Linear Loop Synthesis for Quadratic Invariants

S. Hitarth
Hong Kong University of Science and Technology, Hong Kong

George Kenison
Liverpool John Moores University, UK

Laura Kovács
TU Wien, Austria

Anton Varonka
TU Wien, Austria

Abstract

Invariants are key to formal loop verification as they capture loop properties that are valid before and after each loop iteration. Yet, generating invariants is a notorious task already for syntactically restricted classes of loops. Rather than generating invariants for given loops, in this paper we synthesise loops that exhibit a predefined behaviour given by an invariant. From the perspective of formal loop verification, the synthesised loops are thus correct by design and no longer need to be verified.

To overcome the hardness of reasoning with arbitrarily strong invariants, in this paper we construct simple (non-nested) **while** loops with linear updates that exhibit polynomial equality invariants. Rather than solving arbitrary polynomial equations, we consider loop properties defined by a single quadratic invariant in any number of variables. We present a procedure that, given a quadratic equation, decides whether a loop with affine updates satisfying this equation exists. Furthermore, if the answer is positive, the procedure synthesises a loop and ensures its variables achieve infinitely many different values.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Computing methodologies → Equation and inequality solving algorithms; Computing methodologies → Algebraic algorithms

Keywords and phrases program synthesis, loop invariants, verification, Diophantine equations

Digital Object Identifier 10.4230/LIPIcs.STACS.2024.41

Related Version *Extended Version*: `https://doi.org/10.48550/arXiv.2310.05120` [15]

Funding The project leading to this publication has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101034440. The project was also partially supported by the ERC consolidator grant ARTIST 101002685 and by the Vienna Science and Technology Fund (WWTF) [10.47379/ICT19018].

1 Introduction

Linear loops, in their simplicity, constitute a convenient and yet expressive model. From an algebraic point of view, a linear loop corresponds to a system of recurrence relations; solutions of such systems form a robust class in algorithmic combinatorics and algebraic number theory [12, 20]. Linear loops are particularly common in control and digital signal processing software [19]. Note also that the problem of studying the functional behaviour of affine loops (loops with update polynomials of degree 1) can be reduced to that of studying linear loops [28]. Moreover, linear loops can be used to overapproximate the behaviour of more expressive numerical programs, including those with unrestricted control flow and recursive procedures [22].

41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024).
Editors: Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov;
Article No. 41; pp. 41:1–41:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Loop Invariants. While variable updates of linear loops are restricted to linear assignments, it is quite common that linear loops exhibit intricate polynomial properties in the form of polynomial invariants. Non-linear polynomial invariant assertions might come in handy for the verification of safety properties; by approximating the program's behaviour more accurately, they admit fewer false positives. That is, a program verifier using polynomial loop invariants infers less frequently that a true assertion can be violated [7].

Loop Synthesis. Generating invariants, in particular polynomial invariants, is a notorious task, shown to be undecidable for loops with arbitrary polynomial arithmetic [16]. Rather than generating invariants for loops, *in this paper we work in the reverse direction: generating loops from invariants.* Thus we ensure that the constructed loops exhibit intended invariant properties and are thus correct by design. Loop synthesis therefore provides an alternative approach for proving program correctness. If intermediate assertions of an involved program are written in terms of polynomial equalities, automated loop synthesis can provide a code fragment satisfying that assertion, while being correct by construction with respect to the specification.

To overcome hardness of polynomial reasoning and solving arbitrary polynomial equations, we restrict our attention to linear loops, and provide a decision procedure for computing linear loops from (quadratic) polynomial invariants (Algorithm 1).

Linear loop synthesis showcases how a simple model (a linear loop) can express complicated behaviours (quadratic invariants), as also witnessed in sampling algorithms of real algebraic geometry [2, 11]. A non-trivial linear loop for a polynomial invariant allows to sample infinitely many points from the algebraic variety defined by the polynomial. Moreover, the computational cost to generate a new sample point only involves a matrix-vector multiplication. We give further comment on why we do not accept trivial loops in the synthesis process in Remark 2.8.

Thus the result of a loop synthesis process for a polynomial equation (invariant) is an infinite family of solutions defined by recurrence relations. This family is parameterised by n, the number of loop iterations: nth terms of the synthesised recurrence sequences yield a solution of the polynomial equation. Whether the solution set of an equation admits a parameterisation of a certain kind is, in general, an open problem [34, 36].

Our Contributions. The main contributions of this work are as follows:

1. We present a procedure that, given a quadratic equation $P(x_1, \ldots, x_d) = 0$ with an arbitrary number of variables and rational coefficients, generates an affine loop such that $P = 0$ is invariant under its execution; i.e., the equality holds after any number of loop iterations. If such a loop does not exist, the procedure returns a negative answer.
   The values of the loop variables are rational. Moreover, the state spaces of the loops synthesised by this procedure are infinite and, notably, the same valuation of loop variables is never reached twice. The correctness of this procedure is established in Theorem 5.4.
2. If the equation $Q(x_1, \ldots, x_d) = c$ under consideration is such that Q is a quadratic form, we present a stronger result: a procedure (Algorithm 1) that generates a *linear* loop with d variables satisfying the invariant equation.

Paper Outline. Section 2 introduces relevant preliminary material. We defer the discussion of *polynomial equation solving*, a key element of loop synthesis, to Section 3. Then, in Section 4, we provide a method to synthesise *linear loops* for invariants, where the invariants restricted to be equations with *quadratic forms*. We extend these results in Section 5 and present a procedure that synthesises *affine loops*, and hence also linear loops, for invariants that are *arbitrary quadratic* equations. We discuss aspects of our approach and propose further directions in Section 6, in relation to known results.

An extended version of this paper, containing further details on our approach, is available online [15]. In Appendix A of [15], we summarise the procedure for finding isotropic solutions to quadratic forms (which we employ in our synthesis procedure). The abstract arithmetic techniques contained therein are beyond the scope of this short paper and detail the contributions of many sources [18, 9, 33, 32, 26, 6]. In Appendix B of [15], we summarise the synthesis procedure underlying Theorem 5.4.

2 Preliminaries

2.1 Linear and quadratic forms

▶ **Definition 2.1** (Quadratic form). *A d-ary* quadratic form *over the field $\mathbb{K}$ is a homogeneous polynomial of degree 2 with d variables:*

$$Q(x_1, \dots, x_d) = \sum_{i \le j} c_{ij} x_i x_j,$$

where $c_{ij} \in \mathbb{K}$. It is convenient to associate a quadratic form Q with the symmetric matrix:

$$A_Q := \begin{pmatrix} c_{11} & \frac{1}{2}c_{12} & \cdots & \frac{1}{2}c_{1d} \\ \frac{1}{2}c_{12} & c_{22} & \cdots & \frac{1}{2}c_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}c_{1d} & \frac{1}{2}c_{2d} & \cdots & c_{dd} \end{pmatrix}.$$

We note that since A_Q is symmetric, its eigenvalues are all real-valued. Further, $Q(\boldsymbol{x}) = \boldsymbol{x}^\top A_Q \boldsymbol{x}$ for a vector $\boldsymbol{x} = (x_1, \dots, x_d)$ of variables.

We consider quadratic forms over the field $\mathbb{Q}$ of rational numbers by default. Therefore, a quadratic form has a rational quadratic matrix associated with it.

A quadratic form Q is *non-degenerate* if its matrix A_Q is not singular; that is, $\det A_Q \neq 0$. A quadratic form Q over $\mathbb{Q}$ *represents* the value $a \in \mathbb{Q}$ if there exists a vector $\boldsymbol{x} \in \mathbb{Q}^d$ such that $Q(\boldsymbol{x}) = a$. A quadratic form Q over $\mathbb{Q}$ is called *isotropic* if it represents 0 non-trivially; i.e., there exists a non-zero vector $\boldsymbol{x} \in \mathbb{Q}^d$ with $Q(\boldsymbol{x}) = 0$. The vector itself is then called *isotropic*. If no isotropic vector exists, the form is *anisotropic*. A quadratic form Q is called *positive* (resp. *negative*) *definite* if $Q(\boldsymbol{x}) > 0$ (resp. $Q(\boldsymbol{x}) < 0$) for all $\boldsymbol{x} \neq \boldsymbol{0}$. Note that definite forms are necessarily anisotropic.

▶ **Definition 2.2.** *Let Q_1 and Q_2 be d-ary quadratic forms. The forms Q_1 and Q_2 are* equivalent*, denoted by $Q_1 \sim Q_2$, if there exists $\sigma \in \mathrm{GL}_d(\mathbb{Q})$ such that $Q_2(\boldsymbol{x}) = Q_1(\sigma \cdot \boldsymbol{x})$.*

From the preceding definition, there exists an (invertible) linear change of variables over $\mathbb{Q}$ under which representations by Q_2 are mapped to the representations by Q_1. It is clear that two equivalent quadratic forms represent the same values. In terms of matrices, we have $(\sigma\boldsymbol{x})^\top A_{Q_1} \sigma\boldsymbol{x} = \boldsymbol{x}^\top A_{Q_2}\boldsymbol{x}$, and hence $A_{Q_2} = \sigma^\top A_{Q_1} \sigma$.

▶ **Definition 2.3** (Linear form). *A* linear form *in d variables over the field $\mathbb{Q}$ is a homogeneous polynomial $L(x_1, \dots, x_d) = \sum_{i=1}^{d} b_i x_i$ of degree 1, where $b_1, \dots, b_d \in \mathbb{Q}$.*

Note that each linear form admits a vector interpretation: $L(\boldsymbol{x}) = \boldsymbol{b}^\top \boldsymbol{x}$, where $\boldsymbol{b} = (b_1, \dots, b_d)^\top \in \mathbb{Q}^d$ is a non-zero vector of the linear form.

2.2 Loops and Loop Synthesis

Linear loops are a class of single-path loops whose update assignments are determined by a homogeneous system of linear equations in the program variables.

▶ **Definition 2.4** (Linear loop). *A* linear loop $\langle M, \boldsymbol{s}\rangle$ *is a loop program of the form*

$$\boldsymbol{x} \leftarrow \boldsymbol{s};\;\; \textbf{\textit{while}}\; \star\; \textbf{\textit{do}}\; \boldsymbol{x} \leftarrow M\boldsymbol{x},$$

where $\boldsymbol{x}$ is a d-dimensional column vector of program variables, $\boldsymbol{s}$ is an *initial* d-dimensional *vector*, and M is a $d \times d$ *update matrix*. For the procedures, which we introduce here, to be effective, we assume that the entries of M and $\boldsymbol{s}$ are rational.

We employ the notation $\star$, instead of using `true` as loop guard, as our focus is on loop synthesis rather than proving loop termination.

▶ **Definition 2.5** (Affine loop). *An* affine loop $\langle M, \boldsymbol{s}, \boldsymbol{t}\rangle$ *is a loop program of the form*

$$\boldsymbol{x} \leftarrow \boldsymbol{s};\;\; \textbf{\textit{while}}\; \star\; \textbf{\textit{do}}\; \boldsymbol{x} \leftarrow M\boldsymbol{x} + \boldsymbol{t},$$

where, in addition to the previous definition, $\boldsymbol{t} \in \mathbb{Q}^d$ *is a translation vector.*

▶ Remark 2.6 (Linear and Affine Loops). A standard observation permits the simulation of affine loops by linear ones at a cost of one additional variable constantly set to 1. An *augmented matrix* of an affine loop with d variables is a matrix $M' \in \mathbb{Q}^{(d+1)\times(d+1)}$ of the form

$$M' := \left(\begin{array}{c|c} 1 & 0_{1,d} \\ \hline \boldsymbol{t} & M \end{array}\right).$$

It follows that a linear loop $\langle M', (1, \boldsymbol{s})^\mathsf{T}\rangle$ simulates the affine loop in its last d variables.

A linear (or affine) loop with variables $\boldsymbol{x} = (x_1, \ldots, x_d)$ generates d sequences of numbers. For each loop variable x_j, let $\langle x_j(n)\rangle_{n=0}^{\infty} \subseteq \mathbb{Q}$ denote the sequence whose nth term is given by the value of x_j after the nth loop iteration. Similarly, define the sequence of vectors $\langle \boldsymbol{x}(n)\rangle_n \subseteq \mathbb{Q}^d$. For a given loop, we refer to its reachable set of states in $\mathbb{Q}^d$ as the loop's *orbit*. A loop with variables $x_1, \ldots, x_d$ is *non-trivial* if the orbit

$$\mathcal{O}_{\boldsymbol{x}} := \{(x_1(n), \ldots, x_d(n)) : n \geq 0\} \subseteq \mathbb{Q}^d$$

is infinite. A *polynomial invariant* of a loop is a polynomial $P \in \mathbb{Q}[\boldsymbol{x}]$ such that

$$P(x_1(n), \ldots, x_d(n)) = 0$$

holds for all $n \geq 0$.

▶ **Problem 2.7** (Loop Synthesis). *Given a polynomial invariant $P \in \mathbb{Q}[x_1, \ldots, x_d]$, find a non-trivial linear (affine) loop with vector sequence $\langle \boldsymbol{x}(n)\rangle_n$ such that*

$$P(x_1(n), \ldots, x_d(n)) = 0$$

holds for any $n \geq 0$.

We emphasise that, unless stated otherwise, the objective of the loop synthesis process from Problem 2.7 is to find a loop with the same number of variables d as in the input invariant. That is, $\langle \boldsymbol{x}(n)\rangle_n = (\langle x_1(n)\rangle_n, \ldots, \langle x_d(n)\rangle_n)$

Note that $P = 0$ in Problem 2.7 does not need to be an *inductive invariant* for the synthesised loop: We do not require the matrix M to preserve the equality for all vectors $\boldsymbol{x}$. There might still exist a vector $\boldsymbol{s'}$ such that $P(\boldsymbol{s'}) = 0$ but $P(M \cdot \boldsymbol{s'}) \neq 0$. Observe that the search space only expands when we allow non-inductive invariants, thus making our loop synthesis procedures more general.

In summary, the search for an update matrix M (or the augmented matrix M' in the affine loop version of Problem 2.7), is integrally linked to the search of $\boldsymbol{s}$, a solution of the polynomial $P = 0$.

▶ Remark 2.8 (Loop Synthesis and Polynomial Equation Solving). We note that Problem 2.7, Loop Synthesis, relies on, but it is not equivalent to, solving polynomial equations. Indeed, we focus on *non-trivial* loops in Problem 2.7. Allowing loops with finite orbits would mean that a loop with an identity matrix update I_d is accepted as a solution:

$$\boldsymbol{x} \leftarrow \boldsymbol{s};\ \ \textbf{while}\ \star\ \textbf{do}\ \boldsymbol{x} \leftarrow I_d \cdot \boldsymbol{x}.$$

Then, the loop synthesis problem would be equivalent to the problem of finding a rational solution of a polynomial equation $P = 0$ (see Problem 3.1). The problem, as we define it in Problem 2.7, neglects loops that satisfy a desired invariant but reach the same valuation of variables twice. Due to this, the Problem 2.7 of loop synthesis is different from the Problem 3.1 of solving polynomial equations.

3 Solving Quadratic Equations

As showcased in Problem 2.7 and discussed in Remark 2.8, loop synthesis for a polynomial invariant $P = 0$ is closely related to the problem of solving a polynomial equation $P = 0$.

▶ **Problem 3.1** (Solving Polynomial Equations). *Given a polynomial $P \in \mathbb{Q}[x_1, \dots, x_d]$, decide whether there exists a rational solution $(s_1, \dots, s_d) \in \mathbb{Q}^d$ to the equation $P(x_1, \dots, x_d) = 0$.*

We emphasise that determining whether a given polynomial equation has a rational solution, is a fundamental open problem in number theory [29], see also Section 6.1.

Clearly, this poses challenges to our investigations of loops satisfying arbitrary polynomial invariants. In light of this, it is natural to restrict Problem 2.7 to loop invariants given by *quadratic* equations. Given a single equation $P(\boldsymbol{x}) = 0$ of degree 2, the challenge from now on is to find a rational solution $\boldsymbol{s}$ and an update matrix M such that iterative application of M to $\boldsymbol{s}$ of the equation does not violate the invariant: $P(M^n \boldsymbol{s}) = 0$ for all $n \geq 0$.

In this section, we recall well-known methods for solving quadratic equations. In the sequel, we will employ said methods in the novel setting of loop synthesis for quadratic polynomial invariants (Sections 4 and 5).

▶ **Problem 3.2** (Solving Quadratic Equations). *Given a quadratic equation in d variables with rational coefficients, decide whether it has rational solutions. If it does, generate one of the solutions.*

3.1 Solutions of Quadratic Equations in Two Variables

We first prove two lemmas that discuss the solutions of binary quadratic forms in preparation for Section 4.

▶ **Lemma 3.3.** *For all $a, b \in \mathbb{Q} \setminus \{0\}$, Pell's equation $x^2 + \frac{b}{a} y^2 = 1$ has a rational solution (α, β) with $\alpha \notin \{\pm 1, \pm \frac{1}{2}, 0\}$ and $\beta \neq 0$.*

Proof. So long as $a \neq -b$, it is easy to see that $\left(\frac{b-a}{a+b}, \frac{2a}{a+b}\right)$ is a rational solution to Pell's equation. Recall that $a \neq 0$, hence $\beta \neq 0$ and $\alpha \neq \pm 1$. However, the generic solution might have $\alpha = 0$ or $|\alpha| = \frac{1}{2}$. We thus explicitly pick alternative solutions for the cases when it occurs: (i) $x^2 + y^2 = 1$ has another rational point, e.g., $(\frac{3}{5}, \frac{4}{5})$; (ii) $x^2 + 3y^2 = 1$ has a rational point $(-\frac{11}{13}, \frac{4}{13})$; (iii) $x^2 + \frac{1}{3}y^2 = 1$ has a rational point $(\frac{1}{7}, \frac{12}{7})$.

Finally, if $a = -b$, we can take a rational point $(\frac{5}{3}, \frac{4}{3})$ on the hyperbola $x^2 - y^2 = 1$. ◀

▶ **Lemma 3.4.** *An equation $ax^2 + by^2 = c$ with $a, b \in \mathbb{Q} \setminus 0$ has either no rational solutions different from $(0,0)$, or infinitely many rational solutions different from $(0,0)$.*

Proof. Define $R := \begin{pmatrix} \alpha & -\frac{b}{a}\beta \\ \beta & \alpha \end{pmatrix}$ where $(\alpha, \beta) \in \mathbb{Q}^2 \setminus \mathbf{0}$ satisfies $\alpha^2 + \frac{b}{a}\beta^2 = 1$ (is a solution to Pell's equation) for which $\alpha \notin \{\pm 1, \pm\frac{1}{2}, 0\}$ (as in Lemma 3.3). What follows can be viewed as an application of the multiplication principle for the generalised Pell's equation [1]. Observe that if $\boldsymbol{v} = (x, y)^\mathsf{T}$ is a solution to $ax^2 + by^2 = c$, then so is $R\boldsymbol{v}$.

We now show how to generate infinitely many rational solutions to $ax^2 + by^2 = c$ from a single rational solution. Assume, towards a contradiction, that $R^{n+k}\boldsymbol{v} = R^n\boldsymbol{v}$ holds for some $n \geq 0$, $k \geq 1$. Therefore, there exists an integer k such that 1 is an eigenvalue of R^k. Equivalently, there exists a root of unity ω which is an eigenvalue of R. We proceed under this assumption.

By construction, the eigenvalues of R are ω and ω^{-1}. Let φ be the argument of ω. Then the real part of ω, $\cos(\varphi)$, is equal to α (and thus rational). Since ω is a root of unity, φ is a rational multiple of 2π. By Niven's theorem [27], the only rational values for $\cos(\varphi)$ are 0, $\pm\frac{1}{2}$ and ± 1. We arrive at a contradiction, as α was carefully picked to avoid these values.

In summary, we have shown that R has no eigenvalues that are roots of unity, from which we deduce the desired result. ◀

3.2 Solving Isotropic Quadratic Forms

We next present an approach to solving Problem 3.2 that uses the theory of representations of quadratic forms. First, we prove a lemma concerning the representations of 0.

▶ **Lemma 3.5.** *Let $Q(x_1, \ldots, x_n) = a_1x_1^2 + \cdots + a_nx_n^2$ be an isotropic quadratic form with $a_1, \ldots, a_n \neq 0$. There exists a representation $(\alpha_1, \ldots, \alpha_n)$ of 0; i.e., $a_1\alpha_1^2 + \cdots + a_n\alpha_n^2 = 0$ such that $\alpha_1, \ldots, \alpha_n \neq 0$.*

Proof. Let $(\beta_1, \ldots, \beta_n) \in \mathbb{Q}^n$ be a representation of 0 by Q. We further assume that $\beta_1, \ldots, \beta_r \neq 0$ while $\beta_{r+1} = \cdots = \beta_n = 0$, and $r < n$. Moreover, let $\lambda := a_r\beta_r^2 + a_{r+1}\beta_{r+1}^2$.

Consider the equation $x^2 + \frac{a_{r+1}}{a_r}y^2 = 1$. From Lemma 3.3, it has a rational solution (α, β) such that $\alpha, \beta \neq 0$. This implies $a_r\alpha^2 + a_{r+1}\beta^2 = a_r$. The pair $(\beta_r, 0)$ is one solution to $a_rx_r^2 + a_{r+1}x_{r+1}^2 = \lambda$. Following the steps in the proof of Lemma 3.4, we can construct a matrix R for which $R \cdot (\beta_r, 0)^\mathsf{T} = (\alpha\beta_r, \beta\beta_r)^\mathsf{T}$ where $(\alpha\beta_r, \beta\beta_r)$ is a solution of $a_rx_r^2 + a_{r+1}x_{r+1}^2 = \lambda$ with both components being non-zero. Therefore, $(\beta_1, \ldots, \beta_{r-1}, \alpha\beta_r, \beta\beta_r, \beta_{r+2}, \ldots, \beta_n)$ is an isotropic vector of Q with fewer zero entries. By repeating the process, we obtain an isotropic vector $(\alpha_1, \ldots, \alpha_n)$ as desired. ◀

We emphasise that the process of eliminating zeros from the isotropic vector is effective. A similar proof is given in [3, p.294, Theorem 8].

In this discussion, we focus on solving equations of the form $Q(x_1, \ldots, x_d) = c$, where Q is a quadratic form. As it will be shown later in Section 4, it is always possible to find an equivalent diagonal quadratic form $D \sim Q$. Therefore, we restrict our attention to equations of the form $a_1x_1^2 + \cdots + a_dx_d^2 = c$. Assuming $c \neq 0$, we start by homogenising the equation, and so consider the solutions of

$$a_1x_1^2 + \cdots + a_dx_d^2 - cx_{d+1}^2 = 0. \tag{1}$$

In other words, we are searching for a rational isotropic vector of a quadratic form.

▶ **Proposition 3.6.** *An equation*

$$a_1x_1^2 + \cdots + a_dx_d^2 = c \tag{2}$$

has a rational solution different from $(0, \dots, 0)$ *if and only if the quadratic form* $Q = a_1x_1^2 + \cdots + a_dx_d^2 - cx_{d+1}^2$ *has an isotropic vector.*

Proof. For $c = 0$, the statement is a recitation of a definition. We continue under the assumption $c \neq 0$. Recall from Lemma 3.5 that if the form Q is isotropic, then there is an isotropic vector $(\alpha_1, \dots, \alpha_{d+1})$ with $\alpha_i \neq 0$ for all $i \in \{1, \dots, d+1\}$. Therefore, we can find a non-zero solution $(\alpha_1/\alpha_{d+1}, \dots, \alpha_d/\alpha_{d+1})$ to Equation (2). Conversely, if (2) has a non-trivial solution $(\beta_1, \dots, \beta_d)$, it follows that $(\beta_1, \dots, \beta_d, 1)$ is an isotropic vector for Q. ◀

3.3 Finding Isotropic Vectors

Proposition 3.6 implies that solving Problem 3.1, and hence also loop synthesis in Problem 2.7, requires detecting whether a certain quadratic form is isotropic. Effective isotropy tests are known for quadratic forms $Q(x_1, \dots, x_{d+1})$ as in Equation (1). A more difficult task is the problem of finding an isotropic vector for such a form.

The abstract arithmetic techniques employed in finding an isotropic vector are beyond the scope of this paper; however, we give a brief overview of the computational task and a number of references to the literature in the extended version [15, Appendix A]. Our takeaways from the theory, summarised there [18, 9, 33, 32, 26, 6], are the following functions:

- isIsotropic: a function that, given an indefinite quadratic form over the rationals as an input, determines whether the input is isotropic and duly returns the answers yes and no (as appropriate).
- findIsotropic: a function that accepts isotropic quadratic forms over the rationals as inputs and returns an isotropic vector for each such form.
- solve: a function that takes Equation (2) as an input and returns a non-zero solution if the form $a_1x_1^2 + \cdots + a_dx_d^2 - cx_{d+1}^2$ is isotropic; otherwise solve returns "no solutions". The function solve calls both isIsotropic and findIsotropic, see [15] for details.

We note the solve subroutine in the sequel: the function linLoop defined in Algorithm 1, calls on solve; and, in turn, the function linLoop is called by the procedure in Section 5.

4 Quadratic Forms: Linear Loops

The core of this section addresses equations, and hence loop invariants, that involve quadratic forms. The equations (invariants) of this section do not have a linear part; they are quadratic forms equated to constants; that is, equations of the form

$$Q(x_1, \dots, x_d) = c, \tag{3}$$

where Q is an arbitrary d-ary quadratic form with rational coefficients, c is a rational number.

The main result of this section is the following theorem, which establishes a decision procedure that can determine if a given quadratic invariant admits a linear loop and, if so, constructs that loop.

▶ **Theorem 4.1** (Linear Loops for Quadratic Forms). *There exists a procedure that, given an equation $Q(x_1, \dots, x_d) = c$ of the form* (3)*, decides whether a non-trivial linear loop satisfying $Q(x_1, \dots, x_d) = c$ exists and, if so, synthesises a loop.*

We prove Theorem 4.1 in several steps. The first of them is to diagonalise the quadratic form Q and thus reduce to Equation (3) without mixed terms on the left-hand side.

4.1 Rational Diagonalisation

A rational quadratic form can be diagonalised by an invertible change of variables with only rational coefficients.

▶ **Proposition 4.2.** *Let Q be a (possibly degenerate) d-ary quadratic form. There exists an equivalent quadratic form D with a diagonal matrix $A_D \in \mathbb{Q}^{d\times d}$, i.e., $Q \sim D$. Furthermore, $A_D = \sigma^{\mathsf{T}} A_Q \sigma$ holds with $\sigma \in \mathrm{GL}_d(\mathbb{Q})$.*

A diagonalisation algorithm is described in [23, Algorithm 12.1], see also "diagonalisation using row/column operations" in [35, Chapter 7, 2.2]. The idea, as presented in [35], is to perform row operations on the matrix Q. Different from the usual Gauss–Jordan elimination, the analogous column operations are performed after each row operation. We emphasise that the change-of-basis matrix σ is invertible as a product of elementary matrices.

▶ Remark 4.3 (Degeneracy). Let $A_D := \mathrm{diag}(a_1, \dots, a_d)$ be the diagonal matrix of the quadratic form D as in Proposition 4.2. The product $a_1 \cdots a_d$ is zero if and only if the initial quadratic form Q is degenerate.

▶ **Proposition 4.4.** *Let Q_1 and Q_2 be two equivalent d-ary quadratic forms. If there exists a linear loop $\mathcal{L} = \langle M, \boldsymbol{s}\rangle$ with invariant $Q_2 = c$ for a constant $c \in \mathbb{Q}$, then $Q_1 = c$ is an invariant of the linear loop $\mathcal{L}' = \langle \sigma M \sigma^{-1}, \sigma \boldsymbol{s}\rangle$. Here, $\sigma \in \mathrm{GL}_d(\mathbb{Q})$ is a change-of-basis matrix such that $Q_2(\boldsymbol{x}) = Q_1(\sigma \cdot \boldsymbol{x})$.*

Proof. If $(M^n \boldsymbol{s})^{\mathsf{T}} A_{Q_2} (M^n \boldsymbol{s}) = c$ for all $n \geq 0$, then

$$\begin{aligned}\left((\sigma M \sigma^{-1})^n \sigma \boldsymbol{s}\right)^{\mathsf{T}} A_{Q_1} \left((\sigma M \sigma^{-1})^n \sigma \boldsymbol{s}\right) &= (\sigma M^n \boldsymbol{s})^{\mathsf{T}} A_{Q_1} (M^n \boldsymbol{s}) \\ &= \boldsymbol{s}^{\mathsf{T}} (M^n)^{\mathsf{T}} \sigma^{\mathsf{T}} A_{Q_1} \sigma M^n \boldsymbol{s} = \boldsymbol{s}^{\mathsf{T}} (M^n)^{\mathsf{T}} A_{Q_2} M^n \boldsymbol{s} = (M^n \boldsymbol{s})^{\mathsf{T}} A_{Q_2} M^n \boldsymbol{s} = c\end{aligned}$$

for all $n \geq 0$ as well. We emphasise that σ is a bijection from $\mathbb{Q}^d$ to itself, so the reduction described here preserves the infiniteness of loop orbits. ◀

We conclude from Propositions 4.2 and 4.4 that for a general quadratic form Q, a linear loop with an invariant $Q(\boldsymbol{x}) = c$ exists if and only if a linear loop exists for an invariant $D(\boldsymbol{x}) = c$, where D is an equivalent diagonal form.

4.2 Diagonal Quadratic Forms

In this subsection we consider diagonal quadratic forms $a_1 x_1^2 + \cdots + a_d x_d^2 = c$, where $a_1, \dots, a_d, c \in \mathbb{Q}$ as in Equation (2). If the equation is homogeneous; that is, $c = 0$, then loop synthesis reduces to the problem of searching for a rational solution $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)$. Indeed, a loop with a matrix $\lambda \cdot I_d$ (scaling each variable by $\lambda \in \mathbb{Q} \setminus \{-1, 0, 1\}$) and the initial vector $\boldsymbol{\alpha}$ is a non-trivial linear loop satisfying the invariant $Q(\boldsymbol{x}) = 0$.

From Section 3, we know how to generate a solution (or prove there is no solution) to Equation (2) in its general form, also with $c \neq 0$. The bottleneck of loop synthesis in Problem 2.7 is thus finding an update matrix M for the linear loop. En route to solving this issue, we state the following corollary of Lemma 3.4.

▶ **Corollary 4.5.** *If an equation $ax^2 + by^2 = c$ with $a, b \in \mathbb{Q} \setminus 0$ has infinitely many rational solutions different from $(0,0)$, then there exists a non-trivial linear loop with polynomial invariant $ax^2 + by^2 = c$.*

Proof. We use the construction in the proof of Lemma 3.4, which demonstrates that the orbit of the linear loop $\langle R, \boldsymbol{v}\rangle$ is infinite with polynomial invariant $ax^2 + by^2 = c$. ◀

Proof of Theorem 4.1. Due to Proposition 4.4, we can consider an equation of the form (2):

$$a_1x_1^2 + \cdots + a_dx_d^2 = c.$$

We describe the loop synthesis procedure in this case. If $d = 1$, the equation only has finitely many solutions, hence any loop for Equation (2) is trivial. Hereafter we assume that $d \geq 2$.

In order to generate an initial vector of the loop for Equation (2), we exploit the results of Section 3. Either Equation (2) has no rational solutions and hence no loop exists, or we effectively construct a solution $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{Q}^d$ using procedure SOLVE. Recall that we can guarantee $\alpha_i \neq 0$ for all $i \in \{1, \ldots, d\}$ due to Lemma 3.5.

Note that some of the coefficients a_i, $i \in \{1, \ldots, d\}$, may be zero if the original quadratic form Q is degenerate. We have to consider the case when all coefficients but one are 0, separately. That is, $a_1x_1^2 + 0x_2^2 + \cdots + 0x_d^2 = c$. For this form, a solution exists if and only if c/a_1 is a square of a rational number. Subsequently, if a solution $\boldsymbol{\alpha}$ is found, set $M := \operatorname{diag}(1, 2, \ldots, 2)$ to be a diagonal update matrix. Since $d \geq 2$, we guarantee that the orbit of the linear loop $\langle M, \boldsymbol{\alpha}\rangle$ is infinite.

Without loss of generality, we now assume $a_1 \neq 0$ and $a_2 \neq 0$. Define $\gamma := a_1\alpha_1^2 + a_2\alpha_2^2$, then the equation $a_1x_1^2 + a_2x_2^2 = \gamma$ has a non-trivial solution (α_1, α_2).

From Corollary 4.5, there exists a matrix $R \in \mathbb{Q}^{2\times 2}$ that preserves the value of the quadratic form $a_1x_1^2 + a_2x_2^2$. This matrix can be constructed as in the proof of Corollary 4.5 by considering the equation $x_1^2 + \frac{a_2}{a_1}x_2^2 = 1$. Let M be the matrix given by the direct sum

$$R \oplus I_{d-2} = \left(\begin{array}{c|c} R & 0 \\ \hline 0 & I_{d-2} \end{array}\right)$$

where I_n is an identity matrix of size n.

A desired loop is $(M, \boldsymbol{\alpha})$ as for each $n \geq 0$, $M^n\boldsymbol{\alpha}$ satisfies Equation (2). The loop is non-trivial because its orbit, restricted to x_1, x_2, is infinite. ◀

The process of synthesising a loop for the quadratic invariant $Q(x_1, \ldots, x_n) = c$ is summarised in Algorithm 1. The algorithm starts with a diagonalisation step, proceeds with finding a loop for an equation of the form (2), and applies the inverse transformation to obtain a linear loop for the initial invariant. Whenever Algorithm 1 returns a loop, this loop is linear.

5 Arbitrary Quadratic Equations: Affine Loops

In this section, we leave the realm of quadratic forms and consider general quadratic invariants that may have a linear part. Any quadratic equation can be written in terms of a quadratic form Q, a linear form L, and a constant term c:

$$Q(x_1, \ldots, x_d) + L(x_1, \ldots, x_d) = c. \tag{4}$$

On our way to a complete solution of Problem 2.7 for arbitrary quadratic equations, we carefully analyse Equation (4). A standard technique (see e.g. [14, Proposition 1]) allows to reduce Equation (4) with a non-degenerate quadratic form Q to Equation (3) considered

Algorithm 1 Synthesise a linear loop satisfying a given quadratic form equation.

```
Input: quadratic form Q in d variables and c ∈ ℚ. Assert d ≥ 2.
function LINLOOP(Q, s)
    ⟨M, s⟩ := UNDEFINED.
    compute a rational diagonalisation for Q(x): a σ ∈ GL_d(ℚ) such that Q′(σx) = Q(x)
    with Q′ a diagonal quadratic form
    rewrite the equation Q′ = c as a_1 x_1^2 + ··· + a_d x_d^2 = c with a_1, ..., a_r ≠ 0 and
    a_{r+1} = ··· = a_d = 0
    let α := (α_1, ..., α_r, 1, ..., 1)^T ∈ ℚ^d, where (α_1, ..., α_r) := SOLVE(a_1, ..., a_r, c)
                                                          ▷ for SOLVE see [15, Algorithm 2]
    if r = 1 and α ≠ "NO SOLUTIONS" then
        M := diag(1, 2, ..., 2).
    else if α = "NO SOLUTIONS" then
        return "NO LOOP".
    else
        compute a solution (y_1, y_2) of x_1^2 + (a_2/a_1) x_2^2 = 1.        ▷ see Lemma 3.3
        M := R ⊕ I_{d−2}, where R = ( y_1  −(a_2/a_1) y_2 ; y_2  y_1 ).
    end if
    return ⟨σ^{−1} M σ, σ^{−1} α⟩.
end function
```

in Section 4. We now give the details of this reduction and describe how to synthesise an affine loop for an invariant (4) in the non-degenerate case. Subsequently, we close the gap by discussing the case when Q is degenerate. Using Remark 2.6, our results on affine loop synthesis imply then linear loop synthesis.

5.1 Non-Degenerate Quadratic Forms

For convenience, we rewrite the equation in the matrix-vector form: $\boldsymbol{x}^\top A_Q \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} - c = 0$. Here, A_Q is the non-singular matrix of the quadratic form Q, and $\boldsymbol{b}$ is the vector of the linear form. Let $\delta := \det A_Q \neq 0$ and C be the cofactor matrix of A_Q, i.e., $A_Q \cdot C = C \cdot A_Q = \delta \cdot I_d$. We further define $\boldsymbol{h} := C \cdot \boldsymbol{b}$ and $\tilde{c} = 4\delta^2 c + Q(\boldsymbol{h})$. It can be checked directly that

$$Q(2\delta \cdot \boldsymbol{x} + \boldsymbol{h}) = \tilde{c} \Leftrightarrow Q(\boldsymbol{x}) + L(\boldsymbol{x}) = c. \tag{5}$$

In words, every equation of the form Equation (4) can be reduced to an equation of the form $Q(\boldsymbol{y}) = \tilde{c}$ *by an affine transformation* f that maps each $\boldsymbol{x} \in \mathbb{Q}^d$ to $2\delta \cdot \boldsymbol{x} + \boldsymbol{h} \in \mathbb{Q}^d$. As such, this means that solutions of Equation (4) under the non-degeneracy assumption are in a one-to-one correspondence with representations of $\tilde{c}$ for Q.

▶ **Proposition 5.1.** *Let Q be a non-degenerate quadratic form and L a linear form, both in $d \geq 2$ variables. Define $\delta := \det(A_Q)$, $\boldsymbol{h}$ and $\tilde{c}$, as in the discussion above. The following are equivalent:*

1. *There exists a linear loop $\langle M, \boldsymbol{s}\rangle$ satisfying the invariant $Q(\boldsymbol{x}) = \tilde{c}$.*
2. *There exists an affine loop*

$$\langle M, \frac{1}{2\delta}(\boldsymbol{s} - \boldsymbol{h}), \frac{1}{2\delta}(M - I_d)\boldsymbol{h}\rangle,$$

satisfying the invariant $Q(\boldsymbol{x}) + L(\boldsymbol{x}) = c$.

Proof. Start with the first assumption. For all $n \geq 0$, it holds $Q(M^n \boldsymbol{s}) = \tilde{c}$. Equivalently,

$$Q(f^{-1}(M^n \boldsymbol{s})) + L(f^{-1}(M^n \boldsymbol{s})) = c, \text{ or } Q\left(\frac{1}{2\delta}(M^n \boldsymbol{s} - \boldsymbol{h})\right) + L\left(\frac{1}{2\delta}(M^n \boldsymbol{s} - \boldsymbol{h})\right) = c$$

for all $n \geq 0$.

On the other hand, let $\boldsymbol{x}(n)$ be the variable vector after the nth iteration of an affine loop from the statement. We prove by induction that $\boldsymbol{x}(n) = \frac{1}{2\delta}(M^n \boldsymbol{s} - \boldsymbol{h})$. The base case is true since the initial vector of the affine loop is $\frac{1}{2\delta}(\boldsymbol{s} - \boldsymbol{h}) = \frac{1}{2\delta}(M^0 \boldsymbol{s} - \boldsymbol{h})$. Now, assume that $\boldsymbol{x}(k) = \frac{1}{2\delta}(M^k \boldsymbol{s} - \boldsymbol{h})$ for an arbitrary $k \geq 0$. Then, by applying the loop update once, we have

$$\begin{aligned}\boldsymbol{x}(k+1) = M \cdot \left(\frac{1}{2\delta}(M^k \boldsymbol{s} - \boldsymbol{h})\right) + \frac{1}{2\delta}(M - I_d)\,\boldsymbol{h}& \\ = \frac{1}{2\delta}\left(M^{k+1}\boldsymbol{s} - M\boldsymbol{h} + M\boldsymbol{h} - \boldsymbol{h}\right) &= \frac{1}{2\delta}\left(M^{k+1}\boldsymbol{s} - \boldsymbol{h}\right),\end{aligned}$$

and the inductive step has been shown. By the above work, we conclude that $Q(\boldsymbol{x}(n)) + L(\boldsymbol{x}(n)) = c$ holds for all $n \geq 0$. ◀

▶ **Example 5.2.** Consider an invariant $p(x, y) := x^2 + y^2 - 3x - y = 0$. After an affine change of coordinates $f(x, y) = (2x - 3, 2y - 1)$, it becomes $x^2 + y^2 = 10$ (that corresponds to $\delta = 1$, $\boldsymbol{h} = (-3, -1)^\mathsf{T}$, $\tilde{c} = 10$). There exists a linear loop for this equation:

$$M = \begin{pmatrix} \frac{3}{5} & -\frac{4}{5} \\ \frac{4}{5} & \frac{3}{5} \end{pmatrix} \text{ and } \boldsymbol{s} = \begin{pmatrix} 1 \\ -3 \end{pmatrix}.$$

Next, compute the components of an affine loop. The update matrix is M, whilst the initial and translation vectors are

$$\frac{1}{2}\left[\begin{pmatrix} 1 \\ -3 \end{pmatrix} - \begin{pmatrix} -3 \\ -1 \end{pmatrix}\right] = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \qquad \text{and} \qquad \frac{1}{2}\left[\begin{pmatrix} \frac{3}{5} & -\frac{4}{5} \\ \frac{4}{5} & \frac{3}{5} \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right]\begin{pmatrix} -3 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

respectively. The resulting affine loop is non-trivial with invariant $p(x, y) = 0$ due to Proposition 5.1:

$$\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} 2 \\ -1 \end{pmatrix}; \text{ while } \star \text{ do } \begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} \frac{3}{5}x - \frac{4}{5}y + 1 \\ \frac{4}{5}x + \frac{3}{5}y - 1 \end{pmatrix}.$$

5.2 Degenerate Quadratic Forms

Let $r < d$ be the rank of A_Q. There exist $k := d - r$ linearly independent vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{Q}^d$ such that $A_Q \cdot \boldsymbol{v}_i = \boldsymbol{0}$. Construct a matrix $\tau \in \mathrm{GL}_d(\mathbb{Q})$ such that $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ constitute its first columns. It follows that every non-zero entry $(M)_{ij}$ of a matrix $M := \tau^\mathsf{T} A_Q \tau$ is located in the bottom right corner, that is, $i > k$ and $j > k$. We rewrite $Q(\tau\boldsymbol{x}) = \tilde{Q}(x_{k+1}, \ldots, x_d)$ and $L(\tau\boldsymbol{x}) = \tilde{L}(x_{k+1}, \ldots, x_d) + \lambda_1 x_1 + \cdots + \lambda_k x_k$ in Equation (4). Now we have:

$$\tilde{Q}(x_{k+1}, \ldots, x_d) + \tilde{L}(x_{k+1}, \ldots, x_d) = c - \lambda_1 x_1 - \cdots - \lambda_k x_k, \tag{6}$$

where $\tilde{Q}$ is a non-degenerate quadratic form of r variables.

In the rest of this subsection, we are concerned with finding an affine loop satisfying Equation (6). We emphasise that such a loop $\langle M, \boldsymbol{s}, \boldsymbol{t}\rangle$ exists if and only if $\langle \tau M \tau^{-1}, \tau\boldsymbol{s}, \tau\boldsymbol{t}\rangle$ satisfies Equation (4). The proof is due to τ inducing an automorphism of $\mathbb{Q}^d$, cf. Proposition 4.4.

If $\lambda_1 = \cdots = \lambda_k = 0$, we have arrived at an instance of Equation (4) with a non-degenerate quadratic form and fewer variables. Let δ be the determinant of $\tilde{Q}(x_{k+1}, \ldots, x_d)$ and, as in the non-degenerate setting, define an affine transformation f on the subset of variables $\{x_{k+1}, \ldots, x_d\}$. The constant $\tilde{c}$ and the vector $\boldsymbol{h} \in \mathbb{Q}^r$ are defined similarly to their non-degenerate setting counterparts.

After the change of coordinates that corresponds to f, we have

$$0x_1^2 + \cdots + 0x_k^2 + \tilde{Q}(x_{k+1}, \ldots, x_d) = \tilde{c}. \tag{7}$$

Recall (e.g. from the proof of Theorem 4.1) that once Equation (7) with $k \geq 1$ has a solution, there is a non-trivial linear loop satisfying the polynomial invariant defined by the equation. Now, let $\langle M, \boldsymbol{s} \rangle$ be a linear loop for Equation (7), where $\boldsymbol{s} = (s_1, \ldots, s_d)^\top$. In fact, one can assume $M := \mathrm{diag}(2, \ldots, 2, 1, \ldots, 1)$ with k twos and r ones. Define $\boldsymbol{s'} := \frac{1}{2\delta}\left(\boldsymbol{s} - \left(\begin{smallmatrix}\boldsymbol{0}\\ \boldsymbol{h}\end{smallmatrix}\right)\right)$. It is not hard to see that a non-trivial *linear* loop $\langle M, \boldsymbol{s'} \rangle$ satisfies $\tilde{Q}(x_{k+1}, \ldots, x_d) + \tilde{L}(x_{k+1}, \ldots, x_d) = c$ if and only if $\tilde{Q}(x_{k+1}, \ldots, x_d) = \tilde{c}$ has a solution $(s_{k+1}, \ldots, s_d)$.

From now on, we assume that $k \geq 1$ is the number of non-zero λ_i's on the right-hand side of Equation (6). We show next that the loop synthesis question has a positive answer.

▶ **Proposition 5.3** (Affine Loops for Quadratic Forms). *Given a quadratic equation of the form* (6)*, there exists a non-trivial affine loop in variables $x_1, \ldots, x_d$ for which said equation is a polynomial invariant.*

Proof. Since $k \geq 1$ and $\lambda_1 \neq 0$, the right-hand side $c - \sum_{i=1}^k \lambda_i x_i$ represents every rational number. Set the values of $x_{k+1}, \ldots, x_d$ to some fixed values $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{d-k})$ such that $\boldsymbol{\alpha} \neq \boldsymbol{0}$ and solve the equation for $x_1, \ldots, x_k$ attaining a vector of values $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$. We have $\tilde{Q}(\boldsymbol{\alpha}) + \tilde{L}(\boldsymbol{\alpha}) = A(\boldsymbol{\beta})$, where $A(x_1, \ldots, x_k) := c - \lambda_1 x_1 - \cdots - \lambda_k x_k$.

We introduce the following case distinction.

Case 1. $k > 1$;
Case 2. $r > 1$ and so $\tilde{Q}$ is a non-degenerate quadratic form of at least 2 variables;
Case 3. $r = 1$ and $k = 1$; that is, Equation (6) has the form $ax^2 + bx = c - dy$, $d \neq 0$.
In the rest of the proof, we show that for all these cases, a non-trivial affine loop satisfies Equation (6) and hence, the invariant of Equation (4). Moreover, in Cases 1 and 2 there exist *linear* loops of this sort.

In Case 1, we focus on the vector $\boldsymbol{\beta}$ computed in the previous step. Without loss of generality, $(\beta_1, \beta_2) \neq (0, 0)$. We construct a linear loop that preserves the values of all variables but β_1, β_2. To this end, it suffices to notice that a linear transformation of $\mathbb{Q}^2$ defined by $(x_1, x_2) \mapsto (2x_1, -\frac{\lambda_1}{\lambda_2}x_1 + x_2)$ preserves the value of $\lambda_1 x_1 + \lambda_2 x_2$. The desired linear loop has initial vector $\boldsymbol{s} = (\boldsymbol{\beta}, \boldsymbol{\alpha})^\top$ and an update matrix $M = \left(\begin{smallmatrix} 2 & 0 \\ -\frac{\lambda_1}{\lambda_2} & 1 \end{smallmatrix}\right) \oplus I_{d-2}$.

Let us turn to Case 2 and focus on vector $\boldsymbol{\alpha}$. Clearly, we can now assume $k = 1$. Without loss of generality, we shall assume that $\beta_1 \neq 0$. Consider the equation

$$\tilde{Q}(\boldsymbol{x}) + \tilde{L}(\boldsymbol{x}) = A(\beta_1)$$

over the variables $\boldsymbol{x}$, with $A(y) = c - \lambda_1 y$. Using Equation (5), we argue that its solutions are related to the representations of a certain number $\tilde{c}$ by $\tilde{Q}$. We compute δ, $\tilde{c}$, $\boldsymbol{h}$ for the non-degenerate quadratic form $\tilde{Q}$, linear form $\tilde{L}$ and constant $A(\beta_1)$ such that $\tilde{Q}(2\delta \cdot \boldsymbol{x} + \boldsymbol{h}) = \tilde{c}$. From Theorem 4.1 and, more specifically, its proof, observe that there exists a non-trivial *linear* loop satisfying $\tilde{Q}(\,\cdot\,) = \tilde{c}$. Indeed, there exists at least one solution of this equation, namely $f(\boldsymbol{\alpha})$. Let $\langle M, \boldsymbol{s}\rangle$ be a linear loop satisfying $\tilde{Q}(\,\cdot\,) = \tilde{c}$ with matrix $M \in \mathbb{Q}^{r\times r}$. Proposition 5.1 shows that an affine loop

$$\mathcal{A} := \langle M, \frac{1}{2\delta}\left(\boldsymbol{s} - \boldsymbol{h}\right), \frac{1}{2\delta}\left(M - I_r\right)\boldsymbol{h}\rangle,$$

satisfies the invariant $\tilde{Q}(\boldsymbol{x}) + \tilde{L}(\boldsymbol{x}) = A(\beta_1)$. The sequence $\langle \boldsymbol{x}(n)\rangle_{n=0}^{\infty}$ of $\mathcal{A}$'s variable vectors can be expressed in terms of an augmented matrix (see M' in Section 2) associated with the affine transformation $\boldsymbol{x} \mapsto M\boldsymbol{x} + \boldsymbol{t}$, where $\boldsymbol{t} = \frac{1}{2\delta}\left(M - I_r\right)\boldsymbol{h}$ and $\boldsymbol{s}' = \frac{1}{2\delta}\left(\boldsymbol{s} - \boldsymbol{h}\right)$:

$$\begin{pmatrix} 1 \\ \boldsymbol{x}(n) \end{pmatrix} = \left(\begin{array}{c|c} 1 & 0_{1,r} \\ \hline \boldsymbol{t} & M \end{array}\right)^n \begin{pmatrix} 1 \\ \boldsymbol{s}' \end{pmatrix},$$

satisfies $\tilde{Q}(\boldsymbol{x}) + \tilde{L}(\boldsymbol{x}) = A(\beta_1)$ for all $n \geq 0$. Then,

$$\begin{pmatrix} y(n) \\ \boldsymbol{x}(n) \end{pmatrix} = \left(\begin{array}{c|c} 1 & 0_{1,r} \\ \hline \frac{1}{\beta_1}\boldsymbol{t} & M \end{array}\right)^n \begin{pmatrix} \beta_1 \\ \boldsymbol{s}' \end{pmatrix}$$

satisfies $\tilde{Q}(\boldsymbol{x}(n)) + \tilde{L}(\boldsymbol{x}(n)) = A(y(n))$ as in Equation (6) for all $n \geq 0$. We denote by M_β the d-dimensional square matrix in the preceding displayed equation. Observe that $\langle M_\beta, (\beta_1, \boldsymbol{s}')^\mathsf{T}\rangle$ is a *linear* loop satisfying the invariant of Equation (4).

Finally, we come to the special case, Case 3, that considers quadratic equations of the form $ax^2 + bx = c - dy$ where $d \neq 0$. It suffices to observe that an affine transformation of $\mathbb{Q}^2$ defined by $(x, y) \mapsto (2x, 2\frac{b}{d}x + 4y - 3\frac{c}{d})$ preserves the equation $ax^2 + bx = c - dy$. We conclude that $ax^2 + bx = c - dy$ is a polynomial invariant of the affine loop with initial vector $(1, \frac{c-a-b}{d})^\mathsf{T}$, translation vector $(0, -3\frac{c}{d})^\mathsf{T}$, and update matrix $\begin{pmatrix} 2 & 0 \\ 2\frac{b}{d} & 4 \end{pmatrix}$. ◀

5.3 The Procedure: Affine Loop Synthesis for Quadratic Invariants

▶ **Theorem 5.4** (Affine Loops for Quadratic Equations)**.** *There exists an effective procedure that, given a quadratic equation (i.e. invariant)*

$$Q(x_1, \ldots, x_d) + L(x_1, \ldots, x_d) = c,$$

decides whether a non-trivial affine loop satisfying it exists and, if so, synthesises a loop.

The theorem is essentially proved in Propositions 5.1 and 5.3. If the quadratic form is non-degenerate, Proposition 5.1 reduces the search for an affine loop to the search for a linear loop satisfying $Q(x_1, \ldots, x_d) = \tilde{c}$. The solution of this problem was given in Theorem 4.1. If the quadratic form is degenerate, we consider Equation (6). If at least one of the λ_i's is non-zero, a loop exists, as shown by the ad hoc constructions of Proposition 5.3. In two of the three cases there, the loop is not just affine, but linear. Otherwise, if all of the λ_i's are zero, we obtain a linear loop by essentially testing whether a solution to an equation $\tilde{Q}(x_1, \ldots, x_d) = \tilde{c}$ exists. Finally, in order to obtain an affine loop satisfying the original equation, we apply transformation τ to the loop synthesised for Equation (6).

The synthesis procedure is summarised in the extended version, see [15, Appendix B]. By analysing the algorithm, one can argue that a negative output implies that Equation (4) has no solutions. The problem of *deciding* whether a loop exists for a given invariant, as in Problem 2.7 and opposed to the synthesis of numerical values, is thus solved as follows.

▶ **Corollary 5.5.** *Let Q be a quadratic form, L a linear form over variables $\boldsymbol{x} = (x_1, \ldots, x_d)$.*

1. *A non-trivial affine loop satisfying the quadratic equation $Q(\boldsymbol{x}) + L(\boldsymbol{x}) = c$ exists if and only if the equation has a rational solution different from $\boldsymbol{x} = \boldsymbol{0}$.*
2. *A non-trivial linear loop satisfying the equation $Q(\boldsymbol{x}) = c$ exists if and only if the equation has a rational solution different from $\boldsymbol{x} = \boldsymbol{0}$.*

▶ **Example 5.6.** Let $-11x^2 + y^2 - 3z^2 + 2xy - 12xz + x + z = -1$ be a quadratic invariant in 3 variables. The quadratic form $Q(x, y, z) = -11x^2 + y^2 - 3z^2 + 2xy - 12xz$ is degenerate with rank $r = 2$ and so we can compute $\tau = \left(\begin{smallmatrix} -1 & 0 & 0 \\ 1 & 3 & 0 \\ 2 & 0 & 3 \end{smallmatrix}\right)$ such that $\tau^\intercal A_Q \tau = \mathrm{diag}(0, 9, -27)$ is the matrix of an equivalent form. We have $Q(\tau\boldsymbol{x}) = \tilde{Q}(y, z) = 9y^2 - 27z^2$. For the linear part, $L(x, y, z) = x + z$, the change of coordinates results in $L(\tau\boldsymbol{x}) = \tilde{L}(y, z) + x = 3z + x$. Continue with the equation of the form (6): $9y^2 - 27z^2 + 3z = -1 - x$. Here, $\lambda_1 = 1$, and so we set (y, z) to $(\alpha_1, \alpha_2) = (\frac{1}{3}, 0)$ and find a solution for x: $\beta_1 = -2$. Next, find an affine transformation f associated with $9y^2 - 27z^2 + 3z = 1$. We have $\delta = 243$, $\boldsymbol{h} = (0, 27)^\intercal$ and $\tilde{c} = 216513$. The solutions of $9y^2 - 27z^2 + 3z = 1$ are exactly the solutions of $9y^2 - 27z^2 = 216513$ under the action of f.

Using the LINLOOP procedure, we find a linear loop $\langle M, \boldsymbol{s}\rangle$ for the invariant $9y^2 - 27z^2 = 216513$ with $M = \left(\begin{smallmatrix} 2 & 3 \\ 1 & 2 \end{smallmatrix}\right)$ and $\boldsymbol{s} = (-162, 27)^\intercal$. Therefore, an affine loop

$$\mathcal{A} := \langle M, \frac{1}{2\delta}(\boldsymbol{s} - \boldsymbol{h}), \frac{1}{2\delta}(M - I_2)\boldsymbol{h}\rangle;$$

that is, an affine loop with augmented matrix M' and initial vector $\boldsymbol{s}'$ given by

$$M' = \left(\begin{array}{c|c} 1 & 0_{1,r} \\ \hline \frac{1}{2\delta}(M - I_2)\boldsymbol{h} & M \end{array}\right) = \begin{pmatrix} 1 & 0 & 0 \\ -1/6 & 2 & 3 \\ -1/18 & 1 & 2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{s}' = \frac{1}{2\delta}(\boldsymbol{s} - \boldsymbol{h}) = \begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix},$$

satisfies the invariant $9y^2 - 27z^2 + 3z = 1$. Consequently, a *linear* loop with update matrix

$$M_\beta := \begin{pmatrix} 1 & 0 & 0 \\ 1/12 & 2 & 3 \\ 1/36 & 1 & 2 \end{pmatrix}$$

and initial vector $(-2, 1/3, 0)^\intercal$ satisfies the invariant $9y^2 - 27z^2 + 3z = -1 - x$. We conclude by applying transformation τ: a linear loop with matrix

$$\tau M_\beta \tau^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 27/4 & 2 & 3 \\ 35/12 & 1 & 2 \end{pmatrix} \quad \text{and initial vector} \quad \tau \begin{pmatrix} -2 \\ 1/3 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ -4 \end{pmatrix}$$

satisfies the original invariant $-11x^2 + y^2 - 3z^2 + 2xy - 12xz + x + z = -1$.

6 Conclusion

6.1 Related Work

Loop Synthesis. Work by Humenberger et al. on loop synthesis employs an approach based on algebraic reasoning about linear recurrences and translating loop synthesis into an SMT (Satisfiability Modulo Theory) solving task in non-linear arithmetic [17]. Their approach is relatively complete in the sense that every loop with algebraic values is captured as one of the solutions to the system of constraints. At the same time, no method is known to decide whether such a system has a *rational* solution. In contrast, our approach gives a characterisation of quadratic invariants that have linear loops with rational values.

Another SMT-based algorithm for template-based synthesis of general polynomial programs is given in work by Goharshady et al. [13]. However, loops generated for an invariant $P = 0$ using the latter approach necessarily have $P = 0$ as an inductive invariant and are not guaranteed to have infinite orbits. Recent work by Kenison et al. addresses the loop synthesis problem for multiple polynomial invariants, where each of the polynomials is a binomial of a certain type [21]. In our work, we restrict not the number of monomials in an invariant, but its degree, and thus achieve a complete solution for a single quadratic invariant.

Solving Polynomial Equations. As noted in Remark 2.8, one of the fundamental challenges towards loop synthesis arises from the study of integer and rational solutions to polynomial equations. A *Diophantine equation* $F(x_1, x_2, \ldots, x_d) = 0$ is a polynomial equation with rational coefficients in at least two variables. A general decision procedure for the existence of rational solutions to a Diophantine equation (Problem 3.1) is not known. Over the ring of integers, this is Hilbert's 10th Problem, proven undecidable by Matiyasevich in 1970 [25]. Furthermore, there does not exist an algorithm that for an arbitrary Diophantine equation, decides whether it has infinitely many integer solutions [10].

In contrast to the algorithmic unsolvability of Hilbert's 10th Problem and the open status of Problem 3.1, algorithms exist that allow finding rational solutions for special classes of equations. For instance, there exist procedures [14, 30, 24] completely solving the specialisation of the problem to quadratic equations. Masser introduced an approach based on the effective search bound for rational solutions [24]. A further improvement of this approach for $d \geq 5$ is provided in [5]. An alternative procedure to decide whether an arbitrary quadratic equation has a rational solution is described in [14] (see Corollary, pg. 2 therein). Determining the existence of integer solutions to a *system* of quadratic equations is, however, undecidable [4].

6.2 Discussion

We conclude by sketching some observations and pointing out the directions for future work.

Multiple loops. The approach of Algorithm 1 can be adapted to generate multiple linear loops satisfying a given invariant. Different solutions of the quadratic equation can be found in line 5 (Algorithm 1) and subsequently used as an initial vector. Moreover, in line 11, it is possible to pick two variables (x_1, x_2) in different ways, thus obtaining different matrices M in line 12. Each of the matrices synthesised so is an element of the orthogonal group $\Gamma(Q')$[1] of the quadratic form Q'. Therefore, all possible products of these matrices also preserve the value of Q' and can be used as updates. Other than the default matrix selected by the algorithm, some of these matrices alter more than two variables non-trivially, intuitively making the synthesised loop more specific to the polynomial invariant.

[1] The orthogonal group $\Gamma(Q)$ of a quadratic form Q is the group of all linear automorphisms $M \in \mathrm{GL}_d(\mathbb{K})$ such that $Q(\boldsymbol{x}) = Q(M\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{K}^d$.

Number of loop variables. Let $P(x_1,\ldots,x_d)=0$ be a quadratic invariant in d variables. Note that Theorem 5.4 can be interpreted in terms of linear loops with variables $x_0,x_1,\ldots,x_d$. Specifically, we can redefine the loop synthesis problem (Problem 2.7) by searching for linear loops with $s=d+1$ variables. To this end, update the procedure in Section 5 as follows: if the output of the original algorithm is an affine loop $\langle M,\boldsymbol{s},\boldsymbol{t}\rangle$, then output the linear loop

$$\left\langle \left(\begin{array}{c|c} 1 & 0_{1,d} \\ \hline \boldsymbol{t} & M \end{array}\right), \begin{pmatrix} 1 \\ \boldsymbol{s} \end{pmatrix} \right\rangle.$$

Due to Corollary 5.5, the updated procedure solves the problem of loop synthesis with one additional variable. What follows is a reinterpretation of Theorem 5.4:

▶ **Corollary 6.1.** *There exists an effective procedure for the following problem: given a quadratic equation*

$$Q(x_1,\ldots,x_d)+L(x_1,\ldots,x_d)=c,$$

decide whether there exists a non-trivial linear loop in $d+1$ variables $\{x_0,x_1,\ldots,x_d\}$ that satisfies it. Furthermore, the procedure synthesises a loop, if one exists.

Increasing the number of variables in the loop template leads to the following question, also raised in [17]:

▶ **Question.** *Let P be an arbitrary polynomial in d variables. Does there exist an upper bound N such that if a non-trivial linear loop satisfying $P=0$ exists, then there exists a non-trivial linear loop with at most N variables satisfying the same invariant?*

Corollary 6.1 (together with Corollary 5.5) shows that, *for quadratic polynomials, N is at most $d+1$.* Moreover, we show in Section 4 that in the class of polynomial equations $Q(\boldsymbol{x})-c$, where Q is a quadratic form, the bound $N=d$ is tight. A full characterisation of quadratic equations for which linear loops with d variables exist would also be of interest.

Sufficient conditions. The results of Sections 4 and 5 witness another class of polynomial invariants for which non-trivial linear (or affine) loops always exist. Similar to the setting of equations with pure difference binomials in [21], we can claim this for invariants $Q(x_1,\ldots,x_d)=c$ with isotropic quadratic forms Q. In particular, for every equation of the form $a_1x_1^2+\cdots+a_dx_d^2+c=0$ with $d\geq 4$ and $a_1,\ldots,a_d,c$ not all possessing the same sign, there exists a non-trivial linear loop with d variables. This fact is due to Meyer's Theorem on isotropy of indefinite forms [26] and Corollary 5.5(2).

Beyond quadratic. One future work direction concerns loop synthesis from invariants that are polynomial equalities of higher degrees, and, in particular, algebraic forms. However, we are limited by the hardness of Problem 3.1, as before. For Diophantine equations defined with homogeneous polynomials of degree 3, the loop synthesis is related to the study of rational points on elliptic curves, a central topic in computational number theory [31, 8].

References

1 T. Andreescu and D. Andrica. *Quadratic Diophantine Equations.* Developments in Mathematics. Springer New York, 2016.

2 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics).* Springer-Verlag, Berlin, Heidelberg, 2006.

3 Zenon I. Borevich and Igor R. Shafarevich. *Number Theory*. Pure and Applied Mathematics. Academic Press, New York, NY, 1966.

4 J. L. Britton. Integer solutions of systems of quadratic equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 86(3):385–389, 1979. `doi:10.1017/S0305004100056218`.

5 T. D. Browning and R. Dietmann. On the representation of integers by quadratic forms. *Proceedings of the London Mathematical Society*, 96(2):389–416, 2008. `doi:10.1112/plms/pdm032`.

6 Pierre Castel. Solving quadratic equations in dimension 5 or more without factoring. In *ANTS X—Proceedings of the Tenth Algorithmic Number Theory Symposium*, volume 1 of *Open Book Ser.*, pages 213–233. Math. Sci. Publ., Berkeley, CA, 2013. `doi:10.2140/obs.2013.1.213`.

7 Krishnendu Chatterjee, Hongfei Fu, Amir Kafshdar Goharshady, and Ehsan Kafshdar Goharshady. Polynomial invariant generation for non-deterministic recursive programs. In *PLDI 2020*, pages 672–687, 2020. `doi:10.1145/3385412.3385969`.

8 H. Cohen. *Number Theory: Volume I: Tools and Diophantine Equations*. Graduate Texts in Mathematics. Springer New York, 2007.

9 J. E. Cremona and D. Rusin. Efficient solution of rational conics. *Math. Comp.*, 72(243):1417–1441, 2003. `doi:10.1090/S0025-5718-02-01480-1`.

10 Martin Davis. On the number of solutions of diophantine equations. *Proceedings of the American Mathematical Society*, 35(2):552–554, 1972.

11 Emilie Dufresne, Parker B. Edwards, Heather A. Harrington, and Jonathan D. Hauenstein. Sampling real algebraic varieties for topological data analysis, 2018. `arXiv:1802.07716`.

12 G. Everest, A. van der Poorten, I. Shparlinski, and T. Ward. *Recurrence sequences*, volume 104 of *Math. Surveys Monogr.* Amer. Math. Soc., Providence, RI, 2003. `doi:10.1090/surv/104`.

13 Amir Kafshdar Goharshady, S. Hitarth, Fatemeh Mohammadi, and Harshit Jitendra Motwani. Algebro-geometric algorithms for template-based synthesis of polynomial programs. *Proc. ACM Program. Lang.*, 7(OOPSLA1), April 2023. `doi:10.1145/3586052`.

14 Fritz J. Grunewald and Daniel Segal. How to solve a quadratic equation in integers. *Mathematical Proceedings of the Cambridge Philosophical Society*, 89(1):1–5, 1981. `doi:10.1017/S030500410005787X`.

15 S. Hitarth, George Kenison, Laura Kovács, and Anton Varonka. Linear loop synthesis for quadratic invariants, 2023. Extended version. `arXiv:2310.05120`.

16 Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, and James Worrell. On strongest algebraic program invariants. *J. ACM*, August 2023. Just Accepted. `doi:10.1145/3614319`.

17 Andreas Humenberger, Daneshvar Amrollahi, Nikolaj Bjørner, and Laura Kovács. Algebra-based reasoning for loop synthesis. *Form. Asp. Comput.*, 34(1), July 2022. `doi:10.1145/3527458`.

18 K. Ireland and M.I. Rosen. *A Classical Introduction to Modern Number Theory*. Graduate Texts in Mathematics. Springer, 1990.

19 Bertrand Jeannet, Peter Schrammel, and Sriram Sankaranarayanan. Abstract acceleration of general linear loops. *SIGPLAN Not.*, 49(1):529–540, January 2014. `doi:10.1145/2578855.2535843`.

20 Manuel Kauers and Peter Paule. *The Concrete Tetrahedron - Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates*. Texts & Monographs in Symbolic Computation. Springer, 2011.

21 George Kenison, Laura Kovács, and Anton Varonka. From polynomial invariants to linear loops. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, ISSAC '23, pages 398–406, New York, NY, USA, 2023. Association for Computing Machinery. `doi:10.1145/3597066.3597109`.

22 Zachary Kincaid, Jason Breck, John Cyphert, and Thomas W. Reps. Closed Forms for Numerical Loops. In *Proc. of POPL*, pages 55:1–55:29, 2019.

23 Seymour Lipschutz and Marc Lipson. *Schaum's Outline of Linear Algebra*. Schaum's Outline Series. McGraw-Hill, New York, fourth edition, 2009.

24 D. W. Masser. How to solve a quadratic equation in rationals. *Bulletin of the London Mathematical Society*, 30(1):24–28, 1998. doi:10.1112/S0024609397003913.

25 Yuri Matiyasevich. Enumerable sets are Diophantine. *Soviet Math. Dokl.*, 11:354–358, 1970.

26 A. Meyer. Mathematische Mittheilungen. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 29:209–222, 1884.

27 Ivan Niven. *Irrational numbers*. Mathematical Association of America; distributed by John Wiley & Sons, Inc., New York, N.Y.,,, 1956.

28 Joël Ouaknine and James Worrell. On linear recurrence sequences and loop termination. *ACM SIGLOG News*, 2(2):4–13, April 2015. doi:10.1145/2766189.2766191.

29 Alexandra Shlapentokh. Defining integers. *The Bulletin of Symbolic Logic*, 17(2):230–251, 2011.

30 Carl Ludwig Siegel. Zur theorie der quadratischen formen. *Nachrichten der Akademie der Wissenschaften in Göttingen, Mathematisch-Physikalische Klasse*, 3:21–46, 1972.

31 Joseph H. Silverman and John T. Tate. *Rational Points on Elliptic Curves*. Springer Publishing Company, Incorporated, 2nd edition, 2015.

32 Denis Simon. Quadratic equations in dimensions 4, 5 and more. 2005.

33 Denis Simon. Solving quadratic equations using reduced unimodular quadratic forms. *Math. Comp.*, 74(251):1531–1543, 2005. doi:10.1090/S0025-5718-05-01729-1.

34 T. Skolem. *Diophantische Gleichungen*. Ergebnisse der Mathematik und ihrer Grenzgebiete. J. Springer, 1938.

35 Sergei Treil. Linear algebra done wrong, 2004.

36 Leonid Vaserstein. Polynomial parametrization for the solutions of diophantine equations and arithmetic groups. *Annals of Mathematics*, 171:979–1009, 2010.