A,B,C}(J_I)$ also has an empty intersection with $V_{A,B,C}(J_B)$.*

Theorem 4.2. The interpolant $V_{A,B,C}(J_S)$ corresponding to the ideal $J_S = J_A \cap \mathbb{F}_q[C]$ is the smallest interpolant.

Proof. The proof is given in the appendix.

□

Now we discuss how the largest interpolant can be computed. For this, we will make use of quotients of ideals.

Definition 4.2. (Quotient of Ideals) If J_1 and J_2 are ideals in a ring R, then $J_1 : J_2$ is the set $\{f \in R \mid f \cdot g \in J_1, \forall g \in J_2\}$ and is called the **ideal quotient** of J_1 by J_2.

We use ideal quotients to compute the complement of a variety. Given an ideal $J' \subset R$ containing the vanishing polynomials, suppose we need to find an ideal J such that $V(J) = \mathbb{F}_q^n - V(J') = V(J_0) - V(J')$, where "$-$" corresponds to the set difference operation. Then $J = J_0 : J'$ (see Theorem III.2 and Corollary III.1 in [10] for a proof outline). Once again, the Gröbner basis algorithm can be used to compute $J_0 : J'$ [23].

Theorem 4.3. Consider the elimination ideal $J'_L = J_B \cap \mathbb{F}_q[C]$. The complement of the variety $V_C(J'_L)$, computed as $\mathbb{F}_q^C - V_C(J'_L)$, is the largest interpolant.

Proof. Proof is given in the appendix.

□

Let J_L be the radical ideal corresponding to the largest interpolant $V_C(J_L) = \mathbb{F}_q^C - V_C(J'_L)$. This ideal-interpolant J_L can be computed as $J_L = (J_{0,C} : J'_L)$, where $J_{0,C}$ is ideal of vanishing polynomials in C-variables.

Example 4.3. *The ideal-interpolant $J_L = \langle bd + b + d + 1\rangle$ in Example 4.1 is computed as:*

- *First compute the ideal $J'_L = J_B \cap \mathbb{F}_q[C]$ which results in $J'_L = \langle b, d\rangle$.*
- *Then compute J_L as $J_L = J_{0,C} : J'_L$ which results in $J_L = \langle bd + b + d + 1\rangle$*

The variety $V_C(J_L) = (bcd) : \{001, 011, 100, 101, 110, 111\}$ and it is the largest interpolant for the given pair (J_A, J_B).

Lemma 4.1. The total number of interpolants for the pair (J_A, J_B) is $2^{|SM(J_D)|}$, where $J_D = (J_L : J_S)$.

Proof. The proof is given in the appendix.

□

Example 4.4. *From Example 4.1 $J_L = \langle bd + b + d + 1\rangle$ and $J_S = \langle cd, b + d + 1\rangle$. Computing $J_D = J_L : J_S$ gives $J_D = \langle d + 1, bc + b + c + 1, c^2 + c, b^2 + b\rangle$, where the variety $V_C(J_D) = V_C(J_L) - V_C(J_S) = (bcd) : \{011, 101, 111\}$.*

The standard monomials for J_D are $SM(J_D) = \{1, b, c\}$. Therefore, the total number of interpolants for the given pair (J_A, J_B) is $2^{|\{1,b,c\}|} = 2^3 = 8$.

The structure of the interpolant lattice: Note that our results do provide some insights into the structure of the interpolant lattice. Let $l = |SM(J_D)|$. Then, the height of the interpolant lattice is $l + 1$, and the number of elements (interpolants) at each level i is $\binom{l}{i}$, $0 \le i \le l$. Notice also that the size (height and width) of the interpolant lattice is independent of the number of variables in the set C, and depends only on $|SM(J_D)|$.

5 Enumerating the Interpolants in $\mathbb{F}_2[A, B, C]$

Lemma 4.1 gives us the number of interpolants that exist for the given pair (J_A, J_B). This section presents procedures for enumerating these interpolants using $SM(J_D)$. Note that these procedures can only be applied while operating over the field $\mathbb{F}_2$. First we describe a procedure for enumerating all the interpolants. This is made possible by exploiting the relationship between the interpolants and $SM(J_D)$.

Theorem 5.1. Given the interpolant setup over $\mathbb{F}_2[A,B,C]$, let $SM(J_D) = \{m_1,\dots,m_l\}$. Construct a polynomial f_i using any linear combination of $\{m_1,\dots,m_l\}$ as,

$$f_i = \lambda_1 \cdot m_1 + \lambda_2 \cdot m_2 + \cdots + \lambda_l \cdot m_l \tag{1}$$

where each $\lambda_j \in \mathbb{F}_2 = \{0,1\}$. Then all the ideal-interpolants J_I can be obtained as,

$$J_I = J_S \cdot (J_D + \langle f_i \rangle). \tag{2}$$

There can be 2^l such f_i, and as $|SM(J_D)| = l$, the number of interpolants is also 2^l. Therefore, each f_i in Eqn. (2) will result in a distinct interpolant.

Proof. We want to prove that each f_i will result in a distinct interpolant when used in Eqn. (2). Consider the variety $V_C(J_D)$ and its cardinality $|V_C(J_D)| = l$. From the proof of Lemma 4.1, we know that the union of the variety $V_C(J_S)$ and each subset of $V_C(J_D)$ produces a new interpolant. Therefore

$$V_C(J_I) = V_C(J_S) \cup W_i \tag{3}$$

where $W_i \in PowerSet(V_C(J_D))$.

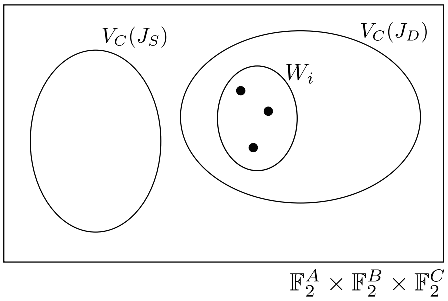
$V_C(J_S)$
$V_C(J_D)$
W_i
$\mathbb{F}_2^A \times \mathbb{F}_2^B \times \mathbb{F}_2^C$

Fig. 2: The variety $V_C(J_D)$ and an element W_i in its power set.

Every W_i is a set of finite number of points as shown in Fig. 2, and therefore it forms a variety. As we are working over finite fields, the ideal of this variety can be constructed using only one polynomial f_i'. For example, f_i' could be constructed by means of Lagrange's interpolation formula over W_i. Therefore, $V_C(\langle f_i' \rangle) = W_i$. Note that there can be multiple polynomials f_i' with variety W_i and they belong to the same equivalence class of polynomials $\pmod{J_D}$. Consider the reduction $f_i' \xrightarrow{GB(J_D)} f_i$. Then $V_C(J_D + \langle f_i' \rangle) = V_C(J_D + \langle f_i \rangle)$. As $V_C(\langle f_i' \rangle) = V_C(J_D + \langle f_i' \rangle)$ because $V_C(\langle f_i' \rangle) \subseteq V_C(J_D)$, we have $V_C(\langle f_i' \rangle) = V_C(J_D + \langle f_i \rangle) = W_i$.

As f_i' is reduced by $GB(J_D)$, the remainder $f_i \in \mathbb{F}_q[C]/J_D$ is a canonical representative for the equivalence class containing f_i' and is composed of only $SM(J_D)$.

There are 2^l such equivalence classes of polynomials $\pmod{J_D}$ and each one of them can be reduced to a unique f_i. As there are l standard monomials $\{m_1,\dots,m_l\}$, they can be combined linearly to form 2^l unique polynomials f_i. Each equivalence class corresponds to a distinct interpolant (Eqn. (3)). Consequently each f_i will also correspond to a distinct interpolant,

$$
\begin{aligned}
V_C(J_I) &= V_C(J_S) \cup W_i \quad \text{(from Eqn. (3))} \\
V_C(J_I) &= V_C(J_S) \cup (V_C(J_d + \langle f_i \rangle)) \\
V_C(J_I) &= V_C(J_S) \cup (V_C(J_d) \cap V_C(\langle f_i \rangle)) \\
J_I &= J_S \cdot (J_D + \langle f_i \rangle) \quad \text{(using ideal-variety duality)}
\end{aligned}
$$

□

Example 5.1. *From Example 4.1 and 4.4 that are setup over $\mathbb{F}_2[A,B,C]$, we have $J_S = \langle cd, b+d+1 \rangle$ and $J_D = \langle d+1, bc+b+c+1 \rangle$ with $SM(J_D) = \{1,b,c\}$. We can enumerate all the interpolants for the pair (J_A, J_B) using $f_i = \lambda_1 \cdot 1 + \lambda_2 \cdot b + \lambda_3 \cdot c$ where $\{\lambda_1, \lambda_2, \lambda_3\} \in \{0,1\}$.*

- *Let $f_1 = c$ with $(\lambda_1,\lambda_2,\lambda_3) = (0,0,1)$.*
  The GB computation of the ideal $J_S \cdot (J_D + \langle f_1 \rangle)$ results in $\langle cd, bd+b+d+1, bc+cd+c \rangle$ (= J_1 from Example 4.1)
- *Let $f_5 = b+1$ with $(\lambda_1,\lambda_2,\lambda_3) = (1,1,0)$.*
  The GB computation of the ideal $J_S \cdot (J_D + \langle f_5 \rangle)$ results in $\langle bc+c, bd+b+d+1 \rangle$ (= J_5 from Example 4.1)
- *Let $f_6 = b+c+1$ with $(\lambda_1,\lambda_2,\lambda_3) = (1,1,1)$.*
  The GB computation of the ideal $J_S \cdot (J_D + \langle f_6 \rangle)$ results in $\langle bd+b+d+1, bc+cd+c \rangle$ (= J_6 from Example 4.1)
- *Similarly, all the 8 interpolants can be obtained from the 8 possible f_i.*

The reason why Theorem 5.1 requires the interpolant setup over $\mathbb{F}_2[A,B,C]$ is as follows. If we assume that the setup was over some other finite field $\mathbb{F}_q$, then Eqn. (1) will produce q^l ($l = |SM(J_D|$) polynomials f_i. However, the number of interpolants is exactly equal to 2^l irrespective of the field we are working on (Lemma 4.1). As a result, multiple f_i can produce same interpolant unlike the case when $q = 2$ (each f_i produces a distinct interpolant). The study to enumerate all the interpolants for any finite field using the $SM(J_D)$ is a part of our future work.

In practice, we don't need to compute all interpolants. However, given an interpolant it may be desirable to obtain a larger interpolant that provides a better abstraction. Given an ideal-interpolant J_I we discuss how a larger ideal-interpolant J_K can be obtained so that $V_C(J_I) \subset V_C(J_K)$.

Let J_I and J_K be two ideal-interpolants obtained using polynomials f_i and f_k respectively (using Eqn. (2)). Assuming that $V_C(J_I) \subset V_C(J_K)$ consider,

$$
\begin{gathered}
V_C(J_I) \subset V_C(J_K) \\
V_C(J_S \cdot (J_D + \langle f_i \rangle)) \subset V_C(J_S \cdot (J_D + \langle f_k \rangle)) \\
V_C(J_S) \cup V_C(J_D + \langle f_i \rangle) \subset V_C(J_S) \cup V_C(J_D + \langle f_k \rangle)
\end{gathered}
$$

as the sets $V_C(J_S)$ and $V_C(J_D)$ are disjoint (Fig. 2) we can write

$$\begin{aligned} V_C(J_D + \langle f_i \rangle) &\subset V_C(J_D + \langle f_k \rangle) \\ I(V_C(J_D + \langle f_i \rangle)) &\supset I(V_C(J_D + \langle f_k \rangle)) \end{aligned}$$

using Lemma 3.1 and Theorem 3.4 we have

$$\begin{aligned} J_D + \langle f_i \rangle &\supset J_D + \langle f_k \rangle \\ J_D + \langle f_i \rangle &\supset f_k \end{aligned} \tag{4}$$

Now that we know that f_k is contained in $J_D + \langle f_i \rangle$, we will show how f_k can be obtained from the GB of $J_D + \langle f_i \rangle$.

Theorem 5.2. Given an ideal-interpolant J_I computed as $J_I = J_S \cdot (J_D + \langle f_i \rangle)$. Obtain the reduced *GB* $G_{Di} = GB(J_D + \langle f_i \rangle)$. Then there must exist at least one $g_j \in G_{Di}$ which is a linear combination of $SM(J_D)$. Each $g_j \neq f_i$ can be used to obtain a new interpolant J_K such that $V_C(J_I) \subset V_C(J_K)$.

Proof. We need to show that G_{Di} will contain at least one polynomial that is a linear combination of $SM(J_D)$. As a reduced *GB* is a canonical representation, G_{Di} can also be computed as $GB(GB(J_D) + \langle f_i \rangle)$. Consider the set $GB(J_D)$ where each polynomial p_r can be written as $lt(p_r) + (p_r - lt(p_r))$. The monomials in $p_r - lt(p_r)$ can only contain the elements of $SM(J_D)$ (otherwise they can be divided by the leading terms of polynomials in J_D).

Construct $GB(J_D) + \langle f_i \rangle$ and compute the reduced *GB* of this ideal. As the set of polynomials $GB(J_D)$ is already a *GB*, Buchberger's algorithm will pair f_i and each polynomial from the set $GB(J_D)$ for the *S-poly* computation $S\text{-}poly(p_r, f_i) \xrightarrow{GB(J_D), f_i}_+ h_r$. The *S-poly* is reduced modulo $\{GB(J_D), f_i\}$ and as a result h_r will only be composed of $SM(J_D)$. This implies that there will be at least one polynomial in the $GB(J_D + \langle f_i \rangle))$ containing monomials only from the set $SM(J_D)$. This polynomial can then be used to compute an ideal-interpolant J_K such that $V_C(J_I) \subset V_C(J_K)$.

□

If there is only one polynomial in G_{Di} which is a linear combination of $SM(J_D)$ and is f_i itself, then using this polynomial in Eqn. (2) will result in J_I itself. In that case, the only larger interpolant is J_L.

Theorem 5.2 gives an approach to devise an algorithm that computes a chain of progressively larger interpolants starting from J_S. The following steps explain the algorithm.

1. Given the pair (J_A, J_B) in $\mathbb{F}_2[A, B, C]$, compute J_S, J_D, and $SM(J_D)$. Store J_S in some list L.
2. Pick a polynomial $f_i = \sum_{i=1}^{i=l} \lambda_i \cdot m_i$, where $\{m_1, \ldots, m_l\} = SM(J_D)$ and $\lambda_i \in \{0, 1\}$. ($f_i \neq 1$ otherwise, $GB(J_D + \langle 1 \rangle) = 1$, so $J_I = J_S$).
3. Compute $G_{Di} = GB(J_D + \langle f_i \rangle)$. Append $J_I = J_S \cdot G_{Di}$ to L.
4. In G_{Di} find polynomials g_j which are of the form $\sum_{i=1}^{i=l} \lambda_i \cdot m_i$.
5. Pick a $g_j \neq f_i$ and goto step 3 where g_j replaces f_i in the computation of G_{Di}.

6. If in step 4, there is only one g_j and $g_j = f_i$, terminate the algorithm after appending $J_L = J_S \cdot J_D$ to L.

The algorithm returns the list L whose first element is J_S and last element is J_L with a chain of progressively larger interpolants in between. The pseudo-code for this algorithm is presented in Algorithm 1.

Algorithm 1 Compute larger ideal-interpolants given $J_I \neq J_S$

```
procedure get_larger_interpolant($J_S, J_D, SM(J_D)$)
    Initialize list L for storing interpolants
    Append $J_S$ to L
    Pick $f_i = \sum_{i=1}^{i=l} \lambda_i \cdot m_i \quad (f_i \neq 1)$
    while (1) do
        Compute $G_{Di} = GB(J_D + \langle f_i \rangle)$
        Append $J_S \cdot G_{Di}$ to L
        Find $g_j \in G_{Di}$ s.t. $g_j = \sum_{i=1}^{i=l} \lambda_i \cdot m_i$
        if $|\{g_j\}| = 1$ and $g_j = f_i$ then
            Append $J_S \cdot J_D$ to L
            return L  //Reached largest interpolant
        else
            Choose a $g_j \neq f_i$
            $f_i = g_j$
```

Example 5.2. *The algorithm can be understood with this example. Consider the following steps.*

- *From Example 4.4, $J_S = \langle cd, b+d+1 \rangle$, $J_D = \langle d+1, bc+b+c+1 \rangle$ with $SM(J_D) = \{1, b, c\}$. The ideal-interpolant J_S is appended to L so that $L = [J_S]$.*
- *We need to pick a $f_i = \lambda_1 \cdot 1 + \lambda_2 \cdot b + \lambda_3 \cdot c$ and $f_i \neq 1$. Let $f_1 = c$ with $(\lambda_1, \lambda_2, \lambda_3) = (0, 0, 1)$.*
- *The computation $G_{Di} = GB(J_D + \langle f_1 \rangle)$ results in the ideal $G_{Di} = \langle d+1, b+1, c \rangle$. The ideal-interpolant $J_1 = J_S \cdot G_{Di}$ (same as the J_1 in Example 5.1) is appended to L so that $L = [J_S, J_1]$.*
- *In G_{Di}, the only g_j of the form $\sum_{i=1}^{i=l} \lambda_i \cdot m_i$ are $g_1 = b+1$ and $g_2 = c$.*
- *We select g_1 for the computation of larger interpolant as $g_2 = f_1$. Notice that g_1 is equal to f_5 from Example 5.1.*
- *Using g_1 in the computation $GB(J_D + \langle g_1 \rangle)$ results in the ideal $\langle d+1, b+1 \rangle$. Also the ideal-interpolant $J_5 = J_S \cdot (J_D + \langle g_1 \rangle) = J_S \cdot (J_D + \langle f_5 \rangle)$ is appended to L so that $L = [J_S, J_1, J_5]$.*
- *The only g_i in $GB(J_D + \langle g_1 \rangle) = \langle d+1, b+1 \rangle$ of the form $\sum_{i=1}^{i=l} \lambda_i \cdot m_i$ is $b+1$. As $b+1 = f_5$, the only larger interpolant for J_5 is J_L. Therefore, the algorithm returns the $L = [J_S, J_1, J_5, J_L]$.*

6 Experiments and Discussions

We have implemented Algorithm 1, including the procedures for the computation of J_S, J_L, J_D, and $SM(J_D)$, using the SINGULAR symbolic algebra computation system [ver. 4-1-0][24]. The experiments were conducted on a desktop computer with a 3.5GHz Intel Core$^{\text{TM}}$ i7-4770K Quad-core CPU, 16 GB RAM, running 64-bit Linux OS. The results are presented in Table 1.

We experimented with a set of benchmarks that contain i) random subset-sum problems (*subset-i*), ii) equivalence checking instances between combinational finite field multiplier circuits (Mastrovito vs. Montgomery) [25], and iii) equivalence checking instances between sequential finite field multiplier circuits (Sequential Multipliers with Parallel Outputs (SMPO) [26]). Some of the benchmarks are available as CNF formulas, which were converted to polynomials in $\mathbb{F}_2$, whereas others were directly available as polynomials in $\mathbb{F}_2$ (from [27]).

For the circuit benchmarks, the initial set of polynomial constraints are partitioned into J_A and J_B such that the primary input variables are common to both J_A and J_B, while keeping the generating sets of J_A and J_B as balanced as possible. For the subset-sum problems, the generators of ideal J are randomly partitioned into generators for J_A and J_B such that $C \neq \emptyset$.

Table 1: Results of our experiment with the benchmarks. (*) denotes that the benchmark was available as CNF formulas. T denotes time and is given in seconds.

Benchmarks	#Vars	($\lvert J_A\rvert$,$\lvert J_B\rvert$)	T(J_D)	#$SM(J_D)$	$\lvert L\rvert$ in Alg. 1	T(Alg. 1)	Total T
subset-1*	(8,8,12)	(50,50)	12.7	4,008	3	0.8	15.2
subset-2*	(14,3,16)	(70,71)	101.0	65,501	4	19.6	370.6
subset-3*	(11,8,9)	(59,59)	0.2	505	4	0.1	0.4
MasVMont 2×2*	(3,3,5)	(13,14)	0.0	8	3	0.0	0.0
MasVMont 3×3	(35,39,13)	(42,42)	20.1	7,104	3	7.8	34.2
SMPO 3×3	(18,20,10)	(22,23)	7,749.4	832	3	3.7	7,803.8
SMPO 4×4*	(18,16,17)	(88,88)	1,106.7	122,816	3	27.7	2,281.6
SMPO 5×5*	(24,25,17)	(120,120)	1.6	110,592	3	711.6	1727

In Table 1, #Vars is a 3-tuple $(|A|,|B|,|C|)$ which denotes the sizes of the variable partitions. The number of initial generators of J_A and J_B are presented in column 3. Columns 4 and 5 represent the time for computing (the GB of) J_D, and the number of standard monomials of J_D, respectively. In line 4 in Algorithm 1, we need to pick a polynomial f_i of the form $\sum_{i=1}^{i=l} \lambda_i \cdot m_i$. In our implementation, we randomly create a polynomial $f_i = sm[|sm|/2]+1$, where sm is the list containing all $SM(J_D)$. In line 8 of Algorithm 1, we need to find $g_j \in G_{Di}$, which is again of the form $\sum_{i=1}^{i=l} \lambda_i \cdot m_i$ and is not equal to the current f_i. We go through the polynomials in G_{Di} and pick the first polynomial satisfying these conditions. Columns 6 and 7 in Table 1 denote the number of interpolants returned by our algorithm, and its execution time, respectively. The last column denotes the total time taken by the implementation, which is the aggregate of times required to compute $GB(J_s), GB(J_L), GB(J_D), SM(J_D)$ and Algorithm 1.

The execution time for the benchmark MasVMont 2×2 is mentioned as 0.0 because it is less than 50ms. For most of the benchmarks, a large part of the total time is consumed in computing J_L and J_D, that involves the quotient of ideal operation. From the experiments, we notice that only a few interpolants (besides J_S and J_L) are generated by our algorithm. This depends on the polynomials that we use in lines 4 and 8 of the Algorithm 1. This implies that the algorithm makes huge "jumps" along the height of interpolant lattice, which is equal to $l+1$ ($l = |SM(J_D)|$). We are currently investigating in more detail the relationship between the standard monomial of J_D and $V_C(J_D)$ so that we can develop a better heuristic that provides for a more guided exploration of the interpolant lattice.

These experiments are made available to the reviewers from the website:

`http://eng.utah.edu/~utkarshg/tools.html`, where these design benchmarks and the SINGULAR code is released.

7 Conclusion

This paper has presented a detailed theory and algorithm describing the notion of Craig interpolants for a pair of polynomial ideals in finite fields with no common zeros. The approach utilizes concepts from computational algebraic geometry. Interpolants always exist in this setting, and they correspond to the variety of an elimination ideal. In addition to defining the smallest and the largest interpolants, techniques are described to compute them using Gröbner basis concepts. The total number of interpolants is also determined by counting the number of points in the variety of (set) difference of the largest and the smallest interpolants. Over the field $\mathbb{F}_2$, a technique is presented that can enumerate all possible interpolants. Given an interpolant, a heuristic algorithm is provided that returns a list of progressively larger interpolants, terminating in the largest one. Experiments conducted demonstrate the validity of our results. As part of future work, we are pursuing heuristic based methods to compute interpolants in a more controlled fashion and classify them according to their capability of abstraction.

References

1. W. Craig, "Linear reasoning: A new form of the Herbrand-Gentzen theorem," *Journal of Symbolic Logic*, vol. 22, no. 3, pp. 250–268, 1957.
2. K. L. McMillan, "Interpolation and SAT-Based Model Checking," in *Computer Aided Verification*, July 2003, pp. 1–13.
3. R.-R. Lee, J.-H. R. Jiang, and W.-L. Hung, "Bi-Decomposing Large Boolean Functions via Interpolation and Satisfiability Solving," in *Proc. Design Automation Conference (DAC)*, 2008, pp. 636–641.
4. K. L. McMillan, "An Interpolating Theorem Prover," in *Tools and Algorithms for the Construction and Analysis of Systems*, ser. TCS, vol. 345, no. 1, 2004, pp. 101–121.
5. D. Kapur, R. Majumdar, and G. Zarba, "Interpolation for data-structures," in *Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2006, pp. 105–116.

6. A. Cimatti, A. Griggio, and R. Sebastiani, “Efficient Interpolant Generation in Satisfiability Modulo Theories,” in *Tools Alg. Const. Anal. Sys. (TACAS)*, ser. LNCS, vol. 4963, 2008, pp. 397–412.
7. A. Griggio, “Effective Word-Level Interpolation for Software Verificaion,” in *Formal Methods in CAD (FMCAD)*, 2011, pp. 28–36.
8. S. Gao, A. Platzer, and E. Clarke, “Quantifier Elimination over Finite Fields with Gröbner Bases,” in *Algebraic Informatics: 4th International Conference, CAI*, 2011, pp. 140–157.
9. T. Pruss, P. Kalla, and F. Enescu, “Efficient Symbolic Computation for Word-Level Abstraction from Combinational Circuits for Verification over Finite Fields,” *IEEE Trans. on CAD*, vol. 35, no. 7, pp. 1206–1218, July 2016.
10. X. Sun, P. Kalla, and F. Enescu, “Word-level Traversal of Finite State Machines using Algebraic Geometry,” in *Proc. High-Level Design Validation and Test*, 2016.
11. S. Gopalakrishnan and P. Kalla, “Optimization of Polynomial Datapaths using Finite Ring Algebra,” *ACM Trans. on Design Automation of Electronic Systems, ACM-TODAES*, vol. 7, 2007, article 49.
12. R.-R. Lee, J.-H. R. Jiang, and W.-L. Hung, “To SAT or not to SAT: Ashenhurst Decomposition in a Large Scale,” in *Proc. Intl. Conf. on CAD (ICCAD)*, 2008, pp. 32–37.
13. P. Pudlák, “Lower bounds for resolution and cutting plane proofs and monotone computations,” *J. Symbolic Logic*, vol. 62, no. 2, pp. 981–998, 1997.
14. P. Pudlák and J. Sgall, “Algebraic models of computation and interpolation for algebraic proof systems,” in *Proof Complexity and Feasible Arithmetics, American Mathematical Society*, 1998, pp. 279–296.
15. M. Clegg, J. Edmonds, and R. Impagliazzo, “Using the Gröbner Basis Algorithm to Find Proofs of Unsatisfiability,” in *ACM Symposium on Theory of Computing*, 1996, pp. 174–183.
16. A. Rybalchenko and V. Sofronie-Stokkermans, “Constraint Solving for Interpolation,” in *Proc. Verification, Model Checking and Abstract Interpretation (VMCAI)*, ser. LNCS, no. 4349, 2007, pp. 346–362.
17. L. Kovács and A. Voronkov, “Interpolation and symbol elimination,” in *Proc. Intl. Conf. on Automated Deduction (CADE)*, 2009, pp. 199–213.
18. S. Gao, “Counting Zeros over Finite Fields with Gröbner Bases,” Master’s thesis, Carnegie Mellon University, 2009.
19. V. D’Silva, D. Kroening, M. Purandare, and G. Weissenbacher, “Interpolant strength,” in *Verification, Model Checking and Abstract Interpretation*, ser. LNCS, vol. 5944, 2010, pp. 129–145.
20. G. Weissenbacher, “Interpolant strength revisited,” in *Proc. Intl. Conf. Theory and Applications of Satisfiability Testing*, 2012, pp. 312–326.
21. P. Rümmer and P. Subotić, “Exploring interpolants,” in *Proc. Formal Methods in CAD*, 2013, pp. 69–76.
22. W. W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases*. American Mathematical Society, 1994.
23. D. Cox, J. Little, and D. O’Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2007.
24. W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, “SINGULAR 4-1-0 — A computer algebra system for polynomial computations,” http://www.singular.uni-kl.de, 2016.
25. J. Lv, P. Kalla, and F. Enescu, “Efficient Groebner Basis Reductions for Formal Verification of Galois Field Multipliers,” in *IEEE Design, Automation and Test in Europe*, 2012.
26. G. B. Agnew, R. C. Mullin, I. Onyszchuk, and S. A. Vanstone, “An implementation for a fast public-key cryptosystem,” *Journal of CRYPTOLOGY*, vol. 3, no. 2, pp. 63–79, 1991.
27. X. Sun, I. Ilioaea, P. Kalla, and F. Enescu, “Finding unsatisfiable cores of a set of polynomials using the Gröbner basis algorithm,” in *Intl. Conf. Principles and Practice of Constraint Programming*, ser. LNCS, vol. 9892, Sept. 2016, pp. 859–875.

Appendix: Omitted Proofs

Proof of Theorem 4.2.

Proof. Let $J_I \subseteq \mathbb{F}_q[C]$ be any another ideal-interpolant $\neq J_S$. We show that $V_C(J_S) \subseteq V_C(J_I)$. For $V_C(J_I)$ to be an interpolant it must satisfy

$$V_{A,B,C}(J_A) \subseteq V_{A,B,C}(J_I)$$

which is equivalent to

$$\begin{aligned} I(V_{A,B,C}(J_A)) &\supseteq I(V_{A,B,C}(J_I)) \\ \implies J_A &\supseteq J_I \end{aligned}$$

due to Theorem 3.4. As the generators of J_I only contain polynomials in C-variables, this relation also holds for the following

$$\begin{aligned} J_A \cap \mathbb{F}_q[C] &\supseteq J_I \\ \implies J_S &\supseteq J_I \\ \implies V_C(J_S) &\subseteq V_C(J_I). \end{aligned}$$

□

Proof of Theorem 4.3.

Proof. We first prove that the interpolant computed by complementing $V_C(J'_L)$ as $\mathbb{F}_q^C - V_C(J'_L)$ is indeed a valid interpolant. As J'_L is the elimination ideal computed from J_B, $V_{B,C}(J'_L) \supseteq V_{B,C}(J_B)$. This in turn implies that the complement of $V(J'_L)$ cannot intersect with $V(J_B)$ at any point. This proves condition 2 for $\mathbb{F}_q^C - V_C(J'_L)$ to be a valid interpolant.

For condition 1, we need to prove that

$$V_{A,C}(J_A) \subseteq \mathbb{F}_q^A \times (\mathbb{F}_q^C - V_C(J'_L))$$

This can be restated as

$$V_{A,C}(J_A) \cap \mathbb{F}_q^A \times V_C(J'_L) = \emptyset$$

Let us assume (by contradiction) that there exists a common point $(\mathbf{a},\mathbf{c})$ in $V_{A,C}(J_A)$ and $\mathbb{F}_q^A \times V_C(J'_L)$. As the projection $Pr_B(V_{B,C}(J_B))$ on the C-variables is equal to the variety of the elimination ideal $V_C(J'_L)$, a point $(\mathbf{c}) \in V_C(J'_L)$ can be extended to some point $(\mathbf{b},\mathbf{c})$ in $V_{B,C}(J_B)$. This implies that the point $(\mathbf{a},\mathbf{b},\mathbf{c})$ is a common point in $V_{A,B,C}(J_A)$ and $V_{A,B,C}(J_B)$, which is a contradiction to our initial assumption. Therefore condition 1 of Def. 4.1 is satisfied too and $\mathbb{F}_q^C - V_C(J'_L)$ is indeed an interpolant.

Next we prove that $\mathbb{F}_q^C - V_C(J'_L)$ is the largest interpolant. Consider an arbitrary ideal-interpolant J_I. We want to prove $V_C(J_I) \subseteq \mathbb{F}_q^C - V_C(J'_L)$, or equivalently to prove $V_C(J_I) \cap V_C(J'_L) = \emptyset$. Let us assume (by contradiction) that there exists a common point $(\mathbf{c})$ in $V_C(J_I)$ and $V_C(J'_L)$. As J'_L is the elimination ideal of J_B, this point can be extended to some point $(\mathbf{b},\mathbf{c})$ in $V_{B,C}(J_B)$. This in turn implies that $(\mathbf{b},\mathbf{c})$ is a common point in $V_{B,C}(J_B)$ and $\mathbb{F}_q^B \times V_C(J_I)$. This is a contradiction as an interpolant cannot intersect with the variety of J_B. Hence, $\mathbb{F}_q^C - V_C(J_L')$ is the largest interpolant and it contains all other interpolants.

$\square$

Proof of Lemma 4.1.

Proof. The smallest and the largest interpolants are $V_C(J_S)$ and $V_C(J_L)$, respectively. The set difference $V_C(J_L) - V_C(J_S)$ is also a variety of some ideal J_D, which can be computed as $J_D = (J_L : J_S)$. By selecting different subsets of $V_C(J_D)$ and adding them to $V_C(J_S)$, we can generate all the interpolants. Consider,

$$\binom{|V_C(J_D)|}{0} + \binom{|V_C(J_D)|}{1} + \cdots + \binom{|V_C(J_D)|}{|V_C(J_D)|} = 2^{|V_C(J_D)|}$$

where the term $\binom{|V_C(J_D)|}{0}$ denotes that no point is selected from $V_C(J_D)$ and results in $V_C(J_S)$ as the ideal-interpolant. On the other hand, the term $\binom{|V_C(J_D)|}{|V_C(J_D)|}$ is equivalent to selecting all the points from $V_C(J_D)$ and results in J_L as the ideal-interpolant. So the number of interpolants is equal to $2^{|V_C(J_D)|}$. Theorem 3.3 further tells us that the cardinality of a variety of an ideal is equal to the number of standard monomials of that ideal, therefore, number of interpolants $= 2^{|SM(J_D)|}$.

$\square$