Generating Program Invariants via Interpolation

Marc Moreno Maza[1] and Rong Xiao[2]

[1] University of Western Ontario, Canada moreno@csd.uwo.ca
[2] University of Western Ontario, Canada rong@csd.uwo.ca

Abstract. This article focuses on automatically generating polynomial equations that are inductive loop invariants of computer programs. We propose a new algorithm for this task, which is based on polynomial interpolation. Though the proposed algorithm is not complete, it is efficient and can be applied to a broader range of problems compared to existing methods targeting similar problems. The efficiency of our approach is testified by experiments on a large collection of programs. The current implementation of our method is based on dense interpolation, for which a total degree bound is needed. On the theoretical front, we study the degree and dimension of the invariant ideal of loops which have no branches and where the assignments define a P-solvable recurrence. In addition, we obtain sufficient conditions for non-trivial polynomial equation invariants to exist (resp. not to exist).

1 Introduction

Many researchers have been using computer algebra to compute polynomial loop invariants, see for instance [21, 22, 19, 17, 14, 15, 9, 3, 1, 20, 5, 12]. In this article, we propose an alternative method, based on interpolating polynomials at finitely many points on the reachable set of the loop under study. This interpolation process[3] yields "candidate loop invariants" which are checked by a new criterion based on polynomial ideal membership testing.

Our paper proposes the following original results. On the theoretical front, for P-solvable loops with no branches, we supply a sharp degree bound (Theorem 1) for the invariant ideal, as well as dimension analysis (Theorem 3), of the invariant ideal. We establish a new criterion (Corollary 1) based on polynomial system solving for checking whether or not a given conjunction of polynomial equations is indeed a loop invariant. Meanwhile, Corollary 2 states a sufficient condition for the invariant ideal of a loop to be trivial.

On the algorithmic front, we propose a modular method (Algorithm 2) for generating polynomial loop invariants. Thanks to polynomial interpolation, most of our calculations reduce to linear algebra. As a consequence, the proposed method works in time $n^{d^{\mathcal{O}(1)}}$, where n is the number of loop variables and d is the total degree of polynomials to interpolate.

[3] Note that polynomial interpolation is different from the interpolation in [13], which is called *Craig interpolation* in first order logic.

Our method is probabilistic and may not compute the whole invariant ideal. However, the implementation (in Maple) of our method computes all the invariants given in each example proposed by Enric Rodríguez Carbonell on his page[4]. Moreover, the degree and dimension estimates can help certifying that whole invariant ideal has been obtained. For instance in co-dimension one, the invariant ideal is necessarily principal.

Our method needs not to solve the recurrence relations associated with the loops and thus does not need to manipulate the algebraic numbers arisen as eigenvalues of these recurrence relations. Therefore, all polynomials and matrices involved in our method have their coefficients in the base field. Our method applies to all loops which can be modeled as algebraic transition systems [21] and can be generalized to handle loops which can be modeled as semi-algebraic transition systems [3]. It can be applied to compute all kinds of invariants (see the notions of different loop invariants presented in Section 2), which is not the case for the methods based on "recurrence solving" [20, 12]. In particular, the methods in [20] and [12] apply only to compute absolute inductive invariants, that is, the loop guard and branch conditions are ignored (thus the loop goes to the branches randomly). This means that those methods can not find loop invariants which are not absolute inductive invariants.

Our implementation is tested against the implementation of 3 other methods [20, 5, 12] which were kindly made available to us by their authors. The experimental results are shown in Section 4. While the performance of our method is comparable to that of [5] on the tested (somehow simple) examples, our method has less restrictive specifications than the methods of [20, 12] which target only on absolute polynomial loop invariants for P-solvable loops/solvable mappings. In addition, the method in [20] applies only when all assignments are invertible and the eigenvalues of coefficient matrix for the linear part are positive rationals, and it is claimed to be complete, that is, to compute all absolute polynomial loop invariants, when it is applicable. The method in [12] can be applied all P-solvable loops in theory and is complete for loops without branches. However, in the implementation, the assignment in the loop are required to be not "coupled" together.

Let us conclude this introduction with a brief review of other works on loop invariant computation. In [10], linear equations as invariants of a linear program at each location is considered, by tracking the reachable states with a method based on linear algebra. In [15], the method in [10] is improved and generalized to generate polynomial equations as invariants; it is also shown there that checking whether or not a linear equation is invariant is undecidable in general. In [14], for polynomial programs, the Authors discuss methods based on abstract interpretation, on checking whether or not a given polynomial equation is invariant, as well as generating all polynomial invariants to a given total degree. In [17, 5], a different abstract interpretation technique is developed, which uses polynomial ideal operations (e.g. intersection, quotient) as widen operators. In [9] and [3], quantifier elimination techniques are used to infer invariants from a

[4] `http://www.lsi.upc.edu/~erodri/webpage/polynomial_invariants/list.html`

given template; these methods requires expertise on supplying meaningful templates, while the complexity of quantifier elimination also restrict their practical efficiency. In [21], Gröbner basis together with linear constraint solving is used to infer polynomial equations as invariants.

2 Preliminaries

Let $\mathbb{Q}$ denote the rational numbers and $\overline{\mathbb{Q}}$ the algebraic closure of $\mathbb{Q}$. Let $\mathbb{Q}^*$ (resp. $\overline{\mathbb{Q}}^*$) denote the non zero elements in $\mathbb{Q}$ (resp. $\overline{\mathbb{Q}}$) .

2.1 Notions on loop and loop invariants

We will use the following simple loop (in MAPLE-like syntax) to introduce some notions related to loop and loop invariant that we are going to use.

```
x := a;
y := b;
while x < 10 do
    x := x + y^5;
    y := y + 1;
end do;
```

A *loop variable* of a loop is a variable that is either updated in the loop; or used to initialize/update the values of other loop variables, e.g. x, y, a, b are loop variables. Without loss of generality, we assume that all variables take only rational number values, i.e. from $\mathbb{Q}$. By *initial values* of a loop, we mean all possible tuples of the loop variables before executing the loop; the set of the initial values of the above loop is

$$\{(x, y, a, b) \mid x = a, y = b, (a, b) \in \mathbb{Q}^2\}.$$

Given an initial value $\boldsymbol{v}$, the *trajectory* of the loop starting at $\boldsymbol{v}$, is the sequence of all tuples of of loop variable values *at each entry* of the loop during the execution, with the loop variable being initialized by $\boldsymbol{v}$; the trajectory of the above loop starting at $(x, y, a, b) = (1, 0, 1, 0)$ is

$$(1, 0, 1, 0), (1, 1, 1, 0), (2, 2, 1, 0), (34, 3, 1, 0).$$

The collection of value tuples of all trajectories is called *reachable set* of the loop. Note that, in general, it is hard to describe a reachable set of a loop precisely. A *loop invariant* (or *plain loop invariant*) of a loop is a condition on the loop variables satisfied by all the values in the reachable set of the loop.

By *inductive reachable set* of a loop, we mean the reachable set of the loop while ignoring the guard condition, while by *absolute reachable set* of a loop, we mean the reachable set of the loop while ignoring the guard conditions, the branch conditions and viewing branches to be selected randomly. Then, by an *inductive (loop) invariant* (resp. *absolute (loop) invariant*) of a loop is a condition on the loop variables satisfied by all the tuple values in the inductive (resp. absolute) reachable set of the loop.

It is easy to deduce that an absolute invariant is always an inductive invariant, and an inductive invariant is always a loop invariant. In principle, absolute inductive invariants are easier to study and compute than the inductive invariants and plain invariants. However, the absolute invariants can be trivial, which is not of practical interest to program analysis. See the following example [21] for the case of a trivial absolute invariant while inductive loop invariants in not trivial.

```
y1 := 0;
y2 := 0;
y3 := x1;
while y2 ≠ 0 do
    if y2 + 1 = x2
    then
        y1 := y1 + 1;
        y2 := 0;
        y3 := y3 − 1;
    else
        y2 := y2 + 1;
        y3 := y3 − 1;
    end if
end do
```

Indeed, the condition $y_1x_2 + y_2 + y_3 = x_1$ is an inductive invariant of the above loop. Note there are also loop invariants which are not inductive invariants, e.g. $x - 1 = 0$ is an invariant but not an inductive of the following loop.

```
x := 1;
while x ≠ 1 do
    x := x + 1;
end do
```

On the other hand, the inductive invariants are less likely to be trivial and easier to handle than the loop invariants. In this article, we are interested in the inductive invariants that are given by polynomial equations and that we call *polynomial equation invariants*, or simply polynomial invariants when there is no possible confusion. It is not hard to deduce that all polynomials that are inductive invariants (or loop invariants, or absolute invariants) of a loop form an ideal (which is indeed the ideal of the points in the inductive reachable set), one can also refer [18] for an alternative proof. We call the ideal of polynomials which are inductive invariants of a given loop the *invariant ideal* of the loop.

In this paper, we consider loops of the following shape.

while C_0 **do**
  if C_1
  then
    $X := A_1(X)$;
  elif C_2
  then
    $X := A_2(X)$;
  ...
  elif C_m
  then
    $X := A_m(X)$;
  end if
end do

where

1. $X = x_1, x_2, \ldots, x_s$ is a list of s scalar loop variables, taking values from $\mathbb{Q}$;
2. the initial values of the loop are constrained by polynomial equations and polynomial inequations;
3. the C_i's are pairwise exclusive algebraic conditions (polynomial equations and polynomial inequations) on X;
4. the A_i's are polynomial functions of X with coefficients from $\mathbb{Q}$.

In our loop model, when the loop body contains assignments only (thus no branches), the assignment indeed induces a recurrence relation among the loop variables, which are viewed as recurrence variables. In this case, we shall simply refer to the loop as this recurrence relation and the initial values of the loop. Here we will show briefly how the loop invariant can be computed by explicitly solving the recurrence relation. Later, in our theoretical analysis, presented in Sections 3, we will focus on the study degree and dimension of the invariant ideal of such kind of loops, where the induced recurrence relation is so-called P-solvable recurrence.

Example 1 *Consider the loop computing the sequence of the Fibonacci numbers:*

```
y := 1;
x := 0;
while true do
        (x, y) := (y, x + y);
end while
```

Viewing (x, y) as two recurrences variables, the loop is actually computing the two recurrence sequences of values of x and y defined by the following recurrence relation and initial condition:

$$x(n+1) = y(n), y(n+1) = x(n) + y(n), \text{ with } x(0) = 0, y(0) = 1.$$

We can write down the closed form for $x(n)$ and $y(n)$ as follows:

$$\begin{aligned} x(n) &= \frac{(\frac{\sqrt{5}+1}{2})^n}{\sqrt{5}} - \frac{(\frac{-\sqrt{5}+1}{2})^n}{\sqrt{5}}, \\ y(n) &= \frac{\sqrt{5}+1}{2} \frac{(\frac{\sqrt{5}+1}{2})^n}{\sqrt{5}} - \frac{-\sqrt{5}+1}{2} \frac{(\frac{-\sqrt{5}+1}{2})^n}{\sqrt{5}}. \end{aligned}$$

Let a, u, v be 3 variables. Replace $(\frac{\sqrt{5}+1}{2})^n$ (resp. $(\frac{-\sqrt{5}+1}{2})^n$) by u (resp. by v); replace $\sqrt{5}$ by a. Taking the dependencies $u^2 v^2 = 1, a^2 = 5$ on the new variables into account, the invariant ideal of the loop is

$$\langle x - \frac{au}{5} + \frac{av}{5}, y - a\frac{a+1}{2}\frac{u}{5} + a\frac{-a+1}{2}\frac{v}{5}, a^2 - 5, u^2v^2 - 1\rangle \cap \mathbb{Q}[x, y],$$

which turns out to be $\langle 1 - y^4 + 2xy^3 + x^2y^2 - 2x^3y - x^4\rangle$.

2.2 Poly-geometric summation

As we discussed in the previous subsection, the study of loops without branches can be reduced to the study of recurrence sequences. In this subsection, we recall several well-known notions together with related results adapted to our needs.

Those notions and results can usually be stated in a more general context, e.g. the notion of multiplicative relation can be defined among elements of an arbitrary Abelian group, whereas we define it for a multiplicative group of algebraic numbers.

Definition 1 *Let $\alpha_1, \ldots, \alpha_k$ be k elements of $\overline{\mathbb{Q}}^* \setminus \{1\}$. Let n be a variable taking non-negative integer values. We regard $n, \alpha_1^n, \ldots, \alpha_k^n$ as independent variables and we call $\alpha_1^n, \ldots, \alpha_k^n$ n-exponential variables. Any polynomial of $\overline{\mathbb{Q}}[n, \alpha_1^n, \ldots, \alpha_k^n]$ is called a* poly-geometrical expression in n over $\overline{\mathbb{Q}}$ *w.r.t. $\alpha_1, \ldots, \alpha_k$.*

Let f, g be two poly-geometrical expressions n over $\overline{\mathbb{Q}}$ w.r.t. $\alpha_1, \ldots, \alpha_k$. Given a non-negative integer number i, we denote by $f|_{n=i}$ the evaluation *of f at i, which is obtained by substituting all occurrences of n by i in f. We say that f and g are* equal *whenever $f|_{n=i} = g|_{n=i}$ holds for all non-negative integer i.*

We say that $f(n)$ is in canonical form *if there exist*

(*i*) *finitely many numbers* $c_1, \ldots, c_m \in \overline{\mathbb{Q}}^*$, *and*
(*ii*) *finitely many pairwise different couples* $(\beta_1, e_1), \ldots, (\beta_m, e_m)$ *all in* $(\overline{\mathbb{Q}}^* \setminus \{1\}) \times \mathbb{Z}_{\geq 0}$, *and*
(*iii*) *a polynomial* $c_0(n) \in \mathbb{Q}[n]$,

such that each $\beta_1, \ldots, \beta_m$ *is a product of some of the* $\alpha_1, \ldots, \alpha_k$ *and such that the poly-geometrical expressions* $f(n)$ *and* $\sum_{i=1}^{m} c_i\, \beta_i^n\, n^{e_i} \; + \; c_0(n)$ *are equal. When this holds, the polynomial* $c_0(n)$ *is called the* exponential-free part *of* $f(n)$.

Remark 1 *Note that sometime when referring to poly-geometrical expressions, for simplicity, we allow* n*-exponential terms with base* 0 *or* 1*, that is, terms with* 0^n *or* 1^n *as factors. Such terms will always be evaluated to* 0 *or* 1 *respectively.*

Proving the following result is routine.

Lemma 1 *With the notations of Definition 1. Let* f *a poly-geometrical expression in* n *over* $\overline{\mathbb{Q}}$ *w.r.t.* $\alpha_1, \ldots, \alpha_k$. *There exists a unique poly-geometrical expression* c *in* n *over* $\overline{\mathbb{Q}}$ *w.r.t.* $\alpha_1, \ldots, \alpha_k$ *such that* c *is in canonical form and such that* f *and* c *are equal. We call* c *the* canonical form *of* f.

Example 2 *The closed form* $f := \frac{(n+1)^2\, n^2}{4}$ *of* $\sum_{i=0}^{n} i^3$ *is a poly-geometrical expression in* n *over* $\overline{\mathbb{Q}}$ *without* n*-exponential variables. The expression* $g := n^2\, 2^{(n+1)} - n\, 2^n\, 3^{\frac{n}{2}}$ *is a poly-geometrical in* n *over* $\overline{\mathbb{Q}}$ *w.r.t.* 2, 3. *Some evaluations are:* $f|_{(n=0)} = 0, f|_{n=1} = 1, g|_{n=0} = 0, g|_{n=2} = 8$.

Notation 1 *Let* x *be an arithmetic expression and let* $k \in \mathbb{N}$. *Following [6], we call* k-th falling factorial of x *and denote by* $x^{\underline{k}}$ *the product*

$$x\,(x-1)\,\cdots\,(x-k+1).$$

We define $x^{\underline{0}} := 0$. *For* $i = 1, \ldots, k$, *we denote by* $\left\{ {k \atop i} \right\}$ *the number of ways to partition* k *into* i *non-zero summands, that is, the* Stirling number of the second kind *also denoted by* $S(n,k)$. *We define* $\left\{ {k \atop 0} \right\} := 0$. *Finally, we shall make use of the convention* $0^0 = 1$.

Example 3 *The expression* $n^2\, 2^{(n+1)} - n\, 2^n\, 3^{(n/2)}$ *is clearly poly-geometrical in* n *over* $\overline{\mathbb{Q}}$. *Consider now a fixed non-negative integer* k. *The sum* $\sum_{i=1}^{n-1} i^k$ *has* $n-1$ *terms while its closed form [6] below*

$$\sum_{i=1}^{k} \left\{ {k \atop i} \right\} \frac{n^{\underline{i+1}}}{i+1}$$

has a fixed number of terms and thus is poly-geometrical in n *over* $\overline{\mathbb{Q}}$.

The following result is proved in [6].

Lemma 2 *Let* x *be an algebraic expression and let* $k \in \mathbb{N}$. *Then we have*

$$x^k = \sum_{i=1}^{k} \left\{ {k \atop i} \right\} x^{\underline{i}}.$$

Notation 2 *Let $r \in \overline{\mathbb{Q}}$ and let k be a non-negative integer. We denote by $H(r,k,n)$ the following symbolic summation*

$$H(r,k,n) := \sum_{i=0}^{n-1} r^i \, i^{\underline{k}}.$$

One can easily check that $H(r,0,n) = \frac{r^n-1}{r-1}$ holds for $r \neq 1$. Moreover, we have the following result.

Lemma 3 *Assume $r \neq 0$. Then, we have*

$$(r-1)\, H(r,n,k) \;=\; (n-1)^{\underline{k}}\, r^n - r\, k \; H(r,k-1,n-1). \tag{1}$$

Moreover, we have

(i) if $r = 1$, then $H(r,n,k)$ equals to $\frac{n^{\underline{k+1}}}{k+1}$, which is a polynomial in n over $\overline{\mathbb{Q}}$ of degree $k+1$.
(ii) if $r \neq 1$, then $H(r,n,k)$ has a closed form like $r^n\, f(n) + c$, where $f(n)$ is a polynomial in n over $\overline{\mathbb{Q}}$ of degree k and c is a constant in $\overline{\mathbb{Q}}$.

Proof: We can verify Relation (1) by expanding $H(r,n,k)$ and $H(r,k-1,n-1)$. Now let us show the rest of the conclusion. First, assume $r = 1$. With Relation (1), we have

$$k\, H(r,k-1,n-1) = (n-1)^{\underline{k}}.$$

Therefore, we deduce

$$H(r,n,k) = \frac{n^{\underline{k+1}}}{k+1}.$$

One can easily check that $\frac{n^{\underline{k+1}}}{k+1}$ is a polynomial in n over $\overline{\mathbb{Q}}$ and $\deg(s,n) = k+1$.

From now on assume $r \neq 1$. We proceed by induction on k. When $k = 0$, we have $H(r,0,n) = \frac{r^n-1}{r-1}$. We rewrite $\frac{r^n-1}{r-1}$ as

$$r^n \, \frac{1}{r-1} - \frac{1}{r-1},$$

which is such a closed form. Assume there exists a closed form $r^n\, f_{k-1}(n) + c_{k-1}$ for $H(r,k-1,n)$, where $f_{k-1}(n)$ is a polynomial in n over $\overline{\mathbb{Q}}$ of degree $k-1$. Define

$$s := \frac{(n-1)^{\underline{k}}\, r^n - r\, k\, (r^{n-1}\, f_{k-1}(n-1) + c_{k-1})}{r-1}.$$

It is easy to verify that s is a closed form of $H(r,n,k)$. We rewrite s as

$$r^n \, \frac{(n-1)^{\underline{k}} - k\, f_{k-1}(n-1)}{r-1} - \frac{r\, k\, c_{k-1}}{r-1},$$

and one can check the later form satisfies the requirements of (ii) in the conclusion. This completes the proof. $\square$

Lemma 4 *Let $k \in \mathbb{N}$ and let λ be a non zero algebraic number over $\mathbb{Q}$. Consider the symbolic summation*

$$S := \sum_{i=1}^{n} i^k \lambda^i.$$

1. *if $\lambda = 1$, then there exists a closed form $s(n)$ for S, where s is a polynomial in n over $\overline{\mathbb{Q}}$ of degree $k+1$.*
2. *if $\lambda \neq 1$, then there exists a closed form $\lambda^n s(n) + c$ for S, where s is a polynomial in n over $\overline{\mathbb{Q}}$ of degree k and $c \in \overline{\mathbb{Q}}$ is a constant.*

PROOF: By Lemma 2, we deduce

$$\begin{aligned} \textstyle\sum_{i=1}^{n} i^k \lambda^i &= \textstyle\sum_{i=1}^{n} \left(\sum_{j=1}^{k} \left\{ {k \atop j} \right\} i^{\underline{j}}\right) \lambda^i \\ &= \textstyle\sum_{j=1}^{k} \left(\left\{ {k \atop j} \right\} \sum_{i=1}^{n} i^{\underline{j}} \lambda^i\right) \\ &= \textstyle\sum_{j=1}^{k} \left(\left\{ {k \atop j} \right\} H(\lambda, j, n)\right) \end{aligned}$$

Then, the conclusion follows from Lemma 3. □

The following definition of multiplicative relation specializes the general definition of multiplicative relation to non-zero algebraic numbers.

Definition 2 (Multiplicative relation) *Let k be a positive integer. Let $A := (\alpha_1, \ldots, \alpha_k)$ be a sequence of k non-zero algebraic numbers over $\mathbb{Q}$ and $\mathbf{e} := (e_1, \ldots, e_k)$ be a sequence of k integers. We say that $\mathbf{e}$ is a* multiplicative relation *on A if $\prod_{i=1}^{k} \alpha_i^{e_i} = 1$ holds. Such a multiplicative relation is said* non-trivial *if there exists $i \in \{1, \ldots, n\}$ such that $e_i \neq 0$ holds. If there exists a non-trivial multiplicative relation on A, then we say that A is* multiplicatively dependent*; otherwise, we say that A is* multiplicatively independent*.*

All multiplicative relations of A form a lattice, called the *multiplicative relation lattice* on A, which can effectively be computed, for instance with the algorithm proposed by G. Ge in his PhD thesis [7].

For simplicity, we need the following generalized notion of multiplicative relation ideal, which is defined for a sequence of algebraic numbers that may contain 0 and repeat elements.

Definition 3 *Let $A := (\alpha_1, \ldots, \alpha_k)$ be a sequence of k algebraic numbers over $\mathbb{Q}$. Assume w.l.o.g. that there exists an index ℓ, with $1 \leq \ell \leq k$, such that $\alpha_1, \ldots, \alpha_\ell$ are non-zero and $\alpha_{\ell+1}, \ldots, \alpha_k$ are all zero. We associate each α_i with a variable y_i, where $y_1, \ldots, y_k$ are different from each other. We call the* multiplicative relation ideal of A associated with variables $y_1, \ldots, y_k$*, the binomial ideal of $\mathbb{Q}[y_1, y_2, \ldots, y_k]$ generated by*

$$\{ \prod_{j \in \{1,\ldots,\ell\},\, v_j > 0} y_j^{v_j} - \prod_{i \in \{1,\ldots,\ell\},\, v_i < 0} y_i^{-v_i} \;\mid\; (v_1, \ldots, v_\ell) \in Z\}$$

and $\{y_{\ell+1}, \ldots, y_k\}$, *denoted by* $\mathrm{MRI}(A; y_1, \ldots, y_k)$, *where* Z *is the multiplicative relation lattice on* $(\alpha_1, \ldots, \alpha_\ell)$. *When no confusion is possible, we shall omit writ-ting down the associated variables* $y_1, \ldots, y_k$.

Lemma 5 *Let* $\alpha_1, \ldots, \alpha_k$ *be* k *multiplicatively independent elements of* $\overline{\mathbb{Q}}$ *and let* n *be a non-negative integer variable. Let* $f(n)$ *be a poly-geometrical expression in* n *w.r.t.* $\alpha_1, \ldots, \alpha_k$. *Assume that* $f|_{(n=i)} = 0$ *holds for all* $i \in \mathbb{N}$. *Then,* f *is the zero polynomial of* $\overline{\mathbb{Q}}[n, \alpha_1^n, \ldots, \alpha_k^n]$.

The following definition will be convenient in later statements.

Definition 4 (Weakly multiplicative independence) *Let* $A := (\alpha_1, \ldots, \alpha_k)$ *be a sequence of* k *non-zero algebraic numbers over* $\mathbb{Q}$ *and let* $\beta \in \overline{\mathbb{Q}}$. *We say* β *is* weakly multiplicatively independent *w.r.t.* A, *if there exist no non-negative integers* $e_1, e_2, \ldots, e_k$ *such that* $\beta = \prod_{i=1}^{k} \alpha_1^{e_i}$ *holds. Furthermore, we say that* A *is* weakly multiplicatively independent *if*

(*i*) $\alpha_1 \neq 1$ *holds, and*
(*ii*) α_i *is weakly multiplicatively independent w.r.t.* $\{\alpha_1, \ldots, \alpha_{i-1}, 1\}$, *for all* $i = 2, \ldots, s$.

It is not hard to prove the following lemma on the shape of closed form solutions of single-variable linear recurrences involving poly-geometrical expressions. For the proof, we need the following lemma, which is easy to check, see for instance [16].

Lemma 6 *Let* n *a variable holding non-negative integer values. Let* a *and* b *be two sequences in* $\mathbb{Q}$ *indexed by* n. *Consider the following recurrence equation of variable* x:

$$x(n) = a(n-1)\, x(n-1) + b(n-1).$$

Then we have

$$x(n) = \prod_{i=0}^{n-1} a(i) \left(x(0) + \sum_{j=0}^{n-1} \frac{b(j)}{\prod_{s=0}^{j} a(s)} \right).$$

Lemma 7 *Let* $\alpha_1, \ldots, \alpha_k$ *be* k *elements in* $\overline{\mathbb{Q}}^* \setminus \{1\}$. *Let* $\lambda \in \overline{\mathbb{Q}}^*$. *Let* $h(n)$ *be a poly-geometrical expression in* n *over* $\overline{\mathbb{Q}}$ *w.r.t.* $\alpha_1, \ldots, \alpha_k$. *Consider the following single-variable recurrence relation* R:

$$x(n+1) = \lambda x(n) + h(n).$$

Then, there exists a poly-geometrical expression $s(n)$ *in* n *over* $\overline{\mathbb{Q}}$ *w.r.t.* $\alpha_1, \ldots, \alpha_k$ *such that we have*

$$\deg(s(n), \alpha_i^n) \leq \deg(h(n), \alpha_i^n) \quad \text{and} \quad \deg(s(n), n) \leq \deg(h(n), n) + 1,$$

and such that

– *if* $\lambda = 1$ *holds, then* $s(n)$ *solves* R,
– *if* $\lambda \neq 1$ *holds, then there exists a constant* c *depending on* $x(0)$ *(that is, the initial value of* x*) such that* $c\,\lambda^n + s(n)$ *solves* R.

Moreover, in both cases, if the exponential-free part of the canonical form of $(\frac{1}{\lambda})^n\,h(n)$ *is* 0*, then we can further require that* $\deg(s(n),n) \leq \deg(h(n),n)$ *holds.*

Proof: By Lemma 6, we have

$$x(n) = \lambda^n \left(x(0) + \sum_{j=0}^{n-1} \frac{h(j)}{\lambda^{j+1}} \right). \tag{2}$$

Denote by $\texttt{terms}(h)$ all the terms of the canonical form of $h(n)$. Assume each $t \in \texttt{terms}(h)$ is of form

$$c_t\, n^{q_t}\, \beta_t^n,$$

where c_t is a constant in $\overline{\mathbb{Q}}$, q_t is a non-negative integer and β_t is a product of finitely many elements (with possible repetitions) from $\{\alpha_1, \ldots, \alpha_k\}$. Define $g(n) := \frac{h(n)}{\lambda^{n+1}}$. Then $g(n)$ is a poly-geometrical expression in n w.r.t. $\{\beta_t\}_{t\in\texttt{terms}(h)}, \frac{1}{\lambda}$. Clearly we have

$$g(n) = \sum_{t\in\texttt{terms}(h(n))} \frac{c_t}{\lambda}\, n^{q_t}\, (\frac{\beta_t}{\lambda})^n.$$

Therefore, we have

$$\sum_{j=0}^{n-1} \frac{h(j)}{\lambda^{j+1}} = \sum_{t\in\texttt{terms}(h)} \sum_{j=0}^{n-1} \frac{c_t}{\lambda}\, j^{q_t}\, (\frac{\beta_t}{\lambda})^j. \tag{3}$$

According to Lemma 4, for each $t \in \texttt{terms}(h)$, we can find a poly-geometrical expression

$$s_t := (\frac{\beta_t}{\lambda})^n f_t(n) + a_t$$

in n over $\overline{\mathbb{Q}}$ w.r.t. $\frac{\beta_t}{\lambda}$ satisfying

1. $s_t = \sum_{j=0}^{n-1} \frac{c_t}{\lambda}\, j^{q_t}\, (\frac{\beta_t}{\lambda})^j$;
2. f_t is a polynomial in n over $\overline{\mathbb{Q}}$ of degree q_t (if $\beta_t \neq \lambda$) or $q_t + 1$ (if $\beta_t = \lambda$), and a_t is a constant in $\overline{\mathbb{Q}}$; note in the later case, $c_t\, n^{q_t}\, (\frac{\beta_t}{\lambda})^n$ is a summand of the constant term of the canonical form of $(\frac{1}{\lambda})^n\, h(n)$ is 0 when viewed as a polynomial of the n-exponential variables.

Therefore, using s_t ($\forall t \in \texttt{terms}(h)$), we can simplify the right hand side of Equation (2) to

$$\left(x(0) + \sum_{t\in\texttt{terms}(h)} a_t \right) \lambda^n + \sum_{t\in\texttt{terms}(h)} f_t(n)\, \beta_t^n. \tag{4}$$

Assume for each $t \in \texttt{terms}(h)$, we have $\beta_t = \alpha_1^{e_{t,1}}\, \alpha_1^{e_{t,2}} \cdots \alpha_1^{e_{t,k}}$.
Define

$$\beta_t(n) := (\alpha_1^n)^{e_{t,1}}\, (\alpha_1^n)^{e_{t,2}} \cdots (\alpha_1^n)^{e_{t,k}},$$

$$c := x(0) + \sum_{t \in \texttt{terms}(h)} a_t \quad \text{and} \quad s(n) := \sum_{t \in \texttt{terms}(h)} f_t(n)\, \beta_t(n).$$

It is easy to deduce $\deg(s(n), \alpha_i^n) = \max_{t \in \texttt{terms}(h)}(\deg(\beta_t(n), \alpha_i^n) \le \deg(h(n), \alpha_i^n)$. Finally, one can easily verify that c and $s(n)$ satisfy the requirements in the conclusion. □

Remark 2 *In Lemma 7, if λ is weakly multiplicatively independent w.r.t. $\alpha_1, \ldots, \alpha_k$, then we know that the exponential-free part of the canonical form of $(\frac{1}{\lambda})^n\, h(n)$ is 0, without computing the canonical form explicitly.*

2.3 Degree preliminaries

In this subsection, we review some notions and results on the degree of algebraic varieties. Up to our knowledge, Proposition 1 is a new result which provides a degree estimate for an ideal of a special shape and which can be applied to degree estimate of loop invariant ideals. Throughout this subsection, let $\mathbb{K}$ be an algebraically closed field. Let F be set of polynomials of $\mathbb{K}[x_1, x_2, \ldots, x_s]$. We denote by $V_{\mathbb{K}^s}(F)$ (or simply by $V(F)$ when no confusion is possible) the zero set of the ideal generated by $F \subset \mathbb{K}[x_1, x_2, \ldots, x_s]$ in $\mathbb{K}^s$.

Definition 5 *Let $V \subset \mathbb{K}^s$ be an r-dimensional equidimensional algebraic variety. The number of points of intersection of V with an $(n-r)$-dimensional generic linear subspace $L \subset \mathbb{K}^s$ is called the* degree *of V [4], denoted by* $\deg(V)$*. The degree of a non-equidimensional variety is defined to be the sum of the degrees of its equidimensional components. The degree of an ideal $I \subseteq \mathbb{K}[x_1, x_2, \ldots, x_s]$ is defined to be the degree of the variety of I in $\mathbb{K}^s$.*

We first review a few well-known lemmas. Note that, for a zero-dimensional algebraic variety, the degree is just the number of points in that variety.

Lemma 8 *Let $V \subset \mathbb{K}^s$ be an r-dimensional equidimensional algebraic variety of degree δ. Let L be an $(n-r)$-dimensional linear subspace. Then, the intersection of L and V is either of positive dimensional or consists of no more than δ points.*

Lemma 9 *Let $V \subset \mathbb{K}^s$ be a algebraic variety. Let L be a linear map from $\mathbb{K}^s$ to $\mathbb{K}^k$. Then we have $\deg(L(V)) \le \deg(V)$.*

Lemma 10 ([8]) *Let $I \subset \mathbb{Q}[x_1, x_2, \ldots, x_s]$ be a radical ideal of degree δ. Then there exist finitely many polynomials in $\mathbb{Q}[x_1, x_2, \ldots, x_s]$ generating I and such that each of this polynomial has total degree less than or equal to δ.*

Lemma 11 *Let* $V := W \cap_{i=1}^{e} V_i$ *with* $\dim(W) = r$. *Then we have*

$$\deg(V) \leq \deg(W) \max(\{\deg(V_i) \mid i = 1 \cdots e\})^r.$$

Proposition 1 *Let* $X = x_1, x_2, \ldots, x_s$ *and* $Y = y_1, y_2, \ldots, y_t$ *be pairwise different* $s + t$ *variables. Let* M *be an ideal in* $\mathbb{Q}[Y]$ *of degree* d_M *and dimension* r. *Let* $f_1, f_2, \cdots, f_s$ *be* s *polynomials in* $\mathbb{Q}[Y]$, *with maximum total degree* d_f. *Denote by* I *the ideal* $\langle x_1 - f_1, x_2 - f_2, \ldots, x_s - f_s \rangle$. *Then the ideal* $J := I + M$ *has degree upper bounded by* ${d_M\, d_f}^r$.

PROOF: We assume first that M is equidimensional. Let $L := l_1, l_2, \ldots, l_r$ be r linear forms in X, Y such that the intersection of the corresponding r hyperplanes and $V(J)$ consists of finitely many points, i.e. $H_L := J + \langle L \rangle$ is zero dimensional. By virtues of Lemma 8, the degree of J equals the maximal degree of H_L among all possible choices of linear forms $l_1, l_2, \ldots, l_r$ satisfying the above conditions.

Let $L^* := l_1^*, l_2^*, \ldots, l_r^*$, where each l_j^* $(j = 1 \cdots r)$ is the polynomial obtained by substituting x_i with f_i, for $i = 1 \cdots s$, in the polynomials l_j. Consider the ideal $L^* + M$ in $\mathbb{Q}[Y]$. It is easy to show that the canonical projection map Π_Y onto the space of Y coordinates is a one-one-map between $V_{\mathbb{C}^t}(M + L^*)$ and $\Pi_Y(V_{\mathbb{C}^{t+s}}(H_L))$. Therefore, $V_{\mathbb{C}^t}(M + L^*)$ is zero dimensional and $\deg(M + L^*) = \deg(H_L)$. Hence, viewing $V_{\mathbb{C}^t}(M + L^*)$ as

$$V_{\mathbb{C}^t}(M) \bigcap_{j=1}^{r} V_{\mathbb{C}^t}(l_j^*)$$

and thanks to Lemma 11, we have $\deg(V_{\mathbb{C}^t}(M + L^*)) \leq d_M\, d_f^r$. Therefore, we deduce that $\deg(J) = \max_L \deg(M + L^*) \leq d_M\, d_f^r$ holds, by Lemma 8.

Assume now that $V_{\mathbb{C}^t}(M)$ is not necessarily equidimensional. Let $V_1, V_2, \cdots, V_k$ be an irredundant equidimensional decomposition of $V_{\mathbb{C}^t}(M)$, with corresponding radical ideals $P_1, P_2, \ldots, P_k$. Then, applying the result proved in the first part of the proof to each $I + P_i$ $(i = 1 \cdots k)$, we deduce

$$\begin{aligned} \deg(J) &= \textstyle\sum_{i=1}^{k} \deg(I + P_i) \\ &\leq \textstyle\sum_{i=1}^{k} \deg(P_i)\, d_f^{r_i} \\ &\leq \textstyle\sum_{i=1}^{k} \deg(P_i)\, d_f^{r} \\ &= d_M\, d_f^r, \end{aligned}$$

where r_i is the dimension of P_i in $\mathbb{Q}[Y]$. This completes the proof. □

Remark 3 *For* J *in Proposition 1, a less tight degree bound*

$$d_M\, d_f^{r+s}$$

can easily be deduced from a generalized form of Bezout's bound, since $\deg(V_{\mathbb{C}^{t+s}}(M))$ *has degree* d_M *and if of dimension* $r + s$ *in* $\mathbb{C}^{t+s}$.

Example 4 *Consider* $M := \langle n^2 - m^3 \rangle$, $g_1 := x - n^2 - n - m$, $g_2 := y - n^3 - 3n + 1$, *and the ideal* $J := M + \langle g_1, g_2 \rangle$*. The ideal* M *has degree* 3*, and is of dimension* 1 *in* $\mathbb{Q}[n, m]$*. The degree of* J *is* 9*, which can be obtained by computing the dimension of*

$$\mathbb{Q}(a, b, c, d, e)[x, y, m, n]/(J + \langle a\,x + b\,y + c\,n + d\,m + e \rangle),$$

where a, b, c, d, e *are indeterminates. The degree bound estimated by Proposition 1 is* 3×3*, which agrees with the true degree.*

3 Invariant ideal of P-solvable recurrences

In this section, we focus on loops with no branches, where the study of loop invariants of such loops reduces to the study of algebraic relations among the recurrence variables. In particular, we are interested in those whose assignments induce a called P-solvable recurrence. We will first formalize the notion of P-solvable recurrence. Then in the rest of this section, we will investigate the shape of the closed form solutions of a P-solvable recurrence equation, for studying the degree and the dimension of invariant ideal. We will provide degree estimates for the the invariant ideal, which is useful for all invariant generation methods which need a degree bound, like the proposed polynomial interpolation based method and those in [14, 15, 5]. Last but not least, we will investigate the dimension of the invariant ideal. So that we can get a sufficient for non-trivial polynomial invariants of a given P-solvable recurrence to exist. Note that in our invariant generation method, we do not need (thus never compute) the closed form solutions explicitly.

A "solvable" recurrence relation is, literally, a recurrence relation which can be solved by a closed formula depending only on the index number. The P-solvable recurrence relations have poly-geometrical expressions (Definition 1) as closed form solutions, which is equivalent to the notion of *solvable mapping* in [18] or *solvable loop* in [12] in the respective contexts.

Definition 6 (Univariate P-solvable recurrence) *Given a recurrence* R : $x(n+1) = \lambda\, x(n) + f(n)$ *in* $\mathbb{K}$*, if* $f(n)$ *is a poly-geometrical expression in* n *over* $\mathbb{K}$*, then* R *is called* univariate P-solvable recurrence*.*

A multivariate recurrence is called P-solvable recurrence, if the recurrence variables can essentially (may need a linear coordinate change) be solved out one by one from P-solvable univariate recurrences We can define multivariate P-solvable recurrence as follows.

Definition 7 (P-solvable recurrence) *Let* $n_1, \ldots, n_k$ *be positive integers and define* $s := n_1 + \cdots + n_k$*. Let* M *be a square matrix over* $\mathbb{Q}$ *of order* s*. We assume* *that M is block-diagonal with the following shape:*

$$M := \begin{pmatrix} \mathbf{M}_{n_1\times n_1} & \mathbf{0}_{n_1\times n_2} & \ddots & \mathbf{0}_{n_1\times n_k} \\ \mathbf{0}_{n_2\times n_1} & \mathbf{M}_{n_2\times n_2} & \ddots & \mathbf{0}_{n_2\times n_k} \\ \ddots & \ddots & \ddots & \ddots \\ \mathbf{0}_{n_k\times n_1} & \mathbf{0}_{n_k\times n_2} & \ddots & \mathbf{M}_{n_k\times n_k} \end{pmatrix}.$$

Consider an s-variable recurrence relation R in the variables $x_1, x_2, \ldots, x_s$ and with the following form:

$$\begin{pmatrix} x_1(n+1) \\ x_2(n+1) \\ x_3(n+1) \\ \vdots \\ x_s(n+1) \end{pmatrix} = M \times \begin{pmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ \vdots \\ x_s(n) \end{pmatrix} + \begin{pmatrix} \mathbf{f}_{1_{n_1\times 1}} \\ \mathbf{f}_{2_{n_2\times 1}} \\ \mathbf{f}_{3_{n_3\times 1}} \\ \vdots \\ \mathbf{f}_{k_{n_k\times 1}} \end{pmatrix},$$

*where $\mathbf{f}_1$ is a vector of length n_1 with coordinates in $\mathbb{Q}$ and where $\mathbf{f}_i$ is a tuple of length n_i with coordinates in the polynomial ring $\mathbb{Q}[x_1, \ldots, x_{n_1+\cdots+n_{i-1}}]$, for $i = 2, \ldots, k$. Then, the recurrence relation R is called P-*solvable *over $\mathbb{Q}$ and the matrix M is called the* coefficient matrix *of R.*

It is known that the solutions to P-solvable recurrences are poly-geometrical expressions in n w.r.t. the eigenvalues of the matrix M, see for example [18]. However, we need to estimate the "shape", e.g. the degree of those poly-geometrical expression solutions, with the final goal of estimating the "shape" (e.g. degree, height, dimension) the invariant ideal. In this paper, we focus on degree and dimension estimates.

We first generalize the result of Lemma 7 to the multi-variable case.

Proposition 2 *Let $\alpha_1, \ldots, \alpha_m \in \overline{\mathbb{Q}}^* \setminus \{1\}$. Let $\lambda \in \overline{\mathbb{Q}}$ and $M \in \overline{\mathbb{Q}}^{s\times s}$ be a matrix in the following Jordan form*

$$\begin{pmatrix} \lambda & 0 & 0 & \cdots & 0 & 0 \\ 1 & \lambda & 0 & \cdots & 0 & 0 \\ 0 & 1 & \lambda & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \lambda & 0 \\ 0 & 0 & 0 & \cdots & 1 & \lambda \end{pmatrix}.$$

Consider an s-variable recurrence R defined as follows:

$$X(n+1)_{s\times 1} = M_{s\times s}\, X(n)_{s\times 1} + F(n)_{s\times 1}, \text{ where}$$

(a) *$X := x_1, x_2, \ldots, x_s$ are the recurrence variables;*

(b) $F := (f_1, f_2, \ldots, f_s)$ *is a list of poly-geometrical expression in* n *w.r.t.* $\alpha_1, \ldots, \alpha_m$, *with maximal total degree* d.

Then we have:

1. *if* $\lambda = 0$, *then* $(f_1, f_1 + f_2, \ldots, f_1 + f_2 + \cdots + f_s)$ *solves* R.
2. *if* $\lambda = 1$, *then there exist* s *poly-geometric expressions* $(g_1, g_2, \ldots, g_s)$ *in* $\alpha_1, \ldots, \alpha_m$ *such that for each* $i \in 1 \cdots s$, g_i *is a poly-geometrical expression in* n *w.r.t.* $\alpha_1, \ldots, \alpha_m$ *with total degree less or equal than* $d + i$.
3. *if* $\lambda \notin \{0, 1\}$, *then there exists a solution of* R, *say* $(y_1, y_2, \ldots, y_s)$, *such that for each* $i = 1, \ldots, s$ *we have*

$$y_i := c_i \lambda_i^n + g_i, \text{ where} \tag{5}$$

for each $i \in 1 \cdots s$: *(a)* c_i *is a constant depending only on the initial value of the recurrence; and (b)* g_i *is like in the case of* $\lambda = 1$. *Moreover, assume further more that the following conditions hold:*
   (i) λ *is weakly multiplicatively independent w.r.t.* $\alpha_1, \ldots, \alpha_m$;
   (ii) $\deg(f_j, n) = 0$ *holds for all* $j \in \{1, 2, \ldots, s\}$.
   Then, for all $i = 1, \ldots, s$, *we can further choose* g_i *such that* $\deg(g_i, n) = 0$ *holds and the total degree of* g_i *is less or equal than* $\max(d, 1)$.

Proof: We observe that the recurrence variables of R can be solved one after the other, from x_1 to x_s. When $\lambda = 0$, the conclusion is easy to verify. The case $\lambda \neq 0$ is easy to prove by induction on s with Lemma 7. □

Proposition 3 *Let* $\lambda_1, \ldots, \lambda_s, \alpha_1, \ldots, \alpha_m \in \overline{\mathbb{Q}}^* \setminus \{1\}$. *Let* $M \in \overline{\mathbb{Q}}^{s\times s}$ *be a matrix in the following Jordan form*

$$\begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 & 0 \\ \epsilon_{2,1} & \lambda_2 & 0 & \cdots & 0 & 0 \\ 0 & \epsilon_{3,2} & \lambda_3 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \lambda_{s-1} & 0 \\ 0 & 0 & 0 & \cdots & \epsilon_{s,s-1} & \lambda_s \end{pmatrix},$$

where for $i = 2, \ldots, s$, $\epsilon_{i,i-1}$ *is either* 0 *or* 1. *Consider an* s*-variable recurrence* R *defined as follows:*

$$X(n+1)_{s\times 1} = M_{s\times s}\, X(n)_{s\times 1} + F(n)_{s\times 1},$$

where

1. $X := x_1, x_2, \ldots, x_s$ *are the recurrence variables;*
2. $F := (f_1, f_2, \ldots, f_s)$ *is a list of poly-geometrical expression in* n *w.r.t.* $\alpha_1, \ldots, \alpha_m$, *with maximal total degree* d.

Then there exists a solution of R, say $(y_1, y_2, \ldots, y_s)$, such that for each $i = 1, \ldots, s$ we have

$$y_i := c_i \lambda_i^n + g_i, \tag{6}$$

where

(a) c_i *is a constant depending only on the initial value of the recurrence and*
(b) g_i *is a poly-geometrical expression in n w.r.t.* $\lambda_1, \ldots, \lambda_{i-1}, \alpha_1, \ldots, \alpha_m$ *with total degree less or equal than* $d + i$.

Assume further more that the following conditions hold:

(i) $\lambda_1, \lambda_2, \ldots, \lambda_s$ *is weakly multiplicatively independent;*
(ii) $\deg(f_j, n) = 0$ *holds for all* $j \in \{1, 2, \ldots, s\}$.

Then, for all $i = 1, \ldots, s$, we can further choose y_i such that $\deg(g_i, n) = 0$ *holds and the total degree of g_i is less or equal than* $\max(d, 1)$.

Proof: We observe that the recurrence variables of R can be solved one after the other, from x_1 to x_s. We proceed by induction on s. The case $s = 1$ follows directly from Lemma 7. Assume from now on that $s > 1$ holds and that we have found solutions $(y_1, y_2, \ldots, y_{s-1})$ for the first $s - 1$ variables satisfying the requirements, that is, Relation (6) with (a) and (b). We define

$$\tilde{f}(n) = f_s(n) - \epsilon_{s,s-1}\, y_{s-1}(n+1). \tag{7}$$

Note that $\tilde{f}(n)$ is a poly-geometrical expression in n w.r.t. $\lambda_1, \ldots, \lambda_{s-1}, \alpha_1, \ldots, \alpha_m$ with total degree less than or equal to $d+s-1$. Moreover, for $v \in \{n, \lambda_1^n, \ldots, \lambda_{s-1}^n, \alpha_1^n, \ldots, \alpha_m^n\}$ we have

$$\deg(\tilde{f}(n), v) \leq \max\left(\deg(f_s(n), v), \deg(y_{s-1}(n), v)\right). \tag{8}$$

It remains to solve x_s from

$$x_s(n+1) = \lambda_s\, x_s(n) + \tilde{f}(n) \tag{9}$$

in order to solve all the variables $x_1, \ldots, x_s$. Again, by Lemma 7, there exists a poly-geometrical expression

$$y_s := c_s\, \lambda_s^n + g_s(n),$$

where $g_s(n)$ is poly-geometrical expression in n w.r.t. $\lambda_1, \ldots, \lambda_{s-1}, \alpha_1, \ldots, \alpha_m$, of total degree upper bounded by $d+s$. This completes the proof of the properties (a) and (b) for y_s.

Now we assume that (i), (ii) hold and we prove the second half of the conclusion. Observe that we have $\deg(g_s(n), n) = \deg(\tilde{f}(n), n)$, which is 0, according to Relation (8) and the fact that we can choose y_{s-1} such that $\deg(y_{s-1}(n), n) = 0$ holds. Next, we observe that for each

$$v \in \{n, \lambda_1^n, \ldots, \lambda_{s-1}^n, \alpha_1^n, \ldots, \alpha_m^n\},$$

we have $\deg(g_s(n), v) = \deg(\tilde{f}(n), v)$, which is less or equal to $\deg(y_{s-1}(n), v)$ by Relation (8). Therefore, the total degree of g_s is less or equal than the total degree of y_{s-1}, which is less or equal than $\max(d, 1)$ by our induction hypothesis. This completes the proof. □

Theorem 1 *Let R be a P-solvable recurrence relation. Using the same notations $M, k, s, F, n_1, n_2, \ldots, n_k$ as in Definition 7. Assume M is in a Jordan form. Assume the eigenvalues $\lambda_1, \ldots, \lambda_s$ of M (counted with multiplicities) are different from $0, 1$, with λ_i being the i-th diagonal element of M. Assume for each block j the total degree of any polynomial in $\mathbf{f}_j$ (for $i = 2 \cdots k$) is upper bounded by d_j. For each i, we denote by $b(i)$ the block number of the index i, that is,*

$$\sum_{j=1}^{b(i)-1} n_j < i \leq \sum_{j=1}^{b(i)} n_j. \tag{10}$$

Let $D_1 := n_1$ and for all $j \in \{2, \ldots, k\}$ let $D_j := d_j\, D_{j-1} + n_j$. Then, there exists a solution $(y_1, y_2, \ldots, y_s)$ for R of the following form:

$$y_i := c_i \lambda_i^n + g_i, \tag{11}$$

for all $i \in 1 \cdots s$, where

(a) *c_i is a constant depending only on the initial value of the recurrence;*
(b) *g_i is a poly-geometrical expression in n w.r.t. $\lambda_1, \ldots, \lambda_{i-1}$, and with total degree less or equal than $D_{b(i)}$.*

Moreover, if $\{\lambda_1, \ldots, \lambda_s\}$ is weakly multiplicatively independent, then, for all $i = 1, \ldots, k$, we can further choose y_i such that $\deg(g_i, n) = 0$ holds and the total degree of g_i is less or equal than $\prod_{2 \leq t \leq b(i)} \max(d_t, 1)$.

Proof: We proceed by induction on the number of blocks, that is, k. The case $k = 1$ follows immediately from Proposition 3. Assume from now on that the conclusion holds for a value $k = \ell$, with $\ell \geq 1$ and let us prove that it also holds for $k = \ell + 1$. We apply the induction hypothesis to solve the first ℓ blocks of variables, and suppose that $\mathbf{y}_\ell$ is a solution satisfying the properties in the conclusion. For solving the variables in the $(\ell+1)$-th block, we substitute $\mathbf{y}_\ell$ to $f_{\ell+1}$ and obtain a tuple of poly-geometrical expressions in n w.r.t the eigenvalues of the first ℓ blocks and with total degree bounded by $d_\ell\, D_\ell$. Therefore, applying again Proposition 3, we can find solutions for the variables in the $(\ell+1)$-th block satisfying the properties required in the conclusion. This completes the proof. □

Note that the degree estimate in Theorem 1 depends on how the block structure of the recurrence is exploited, for example, a 2×2 diagonal matrix can be viewed as a matrix with a single block or a matrix with two 1×1 diagonal blocks.

In practice, one might want to decouple the recurrence first, and then study the recurrence variable one by one (after a linear coordinate change) to get better degree estimates for the poly-geometrical expression solutions, regarded as polynomials of n-exponential terms as the eigenvalues of the coefficient matrix. We will just use a simple example to illustrate this idea.

Example 5 *Consider the recurrence:*

$$\begin{pmatrix} x(n+1) \\ y(n+1) \\ z(n+1) \end{pmatrix} := \begin{pmatrix} 2\,0\,0 \\ 0\,3\,0 \\ 0\,0\,3 \end{pmatrix} \times \begin{pmatrix} x(n) \\ y(n) \\ z(n) \end{pmatrix} + \begin{pmatrix} 0 \\ x(n)^2 \\ x(n)^3 \end{pmatrix}$$

Viewing the recurrence as two blocks (x) *and* (y, z)*, the degree estimate according to Theorem 1 would be bounded by* 5 *(*$3 \times 1 + 2$*).*

If we decouple the (y, z) *block to the following two recurrences*

$$y(n+1) = 3\,y(n) + x(n)^2 \text{ and } z(n+1) = 3\,z(n) + x(n)^3,$$

the we can easily deduce that the degree of the poly-geometrical expression for y *and* z *are upper bounded by* 2 *and* 3 *respectively, again according to Theorem 1.*

It is easy to generalize the previous results to the case of a matrix M which is not in Jordan form. Let Q be a non-singular matrix such that $J := Q\,M\,Q^{-1}$ is a Jordan form of M. Let the original recurrence R be

$$X(n+1) = M\,X(n) + F.$$

Consider the following recurrence R_Q

$$Y(n+1) = J\,Y(n) + QF.$$

It is easy to check that if

$$(y_1(n), y_2(n), \ldots, y_s(n))$$

solves R_Q, then

$$Q^{-1}\ (y_1(n), y_2(n), \ldots, y_s(n))$$

solves R. Note that an invertible matrix over $\overline{\mathbb{Q}}$ maps a tuple of poly-geometrical expressions to another tuple of poly-geometrical expressions; moreover it preserves the highest degree among the expressions in the tuple.

We turn now our attention to the question of estimating the degree of the invariant ideal of a P-solvable recurrence relation.

Proposition 4 *Let* R *be an* s*-variable* P*-solvable recurrence relation, with recurrence variables* $(x_1, x_2, \ldots, x_s)$*. Let* $\mathcal{I} \subset \mathbb{Q}[x_1, x_2, \ldots, x_s]$ *be the invariant ideal of* R*. Denote by* $\mathcal{I}^e$ *the extension of* $\mathcal{I}$ *in* $\overline{\mathbb{Q}}[x_1, x_2, \ldots, x_s]$*. Let* $A = \alpha_1, \alpha_2, \ldots, \alpha_s$ *be the eigenvalues (counted with multiplicities) of the coefficient matrix of* R*. Let* $\mathcal{M}$ *be the multiplicative relation ideal of* A *associated with variables* $y_1, \ldots, y_s$*. Then, there exists a sequence of* s *poly-geometrical expressions in* n *w.r.t.* $\alpha_1, \alpha_2, \ldots, \alpha_s$*, say*

$$f_1(n, \alpha_1^n, \ldots, \alpha_k^n), \ldots, f_s(n, \alpha_1^n, \ldots, \alpha_k^n),$$

which solves R. Moreover, we have

$$\mathcal{I}^e = (\mathcal{S}+\mathcal{M}) \cap \overline{\mathbb{Q}}[x_1, x_2, \ldots, x_s],$$

where $\mathcal{S}$ is the ideal generated by $\langle x_1 - f_1(n, y_1, \ldots, y_s), \ldots, x_s - f_s(n, y_1, \ldots, y_s)$ in $\overline{\mathbb{Q}}[x_1, x_2, \ldots, x_s, n, y_1, \ldots, y_s]$.

PROOF: The existence of $f_1, f_2, \ldots, f_s$ follows by Theorem 1 and the fact that linear combination of poly-geometrical expressions w.r.t. n are still poly-geometrical expressions. The conclusion follows from Lemma 5. □

The following lemma is not hard to prove and one can find a proof in [11].

Lemma 12 *Let R be a P-solvable recurrence relation defining s sequences in $\mathbb{Q}^s$, with recurrence variables $(x_1, x_2, \ldots, x_s)$. Let $\mathcal{I}$ be the invariant ideal of R in $\mathbb{Q}[x_1, x_2, \ldots, x_s]$; and let $\overline{\mathcal{I}}$ be the invariant ideal of R in $\overline{\mathbb{Q}}[x_1, x_2, \ldots, x_s]$. Then $\overline{\mathcal{I}}$ equals to $\mathcal{I}^e$, the extension of $\mathcal{I}$ in $\overline{\mathbb{Q}}[x_1, x_2, \ldots, x_s]$.*

With Proposition 4 and Proposition 1, we are able to estimate the degree of polynomials in a generating system of the invariant ideals. Now we are able to estimate the total degree of closed form solutions of a P-solvable recurrence without solving the recurrence explicitly.

Theorem 2 *Let R be a P-solvable recurrence relation defining s sequences in $\mathbb{Q}^s$, with recurrence variables $(x_1, x_2, \ldots, x_s)$. Let $\mathcal{I} \subset \mathbb{Q}[x_1, x_2, \ldots, x_s]$ be the invariant ideal of R. Let $A = \alpha_1, \alpha_2, \ldots, \alpha_s$ be the eigenvalues (counted with multiplicities) of the coefficient matrix of R. Let $\mathcal{M}$ be the multiplicative relation ideal of A associated with variables $y_1, \ldots, y_k$. Let r be the dimension of $\mathcal{M}$. Let $f_1(n, \alpha_1^n, \ldots, \alpha_k^n), \ldots, f_s(n, \alpha_1^n, \ldots, \alpha_k^n)$ be a sequence of s poly-geometrical expressions in n w.r.t. $\alpha_1, \alpha_2, \ldots, \alpha_s$ that solves R. Suppose R has a k block configuration as $(n_1, 1), (n_2, d_2), \ldots, (n_k, d_k)$. Let $D_1 := n_1$; and for all $j \in \{2, \ldots, k\}$, let $D_j := d_j\, D_{j-1} + n_j$. Then we have*

$$\deg(\mathcal{I}) \leq \deg(\mathcal{M})\, D_k^{r+1}.$$

Moreover, if the degrees of n in f_i $(i = 1 \cdots s)$ are 0, then we have

$$\deg(\mathcal{I}) \leq \deg(\mathcal{M})\, D_k^{r}.$$

PROOF:

Denoting by Π the standard projection from $\overline{\mathbb{Q}}^{s+1+s}$ to $\overline{\mathbb{Q}}^{s}$:

$$(x_1, x_2, \ldots, x_s, n, y_1, \ldots, y_s) \mapsto (x_1, x_2, \ldots, x_s),$$

we deduce by Proposition 4 that

$$V(\mathcal{I}) = \overline{\Pi(V(\mathcal{S}+\mathcal{M}))}, \tag{12}$$

where $\mathcal{S}$ is the ideal generated by $\langle x_1 - f_1(n, y_1, \ldots, y_s), \ldots, x_s - f_s(n, y_1, \ldots, y_s)$ in $\overline{\mathbb{Q}}[x_1, x_2, \ldots, x_s, n, y_1, \ldots, y_s]$.

Thus, by Lemma 9, we have

$$\deg(\mathcal{I}) \leq \deg(\mathcal{S} + \mathcal{M}).$$

It follows from Proposition 1 that

$$\deg(\mathcal{S} + \mathcal{M}) \leq \deg(\mathcal{M})\, D_k^{r+1},$$

since the total degree of f_i of R is bounded by D_k according to Theorem 1 and the dimension of $\mathcal{M}$ is $r+1$ is in $\mathbb{Q}[n, y_1, \ldots, y_s]$.

With similar arguments, the second part of the conclusion follows from the fact that $S + M$ can be viewed as an ideal in in $\overline{\mathbb{Q}}[x_1, x_2, \ldots, x_s, n, y_1, \ldots, y_s]$, where M has dimension r. $\square$

Indeed, the degree bound in Theorem 2 is "sharp" in the sense that it is reached by many of the examples we have considered. Let show two of such examples below.

Example 6 (Example 1 Cont.) *The corresponding recurrence only* 1 *block. Denote by* $A := \frac{-\sqrt{5}+1}{2}, \frac{\sqrt{5}+1}{2}$. *One can easily check that* A *is weaklymultiplicatively independent. Note the multiplicative relation ideal of* A *associated with variables* u, v *is generated by* $u^2v^2 - 1$ *and thus has degree* 4 *and dimension* 1 *in* $\mathbb{Q}[u, v]$. *Therefore, by Theorem 2, the degree of invariant ideal bounded by* 4×1^1. *This implies that the degree bound given by Theorem 2 is sharp.*

In the rest of this section, we are going to investigate the dimension of the invariant ideal of P-solvable recurrences. This can help to answer the following natural question: whether or not the invariant ideal of a P-solvable recurrence over $\mathbb{Q}$ is the trivial ideal of $\mathbb{Q}[x_1, \ldots, x_s]$? Note that it is obvious that the invariant ideal is not the whole polynomial ring.

Theorem 3 *Using the same notations as in Definition 7. Let* $\lambda_1, \lambda_2, \ldots, \lambda_s$ *be the eigenvalues of* M *counted with multiplicities. Let* $\mathcal{M}$ *be the multiplicative relation ideal of* $\lambda_1, \lambda_2, \ldots, \lambda_s$. *Let* r *be the dimension of* $\mathcal{M}$. *Let* $\mathcal{I}$ *be the invariant ideal of* R. *Then* $\mathcal{I}$ *is of dimension at most* $r+1$. *Moreover, for generic initial values,*

1. *the dimension of* $\mathcal{I}$ *is at least* r;
2. *if* 0 *is not an eigenvalue of* M *and* $\lambda_1, \lambda_2, \ldots, \lambda_s$ *is weakly multiplicatively independent, then* $\mathcal{I}$ *has dimension* r.

Proof**:** Assume without loss of genericity that M is in Jordan form. By Theorem 1, we deduce that R has a solution $(f_1, f_2, \ldots, f_s)$ as follows

$$\left(c_1\, \lambda_1^n + h_1(n), c_2\, \lambda_2^n + h_2(n), \ldots, c_s\, \lambda_s^n + h_s(n)\right),$$

where for each $i \in 1 \cdots s$, c_i is a constant in $\overline{\mathbb{Q}}$ depending only on the initial value of R, and h_i is a poly-geometrical expression in n w.r.t. $\lambda_1, \ldots, \lambda_{i-1}$. Moreover, we have

1. for generic initial values, none of $c_1, c_2, \ldots, c_s$ is 0;
2. if the eigenvalues of M can be ordered in $\lambda_1, \lambda_2, \ldots, \lambda_s$ s.t. $\lambda_1 \neq 1$ and for each $i \in 2 \cdots s$, λ_i is weakly multiplicatively independent w.r.t. $\lambda_1, \lambda_2, \ldots, \lambda_{i-1}$, then we can require that, for all $i \in 1 \cdots s$, we have $\deg(f_i, n) = 0$.

Viewing n, λ_i^n (for $i = 1, \ldots, s$) as indeterminates, let us associate coordinate variable u_0 to n, u_i to λ_i^n (for $i = 1, \ldots, s$). Denote by V the variety of $\mathcal{I}$ in $\overline{\mathbb{Q}}^s$ (with coordinates $x_1, x_2, \ldots, x_s$). Note that we have

$$\dim(V) = \dim(\mathcal{I}).$$

Denote by W_1, W_2 respectively the variety of $\mathcal{M}$ in $\overline{\mathbb{Q}}^s$ (with coordinates $u_1, u_2, \ldots, u_s$) and in $\overline{\mathbb{Q}}^{s+1}$ (with coordinates $u_0, u_1, u_2, \ldots, u_s$). Note that we have

$$\dim(W_1) = r \text{ and } \dim(W_2) = r + 1.$$

Consider first the map F_0 defined below:

$$F_0 : \overline{\mathbb{Q}}^{s+1} \mapsto \overline{\mathbb{Q}}^{s+1}$$
$$(u_0, u_1, \ldots, u_s) \to (c_1\, u_1 + f_1, \ldots, c_s\, u_s + f_s).$$

By Theorem 2, we have $V = \overline{F_0(W_2)}$. Therefore, we have we have $\dim(\mathcal{I}) = \dim(V) \leq \dim(W_2) = r + 1$.

Now assume the initial value of R is generic, thus we have $c_i \neq 0$, for all $i \in 1 \cdots s$. Let us consider the map F_1 defined below:

$$F_1 : \overline{\mathbb{Q}}^{s+1} \mapsto \overline{\mathbb{Q}}^{s+1}$$
$$(u_0, u_1, \ldots, u_s) \to (u_0, c_1\, u_1 + f_1, \ldots, c_s\, u_s + f_s).$$

Let us denote by V_2 the variety $\overline{F_1(W_2)}$. By virtue of Theorem 2, we have $\dim(V_2) = \dim(W_2) = r + 1$. Denote by Π the standard projection map that forgets the first coordinate, that is, u_0. We observe that $V = \overline{\Pi(V_2)}$. Therefore, we have $\dim(V) \geq \dim(\overline{\Pi(V_2)}) - 1 = r$.

Now we further assume $\lambda_1 \neq 1$ and for each $i \in 2 \cdots s$, λ_i is weakly multiplicatively independent w.r.t. $\lambda_1, \lambda_2, \ldots, \lambda_{i-1}$ the invariant ideal of R. In this case, we have that for all $i \in 1 \cdots s$, $\deg(f_i, n) = 0$. Let us consider the map F_2 defined below:

$$F_2 : \overline{\mathbb{Q}}^s \mapsto \overline{\mathbb{Q}}^s$$
$$(u_1, \ldots, u_s) \to (c_1\, u_1 + f_1, c_2\, u_2 + f_2, \ldots, c_s\, u_s + f_s).$$

By Theorem 2, we have $V = \overline{F_2(W_1)}$. Therefore, we have $\dim(\mathcal{I}) = \dim(V) = \dim(W_1) = r$. This completes the proof. $\square$

The following result, which is a direct consequence of Theorem 3, can serve as a sufficient condition for the invariant ideal to be non-trivial. This condition is often satisfied when there are eigenvalues with multiplicities or when 0 and 1 are among the eigenvalues.

Corollary 1 *Using the same notations as in Theorem 3. If* $r+1 < s$ *holds, then* $\mathcal{I}$ *is not the zero ideal in* $\mathbb{Q}[x_1, x_2, \ldots, x_s]$.

The following corollary indicates that, the fact that the inductive loop invariant is trivia could be determined by just investigating the multiplicative relation among the eigenvalues of the underlying recurrence.

Corollary 2 *Using the same notations as in Theorem 3, consider the corresponding loop* $\mathcal{L}$ *with* $x_1(0) := a_1, \ldots, x_s(0) := a_s$*, where* $a_1, \ldots, a_s$ *are indeterminates. If the eigenvalues of* R *are multiplicatively independent, then the inductive invariant ideal of* $\mathcal{L}$ *is the zero ideal in* $\mathbb{Q}[a_1, \ldots, a_s, x_1, x_2, \ldots, x_s]$.

Proof: Since there is only trivial multiplicative relation, the multiplicative relation ideal of the eigenvalues is 0, which is of dimension s. By Theorem 3, the invariant of R must be zero ideal in $\mathbb{Q}(a_1, \ldots, a_s)[x_1, \ldots, x_s]$, since its dimension must be at least s.

Assume there exists a non-zero invariant polynomial p of $\mathcal{L}$, then p must be an invariant polynomial of R since the loop variables $a_1, \ldots, a_s$ are free to take any value. This is a contradiction to the fact that the invariant ideal of R is trivial. Therefore, the inductive invariant ideal of $\mathcal{L}$ is the zero ideal in $\mathbb{Q}[a_1, \ldots, a_s, x_1, x_2, \ldots, x_s]$. □

Example 7 *Consider the recurrence:*

$$(x(n+1), y(n+1)) := (3\,x(n) + y(n), 2\,y(n)) \text{ with } x(0) = a, y(0) = b.$$

On one hand, the two eigenvalues are 2 *and* 3 *which are multiplicatively independent, therefore, by Corollary 2, the invariant ideal of the corresponding loop is trivial.*

On the other hand, for loop variables (a, b, x, y)*, the reachable set of the loop is*

$$\mathfrak{R} := \{(a,\, b,\, (a+b)\,3^i - b\,2^i,\, b\,2^i) \mid (a, b) \in \mathbb{Q}^2,\ i \text{ is a non-negative integer}\}.$$

Therefore, according to Lemma 5, any polynomial vanishes on all points of $\mathfrak{R}$ *must be* 0.

Note in Theorem 3, if we drop the "generic" assumption on the initial values, then the conclusion might not hold. The following example illustrate this for the case when all the eigenvalues are different and multiplicatively independent, but the invariant ideal is not trivial.

Example 8 *Consider the linear recurrence* $x(n+1) = 3\,x(n) - y(n), y(n+1) = 2\,y(n)$ *with* $(x(0), y(0)) = (a, b)$*. The eigenvalues of the coefficient matrix are* $2, 3$*, which are multiplicatively independent. One can check that, when* $a = b$*, the invariant ideal is generated by* $x - y$*. However, generically, that is when* $a \neq b$ *holds, the invariant ideal is the zero ideal.*

4 Algorithm and experimental results

In this section, we shall discuss how to compute invariant ideals of P-solvable recurrences as well as polynomial loop invariants. Our approach is based on polynomial interpolation and consists essentially of three main steps.

1. Sample a list of points S from the trajectory of the recurrence or loop.
2. Compute all the polynomials vanishing on S up to a certain degree, which can be either a known degree bound or a "guessed" bound.
3. Check whether or not the interpolated polynomials are invariants of the loop.

As one can see from our algorithm sketch, we need to check whether or not a given condition (say a polynomial equation or a polynomial inequality) is an invariant. In general, roughly speaking, when a branch condition contains constraints given by inequalities, the problem of checking whether or not a linear equation is a loop invariant is undecidable, see [15] for a more detailed discussion. Nevertheless, criteria showing that a given condition is indeed an invariant are useful in practice. For this reason, we are interesting necessary or sufficient conditions for a conjunction of polynomial equations to be an invariant of a loop.

4.1 Checking invariants

Proposition 5 states a necessary condition for a set of polynomials to be the invariant ideal of a given loop.

Proposition 5 *Given a loop $\mathcal{L}$ with only one branch and let A be the assignment function. Let I be the inductive invariant ideal of $\mathcal{L}$. Then for any point $\alpha \in V(I)$, we have $A(\alpha) \in V(I)$.*

Proof: Denote by $\mathcal{T}$ the inductive trajectory of $\mathcal{L}$. Let W be the Zariski closure of $A(V(I))$. Let W_1 be the Zariski closure of $W \setminus V(I)$. We proceed by contradiction, thus we assume $W_1 \neq \emptyset$. Then we have $V(I) = A^{-1}(W_1) \cup A^{-1}(V(I) \cap W)$ and $\mathcal{T} \subseteq A^{-1}(V(I) \cap W)$ and $\mathcal{T} \not\subseteq A^{-1}(W_1)$, contradicting the fact that $V(I)$ is the Zariski closure of $\mathcal{T}$. $\square$

The following Proposition, which follows directly from the definition of an inductive invariant, can serve as a sufficient condition for a set of polynomials to be inductive invariants of a given loop.

Proposition 6 *Let $\mathcal{L}$ be a loop with variables X and m branches (C_i, A_i) $i = 1, \ldots, m$. Let $P \subset \mathbb{Q}[X]$. If $V(P)$ contains the initial values of $\mathcal{L}$, and if for each $\alpha \in V(P) \cap Z(C_i)$, we have $A_i(\alpha) \in V(P)$, then all the polynomials in P are inductive polynomial equation invariants of $\mathcal{L}$.*

Note Proposition 6 states a sufficient condition for a set of polynomials to be invariant, not to generate the invariant ideal.

We shall use Proposition 6 as a criterion to certify given polynomials are indeed inductive invariants. Actually, most loop invariant checking criteria work in a similar spirit. The proposed criterion is more general than the various "consecutions" conditions in [21], in the sense that all invariants certifiable by those "consecutions" conditions is certifiable by the proposed criterion, but there are invariants certifiable by Proposition 6, which can not be certified by any of the "consecutions" conditions.

4.2 Implementation of the method

We use polynomial interpolation to construct candidate invariants from a given template (which is either all possible dense polynomials up to a certain degree or a specific form guessed by an oracle). To do so, we need to take sufficiently many points from the trajectory of the program execution. This is done by emulating the program and recording the relevant values. To apply the criterion of Proposition 6, we need to compute the image of a variety under a polynomial map. This is where we use state-of-art computer algebra software tools.

In this section, we describe two algorithms for generating polynomial loop invariants that we have implemented. We refer the first one as our *direct* method.

Notation 3 *Notations in the input of our algorithms:*

(*i*) $M := m_1, m_2, \ldots, m_c$ *is a sequence of monomials in the loop variables* X
(*ii*) $S := s_1, s_2, \ldots, s_r$ *is a set of* r *points on the inductive trajectory of the loop*
(*iii*) E *is a polynomial system defining the loop initial values*
(*iv*) B *is the transitions* $(C_1, A_1), \ldots, (C_m, A_m)$ *of the loop*

The subroutines in Algorithm 1 are explained as follows: `BuildLinSys`(M, S) returns an $r \times c$ matrix L, such that $L_{i,j}$ is the evaluation of the i-th monomial in M at the j-th point in S. `LinSolve`(L) returns a matrix N in row echelon form with full row rank, whose rows generate the null space of L in $\mathbb{Q}^c$. `GenPoly`$(M, \mathbf{v})$ returns the polynomial $\sum_{i=1}^{c} v_i\, m_i$, where $\mathbf{v} = (v_1, v_2, \ldots, v_c)$ is a vector in $\mathbb{Q}^c$.

Note that we can find effective tools for all the operations in Algorithm 1, for instance, we can find tools in [2] for computing the intersection of two constructible sets, or the image of a constructible set under a polynomial map as well as testing the inclusion relation.

However, there is a notable challenge with our direct algorithm (Algorithm 1): the coordinates of the points sampled on the trajectory often grow dramatically in size. This has clearly a negative impact on the solving of the linear system L. All this leads to a severe memory consumption issue, so we decided to consider an algorithm based on modular techniques. We opted for a "small prime" approach, see Algorithm 2, as we observed that many invariants of practical program loops have often small coefficients.

Some additional subroutines, used in Algorithm 2, are specified hereafter: `MaxMachinePrime`() returns the maximum machine-word prime; `PrevPrime`(p) returns the largest prime less than p; `BuildLinSysModp`(M, S, p) returns an $r \times c$ matrix L, such that $L_{i,j}$ is the evaluation of the i-th monomial in M at the j-th point in S modulo p. `LinSolveModp`(L, p) returns a matrix N in row echelon form with full row rank, whose rows generate the null space of L in $\mathbb{Z}_p^c$. `RatRecon`$(\mathbf{N}, \mathbf{P})$ returns a matrix N with rational coefficients, such that for each $i = 1 \ldots k$, the i-th matrix in $\mathbf{N}$ equal to the image of N modulo the i-th prime in $\mathbf{P}$ if possible; otherwise returns FAIL.

Proposition 7 *Both Algorithms 1 and 2 terminate for all inputs. Moreover, when the output is not* FAIL*, it is a list of polynomial equation invariants for the target loop.*

Proof**:** The termination is easy to check since all loops iterate on finitely many terms and each operation (sub-algorithm) does terminate. When the output is not FAIL, that means the output satisfies the sufficient conditions for polynomial invariants stated in Proposition 6 and thus the conclusion follows from Proposition 6. □

Remark 4 *We handle "unlucky primes" by checking the dimension of the solution space (lines* $15-18$ *in Algorithm 2): if the dimension of an image increases, then we drop this image; if a new image has a lower dimension, then we drop all previous images. Several points of Algorithm 2) returns the same "*FAIL*" message, for sake a simplicity. However, we could customize the* FAIL *message in each return point, for examples:*

- *the* FAIL *at line* 9 *implies that either the invariant ideal is the zero ideal or the total degree of interpolated polynomials is too low; or the modulus is too small;*
- *the* FAIL *at line* 23 *means that the product of the chosen moduli is still too small and more images are needed.*

Algorithm 1: `PlainInvInterp`(M, S, B, E)

Input: See Notation 3 for M, S, B, E
Output: A set of polynomial inductive invariants of the target loop
1 $L :=$ `BuildLinSys`(M, S);
2 $N :=$ `LinSolve(L)` ;
3 $F := \emptyset$;
4 **foreach** *row vector* $\mathbf{v} \in N$ **do**
5 $\quad F := F \cup \{$`GenPoly`$(T, \mathbf{v})\}$;
6 **if** $Z(E) \not\subseteq V(F)$ **then** **return** FAIL;
7 **foreach** $(C_i, A_i) \in B$ **do**
8 $\quad$ **if** $A_i(V(F) \cap Z(C_i)) \not\subseteq V(F)$ **then** **return** FAIL;
9 **return** F;

Note that Algorithm 1 and Algorithm 2 will sometimes return FAIL even if the bounds for the polynomial degrees and coefficient sizes are known. When these algorithms return a list of non-trivial polynomials, we are not sure whether those polynomial can generate the whole loop invariant ideal or not. However, in practice, these algorithms often find meaningful results quickly. Indeed, both algorithms run in singly exponential time w.r.t. number of variables for a fix total degree bound, which is stated formally as below.

Proposition 8 *Algorithm 2 runs in singly exponential time w.r.t. number of loop variables.*

PROOF**:** The complexity of Algorithm 2 between Lines 1 and 26 is polynomial in the number of monomials in the support.

The number of those monomials is singly exponential t w.r.t. number of loop variables. In addition, applying our criterion to certify the result (Lines 27 to 30) can be reduced to an ideal membership problem, which is singly exponential w.r.t. number of loop variables.□

In particular, if the total degree bound supplied is greater of equal than the degree of invariant ideal and the sample points are sufficiently many, then with a high possibility (depending on the selection of sample points and also on the choice of the moduli for Algorithm 2), a list of polynomials generating the invariant ideal will be computed by out method.

4.3 Experimental results

We have applied Algorithm 2 to the example programs used in the paper [20], and we are able to find the loop invariants by trying total degree up to 4 for most loops within 60 seconds. See the Table 1 for details.

In the following table, we supply experimental results for computing absolute inductive invariants for some well-known programs from literature as well as some homemade examples marked with a star $*$. The first column labeled by "# vars" is the number of loop variables; the second column labeled by "deg" is the total degree tried for the methods which use a degree bound; the third column labeled by "PI" is the timing of the our method; the fourth column labeled by "AI" is the timing of the method described in [5]; the fifth column labeled by "IF" is the timing of the method described in [20]; the sixth column labeled by "ALIGATOR" is the timing of the method described in [12]. The time unit is second; the "NA" symbol in a time field means that the related method does support the input program; the "FAIL" symbol in a time field means that the output is not "correct". All the tests were done using an Intel Core 2 Quad CPU 2.40GHz with 8.0GB memory.

The following example shows how we can use the degree and dimension information:w to assure that we are computing the whole invariant ideal.

[1] For more details, see `http://www.csd.uwo.ca/~rong/loop_inv.tgz` for the source of all the programs.

[2] There might be a bug in the version of Aligator we are using, because the computation can not finished in 1hr in this test; the timing was reported by Laura Kovacs in a demo of Aligator.

Algorithm 2: `ModpInvInterp`(M, S, B, E, n)

Input: See Notation 3 for M, S, B, E; n is the maximal number of modular images to use

Output: A set of polynomial inductive invariants of the target loop

```
p := MaxMachinePrime();
L := BuildLinSysModp(M, S, p);
N := LinSolveModp(L, p) ;
d := dim(N) ;
N := (N);
P := (p);
i := 1;
while i ≤ n and p > 2 do
    if d = 0 then return FAIL;
    N := RatRecon(N, P);
    if N ≠ FAIL then break;
    p := PrevPrime(p);
    L := BuildLinSysModp(M, S, p);
    N := LinSolveModp(L, p) ;
    if d > dim(N) then
        d := dim(N); N := (N);
        P := (p);
        i := 1;
    else if d = dim(B) then
        N := Append(N, N);
        P := Append(P, p);
        i := i + 1;
if i > n or p = 2 then return FAIL;
F := ∅;
foreach row vector v ∈ N do
    F := F ∪ {GenPoly(T, v)};
if Z(E) ⊈ V(F) then return FAIL;
foreach (C_i, A_i) ∈ B do
    if A_i(V(F) ∩ Z(C_i)) ⊈ V(F) then return FAIL;
return F;
```

Table 1. Experiments on selected programs

prog.[1]	# vars	deg	PI	AI	FP	SE
cohencu	4	3	0.6	0.93	0.28	0.13
cohencu	4	2	0.06	0.76	0.28	0.13
fermat	5	4	3.74	0.79	0.37	0.1
prodbin	5	3	1.4	0.74	0.36	0.13
rk07	6	3	3.1	2.23	NA	0.35
kov08	3	3	0.2	0.57	0.22	0.01
sum5	4	5	12	1.60	2.25	0.16[2]
wensley2	3	3	0.4	0.84	0.39	0.21
int-factor	6	3	60.9	1.28	160.7	0.9
fib(coupled)	4	4	2.4	0.71	NA	NA
fib(decoupled)	6	4	4.3	1.28	160.7	FAIL
non-inv2*	4	3	1.2	3.83	NA	FAIL
coupled-5-1*	4	4	1.1	9.58	NA	NA
coupled-5-2*	5	4	5.38	15.8	NA	NA
mannadiv	3	3	0.1	0.83	NA	0.04

Example 9 *Consider the following recurrence relation on* (x, y, z)*:*

$$\begin{pmatrix} x(n+1) \\ y(n+1) \\ z(n+1) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -3 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} x(n) \\ y(n) \\ z(n) \end{pmatrix}$$

with initial value $(x(0), y(0), z(0)) = (1, 2, 3)$*. Denote by* M *the coefficient matrix. Note that the characteristic polynomial of* M *has* 1 *as a triple root and the multiplicative relation ideal of the eigenvalues is zero-dimensional. So the invariant ideal of this recurrence has dimension either* 0 *or* 1*. On the other hand, we can show that for all* $k \in \mathbb{N}$*, we have* $M^k \neq M$*; so there are infinitely many points in the set* $\{(x(k), y(k), z(k)) \mid k \in \mathbb{N}\}$*, whenever* $(x(0), y(0), z(0)) \neq (0, 0, 0)$*.*

With our method, we are able to compute the following invariant polynomials

$$x + y + z - 6, y^2 + 4yz + 4z^2 - 6y - 24z + 20,$$

which generate a prime ideal of dimension 1 *(thus the invariant ideal of this recurrence), in less than* $0.25s$*.*

5 Concluding remarks

In this article, we propose a loop invariant computing method based on polynomial interpolation. We supply a sharp total degree bound for polynomials generating the loop invariant of P-solvable recurrences. We supply also sufficient conditions for inductive loop invariant to be trivial or non trivial.

The current implementation is for dense interpolation. However, we observe that for loops with sparse polynomials in the assignments, the computed invariants are often sparse too. As future work, we will investigate suitable sparse interpolation techniques for interpolating polynomial loop invariant.

Acknowledgement. We thank Laura Kovávs and Enric Rodríguez-Carbonell for providing the implementations of their invariant generation methods for our test.

References

1. Jacques Carette and Ryszard Janicki. Computing properties of numerical imperative programs by symbolic computation. *Fundam. Inf.*, 80:125–146, January 2007.
2. Changbo Chen, François Lemaire, Liyun Li, Marc Moreno Maza, Wei Pan, and Yuzhen Xie. Computing with constructible sets in maple, 2009. Submitted to J. of Symbolic Computation.
3. Yinghua Chen, Bican Xia, Lu Yang, and Naijun Zhan. Generating polynomial invariants with discoverer and qepcad. In *Formal Methods and Hybrid Real-Time Systems*, pages 67–82, 2007.
4. D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Graduate Text in Mathematics, 185. Springer-Verlag, New-York, 1998.
5. D. Kapur E. Rodriguez-Carbonell. Automatic generation of polynomial invariants of bounded degree using abstract interpretation. *Science of Computer Programming*, 64(1):54–75, 2007.
6. J. von zur Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, 1999.
7. G. Ge. *Algorithms related to multiplicative representations of algebraic numbers.* PhD thesis, U.C. Berkeley, 1993.
8. Joos Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theor. Comput. Sci.*, pages 239–277, 1983.
9. Deepak Kapur. Automatically generating loop invariants using quantifier elimination. In *Deduction and Applications*, 2005.
10. Michael Karr. Affine relationships among variables of a program. *Acta Inf.*, 6:133–151, 1976.
11. Manuel Kauers and Burkhard Zimmermann. Computing the algebraic relations of c-finite sequences and multisequences. *J. Symb. Comput.*, 43:787–803, November 2008.
12. Laura Kovács. Invariant generation for p-solvable loops with assignments. In *Proceedings of the 3rd international conference on Computer science: theory and applications*, CSR'08, pages 349–359, Berlin, Heidelberg, 2008. Springer-Verlag.
13. Laura Kovács and Andrei Voronkov. Interpolation and symbol elimination. In Renate Schmidt, editor, *Automated Deduction CADE-22*, volume 5663 of *Lecture Notes in Computer Science*, pages 199–213. Springer Berlin / Heidelberg, 2009.
14. Markus Müller-Olm and Helmut Seidl. Computing polynomial program invariants. *Inf. Process. Lett.*, 91(5):233–244, September 2004.
15. Markus Müller-Olm and Helmut Seidl. A Note on Karr's Algorithm. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming*, volume 3142 of *Lecture Notes in Computer Science*, pages 1016–1028, Turku, Finland, July 2004. Springer.
16. Martin J. Osborne. Math tutorial: first-order difference equations, 2000.

17. E. Rodríguez-Carbonell and D. Kapur. An Abstract Interpretation Approach for Automatic Generation of Polynomial Invariants. In *International Symposium on Static Analysis (SAS 2004)*, volume 3148 of *Lecture Notes in Computer Science*, pages 280–295. Springer-Verlag, 2004.
18. E. Rodríguez-Carbonell and D. Kapur. Automatic generation of polynomial loop invariants: Algebraic foundations. ISSAC '04, pages 266–273. ACM, 2004.
19. E. Rodríguez-Carbonell and D. Kapur. Program Verification Using Automatic Generation of Invariants. In *1st International Colloquium on Theoretical Aspects of Computing (ICTAC'04)*, volume 3407 of *LNCS*, pages 325–340. Springer-Verlag, 2005.
20. Enric Rodríguez-Carbonell and Deepak Kapur. Generating all polynomial invariants in simple loops. *J. Symb. Comput.*, 42(4):443–476, 2007.
21. Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Non-linear loop invariant generation using gröbner bases. *SIGPLAN Not.*, 39:318–329, 2004 2004.
22. Ashish Tiwari, Harald Rueß, Hassen Saïdi, and Natarajan Shankar. A technique for invariant generation. In *Proceedings of the 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS 2001, pages 113–127, London, UK, UK, 2001. Springer-Verlag.