UNIVERSITÉ PIERRE ET MARIE CURIE - PARIS 6
ÉCOLE DOCTORALE EDITE
UNIVERSITÉ BEIHANG - BUAA

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'Université Pierre et Marie Curie - Paris 6

et l'Université Beihang - BUAA

Mention : Informatique/Mathématiques

Présentée et soutenue par

Ting ZHAO

Résolution de Systèmes Polynomiaux Paramétriques par Radicaux et ses Applications Géométriques

Thèse dirigée par

Dongming Wang et Philippe Aubry

soutenue le 3 Mai 2012

Rapporteurs

Thomas STURM	Chercheur – MPII
Bican XIA	Professeur – PKU
Tiegang LIU	Professeur – BUAA
Lihong ZHI	Professeur – AMSS

Directeurs

Dongming WANG	Directeur de Recherche – CNRS
	Professeur – BUAA
Philippe AUBRY	Maître de Conférences – UPMC

Jury

Annick VALIBOUZE	Professeur – UPMC
Tiegang LIU	Professeur – BUAA
Thomas STURM	Chercheur – MPII
Bican XIA	Professeur – PKU
Lihong ZHI	Professeur – AMSS
Zhikun SHE	Professeur – BUAA
Dongming WANG	Directeur de Recherche – CNRS
	Professeur – BUAA
Philippe AUBRY	Maître de Conférences – UPMC

Parametric Polynomial System Solving by Radicals and Its Geometric Applications

To My Family.

Table of Contents

List of Tables . . . . **iii**

List of Figures . . . . **iv**

Abstract . . . . **v**

Acknowledgements . . . . **vii**

Chapter 1 Introduction **1**

1.1 Background, Problems and Motivations 1

1.2 State of the Art 4

1.3 Contributions of This Thesis 15

Chapter 2 Fundamental Tools from Polynomial Algebra **19**

2.1 Solution Formulas of Polynomial Equations 19

2.2 Number of Distinct Real Roots of a Polynomial 23

2.3 Characteristic Sets and Triangular Decomposition 26

2.4 Quantifier Elimination 34

Chapter 3 Solution Formulas for Cubic Equations Without or With Constraints **37**

3.1 Introduction 37

3.2 Incorrectness of the Standard Convention 38

3.3 Real Convention 39

3.4 Proof of the Correctness of the Real Convention 41

3.5 Formulas for Cubic Equations with Constraints 51

3.6 Proof of the Correctness of the Formulas 54

Chapter 4 Solution Formulas for Quartic Equations Without or With Constraints **59**

4.1 Adjusted Solution Formula for Quartic Equations 59

4.2 Configuration of the Solutions of Quartic Equations 61
4.3 Proof of the Correctness of the Real Convention 64
4.4 Formulas for Quartic Equations with Constraints 92

Chapter 5 Generating Dynamic Diagrams with Inequality Constraints 97
5.1 Introduction and Motivation 97
5.2 Representing Real Solutions of Semi-algebraic Systems by Radicals . 99
5.3 Solution Formulas of Univariate Equations with Inequality Constraints 102
5.4 Generation of Dynamic Diagrams 104
5.5 Examples of Geometric Configurations 109

Chapter 6 GeoDraw: A Software Package for Automated Generation of Dynamic Diagrams 121
6.1 Introduction and Motivation 121
6.2 Design and Implementation 122
6.3 Functionalities and Features 126
6.4 Predicate Specifications 128
6.5 Examples and Analysis 129

Chapter 7 Conclusion and Future Work 137

Bibliography 141

Appendices 153

Publications 179

Curriculum Vitae 181

List of Tables

Table 3.1 Signs of Derivatives of Cubic Polynomial f 56
Table 3.2 Truth Values of Φ_j of Solutions of Cubic Equation 57

Table 4.1 Signs of Derivatives of Quartic Polynomial f (I) 95
Table 4.2 Signs of Derivatives of Quartic Polynomial f (II) 95
Table 4.3 Truth Values of Φ_j of Solutions of Quartic Equation (I) 96
Table 4.4 Truth Values of Φ_j of Solutions of Quartic Equation (II) 96

Table 6.1 Data and Timings for 15 Examples 134

List of Figures

Figure 2.1 Standard Convention for the Square and Cubic Roots 20

Figure 3.1 Real Convention for the Square and Cubic Roots 40

Figure 3.2 Solution Indexing for Cubic Equation 42

Figure 4.1 Solution Indexing for Quartic Equation ($p_1 \neq 0$) 62

Figure 4.2 Solution Indexing for Quartic Equation ($p_1 = 0$) 63

Figure 4.3 Another Solution Indexing for Cubic Equation for Case $p_1 < 0$ 72

Figure 5.1 Steiner Configuration 107

Figure 5.2 Apollonius Circles 109

Figure 5.3 Feuerbach Configuration 113

Figure 5.4 Thébault–Taylor Configuration 115

Figure 6.1 Architecture of GeoDraw 123

Figure 6.2 Graphic User Interface 127

Figure 6.3 Simson Configuration 130

Figure 6.4 Napoleon Configuration 132

Abstract

Geometric constraint solving has been extensively studied in computer aided geometric design, modeling, and engineering. This thesis focuses on the design and implementation of efficient algorithms and software tools for solving parametric geometric constraints involving both equalities and inequalities, with application to automated generation of dynamic diagrams.

After a review of the state of the art and an introduction of some basic knowledge in the first two chapters, we present in Chapter 3 a real convention (for square and cubic roots) which provides correct interpretations of the Lagrange formula for cubic polynomial equations with real coefficients. It is a uniform convention that does not depend on the coefficients of the polynomials. Using this real convention, we also present real solution formulas for the general real-coefficient cubic equation under equality and inequality constraints.

In Chapter 4, we generalize the solution formulas from the cubic to the quartic case. We adjust the Lagrange-type solution formulas for quartic equations and prove that the real convention can also yield correct interpretations of the adjusted formulas. We then use the real convention to derive the real solution formulas for the general real-coefficient quartic polynomial equation with equality and inequality constraints. The extension of the solution formulas from the cubic to the quartic case is not straightforward and needs some quite sophisticated techniques including those used in the study of the Sturm–Habicht sequence and discrimination systems. The new real solution formulas of cubic and quartic equations with constraints avoid divisions by small numbers and thus the numerically unstable "near 0/0" case.

In Chapter 5, we enhance the HLLW approach proposed by Hong and others for diagram generation by replacing the Ferro–Cardano-type solution formulas of cubic equations used there with new Lagrange-type real solution formulas under equality and inequality constraints introduced in Chapter 3 and by incorporating new real solution formulas of quartic equations with constraints presented in Chapter 4. The performance of the HLLW approach is improved in terms of stability of numeric computation and quality of generated diagrams. Several examples are presented to illustrate the enhanced approach and to demonstrate the advantages and effectiveness of our new solution formulas.

A software package GeoDraw that has been developed for drawing dynamic diagrams automatically from predicate specifications of given sets of geometric relations involving inequality constraints and in which the enhanced HLLW approach has been implemented is presented in Chapter 6. We discuss its design and implementation, highlight its functionalities and features, give several examples to illustrate its usage, and provide some empirical data in table form to show its performance. Some problems for future investigations are discussed finally in Chapter 7.

Key words: geometric constraint solving, real solution formula, dynamic diagram, inequality constraint, root convention, animation, semi-algebraic system, geometry software.

Acknowledgements

I would like to express my sincere appreciation and gratitude to my supervisors Professor Dongming Wang and Professor Philippe Aubry, who brought me into the field of dynamic geometric constraint solving. Their guidance, ideas, and encouragement have been of great benefit to me through my entire Ph.D study.

Great thanks are due to Professor Hoon Hong for his helpful advices and inspiring discussions, cooperating with me, and inviting me to visit North Carolina State University in USA and Korea Institute for Advance Study in Seoul.

Many thanks to the Key Laboratory of Mathematics, Informatics and Behavioral Semantics in Beihang University for providing me a stimulating and fun working environment. I am deeply indebted to all the teachers in the School of Mathematics and Systems Science, from whose devoted teaching and enlightening lectures I have benefited a lot. Special thanks to my colleagues Yanli Huang, Meng Jin, Chenqi Mou, Jing Yang and Lei Jiang for their help during my Ph.D study. I have a very happy time to work with them. In particular, I would like to thank Xiaoyu Chen and Yang Liu for helping me solving several technical problems and reading part of this thesis.

I would like to thank Daniel Lazard, Fabrice Rouillier, Jean-Charles Faugère, Guénaël Renault, and Ludovic Perret from the SALSA team of the Laboratoire d'Informatique de Paris 6 for their help and encouragement. My thanks are extended to all the Ph.D students in this team for the help they provided when I was in Paris. Especially, I wish to show my great thanks to Wei Niu and Xiaomin Wang for helping me to go through some formality matters with UPMC and EDITE.

I am sincerely grateful to Xiaojuan Zhang, Yajun Cao, Shujuan Ding, Xuemei Dong, Deyuan Meng, Chengyu Jiao, and all my friends for sharing my happiness and helping me when I was in difficulty during these years.

Special thanks to my parents, my husband Haoliang Zhang and my little girl Miaomiao, for their constant support, considerations, and never fading love.

Chapter 1

Introduction

1.1 Background, Problems and Motivations

This thesis presents the results of our studies on real solution formulas for cubic and quartic equations without or with inequality constraints and their application to automated generation of dynamic diagrams. Solving univariate equations of higher degree by radicals is a fundamental problem with a long history which had important impact on the development of algebra. It was studied by many great mathematicians including G. Cardano, R. Descartes, J. L. Lagrange, and N. H. Abel, leading to the invention of Galois theory. Explicit expressions of solutions in terms of radicals are not only of theoretical interest, but also convenient for computing the solutions and for many other applications (such as solving systems of multivariate polynomial equations and parameterizing rational curves by radicals). Our work on real solution formulas was motivated by the need of such formulas for fast computation of solutions to implement efficient programs for real-time diagram animation.

Diagrams have been widely used in geometry to help one to understand geometric configurations and problems intuitively. Modern computers can now be used to produce dynamic diagrams, which contain constrained movable geometric objects and may be animated under mouse actions. By means of animating a dynamic diagram, one can observe and even discover relations among the geometric objects involved in the diagram. A geometric description of the dynamic diagram can be algebraically expressed as a system of polynomial equations and inequalities in certain free or dependent coordinates. Solving this system is a major step for the generation of the dynamic diagram. By using algebraic methods based, e.g., on triangular decomposition and Gröbner bases (see Section 2.3 and [128, 24]), the problem of solving the system can be reduced to that of solving univariate polynomial equations (of low degree in most cases) under inequality constraints. Univariate equations may be solved by using explicit solution formulas. Such formulas for linear and quadratic equations are simple and well known. Our investigations will be devoted to real solution formulas for cubic and quartic equations. Solution formulas for nonlinear equations have been studied since the ancient time and there are plenty of results, but there is little work on such formulas for real solutions or under inequality constraints.

In recent years, a large number of software systems have become available for producing dynamic diagrams interactively or (semi-) automatically. We mention a few of them, Cabri,[1] Cinderella,[2] GCLC,[3] GeoGebra,[4] Geometry Expressions,[5] GEOTHER,[6] JGEX,[7] and Geometer's SketchPad,[8] as examples. There are different approaches proposed for solving the involved polynomial systems based on symbolic computation, numeric computation, and their combination. There are also highly efficient preprocessing methods, developed mostly in the computer aided design community, which transform geometric descriptions into easy-to-draw[9] geometric constructions using graph analysis and rule-based reasoning. The reader may refer to [66] for a review of such methods and approaches.

Most of the approaches cited above were proposed to deal with geometric constraints involving equalities only, despite that inequality constraints occur very frequently in real-world problems of geometry. Many geometric constraints require such concepts as "between", "inside", and "outside" which involved *order*. For example, we often need to deal with geometric constraints on external tangency of circles and internal bisection of angles. When such geometric constraints are translated into algebraic expressions, they show up as inequalities. There are two ways which are usually used to deal with them. One is formulating the problems using equalities only, but then the formulations may become unnatural, involving artificial slack variables and existential quantifiers, thus causing enormous computational blowup and making the problems practically intractable. The other is ignoring the order notion

[1] `http://www.cabri.com/`
[2] `http://www.cinderella.de/`
[3] `http://poincare.matf.bg.ac.rs/~janicic//gclc/`
[4] `http://www.geogebra.org/cms/`
[5] `http://www.geometryexpressions.com/index.php`
[6] `http://www-salsa.lip6.fr/~wang/GEOTHER/`
[7] `http://www.cs.wichita.edu/~ye/`
[8] `http://www.dynamicgeometry.com/`
[9] e.g., with ruler and compass or other basic drawing tools [50]

while formulating geometric constraints, but this often leads to unexpected extraneous or degenerate solutions, causing confusion, especially when the diagram is moving across some critical points. Therefore, it is imperative to develop effective methods that directly tackle geometric constraints involving inequalities.

However, there is limited work on solving geometric constraints involving inequalities as well as equalities. This is mainly due to the inherent practical difficulties of dealing with general inequality constraints. There are methods (based on cylindrical algebraic decomposition and others) which can handle, in principle, arbitrarily general equality/inequality constraints, but the practical complexity is prohibitive for problems even of moderate size [31, 32]. This is partly because general methods need to consider the interaction of arbitrary degree equalities with inequalities. In view of the fact that many interesting and important geometric constraints involve only equalities of low degree, we restrict our studies to *equalities of degree less than* 5 but allow *inequalities of arbitrary degree*. In this restricted case, real solutions can be represented by means of explicit formulas with radicals, which allows efficient evaluation of the solutions for dynamic update of diagrams.

H. Hong, L. Li, T. Liang and D. Wang [63] developed an approach (referred to as the HLLW approach) for solving geometric constraints involving inequalities. This approach proceeds by first decomposing the set of equality constraints into finitely many triangular sets. Then for each triangular set together with inequality constraints, the space of parameters is decomposed into finitely many domains by means of real quantifier elimination, such that associated with each domain there is a set of explicit expressions of the dependent variables in terms of the parameters with radicals. For any given values of the parameters, if they verify the relations of some domains, the values of the dependent variables may be easily computed by direct evaluation of the corresponding explicit expressions.

Nevertheless, there are at least two problems with the HLLW approach that remain for investigation. Firstly, the Ferro–Cardano formula used therein for solving cubic equations involves divisions, which may encounter the numerically unstable "near 0/0" case (i.e., the case when both the numerator and denominator are close to zero). This "near 0/0" case happens in particular during geometric constraint solving because the coefficients of the equations being solved keep changing gradually. Secondly, in the HLLW approach it is not described in detail how to deal with general quartic equations.

We will enhance the HLLW approach by using the real solution formulas we will establish for cubic and quartic equations with real coefficients, which do not involve non-constant divisions. We will also present our software package in which the enhanced HLLW approach has been implemented for automated and efficient generation of dynamic diagrams involving both equality and inequality constraints.

1.2 State of the Art

1.2.1 Polynomial System Solving

Polynomial systems occur almost everywhere in mathematics, science and technology. Solving polynomial systems is the basis for many problems, for example, in effective (real) algebraic geometry [7, 35], robotics [33], and geometric modeling [106]. Systems of polynomial equations can be written in the following form

$$\begin{cases} F_1(x_1, \ldots, x_n) = 0, \\ \quad\cdots\cdots \\ F_s(x_1, \ldots, x_n) = 0, \end{cases}$$

where $F_i(x_1, \ldots, x_n)\,(i = 1, \ldots, s)$ are polynomials in $x_1, \ldots, x_n$ with rational, real or complex coefficients. There are many methods for solving such systems and we introduce some of them briefly below.

Gröbner basis method: B. Buchberger [20, 25] defined some special sets of generators of a polynomial ideal as Gröbner bases,[10] and gave the first algorithm (known as Buchberger's algorithm) to compute them. The basic idea of the algorithm is to reduce leading terms of polynomials with respect to a specified term order (i.e., admissible term ordering). Due to Buchberger and his followers' efforts [21, 22, 96, 23, 82, 84, 85, 47, 45, 46, 62, 61, 138, 139, 132, 93, 135, 105, 156], the theory of Gröbner bases has become a cornerstone of computer algebra with

[10] It was named after Buchberger's advisor Wolfgang Gröbner. Analogous concepts were also proposed independently by Shirshov [113] and Hironaka [58].

many applications. Especially, when a pure lexicographic order is chosen, the Gröbner basis of an ideal has elimination properties, and thus can be used explicitly to work out the zeros of an ideal.

Wu–Ritt's method: This method was developed by W.-T. Wu in the late 1970s for proving geometric theorems mechanically [144, 143] and solving multivariate polynomial equations [142]. It is based on the concept of characteristic set introduced in the late 1940s by J. F. Ritt [109, 110] which is fully independent of Gröbner bases, though the Gröbner basis method may be used to compute characteristic sets. Wu–Ritt's method is powerful for theorem proving in elementary geometry, and provides a complete decision process for certain classes of geometric problems. The main trends of research on Wu–Ritt's method concern systems of polynomial equations of positive dimension and differential algebra where Ritt's results have been made effective. The basic idea of Wu–Ritt's method is to eliminate variables of polynomials by pseudo-divisions. Wu and his followers [140, 123, 125, 126, 127, 128, 5, 6, 30, 51, 28] have done remarkable work to improve this method and successfully applied it to automated geometric reasoning and many other fields.

Resultant methods: Variables of polynomials can also be eliminated by using various resultants such as sparse resultant [42, 36, 26], u-resultant [121, 34, 128, 151], and Macaulay resultant [94, 81, 128, 151]. They use matrix formulations so that their computation can be transformed to problems of linear algebra. However, the resultant methods usually have restrictions (e.g., on the number of equations) for the construction of their corresponding matrixes.

There are also other methods for solving polynomial systems, such as Homotopy method [88, 89], Eigenvalue/Eigenvector method [97, 98], and Rational Univariate Representation (RUR) method [102, 111]. Note finally that most of the methods reduce the problem to solving some univariate polynomial equations. The study of solving polynomial equations of the form

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = 0$$

started at the ancient time by Egyptians who could solve problems equivalent to linear equations in one unknown. Their work was written on the *Rhind Papyrus* which is thought to represent the state of Egyptian mathematics of about 1850 B.C. The research on solving polynomial equations substantially influenced the development of mathematics for centuries [8, 12, 101, 44], leading to the ideas of abstract thinking and the introduction of some fundamental concepts of mathematics including irrational and complex numbers, algebraic groups, fields, and ideals.

The appearance of quadratic equations can be traced back to evidence left in clay tablets by ancient Babylonians. The early study focused on methods to solve quadratic equations of specific forms, such as the form $a_2x^2 + a_0 = 0$. Later in 1145 A.D., the first complete solution was given by Abraham bar Hiyya Ha-Nasi (known also with the Latin name Savasorda) in Europe in his book *Liber Embadorum* about algebra and geometry. Note that the general solution formula for quadratic equations is one of the most fundamental and key principles of mathematics, and a full understanding of this formula requires the introduction of negative, irrational, and complex numbers. For example, about 500 B.C. in ancient Greece, the Pythagoreans formally and rigorously proved that the equation $x^2 = 2$ has no rational solution, that is, its solution cannot be found by using only arithmetic operations but also a radical. This is an important achievement for finding irrational numbers, which is closely related to the history of solving quadratic polynomial equations.

Comparing to the linear and quartic equations, the solution formulas which would involve only arithmetic operations and radicals for polynomial equations of degree 3 and 4 are much more difficult and significant in the history of mathematics. In fact, they were the first mathematical formulas unknown to the ancients, and forced mathematicians to take both complex numbers and negative numbers seriously. The achievement on solving cubic and quartic equations was the first major advance by modern man since the time of ancient Greeks. More importantly, the proof of the insolvability of the quintic in the 19th century finally led to the study of the theory of equations.

In the success of finding the general solution formulas for cubic and quartic equations, there are a number of key figures. We list some of them who gave the main contributions. In the 11th century, O. Khayyam found general geometric solutions of cubic equations with intersections of conic sections. Then by using the Babylonian numerals, L. Pisa, also known as Fibonacci, was able to give an approximate formula for cubic equations with certain form $x^2 + ax^2 + bx = ab$. About 1515, S. D. Ferro discovered a method for finding the solutions of a class of cubic equations with the form $x^3 + mx = n$, and told his achievement to some of his students. In 1535, one of these student named A. M. Flor challenged N. Tartaglia to a problem-solving contest involving the cubic. In this competition, Tartaglia won and revealed his result for solving cubic equations to G. Cardano who published this secret formula and Ferro's prior work in his book *Ars Magna* [27, 55] with credit given to Tartaglia.

The solution to quartic equations was discovered by L. Ferrari, one of Cardano's students, in 1540, and at the same time, a similar solution was found by R. Descartes. However, this solution, like all algebraic solutions of the quartic, requires the solution of a cubic to be found. Therefore, it couldn't be published immediately. The general solution of quartic equations was published together with that of the cubic by Cardano in *Ars Magna*.

For hundreds of years, mathematicians tried to investigate some generalization of the classical solution formulas of any polynomial, but did not succeed for polynomials of fifth and higher degrees. Around 1770, J. L. Lagrange began the groundwork to unify the many different methods which had been used for solving equations in the form of Lagrange resolvents. This innovative work is corresponding to the theory of groups of permutations and also a precursor to Galois theory. However, Lagrange failed to develop solutions for equations of degrees greater than 4, which implied that such solutions might be impossible, but he did not give conclusive proof. Finally, a theorem on the nonexistence of solution formula in radicals for the class of polynomials of degree more than 4 was obtained and proved by R. Ruffini in 1813 and N. H. Abel in 1827. Insights into these issues were also gained using Galois fundamental theory pioneered by E. Galois in 1832, which was motivated by the same problem of solving equations and included the proof of the nonexistence of the solution in the form of formula with radicals, already for simply specific polynomial equations with integer coefficients. This theory also gave a world of major ideas and techniques for the development of modern algebra.

The absence of solution formula in radicals does not means that higher-degree polynomial equations are unsolvable. In fact, the *fundamental theorem of algebra* states that every non-constant polynomial equation in one unknown, with real or complex coefficients, has at least one complex number as solution. However, the equation does not always have a real or a rational solution. In many applications, one usually needs to find the number of distinct real solutions, or even the real solution formulas. How to represent the real solutions of polynomial equations, even for the cubic and quartic ones, is an interesting and difficult question. However, there is not much work on this aspect. Therefore, in this thesis, we are concerned with the problems of solving cubic and quartic polynomial equations, and finding their real solution formulas in radicals with equality and inequality constraints.

The problem of solving polynomial equations has essentially affected the development of mathematics throughout the centuries and still has many useful applications to the theory and practice of present-day computing. It is therefore very meaningful to investigate the methods of solving polynomial equations deeply.

1.2.2 Geometric Constraint Solving

Geometric constraint solving has been studied extensively in various areas, such as computer-aided geometric design and modeling, mechanical engineering, chemical molecular modeling, tolerance analysis and geometric theorem proving. It takes a given set of geometric elements including points, lines, circles and a description of geometric constraints between these elements, such as distances between points, parallel lines, angles between lines, incidence relations between points and lines or between points and circles, tangency relations between lines and circles or between circles, as input. The purpose is to output all placements of the geometric elements which satisfy the given constraints. For example, consider a problem with its input elements to be a set of three lines and the constraints to be that the first two lines are perpendicular and the third line makes a specified angle with the first line. One may find infinitely many solutions for this particular problem. Thus in order to tie down a particular solution, an additional constraint such as the length of one segment between the intersections of two pairs of lines has to be given.

Intuitively a geometric constraint solving problem is considered bo be *structurally well-constrained* if it has a finite number of solutions for non-degenerate configurations. Similarly, if a problem has an infinite number of solutions, it is called *under-constrained*. Finally, if a constraint system still has a finite number of solutions after one of the constraints is deleted, then the problem is *over-constrained*. In the previous example, if the angle between the third line and the second line was given as another constraint, the problem would become over-constrained, because that angle is already determined by the first two angle constraints. An over-constrained problem may have a solution if the additional constraints are consistent with previous constraints. But in most cases, over-constrained problems have no solution. A constraint problem naturally corresponds to a set of (usually nonlinear) algebraic equations.

Many resolution methods have been proposed for solving systems of geometric constraints. In [59], C. M. Hoffmann and R. Joan-Arinyo stated that the general properties a geometric constraint solving method should have include soundness, completeness, competence, persistence, stabilities, efficiency, robustness, intensionality, geometric sense, dimension independence, and generality. However, it is very difficult to obtain methods that possess all these characteristics. So far, there have been some good compromises that achieve each of these methods characteristics to a reasonable degree [74]. The geometric constraint solving methods can be classified in many ways. We classify them in three broad categories: graph-based, logic-based, and algebraic methods.

In graph-based methods, a constraint problem can be translated into a graph (or hypergraph) with vertices. The geometric elements are represented by the graph nodes and the constraints are the graph edges. The class of configurations solved by these methods is typically the ruler and compass constructive problems. The advantages of graph-based methods is that the problem can be decomposed into some small subproblems and a few simple categories by graph analysis. The methods can deal with the over-constrained and under-constrained problems. The disadvantage is that the graph analysis of a fully competent method is rather complicated. The methods can be further subdivided into constructive [48, 49, 86, 104], degree of free analysis [76, 77, 75, 112] and propagation [9, 78, 116, 119, 10].

In logic-based methods, a problem can be translated into a set of assertions and axioms characterizing the constraints and the geometric objects. The constraints are expressed as facts and an inference engine is used to derive the solution by exhaustively applying rules. These methods [1, 17, 115, 145, 118, 122, 67, 68] provide a qualitative study of geometric constraints. Although they are good methods for prototyping and experimentation, the extensive computations involved in the exhaustive searching and matching make them inappropriate for real world applications.

In this thesis, we are mainly concerned with algebraic methods in particular symbolic methods which can translate the problem into a system of polynomial equations and solve the system by using the algorithms we mentioned in the previous section and the solution formulas for solving univariate polynomial equations. The main advantage of algebraic methods are their generality, dimension independence and ability to deal with symbolic constraints naturally. However, if the problem is complicated, the algebraic methods may be quite slow and space-consuming. Even though, algebraic techniques can be extremely useful and very practical for pre-processing and studying specific constraint systems. The algebraic methods can be further classified into numerical and symbolic according to the different technique for solving the system of polynomial equations.

- **Numerical Methods**

  The numerical method is a general approach for solving the geometric constraint problems. It first translate the constraints into a system of nonlinear equations. Then this equation system can be solved by iterative methods.

  A. H. Borning [9], R. C. Hillyary and I. C. Braid [57], and I. Sutherland [119] used a *relaxation* method to solve geometric constraint problems. Basically, the method perturbs the values assigned to the variables and minimizes some measure of the global error. However, this method is quite slow and sometimes convergence cannot be obtained.

  The most commonly used numerical method is the *Newton–Raphson* [69] iteration, which was popularly used in many applications [56, 91, 92, 100]. This method is fast but instable because of the sensitivity to the initial values, and may lead to an unexpected or unwanted solution, or to the iteration divergence by a small deviation in the initial value.

  To overcome the above problems, recently the *homotopy* or *continuation* [2] method has been proposed and experimented. According to the reports in [80, 41], generally the homotopy method works much better in terms of stability.

These two methods generally require the number of variables to be the same as the number of equations. If these two numbers are different, i.e. the constraint system is generally over- or under-constrained, some special techniques, such as linear least square and singular value decomposition of a matrix, are required.

The numerical methods have generality and can handle problems with complicated geometric constraints, but cannot deal with the over-constrained and under-constrained problems well. There are two drawbacks: one is that the efficiency and stability of these methods will decrease fast with respect to the increase of the number of polynomials; the other is that they can only get one solution of the equation system each time, but cannot find out all the solutions and compare them to choose the best one.

- **Symbolic Methods**

  Symbolic algebraic methods solve a system of equations by computer algebra techniques such as the Wu–Ritt method [29, 143] or the Gröbner basis method [24]. In essence, these methods transform the system of polynomial equations into a triangular system whose solutions are those of the given system. In effect, triangularization reduces the problem of solving a simultaneous, nonlinear system to that of univariate root finding. Forward or backward substitution must be used for solving the triangular system.

  Many systems have used a symbolic resolution of the system of equations by a lot of researchers. For example, S. A. Buchanan and others [19] provided a method based on Buchberger's algorithm. In [72, 73], K. Kondo reported a symbolic algebraic method, and improved his work by generating a polynomial that summarizes the changes undergone by the system of equations.

  Symbolic methods use symbols to represent the coefficients of the polynomials which can deal with more problems. They can determine whether a problem is well-, over- or under-constrained, and can compute all the solutions of a system of polynomial equations. The disadvantage is that this kind of methods are of high complexity and cost plenty of computing time and space.

There are many methods available for solving geometric constraints. However, note that these methods are developed mainly for solving geometric constraints that may be expressed algebraically as equalities. For those geometric constraints involving inequalities which occurs very frequently in real-world problems of geometry, the work is very limited. The first approach of solving such geometric constraints was proposed by Hong and his coauthors [63]. However, the stability of numerical computation and quality of generated diagrams in their approach doesn't perform very well and thus it needs to be enhanced. Therefore, it is necessary to develop effective and stable method for solving geometric constraint problems involving inequalities.

1.2.3 Dynamic Geometry Software

In the last 35 years, dynamic geometry software has been studied and developed extensively. It can produce dynamic diagrams interactively or (semi-) automatically by using ruler and compass or other basic drawing tools in a similar way to how they are sketched on paper. A static diagram is a figure of fixed geometric objects satisfying certain geometric constraints or properties. A dynamic diagram is a segment of program or a piece of software that encodes information about movable geometric objects involved in a family of static diagrams and about the constraints among these objects and that can produce sequences of static diagrams of the family by updating the values of some control parameters, usually under mouse action, and show each sequence graphically on computer screen as an animation. The geometric specification of a dynamic diagram by a set of geometric objects and a set of geometric constraints among these objects may be translated, automatically and in most cases, into a semi-algebraic system of equalities and inequalities with parameters. The generation of a dynamic diagram amounts to producing a segment of program that implements the process of solving the corresponding parametric semi-algebraic system and visualizes the geometric configurations corresponding to the real solutions of the system for changing values of the parameters.

Dynamic diagrams have features of interaction, intelligence and real time. By observing the animation of a dynamic diagram, we can learn the relations of the geometric objects more directly or even find out new geometry theorems. Therefore, it has been a common view that dynamic geometric software significantly helps users acquire not only knowledge about geometric objects, but also mathematical rigour more generally. Recently there are a large number of software systems available for creating dynamic diagrams.[11] We briefly introduce some typical ones below.

Geometer's Sketchpad [65] is one of the most popular and complete interactive geometry software systems for investigating Euclidean geometry, algebra, calculus, and some other areas of mathematics. It unifies the traditional geometric construction tools, and allows users create dynamic diagrams that can be constructed with rulers and compass. Furthermore, it can also allows users to produce diagrams that cannot be constructed under the traditional compass-and-straight-edge rules by employing transformations. This system almost includes all the fundamental functions needed in dynamic geometry research, such as measuring lengths of segments, angles, area, perimeter, creating objects in relation to selected objects and points in relation to objects including distance, angle, ratio, and others. By using these functions, users can construct a lot of different geometric objects, measure them, and potentially figure out hard-to-solve math problems.

Cinderella [108] is a proprietary interactive geometry software system, written in Java programming language. Besides all the familiar constructions of Geometer's Sketchpad, Cinderella has its own features such as supporting constructions in spherical and hyperbolic geometry, theorem proving, more general animation features, generating Java applets that paste easily onto web pages, and generating applets to create self-checking construction exercises. It is a versatile and powerful tool for studying and teaching Euclidean, hyperbolic and spherical geometry. However, in some aspects, Cinderella does worse than Sketchpad, for example, less paper-oriented, that is, hard to paster diagrams into paper documents, and less efficiency for quick diagram constructions.

GeoGebra[12] is a system which can construct dynamic diagrams with points, vectors, segments, lines, polygons, conic sections, implicit polynomials and functions, and animate them with mouse click and dragging. In this system, users can not only enter and modify geometric elements directly on screen, but also through the Input Bar. GeoGebra has some features of its own, such as using variables to represent numbers, vectors and points, computing derivatives and integrals of functions, and providing a full complement of commands like Root or Extremum. These features of GeoGebra can help teachers, students and researchers make conjectures and prove

[11] http://en.wikipedia.org/wiki/List_of_interactive_geometry_software

[12] http://www.geogebra.org/cms/

geometric theorems.

GCLC [38] is a geometric tool that can be used for visualizing geometric configurations, teaching geometry and producing mathematical illustrations. A range of easy-to-use functionalities are provided by GCLC, such as geometrical constructions, transformations, parametric curves, flow control, symbolic expressing and automated theorem proving. Knowing that, for the constructions in GCLC, formal procedures are more important than drawings, that is, producing mathematical illustrations is based on "describing configurations" rather than "drawing configurations". Furthermore, figures can be displayed and exported into LaTeX and other formats.

GEOTHER [131] is a module of Epsilon, which implemented in Maple with interface written in Java for manipulating and proving geometric theorems. One of its functions is to generate dynamic diagrams automatically from the predicate specification of a given set of geometric relations among a set of points in the plane. The whole drawing process, based on the combination of symbolic and numerical computations performed in two environments, is automatic. This program has implemented some demos for HLLW approach, showing the possibility for generating dynamic diagrams with inequality constraints while the degrees of the involved equalities are less than or equal to 3.

Gabri 3D[13] is a typical system for the visualization of 3D dynamic diagrams. It helps users to discover the properties of 3D figures, find theories then test them in a advanced way, and better understand three-dimensional space. In order to solve problems, Cabri 3D Geometry includes the use of length, area, geometric configurations, and also illustrates arc length, area of sectors of circles, lateral area, surface area, and lots of two- and three-dimensional figures. This system has implemented some functionalities, such as spatial reasoning and geometric modeling. It can also visualize 2D diagrams and help analyzing characteristics and properties of two-dimensional space.

There are a wide variety of dynamic geometry systems available and each of them has some fundamental functions and its own features. However, most of them can only draw the diagrams that can be expressed as linear or quadratic equalities, and cannot deal with inequality constraints. Therefore, it is imperative to develop effective software tools that can tackle geometric constraints involving high degree equalities,

[13] http://www.cabri.com

and also can handle inequalities as well as equalities. This thesis is devoted to studying and solving these problems.

1.3 Contributions of This Thesis

In this thesis, generation of dynamic diagrams with equality and inequality constraints is mainly researched. The geometric specification of a dynamic diagram by a set of geometric objects and a set of geometric constraints among the objects may be translated into a semi-algebraic system of equalities and inequalities with parameters. The generation of a dynamic diagram amounts to solving the corresponding parametric semi-algebraic system, which includes triangular decomposition for solving systems of polynomial equations with parameters, real solution formulas with radicals for solving univariate equations, and quantifier elimination for dealing with inequality constraints.

Following the HLLW approach, we present an approach and its implementation for automatically generating dynamic diagrams, which are distinct from other existing approaches and implementations by their capabilities of dealing with inequality constraints and ensuring numeric stability in diagram animation. In order to enhance stability, we are concerned with the solution formulas of cubic and quartic polynomial equations.

We discover that Lagrange formula of cubic equations is a bit ambiguous since there are 18 possible interpretations of the formula. However, some interpretations are correct (yielding the solutions), but the others are not. The usual condition for choosing the correct interpretation is depending on the coefficients of the polynomial. It is natural and interesting to ask whether there is a uniform condition which is independent of the polynomial. The question essentially amounts to whether there is a convention for choosing square root and cubic root that will yield correct interpretations for all cubic polynomials. We use a counter example to prove that the standard convention is incorrect. In particular, we provide a real convention (for square and cubic roots) which provides correct interpretations of the Lagrange formula for all cubic polynomial equations with real coefficients, and prove its correctness. Using this convention, we also present real solution formulas in radicals for the general real-coefficient cubic equations under equality and inequality constraints. The real constraints in the formulas are given as three existentially quantified subformulas. If needed, one could eliminate the existential quantifier using, e.g., the method based on partial cylindrical algebraic decomposition.

It is natural to think whether the real convention is correct for the quartic formulas, and how to generalize the real solution formulas from the cubic to the quartic case. With a lot of investigations and experiments, we adjust the Lagrange-type solution formula for the quartic equations and prove that the real convention can also provide correct interpretations of the adjusted formula for all quartic polynomial equations with real coefficients. Using this convention, we also present a real solution formula in radicals for the general quartic equation with real coefficients under equality and inequality constraints. The extension of the solution formulas from the cubic to the quartic case is not straightforward and needs some quite sophisticated techniques including those used in the study of the Sturm-Habicht sequence and discrimination systems.

The real convention we found is a uniform convention that does not depend on the coefficients of the polynomials and that can always yield correct interpretations of the Lagrange-type formulas for all cubic and quartic polynomial equations with real coefficients. The new real solution formulas with constraints we give avoid divisions by small numbers and thus the numerically unstable "near 0/0" case. In various applications, such as geometric constraint solving, one needs to solve equations with gradually changing coefficients, for which a solution formula with divisions can encounter "near 0/0", resulting in significant numeric errors. Therefore, the new solution formulas are computationally better in terms of numeric stability and geometric continuity for dynamic diagram animation.

We enhance the HLLW approach by replacing the Ferro–Cardano-type solution formulas of cubic equations used there with newly introduced Lagrange-type real solution formulas with inequality constraints and by incorporating new real solution formulas of quartic equations with inequality constraints. The new formulas involve no division by small numbers. We show that for generating dynamic diagrams automatically the performance of the HLLW approach can be improved, in terms of stability of numeric computation and quality of generated diagrams, when the used solution formulas are replaced by the new ones.

A software package GeoDraw has been developed for drawing dynamic diagrams automatically based on the predicate specification of a given set of geometric relations involving inequality constraints. This package has implemented the process of generating dynamic diagrams by using the enhanced HLLW approach, and can be used to draw configurations of constructive type based on numerical computation, and configurations of declarative type based on a combination of both symbolic and numeric computations. GeoDraw allows the diagrams to be generated with several independent components and enables users to choose drawing types flexibly to improve the efficiency. The whole process of dynamic diagram generation combining triangular decomposition in Maple, real quantifier elimination in QEPCAD with parsing, numerical computation, graphic drawing, and letter labeling in Java is completely automated. The drawn diagrams may be animated and fine-tuned by mouse click and dragging with the given geometric relations maintained.

This thesis is structured as follows.

In the following chapter, basic information is provided about the algebraic methods for polynomial systems, containing solution formulas of polynomial equations, triangular sets, and real quantifier elimination, together with their implementations.

We present in Chapter 3 a real convention (for square and cubic roots) which provides correct interpretations of the Lagrange formula for all cubic polynomial equations with real coefficients, and prove its correctness. Using this convention, we also present real solution formulas in radicals for the general cubic equations with real coefficients under equalities and inequalities.

In Chapter 4, we generalize the solution formulas from the cubic to the quartic case. We adjust the Lagrange-type formulas for quartic equations and prove that the real convention is also correct for this adjusted quartic formulas. Using this convention, we also present a real solution formula in radicals for the general quartic equation with real coefficients under equality and inequality constraints.

Then in Chapter 5 we recall the original HLLW approach and enhance it by incorporating the Lagrange-type real solution formulas of cubic and quartic equations with inequality constraints introduced in Chapter 3 and Chapter 4. Several examples are presented to illustrate the enhanced approach and to demonstrate the advantages and effectiveness of the new solution formulas.

In Chapter 6, we discuss the design and implementation of GeoDraw, point out the functionalities and features of the current version, provide several examples to show the usage of GeoDraw, and present some empirical data in table form to show the performance of the package for generating the diagrams of 15 theorems in plane Euclidean geometry.

The thesis is concluded with some discussions on future work in Chapter 7.

Chapter 2

Fundamental Tools from Polynomial Algebra

Prior to explaining how to apply the algebraic methods to solve polynomial systems, we review some algebraic knowledge and methods in polynomial algebra as preparation. In this chapter, we provide the definitions of "convention" and "standard convention" for square and cubic roots, the solution formulas of cubic and quartic polynomial equations, an algorithm of determining the number of distinct real roots for a polynomial, and the basic information about the methods of triangular decomposition and quantifier elimination over the real field. Formal descriptions of these methods are available from standard textbooks, see, e.g., [7, 28, 35, 133].

2.1 Solution Formulas of Polynomial Equations

Abel–Ruffini theorem (also known as Abel's impossibility theorem) states that there is no general algebraic solution in radicals to polynomial equations of degree five or higher [99]. Therefore, only linear, quadratic, cubic and quartic equations have explicit solution formulas in radicals. In this section, we introduce the standard conventions for the square and cubic roots, Ferro–Cardano-type and Lagrange-type solution formulas in radicals of cubic and quartic polynomial equations.

2.1.1 Standard Conventions for Square and Cubic Roots

In the following, we will frequently use notions such as "convention" for choosing square and cubic roots. In this section, we make those notions precise.

Every complex number x can be written as $re^{i\phi}$, where $r \geq 0$ and $-\pi < \phi \leq \pi$. We use the notation $\arg x = \phi$, where $\arg x$ is a function that extracts the principal value of the angular component of x. Hence $-\pi < \arg x \leq \pi$.

Definition 2.1.1 (Convention)**.** A *convention* is a pair (C_2, C_3) of functions $C_n : \ (-\pi, \pi] \longrightarrow (-\pi, \pi]$ such that $nC_n(\phi) \equiv \phi \mod 2\pi$. Under the convention, we

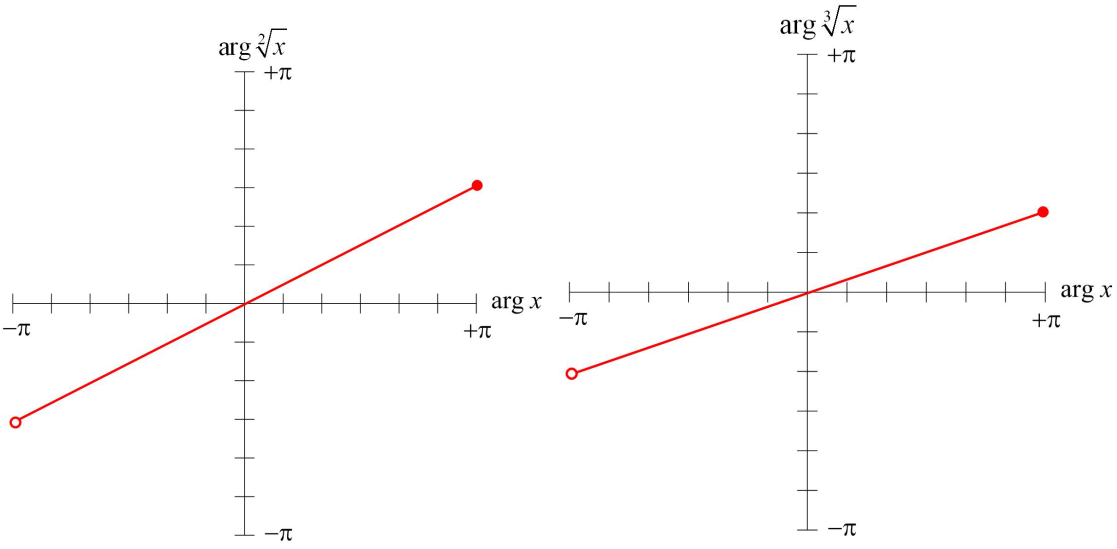

Figure 2.1: Standard Convention for the Square and Cubic Roots

choose square root $\sqrt[2]{x}$ and cubic root $\sqrt[3]{x}$ of x so that

$$\arg \sqrt[2]{x} = C_2(\arg x), \quad \arg \sqrt[3]{x} = C_3(\arg x).$$

Now let us review the "standard" convention.

Definition 2.1.2 (Standard Convention)**.** The *standard convention* chooses the square root $\sqrt[2]{x}$ and cubic root $\sqrt[3]{x}$ of x so that

$$\arg \sqrt[2]{x} = \frac{1}{2} \arg x, \quad \arg \sqrt[3]{x} = \frac{1}{3} \arg x,$$

(See Figures 2.1).

This standard convention is commonly used for solving quadratic,cubic and quartic equations (see, e.g., [107]).

2.1.2 Ferro–Cardano-type Solution Formulas

Throughout the 15th and the early 16th centuries, many mathematicians worked on the problem of giving an algebraic solution to the cubic equation [4, 18, 43, 55]. In early 1500's, Ferro gave a general solution to cubics of the form $x^3 + cx = b$. In fact, all cubic equations can be reduced to this form. In 1540, Ferrari discovered the solution to the quartic equations which, like all algebraic solutions of the quartic, requires the solution of a cubic. Both the solutions of the cubic and quartic equations were published by Ferrari's mentor Cardano in the book *Ars Magna* [27, 55] in 1545.

Let $\mathbb{C}$ denotes the field of complex numbers. In the following, we present the cubic solution formula due to Ferro and communicated by Cardano in Theorem 2.1.3, and the quartic solution formula due to Ferrari in Theorem 2.1.4 with the inside subsidiary cubic equation solved by the method of Ferro–Cardano.

Theorem 2.1.3 (Ferro's Cubic Formula). *Let $f(x) = x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{C}[x]$, the formula for the three solutions x_1, x_2, x_3 of the equation $f(x) = 0$ is*

$$\begin{aligned}
x_1 &= \frac{\omega^0\sqrt[3]{\delta} - 2\,a_2}{6} - \frac{6\,a_1 - a_2^2}{3\,\omega^0\sqrt[3]{\delta}}, \quad x_2 = \frac{\omega^1\sqrt[3]{\delta} - 2\,a_2}{6} - \frac{6\,a_1 - a_2^2}{3\,\omega^1\sqrt[3]{\delta}},\\
x_3 &= \frac{\omega^2\sqrt[3]{\delta} - 2\,a_2}{6} - \frac{6\,a_1 - a_2^2}{3\,\omega^2\sqrt[3]{\delta}}, \quad \omega = \mathrm{e}^{i\frac{2\pi}{3}} = -\frac{1}{2} + \frac{\sqrt{3}}{2}i,\\
\delta &= 36\,a_2a_1 - 108\,a_0 - 8\,a_2^3 + 12\sqrt{-3\,p_1},\\
p_1 &= a_2^2a_1^2 + 18\,a_2a_1a_0 - 4\,a_1^3 - 27\,a_0^2 - 4\,a_2^3a_0.
\end{aligned}$$

Theorem 2.1.4 (Ferrari's Quartic Formula). *Let $f(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{C}[x]$, the formula for the four solutions x_1, x_2, x_3, x_4 of the equation $f(x) = 0$ is,*

$$\begin{aligned}
x_1 &= \frac{-a_3 + d}{4} + \sqrt{\frac{(a_3 - d)^2}{16} - \frac{a_1 + t(-a_3 + d)}{d}},\\
x_2 &= \frac{-a_3 + d}{4} - \sqrt{\frac{(a_3 - d)^2}{16} - \frac{a_1 + t(-a_3 + d)}{d}},\\
x_3 &= \frac{-a_3 - d}{4} + \sqrt{\frac{(a_3 + d)^2}{16} + \frac{a_1 - t(a_3 + d)}{d}},\\
x_4 &= \frac{-a_3 - d}{4} - \sqrt{\frac{(a_3 + d)^2}{16} + \frac{a_1 - t(a_3 + d)}{d}},\\
d &= \sqrt{a_3^2 - 4\,a_2 + 8\,t},
\end{aligned}$$

and t is one of the real solutions of the cubic polynomial

$$l(t) = 8\,t^3 - 4\,a_2t^2 + (2\,a_3a_1 - 8\,a_0)t + 4\,a_2a_0 - a_3^2a_0 - a_1^2,$$

which can be solved using the Ferro's method in Theorem 2.1.3.

Note that Ferro–Cardano and Ferrari's formulas involve divisions. Thus they may encounter a numerically unstable case (i.e., "near 0/0" case), when both the numerator and denominator are close to zero.

2.1.3 Lagrange-type Solution Formulas

Lagrange [114] introduced a new method to solve the equations of low degree in [79]. This method works well for cubic and quartic equations, but Lagrange did not succeed in applying it to a quintic equation, because it requires solving a resolvent polynomial of degree at least six. In the following, we recall the cubic and quartic formulas which are found due to Lagrange.

Theorem 2.1.5 (Lagrange's Cubic Formula). *Let $f(x) = x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{C}[x]$, the formula for the three solutions x_1, x_2, x_3 of the equation $f(x) = 0$ is,*

$$x_1 = (-a_2 + \omega^1 c_1 + \omega^2 c_2)/3, \qquad c_1 = \sqrt[3]{(p_2 + 3\,s)/2}, \qquad s = \sqrt[2]{-3\,p_1},$$

$$x_2 = (-a_2 + \omega^0 c_1 + \omega^0 c_2)/3, \qquad c_2 = \sqrt[3]{(p_2 - 3\,s)/2},$$

$$x_3 = (-a_2 + \omega^2 c_1 + \omega^1 c_2)/3, \qquad \omega = \mathrm{e}^{i\frac{2\pi}{3}} = -\tfrac{1}{2} + \tfrac{\sqrt{3}}{2} i,$$

$$p_1 = a_2^2 a_1^2 + 18\,a_2 a_1 a_0 - 4\,a_1^3 - 27\,a_0^2 - 4\,a_2^3 a_0,$$

$$p_2 = 9\,a_2 a_1 - 27\,a_0 - 2\,a_2^3.$$

Theorem 2.1.6 (Lagrange's Quartic Formula). *Let $f(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{C}[x]$, the formula for the four solutions x_1, x_2, x_3, x_4 of the equation $f(x) = 0$ is,*

$$x_1 = \frac{m_1 + \sqrt{m_1^2 - 4\,h_1}}{2}, \quad x_2 = \frac{m_1 - \sqrt{m_1^2 - 4\,h_1}}{2}, \quad x_3 = \frac{m_2 + \sqrt{m_2^2 - 4\,h_2}}{2},$$

$$x_4 = \frac{m_2 - \sqrt{m_2^2 - 4\,h_2}}{2}, \quad m_1 = \frac{-a_3 + \sqrt{a_3^2 - 4\,y_1}}{2}, \quad m_2 = \frac{-a_3 - \sqrt{a_3^2 - 4\,y_1}}{2},$$

$$h_1 = \frac{(y_2 + y_3 - y_1) + \sqrt{(y_2 + y_3 - y_1)^2 - 16\,a_0}}{4},$$

$$h_2 = \frac{(y_2 + y_3 - y_1) - \sqrt{(y_2 + y_3 - y_1)^2 - 16\,a_0}}{4},$$

and y_1, y_2, y_3 are the three solutions of the cubic polynomial

$$g(y) = y^3 - 2\,a_2 y^2 + (a_2^2 - 4\,a_0 + a_1 a_3)y + (a_1^2 + a_0 a_3^2 - a_1 a_2 a_3),$$

which can be solved using the Lagrange's cubic formula in Theorem 2.1.5.

The given formulas (usually attributed to Lagrange and based on his idea of resolvent) are inspired by but different from the well known formulas due to Ferro ane Ferrari (communicated by Cardano) in Section 2.1.2. Lagrange's formula does not require division and thus avoids the "near 0/0" case. In various applications, such as geometric constraint solving, one needs to solve equations with gradually changing coefficients, for which Ferro–Cardano-type formulas can encounter "near 0/0", resulting in significant numerical errors. Therefore, Lagrange-type formulas are better for such applications.

2.2 Number of Distinct Real Roots of a Polynomial

In this section, we introduce the complete discrimination system, a set of explicit expressions in terms of the coefficients, for determining the number of distinct real roots of a polynomial which is provided by L. Yang and coauthors [90, 147, 148].

Given a polynomial

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n,$$

we write the derivative of $f(x)$ as

$$f'(x) = 0 \cdot x^n + na_0x^{n-1} + (n-1)a_1x^{n-2} + \cdots + a_{n-1}.$$

Definition 2.2.1 (Discriminator Matrix)**.** The discriminant of the Sylvester matrix of $f(x)$ and $f'(x)$,

$$\begin{bmatrix}
a_0 & a_1 & a_2 & \cdots & a_n & & & & \\
0 & na_0 & (n-1)a_1 & \cdots & a_{n-1} & & & & \\
 & a_0 & a_1 & \cdots & a_{n-1} & a_n & & & \\
 & 0 & na_0 & \cdots & 2\,a_{n-2} & a_{n-1} & & & \\
 & & & \cdots & \cdots & & & & \\
 & & & \cdots & \cdots & & & & \\
 & & & a_0 & a_1 & a_2 & \cdots & a_n & \\
 & & & 0 & na_0 & (n-1)a_1 & \cdots & a_{n-1} &
\end{bmatrix}$$

is called the discriminator matrix of $f(x)$, and denoted by Discr(f).

Definition 2.2.2 (Discriminant Sequence)**.** Let D_k denote the discriminator of the submatrix of Discr(f), formed by the first $2\,k$ rows and the first $2\,k$ columns, for $k = 1, \ldots, n$ (note that Discr(f) is a $2\,n \times 2\,n$ matrix).

We call the n-tuple

$$[D_1, D_2, \ldots, D_n]$$

the discriminant sequence of the polynomial $f(x)$.

Sometimes, we denote it by a more detailed notation of specify $f(x)$:

$$[D_1(f), D_2(f), \ldots, D_n(f)].$$

In practice, we usually get the discriminant sequence from the Bezout matrix of $f(x)$ and $f'(x)$, which is the same one as that got from the Sylvester matrix (see [146, 150] for details). The following short program written in Maple was demonstrated to be quite efficient for producing discriminant sequences of polynomials with parametric coefficients.

```
discr:=proc(poly,var)
    local g,h,tt,d,bz,i,ar,j,mm,dd;
        d:=degree(poly,var);
        h:=tt*var^d+diff(poly,var);
        with(linalg);
        bz:=subs(tt=0,bezout(poly,h,var));
        ar:=[ ];
        for i to d do ar:=[op(ar),row(bz,d+1-i..d+1-i)] od;
        mm:=matrix(ar);
        dd:=[ ];
        for j to d do
            dd:=[op(dd),det(submatrix(mm,1..j,1..j))]
        od;
        dd:=map(primpart,dd);
end;
```

Definition 2.2.3 (Sign List)**.** We call the list

$$[\mathrm{sign}(D_1), \mathrm{sign}(D_2), \ldots, \mathrm{sign}(D_n)]$$

the sign list of the discriminant sequence $\{D_1, D_2, \ldots, D_n\}$.

Definition 2.2.4 (Revised Sign List)**.** Given a sign list $[s_1, s_2, \ldots, s_n]$, we construct a new list

$$[\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n]$$

as follows (which is called the revised sign list):

- If $s_i, s_{i+1}, \ldots, s_{i+j}$ is a section of the given list, where $s_i \neq 0, s_{i+1} = s_{i+2} = \cdots = s_{i+j-1} = 0, s_{i+j} \neq 0$, then we replace the subsection

  $$[s_{i+1}, s_{i+2}, \ldots, s_{i+j-1}]$$

  by

  $$[-s_i, -s_i, s_i, s_i, -s_i, -s_i, s_i, s_i, -s_i, \cdots],$$

  i.e.

  $$\varepsilon_{i+r} = (-1)^{[\frac{r+1}{2}]} \cdot s_i$$

  for $r = 1, \ldots, j-1$.

- Otherwise, let $\varepsilon_k = s_k$, i.e. there are no changes for other terms.

Then the following theorem can determine the number of distinct real or imaginary roots (without considering the multiplicities).

Theorem 2.2.5. *Given a polynomial $f(x)$ with real coefficients,*

$$f(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_n,$$

if the number of the sign changes of the revised sign list of

$$\{D_1(f), D_2(f), \ldots, D_n(f)\}$$

is v, then the number of the pairs of distinct conjugate imaginary roots of $f(x)$ equals v, if the number of non-vanishing members of the revised sign list is l, then the number of distinct real roots of $f(x)$ equals $l - 2\,v$.

2.3 Characteristic Sets and Triangular Decomposition

Triangular decomposition is a fundamental and well-developed tool for describing the complex solutions of polynomial systems. Characteristic sets and triangular sets are sets of multivariate polynomials which may be ordered in certain triangular form. The method of characteristic sets, developed by Wu [141, 144] based on the classical work of Ritt [110], provides a powerful tool for solving systems of multivariate polynomial equations by transforming the corresponding sets of polynomials into triangular sets. Part of the material in this section is taken from Wang's book [128].

Let $\mathcal{K}$ be a computable field of characteristic 0. The rational number field $\mathcal{Q}$ is a concrete example of $\mathcal{K}$. And let $\mathcal{K}[\boldsymbol{x}]$ be the ring of polynomial in $\boldsymbol{x} = (x_1, \ldots, x_n)$ with the coefficients in $\mathcal{K}$, and with the ordering for the indeterminates $x_1 \prec \cdots \prec x_n$. For any polynomial $P \in \mathcal{K}[\boldsymbol{x}] \setminus \mathcal{K}$ and an arbitrary indeterminate x_k, the *degree* of P in x_k is denoted by $\deg(P, x_k)$. The biggest index p such that $\deg(P, x_p) > 0$ is called the *class*, x_p the *leading variable*, and $\deg(P, x_p)$ the *leading degree*, denoted by $\mathrm{cls}(P)$, $\mathrm{lv}(P)$ and $\mathrm{ldeg}(P)$ respectively. The leading coefficient of P in x_p, denoted by $\mathrm{lc}(P, x_p)$, is called the *initial* of P, denoted by $\mathrm{ini}(P)$.

Definition 2.3.1. A finite nonempty ordered set of non-constant polynomials in $\mathcal{K}[\boldsymbol{x}]$

$$\mathbb{T} = [T_1, T_2, \ldots, T_r]$$

is called a *triangular set* or a *non-contradictory quasi-ascending set*, if

$$\mathrm{cls}(T_1) < \mathrm{cls}(T_2) < \cdots < \mathrm{cls}(T_r).$$

Any triangular set can be written in the following form

$$\mathbb{T} = \begin{bmatrix} T_1(x_1, \ldots, x_{p_1}), \\ T_2(x_1, \ldots, x_{p_1}, \ldots, x_{p_2}), \\ \cdots\cdots \\ T_r(x_1, \ldots, x_{p_1}, \ldots, x_{p_2}, \ldots, x_{p_r}), \end{bmatrix}, \tag{2.1}$$

where

$$0 < p_1 < p_2 < \cdots < p_r \leq n,$$
$$p_i = \mathrm{cls}(T_i),\ x_{p_i} = \mathrm{lv}(T_i),\ i = 1, 2, \ldots, r.$$

Now we introduce the basic operation on polynomials in the method of triangular sets, that is *pseudo-division*. Let F and G be the polynomials in $\mathcal{K}[\boldsymbol{x}]$, x_k a fixed variable, $m = \deg(F, x_k)$, and $l = \deg(G, x_k)$. For pseudo-dividing G by F, we have a division algorithm as follows. Let $R := G$; Repeat the following process until $r = \deg(R, x_k) < m$:

$$R := F_0 R - R_0 x_k^{r-m} F,$$

where $F_0 = \mathrm{lc}(F, x_k)$, $R_0 = \mathrm{lc}(R, x_k)$. As r strictly decreases for each iteration, the procedure must terminate. Finally, one obtains two polynomials Q and R in $\mathcal{K}[\boldsymbol{x}]$ satisfying the relation

$$I^q G = QF + R, \tag{2.2}$$

where $I = \mathrm{lc}(F, x_k)$, $q = \max(l - m + 1, 0)$, $\deg(R, x_k) < m$, and $\deg(Q, x_k) = \max(l - m, -1)$. In case $m = 0$, we have $R = 0$ and $Q = F^l G$.

The expression (2.2) is called a *pseudo-remainder formula*; Q is called the *pseudo-quotient* and R the *pseudo-remainder* of G with respect to F in x_k, denoted by $\mathrm{pquo}(G, F, x_k)$ and $\mathrm{prem}(G, F, x_k)$ respectively, and they are uniquely determined by F and G.

Let $\mathbb{T}$ be a triangular set as in (2.1) and P any polynomial in $\mathcal{K}[\boldsymbol{x}]$. P is said to be *reduced* with respect to $\mathbb{T}$ if P is reduced with respect to every $T \in \mathbb{T}$, i.e., $\deg(P, x_{p_i}) < \mathrm{ldeg}(T_i)$ for all i. The polynomial

$$R = \mathrm{prem}(\cdots \mathrm{prem}(P, T_r, x_{p_r}), \cdots, T_1, x_{p_1})$$

simply denoted by $\mathrm{prem}(P, \mathbb{T})$ is called the *pseudo-remainder* of P with respect to $\mathbb{T}$. From Expression (2.2), one can easily deduce the following *pseudo-remainder formula*

$$I_1^{q_1} \cdots I_r^{q_r} P = \sum_{i=1}^{r} Q_i T_i + R,$$

where each q_i is a non-negative integer, and $I_i = \mathrm{ini}(T_i)$, $Q_i \in \mathcal{K}[\boldsymbol{x}]$, $i = 1, \ldots r$.

Similarly, $\mathrm{prem}(P, \mathbb{T}) = P$ when P is reduced with respect to $\mathbb{T}$. For any polynomial set $\mathbb{P}$, $\mathrm{prem}(\mathbb{P}, \mathbb{T})$ stands for $\{\mathrm{prem}(P, \mathbb{T})\colon P \in \mathbb{P}\}$.

Let $\tilde{\mathcal{K}}$ be an extension field of $\mathcal{K}$. For any nonempty polynomial sets $\mathbb{P}$ and $\mathbb{Q}$, denote by $\mathrm{Ideal}(\mathbb{P})$ the *ideal* generated by the polynomials in $\mathbb{P}$, and we define

$$\begin{aligned}
&\mathrm{ini}(\mathbb{P}) := \{\mathrm{ini}(P) \mid P \in \mathbb{P}\},\\
&\mathrm{Zero}(\mathbb{P}) := \{\bar{\boldsymbol{x}} \in \tilde{\mathcal{K}}^n \mid P(\bar{\boldsymbol{x}}) = 0, \forall\, P \in \mathbb{P}\},\\
&\mathrm{Zero}(\mathbb{P}/\mathbb{Q}) := \{\bar{\boldsymbol{x}} \in \mathrm{Zero}(\mathbb{P}) \mid Q(\bar{\boldsymbol{x}}) \neq 0, \forall\, Q \in \mathbb{Q}\}.
\end{aligned}$$

Lemma 2.3.2. *For any triangular set* $\mathbb{T}$ *and polynomial* P *in* $\mathcal{K}[\boldsymbol{x}]$*, if* $\mathrm{prem}(P, \mathbb{T}) = 0$ *then* $\mathrm{Zero}(\mathbb{T}/\mathrm{ini}(\mathbb{T})) \subset \mathrm{Zero}(P)$*.*

Definition 2.3.3. A triangular set $\mathbb{T} = [T_1, \ldots, T_r] \subset \mathcal{K}[\boldsymbol{x}]$ is called a *non-contradictory ascending set* if T_i is reduced with respect to $[T_1, \ldots, T_{i-1}]$ for each i, $2 \leq i \leq r$.

$\mathbb{T}$ is said to be *fine* if we have

$$\mathrm{prem}(\mathrm{ini}(T_i), [T_1, \ldots, T_{i-1}]) \neq 0$$

for all $2 \leq i \leq r$.

Any set with a single non-zero constant is called a *contradictory ascending set.*

Apparently, every non-contradictory ascending set is a fine triangular set. The pseudo-remainder of any polynomial with respect to a contradictory ascending set is 0.

Definition 2.3.4. An ascending set $\mathbb{C}$ is called a *characteristic set* of a non-empty polynomial set $\mathbb{P} \subseteq \mathcal{K}[\boldsymbol{x}]$, if

$$\mathbb{C} \subset \mathrm{Ideal}(\mathbb{P}), \quad \mathrm{prem}(\mathbb{P}, \mathbb{C}) = \{0\}.$$

Here, a characteristic set of $\mathbb{P}$ is defined by Wu. Ritt's definition of a characteristic set is for the ideal $\boldsymbol{J}$ (generated by $\mathbb{P}$) and requires that $\mathrm{prem}(\boldsymbol{J}, \mathbb{C}) = \{0\}$; thus for computing $\mathbb{C}$ one has to consider its *irreducibility* or use alternative algorithm (see [95], Sect. 5.6).

Proposition 2.3.5. *Let* $\mathbb{C} = [C_1, \ldots, C_r]$ *be a characteristic set of* $\mathbb{P} \subseteq \mathcal{K}[\boldsymbol{x}]$*, and*

$$\begin{aligned}
&I_i = \mathrm{ini}(C_i), \quad \mathbb{P}_i = \mathbb{P} \cup \{I_i\}, \quad i = 1, \ldots, r,\\
&\mathcal{I} = \mathrm{ini}(\mathcal{C}) = \{I_1, \ldots, I_r\}.
\end{aligned}$$

Then

$$\mathrm{Zero}(\mathbb{C}/\mathbb{I}) \subset \mathrm{Zero}(\mathbb{P}) \subset \mathrm{Zero}(\mathbb{C}),$$
$$\mathrm{Zero}(\mathbb{P}) = \mathrm{Zero}(\mathbb{C}/\mathbb{I}) \cup \bigcup_{i=1}^{r} \mathrm{Zero}(\mathbb{P}_i) \tag{2.3}$$

in $\mathcal{K}$ *or any extension field of* $\mathcal{K}$.

Before presenting the characteristic set algorithm of Ritt–Wu, we introduce the notion of *rank*.

Definition 2.3.6. For two non-zero polynomials F and G in $\mathcal{K}[\boldsymbol{x}]$, F is said to have a *lower rank* than G, which is denoted as $F \prec G$ or $G \succ F$ if either $\mathrm{cls}(F) < \mathrm{cls}(G)$, or $\mathrm{cls}(F) = \mathrm{cls}(G) > 0$ and $\mathrm{ldeg}(F) < \mathrm{ldeg}(G)$. In this case, G is said to have a *higher rank* than F.

If neither $F \prec G$ nor $G \prec F$, F and G are said to have the *same rank*, denoted as $F \sim G$.

We write $F \precsim G$ for "$F \prec G$ or $F \sim G$," and similarly for "$\succsim$."

Definition 2.3.7. For two triangular sets

$$\mathbb{T} = [T_1, \ldots, T_r], \quad \mathbb{T}' = [T'_1, \ldots, T'_{r'}],$$

$\mathbb{T}$ is said to have a *higher rank* than $\mathbb{T}'$, which is denoted as $\mathbb{T} \succ \mathbb{T}'$ or $\mathbb{T}' \prec \mathbb{T}$, if either there exists a $j \leq \min(r, r')$ such that

$$T_1 \sim T'_1, \ldots, T_{j-1} \sim T'_{j-1}, \quad \text{while } T_j \succ T'_j;$$

or $r' > r$ and $T_1 \sim T'_1, \ldots, T_r \sim T'_r$. In this case, $\mathbb{T}'$ is said to have a *lower rank* than $\mathbb{T}$. If neither $\mathbb{T} \prec \mathbb{T}'$ nor $\mathbb{T}' \prec \mathbb{T}$, $\mathbb{T}$ and $\mathbb{T}'$ are said to have the *same rank*, denoted as $\mathbb{T} \sim \mathbb{T}'$. In this case, $r = r'$ and $T_1 \sim T'_1, \ldots, T_r \sim T'_r$.

The above-defined "$\precsim$" is a partial order, under which the collection of all triangular sets is partially ordered. Thus, for any set of ascending sets, one is free to talk about the notion of *minimal ascending set* if it exists.

Lemma 2.3.8. *Let $\mathbb{T}_1 \succsim \mathbb{T}_2 \succsim \cdots \succsim \mathbb{T}_k \succsim \cdots$ be a sequence of triangular sets whose ranks never increase. Then there exists a k' such that $\mathbb{T}_k \sim \mathbb{T}_{k'}$ for all $k \geq k'$.*

Consider any non-empty polynomial set $\mathbb{P}$. Let Φ be the set of all ascending sets contained in $\mathbb{P}$. Since each single polynomial forms by itself an ascending set, $\Phi \neq \emptyset$. Any minimal ascending set of Φ is called a *basic set* of $\mathbb{P}$. Such a basic set exists and can be determined by the following algorithm.

Algorithm BasSet: $\mathbb{B}$:=BasSet($\mathbb{P}$). Given a non-empty polynomial set $\mathbb{P} \subset \mathcal{K}[\boldsymbol{x}]$, this algorithm computes a basic set $\mathbb{B}$ of $\mathbb{P}$.

B1. Set $\mathbb{F} := \mathbb{P}$, $\mathbb{B} := \emptyset$.

B2. While $\mathbb{F} \neq \emptyset$ do:

B2.1. Let B be an element of $\mathbb{F}$ with lowest rank.

B2.2. Set $\mathbb{B} := \mathbb{B} \cup [B]$.

B2.3. If $\mathrm{cls}(B) = 0$ then set $\mathbb{F} := \emptyset$ else set

$$\mathbb{F} := \{F \in \mathbb{F} \setminus \{B\} \mid F \text{ is reduced with respect to } B\}.$$

A basic set of $\mathbb{P}$ is contradictory if and only if $\mathbb{P}$ contains a constant. In this case Algorithm BasSet terminates at the first iteration of the while-loop.

Now we are ready to present the characteristic set algorithm of Ritt–Wu, which points out how to construct a characteristic set from any given polynomial set.

Algorithm CharSet: $\mathbb{C}$:= CharSet($\mathbb{P}$). Given a non-empty polynomial set $\mathbb{P} \subset \mathcal{K}[\boldsymbol{x}]$, this algorithm computes a characteristic set $\mathbb{C}$ of $\mathbb{P}$.

C1. Set $\mathbb{F} := \mathbb{P}$, $\mathbb{R} := \mathbb{P}$.

C2. While $\mathbb{R} \neq \emptyset$ do:

B2.1. Compute $\mathbb{C} := \mathrm{BasSet}(\mathbb{F})$.

B2.2. If $\mathbb{C}$ is contradictory then set $\mathbb{R} := \emptyset$ else compute

$$\mathbb{R} := \mathrm{prem}(\mathbb{F} \setminus \mathbb{C}, \mathbb{C}) \setminus \{0\},$$

and set $\mathbb{F} := \mathbb{F} \cup \mathbb{R}$.

The characteristic set algorithm is the kernel of the zero decomposition algorithms CharSer, IrrCharSer and IrrCharSerE, and has many applications. In this thesis, we mainly use the IrrCharSer algorithm to compute the irreducible characteristic series (Definition 2.3.9) of a given polynomial set.

Let us return to the zero relation (2.3). As each I_i is reduced with respect to $\mathbb{C}$, any basic set of the polynomial set $\mathbb{P}_i \cup \mathbb{C}$ has a lower rank than that of $\mathbb{C}$. Note that $\mathrm{Zero}(\mathbb{P}_i \cup \mathbb{C}) = \mathrm{Zero}(\mathbb{P}_i)$. Therefore, in proceeding further with each $\mathbb{P}_i \cup \mathbb{C}$ by means of Algorithm CharSet, one may arrive after a finite number of steps at a zero decomposition of the form

$$\mathrm{Zero}(\mathbb{P}) = \bigcup_{i=1}^{e} \mathrm{Zero}(\mathbb{C}_i/\mathbb{I}_i), \tag{2.4}$$

in which $\mathbb{C}_i$ is an ascending set and $\mathbb{I}_i = \mathrm{ini}(\mathbb{C}_i)$ for each i.

Definition 2.3.9. A finite set or sequence Ψ of (weak-) ascending sets $\mathbb{C}_1, \ldots, \mathbb{C}_e$ is called a (weak-) *characteristic series* of a polynomial set $\mathbb{P}$ in $\mathcal{K}[\boldsymbol{x}]$ if (2.4) holds and $\mathrm{prem}(\mathbb{P}, \mathbb{C}_i) = \{0\}$ for every i.

If $\Psi = \emptyset$, it is meant that $e = 0$ and thus $\mathrm{Zero}(\mathbb{P}) = \emptyset$.

Any system $[\mathbb{P}, \mathbb{Q}]$ can be decomposed into systems $[\mathbb{T}_i, \mathbb{U}_i]$ with the following zero decomposition

$$\mathrm{Zero}(\mathbb{P}/\mathbb{Q}) = \bigcup_{i=1}^{e} \mathrm{Zero}(\mathbb{T}_i/\mathbb{U}_i), \tag{2.5}$$

where each $\mathbb{T}_i$ is a triangular set, and each $[\mathbb{T}_i, \mathbb{U}_i]$ is called a *triangular system*. Another alternative zero decomposition to (2.4) is

$$\mathrm{Zero}(\mathbb{P}) = \bigcup_{i=1}^{e} \mathrm{Zero}(\mathrm{sat}(\mathbb{T}_i)), \tag{2.6}$$

where $\mathrm{sat}(\mathbb{T}_i)$ denotes the saturated ideal of $\mathbb{T}_i$ for which a finite basis may be computed from $\mathbb{T}_i$. Effective algorithms other than Wu's [144] have been proposed by Lazard [83], Kalkbrener [70], and Wang [124, 126, 127] to decompose any polynomial set $\mathbb{P}$ or system $[\mathbb{P}, \mathbb{Q}]$ into triangular sets $\mathbb{T}_i$ or triangular systems $[\mathbb{T}_i, \mathbb{U}_i]$ of various kinds such that (2.5) and (2.6) holds. Such triangular sets and systems may be *regular*, *simple*, or *irreducible* [5, 6, 128]. Different kinds of triangular sets and systems have different properties, require different costs of computation, and therefore have different applicabilities. The methods of Wu and others have been studied, improved, and extended by different researchers (see [5, 52, 128, 144, 149, 150] and references therein) and some of them may be more efficient than others. As an example let us introduce the *regular set*.

Definition 2.3.10. A triangular set $\mathbb{T} = [T_1, \ldots, T_r] \subseteq \mathcal{K}[\boldsymbol{x}]$ is called a *regular set* if

$$\mathrm{res}(\,\mathrm{ini}(T_i), [T_1, \ldots, T_{i-1}]) \neq 0, \quad \text{for } 2 \leq i \leq r,$$

where $\mathrm{res}(\,\mathrm{ini}(T_i), [T_1, \ldots, T_{i-1}])$ is the Sylvester resultant of $\mathrm{ini}(T_i)$ with respect to the triangular set $[T_1, \ldots, T_{i-1}]$.

A regular set is also called a *regular chain* by Kalkbrener [70]. Let $\mathbb{T} = [T_1, \ldots, T_r]$ be a regular set in $\mathcal{K}[\boldsymbol{x}]$. Denote the variables which are not leading variables in $\mathbb{T}$ by $\boldsymbol{v} = (v_1, \ldots, v_{n-r})$. Note that the variables $v_1, \ldots, v_{n-r}$ are evaluated by the values which form an algebraically independent set over $\mathcal{K}$, i.e., they do not annihilate any nonzero polynomial in $\mathcal{K}[\boldsymbol{v}]$. We keep the components of $\boldsymbol{v}$ as symbols, then we call any zero $(\boldsymbol{v}, \xi_1, \ldots, \xi_r) \in \mathrm{Zero}(\mathbb{T})$ a *regular zero* of $\mathbb{T}$. For any regular zero $\boldsymbol{\xi}_i = (\boldsymbol{v}, \xi_1, \ldots, \xi_i)$ of $[T_1, \ldots, T_i]$ $(1 \leq i \leq r-1)$, we have

$$\mathrm{ini}(T_{i+1}(\boldsymbol{\xi}_i)) \neq 0.$$

From this property, we know that the number of regular zeros of each triangular set $[T_1, \ldots T_r]$ is equal to $\mathrm{ldeg}(T_1) \cdots \mathrm{ldeg}(T_r)$ (with multiplicities).

The method based on triangular sets is particularly efficient for geometric problems, thus it is a powerful tool for mechanical theorem proving in elementary geometry, and provides a complete decision process for certain classes of problems. To prove a class of geometric theorems of *equality type*, one first needs to algebraize the hypotheses and conclusions of the geometric theorems as polynomial equations. Then the relationship between these algebraic equations is studied by the computation of characteristic sets and pseudo-remainders, and thus the non-degeneracy conditions under which the theorem is true may be obtained.

Triangular sets have a nice structure suitable for representing field extensions and zeros of polynomial systems of any dimension. The various zero decompositions may be applied naturally to solving systems of polynomial equations and inequations. Consider the systems of polynomial equations $\{P_1(\boldsymbol{x}) = 0, \ldots, P_s(\boldsymbol{x}) = 0\}$, where P_i are polynomials in $\boldsymbol{x} = (x_1, \ldots, x_n)$ over $\mathcal{K}$ (eg, $\mathcal{Q}$). Let $\mathbb{P} = \{P_1, \ldots, P_s\}$. We write the system as $\mathbb{P} = 0$. Our purpose is to find the solutions of $\mathbb{P} = 0$ over the algebraically closed field extension of $\mathcal{K}$ (eg, $\mathcal{C}$). Some theorems and algorithms enable us to decompose the system into one or several regular sets $[\mathbb{T}_1, \ldots \mathbb{T}_e]$, called a *regular series*. The following theorem, regarded as the principle for polynomial system solving, helps us to determine whether system $\mathbb{P} = 0$ has solutions and whether or not the number of the solutions is finite.

Theorem 2.3.11. *Let $[\mathbb{T}_1, \ldots \mathbb{T}_e]$ be a regular series of any polynomial set $\mathbb{P} \subseteq \mathcal{K}[\boldsymbol{x}]$. Then:*

- (a) $\mathbb{P} = 0$ *has no solution in any extension field of $\mathcal{K}$ if and only if the series is a empty set;*
- (b) $\mathbb{P} = 0$ *has finitely many solutions if and only if the number of polynomials in $\mathbb{T}_i$ ($|\mathbb{T}_i|$) is the number of variables, for $1 \leq i \leq e$. In this case, the solutions of $\mathbb{P} = 0$ may be found by means of computing $\mathrm{Zero}(\mathbb{T}_i / \mathrm{ini}(\mathbb{T}_i))$ for $1 \leq i \leq e$.*

Now we consider the case for the system to have a finite number of solutions, then each $\mathbb{T}_i$ is associated to a system of equations of the following triangular form

$$\begin{cases} T_{i_1}(x_1) = 0, \\ T_{i_2}(x_1, x_2) = 0, \\ \quad \cdots \\ T_{i_n}(x_1, x_2, \ldots, x_n) = 0. \end{cases}$$

The solutions of the system above may be obtained by solving the first univariate equation in x_1, substituting the solutions for x_1 into the other ones, then solving the second equation only in x_2, and so on. Then according to the zero decomposition, the solutions of the system $\mathbb{P} = 0$ can be found after the solutions of each $\mathbb{T}_i$ have been obtained. Similarly, in order to solve systems of polynomial equations and inequations or systems involving parameters, one may decompose the systems into several sets of triangular systems satisfying some additional requirements, such as regular systems and simple systems, and then compute the solutions of each triangular system.

The method of characteristic sets was implemented by Wu and his students, and other researchers in the 1980s mainly for the purpose of automated theorem proving in geometry. In the 1990s, several implementation were made in different languages by members of Wu's extended group (see [52], chap. 20), in Scratchpad II by M. C. Gontard, and in REDUCE[1] by R. Bradford. The Maple package CharSets developed by Wang on the basis of Wu's method now is available as a module in Wang's Epsilon library[2] [130]. Part of this package has been reimplemented in Singular[3] and Macaulay 2[4] by M. W. Messollen and translated into Maxima[5] by D. Stanger. A comparative implementation of the four methods of Wu [141, 142], D. Lazard [83], M. Kalkbrener [70], and Wang [124] for solving polynomial systems in Axiom[6] was realized by P. Aubry and M. Moreno Maza [6]. The algorithm for computing regular chains has been implemented as RegularChains library in Maple by F. Lemaire and others. The above-mentioned Epsilon library includes a complete implementation of Wang's algorithms for computing triangular systems, regular systems, and simple systems.

2.4 Quantifier Elimination

Before describing the problem of quantifier elimination over real closed fields, we give some basic definitions. Below, the abbreviation QE stands for quantifier elimination.

Definition 2.4.1. A *standard atomic formula* is an expression of the form $F \sim 0$, where F is any nonzero polynomial in $\mathbb{Q}[\boldsymbol{x}]$ and $\sim$ is a predicate symbol $(=, >, \geq, <, \leq,$ or $\neq)$.

A *standard quantifier-free formula* is an expression consisting of standard atomic formulas which are combined using the boolean operators $\wedge$ (and), $\vee$ (or), and $\Rightarrow$ (implies).

Given a quantified formula in the prenex form

$$(\mathsf{Q}_{k+1} x_{k+1}) \cdots (\mathsf{Q}_n x_n) \, \Gamma(x_1, \ldots, x_n), \tag{2.7}$$

[1] `http://www.reduce-algebra.com/`
[2] `http://www-calfor.lip6.fr/~wang/epsilon/`
[3] `http://www.singular.uni-kl.de/`
[4] `http://www.math.uiuc.edu/Macaulay2/`
[5] `http://maxima.sourceforge.net/`
[6] `http://www.axiom-developer.org/`

where Q_i is either an existential ($\exists$) or a universal ($\forall$) quantifier and Γ is a standard quantifier-free formula in $x_1, \ldots, x_n$, quantifier elimination is the procedure which transforms the quantified formula (2.7) into an equivalent quantifier-free formula.

Tarski showed that for every first-order quantified formula over real closed fields, there exists an equivalent quantifier-free formula, i.e., the QE problem is solvable. Furthermore, in 1951, he first proposed an explicit algorithm for solving the QE problem [120], which has very high computational complexity. A much more efficient algorithm for QE over real closed fields is *cylindrical algebraic decomposition* (CAD) due to G. E. Collins [31].

The basic idea of the CAD method is to decompose $\mathbb{R}^n$ into finitely many cylindrically arranged regions, called *cells*, such that all the polynomials in Γ are sign-invariant in each cell. The algorithm for computing a CAD also provides a point in each cell, called *sample point*. Since the signs of all the polynomials in each cell of the decomposition can be easily determined by computing the values of the polynomials at the sample point, one is able to eliminate, by computing a CAD, the quantifiers of any quantified formula over real closed fields. The basic CAD construction consists of two key phases: the projection and lifting phases. In the projection phase, one projects n-variate polynomials to $(n-1)$-variate ones by eliminating one variable using an appropriate *projection operator*, then to $(n-2)$-variate ones, and finally to univariate polynomials. A CAD of $\mathbb{R}$ can be constructed from the univariate polynomials, for example, using a method of real root isolation. The second phase (lifting) constructs a decomposition of $\mathbb{R}^2$ from one-dimensional cells, and to a decomposition of $\mathbb{R}^n$ successively. The key problem in the construction of CADs is the definition of the projection operator, and the basic operations consist of the computations of (sub)resultants and (sub)discriminants.

The CAD method has been improved by Collins, his students, and other researchers. One of the notable improvements is made by Collins and Hong [32] for the lifting phase by constructing partial CADs. The QEPCAD package[7], developed originally by Hong, improved and maintained by Brown now, is perhaps the most well-known software tool for QE using partial CAD. Besides, the QE method by partial CAD is also implemented in Mathematica, REDLOG package[8] by A. Dolzmann

[7] `http://www.cs.esna.edu/~qepcad/B/QEPCAD.html`

[8] `http://redlog.dolzmann.de`

and others [39] in REDUCE, and SyNRAC package by A. Anai and H. Yanami [3] in Maple.

The worst case complexity of the CAD method is doubly exponential in the number of variables of the input quantified formula. In order to speed up the QE procedure, some algorithms specialized for particular types of QE problems, concerning the practical applications, have been developed. We may mention in particular three methods for effectively solving the QE problems in special cases. The QE method by *virtual term substitution* for the case that the input formulas have low degrees in their quantified variables, for example, linear, quadratic, or cubic, was developed by V. Weispfenning [136, 137]. This method, though efficient for solving many practical problems, may produce a large number of quantifier-free equivalent formulas which need further simplifications. Implementations of the virtual term substitution method are available in Mathematica as functions `Resolve` and `Reduce`, REDLOG, and SyNRAC. Another technique for QE problems, based on real root counting, also proposed by Weispfenning, is called *Hermitian quantifier elimination* [133]. It focuses on a class of problems whose quantified formulas involve many equations and only few other atomic formulas. For the QE problems concerning positive definiteness of polynomials (i.e., $\forall \boldsymbol{x} > 0\ P(\boldsymbol{x}) > 0$, where $P(\boldsymbol{x}) \in \mathbb{Q}[\boldsymbol{x}]$), L. González-Vega has developed a method by using the Sturm–Habicht sequence for the polynomials. The two above-mentioned methods are implemented in REDLOG and SyNRAC respectively. Each method has its own advantages and limitations and none of them is superior to another one in general. The combinations of some methods, such as that of the methods of partial CAD and virtual term substitution, implemented in REDLOG, are expected to perform well.

To improve the efficiency of quantifier elimination, one may write the quantified formulas in a simple and appropriate way, choose the best elimination order of the quantified variables, or/and perform QE separately on several parts divided from the formulas and then combine the results (see, e.g., [117]). These strategies need a lot of observations, experiences, and good understanding of the problems. Another strategy is formula simplification. Several methods of simplification [13, 14, 16, 39] have been developed for the QE methods of CAD and virtual term substitution, which both speed up the algorithms and reduce the size of the output quantifier-free formulas.

Chapter 3

Solution Formulas for Cubic Equations Without or With Constraints

Cubic equations were studied and discussed since the 5th century B.C., a great number of interesting and important results were found in many research works, especially for the cubic polynomial equations with real coefficients which have a lot of applications in many different areas. In this chapter, we present a convention (for square/cubic root) which provides correct interpretations of Lagrange's formula for all cubic polynomial equations with real coefficients. Using this convention, we also present a real solution formula for the general cubic equation with real coefficients under equality and inequality constraints. Constraints naturally arise in applications such as geometric constraint solving. Most of the material in this chapter is taken from the joint paper [154] with Hong and Wang.

3.1 Introduction

Note that the Lagrange's formula of cubic equations in Theorem 2.1.5, as usually stated, is a bit ambiguous since there are two possible values of s, three possible values of c_1, and three possible values of c_2, depending on which square/cubic roots one takes. Hence there are all together $2 \times 3 \times 3 = 18$ possible interpretations of the formula. It is well known that some interpretations are correct (yielding the solutions), but the others are not.

How to choose a correct interpretation? The usual answer, in the literature, is to choose an interpretation satisfying the condition

$$c_1 c_2 = a_2^2 - 3a_1.$$

Note that the above condition depends on the polynomial f. So we question whether there is a *uniform* condition, i.e., a condition that is *independent* of the polynomial f. The question essentially amounts to whether there is a *convention* for choosing square root and cubic root that will yield correct interpretations for *all* f. We ask the question because it seems to be natural and interesting on its own. We are also motivated by the need of such a convention in geometric constraint solving [63, 131], where it is very desirable to have a uniform way (independent of f) to choose a correct interpretation.

Naturally, we think about whether the standard convention in Section 2.1.1 is correct or not. In the following section, we prove the incorrectness of the standard convention. In Section 3.3, we present the non-standard convention (which we will call "real" convention) that yields correct solutions for all cubic polynomial equations with real coefficients. In Section 3.4, correctness of the real convention is proved. In Section 3.5, using the real convention, we present real solution formulas for the general real-coefficient cubic equation under equality and inequality constraints, and prove its correctness in Section 3.6.

3.2 Incorrectness of the Standard Convention

Proposition 3.2.1. The standard convention is incorrect.

Proof. We only need to find a counter example where the standard convention leads to wrong roots. Consider

$$f = x^3 - 2\,x^2 + x.$$

Direct calculations, following the standard convention, yield

$$\begin{aligned}
p_1 &= a_2^2 a_1^2 + 18\,a_2 a_1 a_0 - 4\,a_1^3 - 27\,a_0^2 - 4\,a_2^3 a_0 = 0,\\
p_2 &= 9\,a_2 a_1 - 27\,a_0 - 2\,a_2^3 = -2,\\
s &= \sqrt[2]{-3\,p_1} = \sqrt[2]{0} = 0,\\
c_1 &= \sqrt[3]{(p_2 + 3\,s)/2} = \sqrt[3]{-1} = \sqrt[3]{\mathrm{e}^{i\pi}} = \mathrm{e}^{i\frac{\pi}{3}},\\
c_2 &= \sqrt[3]{(p_2 - 3\,s)/2} = \sqrt[3]{-1} = \sqrt[3]{\mathrm{e}^{i\pi}} = \mathrm{e}^{i\frac{\pi}{3}},\\
x_1 &= (-a_2 + \omega^1 c_1 + \omega^2 c_2)/3 = \tfrac{1}{2} - \tfrac{\sqrt{3}}{6} i,\\
x_2 &= (-a_2 + \omega^0 c_1 + \omega^0 c_2)/3 = 1 + \tfrac{\sqrt{3}}{3} i,\\
x_3 &= (-a_2 + \omega^2 c_1 + \omega^1 c_2)/3 = \tfrac{1}{2} - \tfrac{\sqrt{3}}{6} i.
\end{aligned}$$

Note that $f = x^3 - 2x^2 + x = (x-1)^2 x$. Therefore the three solutions of $f = 0$ are 1, 1, 0. They are different from x_1, x_2, x_3. Hence the standard convention is incorrect. □

Of course there are infinitely many other (non-standard) conventions. However, we do not yet know if there exists a non-standard but correct convention. Nevertheless, in most applications the polynomials have only real coefficients. So we ask instead whether there is a convention that always yields correct solutions if we restrict the coefficients of the polynomials to *real* numbers. The answer is *Yes.* [1]

3.3 Real Convention

We discovered a correct convention for all cubic equations with real coefficients. The new convention is described in the following definition, under the name of *real convention.*

Definition 3.3.1 (Real Convention)**.** The *real convention* (Figure 3.1) chooses the square root $\sqrt[2]{x}$ and cubic root $\sqrt[3]{x}$ of x so that

$$\arg \sqrt[2]{x} = \frac{1}{2} \arg x,$$

$$\arg \sqrt[3]{x} = -\begin{cases} \frac{1}{3}\arg x - \frac{2}{3}\pi & \text{if } \; -\pi < \arg x < -\frac{\pi}{2}, \\ +\frac{\pi}{2} & \text{if } \; -\frac{\pi}{2} = \arg x, \\ \frac{1}{3}\arg x & \text{if } \; -\frac{\pi}{2} < \arg x < +\frac{\pi}{2}, \\ -\frac{\pi}{2} & \text{if } \; +\frac{\pi}{2} = \arg x, \\ \frac{1}{3}\arg x + \frac{2}{3}\pi & \text{if } \; +\frac{\pi}{2} < \arg x \leq +\pi. \end{cases}$$

[1] One might wonder whether there is any relationship between our question and R. Bombelli's [11, 103], since both address the issue of "complex/real numbers" in the context of solving cubic equations. They are completely different questions. Bombelli asked how to deal with the cases where intermediate results involve square root of negative numbers. He developed a theory of complex numbers by analogy with known rules for real numbers and demonstrated that real roots can be obtained even though some intermediate results are non-real numbers. Our question is to find a "uniform convention" (for square/cubic roots) that does *not* depend on the coefficients of the polynomials and that provides correct interpretations of Lagrange's formula for all cubic polynomial equations with real coefficients.

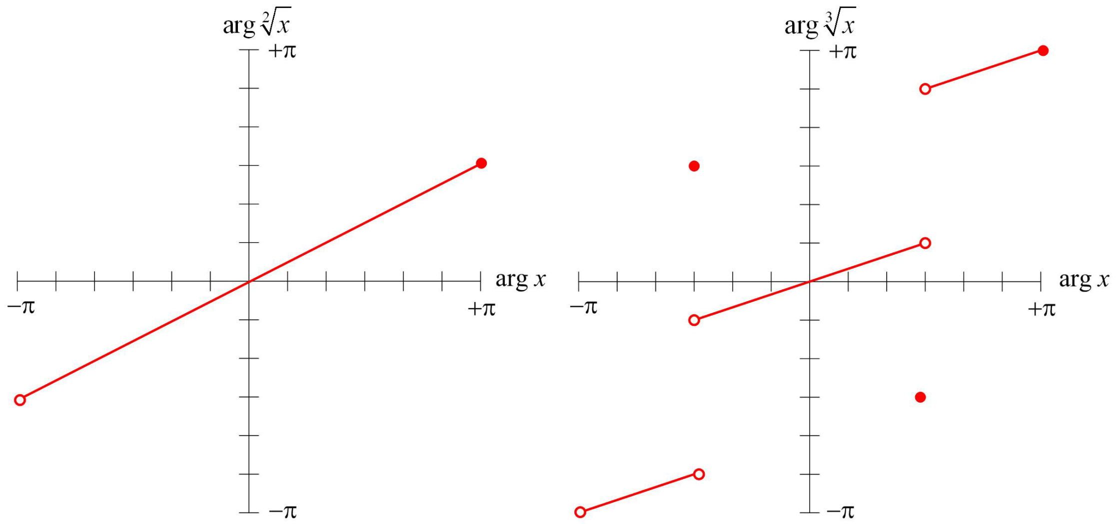

Figure 3.1: Real Convention for the Square and Cubic Roots

Remark 3.3.2. The real convention for the square root is the same as the standard one, but for the cubic root it is quite different from the standard one.

Theorem 3.3.3. *The Lagrange formula under the real convention yields the correct solutions for all cubic polynomial equations with real coefficients, and the solution x_2 is always real.*

Proof. Will be given in the next section. □

Example 3.3.4. We use the example

$$f = x^3 - 2\,x^2 + x = (x-1)^2 x$$

from Section 3.2 to verify the correctness of the real convention. Direct calculations, following the real convention, yield

$$\begin{aligned}
p_1 &= a_2^2 a_1^2 + 18\,a_2 a_1 a_0 - 4\,a_1^3 - 27\,a_0^2 - 4\,a_2^3 a_0 = 0,\\
p_2 &= 9\,a_2 a_1 - 27\,a_0 - 2\,a_2^3 = -2,\\
s &= \sqrt[2]{-3\,p_1} = \sqrt[2]{0} = 0,\\
c_1 &= \sqrt[3]{(p_2 + 3\,s)/2} = \sqrt[3]{-1} = \sqrt[3]{e^{i\pi}} = e^{i\pi} = -1,
\end{aligned}$$

$$\begin{aligned}
c_2 &= \sqrt[3]{(p_2 - 3\,s)/2} = \sqrt[3]{-1} = \sqrt[3]{\mathrm{e}^{i\pi}} = \mathrm{e}^{i\pi} = -1, \\
x_1 &= (-a_2 + \omega^1 c_1 + \omega^2 c_2)/3 = 1, \\
x_2 &= (-a_2 + \omega^0 c_1 + \omega^0 c_2)/3 = 0, \\
x_3 &= (-a_2 + \omega^2 c_1 + \omega^1 c_2)/3 = 1.
\end{aligned}$$

Clearly, x_1, x_2, x_3 are the three solutions of $f = 0$.

Example 3.3.5. Consider another polynomial

$$f = x^3 + x = x(x + i)(x - i).$$

Direct calculations, following the real convention, yield

$$\begin{aligned}
p_1 &= -4,\ p_2 = 0,\ s = 2\sqrt[2]{3}, \\
c_1 &= \sqrt[2]{3},\ c_2 = -\sqrt[2]{3}, \\
x_1 &= (-a_2 + \omega^1 c_1 + \omega^2 c_2)/3 = i, \\
x_2 &= (-a_2 + \omega^0 c_1 + \omega^0 c_2)/3 = 0, \\
x_3 &= (-a_2 + \omega^2 c_1 + \omega^1 c_2)/3 = -i.
\end{aligned}$$

Clearly, x_1, x_2, x_3 are the three solutions of $f = 0$ and x_2 is real.

3.4 Proof of the Correctness of the Real Convention

In this section, we prove Theorem 3.3.3 stated in the previous section. Let f be an arbitrary (monic) cubic polynomial. Let r_1, r_2, r_3 be the three (complex) solutions of $f = 0$. Using the well-known symmetric relations

$$\begin{aligned}
a_2 &= -r_1 - r_2 - r_3, \\
a_1 &= r_1 r_2 + r_1 r_3 + r_2 r_3, \\
a_0 &= -r_1 r_2 r_3,
\end{aligned}$$

we can rewrite p_1 and p_2 as

$$\begin{aligned}
p_1 &= (r_1 - r_2)^2 (r_1 - r_3)^2 (r_2 - r_3)^2, \\
p_2 &= (2r_1 - r_2 - r_3)(2r_2 - r_1 - r_3)(2r_3 - r_1 - r_2).
\end{aligned}$$

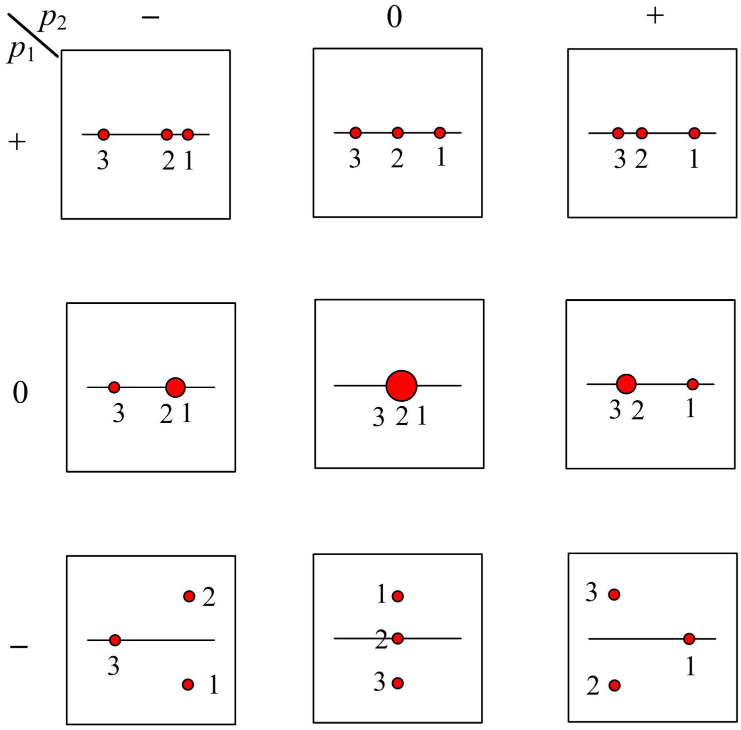

Figure 3.2: Solution Indexing for Cubic Equation

It is easy to verify that the signs of p_1 and p_2 determine the "configuration" of the solutions r_1, r_2 and r_3, as shown in Figure 3.2. Each rectangle denotes a complex plane, in which the horizontal line is the real axis with left-to-right direction. A small disk stands for a simple solution, a bigger disk for a double solution, and the biggest disk for a triple solution. We have also indexed the solutions so that we can refer to them later on. Note that the indexing for the bottom-middle configuration is peculiar (causing solutions jump discontinuously) but it is essential.

Note that p_1 is the discriminant of f. There are three cases according to the sign of p_1:

(1) If p_1 is positive, then $f = 0$ has three real solutions;

(2) If p_1 is zero, then $f = 0$ has one real (triple) solution or two real solutions (of which one is single and the other is double);

(3) If p_1 is negative, then $f = 0$ has one real solution and a pair of conjugated solutions.

The proof proceeds by rewriting, in terms of the solutions, the expressions for s, c_1, c_2 and x_1, x_2, x_3 in Lagrange's formula, taking radicals according to the real convention. It is split into the following several lemmas.

Lemma 3.4.1. $s = i\sqrt{3}\,(r_1 - r_2)(r_1 - r_3)(r_2 - r_3)$.

Proof. Let $q = -3\,p_1$. Then we obviously have

$$q = \left[i\sqrt{3}\,(r_1 - r_2)(r_1 - r_3)(r_2 - r_3)\right]^2.$$

Hence $\sqrt{q}$ is one of the following:

$$q_1 = +i\sqrt{3}\,(r_1 - r_2)(r_1 - r_3)(r_2 - r_3),$$
$$q_2 = -i\sqrt{3}\,(r_1 - r_2)(r_1 - r_3)(r_2 - r_3).$$

We proceed to show that $s = q_1$ in every configuration of the solutions.

(1) $p_1 > 0$. In this case, $f = 0$ has three real solutions indexed as $r_3 < r_2 < r_1$. Note that with this ordering

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(2) $p_1 = 0$. In this case, $f = 0$ has a multiple solution. It follows that $q_1 = q_2 = 0$ and

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(3) $p_1 < 0$ and $p_2 > 0$. In this case, $f = 0$ has a real solution r_1 and a pair of complex conjugates $r_3 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $r_1 > \alpha$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +2\sqrt{3}\,\beta\left[(r_1 - \alpha)^2 + \beta^2\right] > 0,$$
$$q_2 = -2\sqrt{3}\,\beta\left[(r_1 - \alpha)^2 + \beta^2\right] < 0.$$

Then

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(4) $p_1 < 0$ and $p_2 = 0$. In this case, $f = 0$ has a real solution r_2 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 = \alpha$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +2\sqrt{3}\,\beta^3 > 0, \quad q_2 = -2\sqrt{3}\,\beta^3 < 0.$$

Then

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(5) $p_1 < 0$ and $p_2 < 0$. In this case, $f = 0$ has a real solution r_3 and a pair of complex conjugates $r_2 = \alpha + i\beta$ and $r_1 = \alpha - i\beta$ such that $r_3 < \alpha$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +2\sqrt{3}\,\beta\left[(r_3 - \alpha)^2 + \beta^2\right] > 0,$$
$$q_2 = -2\sqrt{3}\,\beta\left[(r_3 - \alpha)^2 + \beta^2\right] < 0.$$

Then

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$. □

Lemma 3.4.2. At least one of the followings is true.

$$\begin{array}{lllllll} c_1 & = & \omega^0 r_1 + \omega^1 r_2 + \omega^2 r_3 & \wedge & c_2 & = & \omega^0 r_1 + \omega^2 r_2 + \omega^1 r_3 \\ c_1 & = & \omega^2 r_1 + \omega^0 r_2 + \omega^1 r_3 & \wedge & c_2 & = & \omega^1 r_1 + \omega^0 r_2 + \omega^2 r_3 \\ c_1 & = & \omega^1 r_1 + \omega^2 r_2 + \omega^0 r_3 & \wedge & c_2 & = & \omega^2 r_1 + \omega^1 r_2 + \omega^0 r_3 \end{array}$$

Proof. Let $q = (p_2 + 3\,s)/2$ and $q' = (p_2 - 3\,s)/2$. Recalling Lemma 3.4.1, substitution and factorization yield

$$q = (\omega^0 r_1 + \omega^1 r_2 + \omega^2 r_3)^3, \quad q' = (\omega^0 r_1 + \omega^2 r_2 + \omega^1 r_3)^3.$$

Hence $\sqrt[3]{q}$ is one of the following:

$$\begin{aligned} q_1 &= \omega^0 r_1 + \omega^1 r_2 + \omega^2 r_3, \\ q_2 &= \omega^2 r_1 + \omega^0 r_2 + \omega^1 r_3, \\ q_3 &= \omega^1 r_1 + \omega^2 r_2 + \omega^0 r_3. \end{aligned}$$

Likewise $\sqrt[3]{q'}$ is one of the following:

$$\begin{aligned} q'_1 &= \omega^0 r_1 + \omega^2 r_2 + \omega^1 r_3, \\ q'_2 &= \omega^1 r_1 + \omega^0 r_2 + \omega^2 r_3, \\ q'_3 &= \omega^2 r_1 + \omega^1 r_2 + \omega^0 r_3. \end{aligned}$$

We can rewrite q_1, q_2, q_3 and q'_1, q'_2, q'_3 as

$$\begin{aligned} q_1 &= \omega^0(r_1 - r_2) - \omega^2(r_2 - r_3) = \mathrm{e}^{i\frac{\pm 0}{6}\pi}(r_1 - r_2) + \mathrm{e}^{i\frac{\pm 2}{6}\pi}(r_2 - r_3), \\ q_2 &= \omega^2(r_1 - r_2) - \omega^1(r_2 - r_3) = \mathrm{e}^{i\frac{-4}{6}\pi}(r_1 - r_2) + \mathrm{e}^{i\frac{-2}{6}\pi}(r_2 - r_3), \\ q_3 &= \omega^1(r_1 - r_2) - \omega^0(r_2 - r_3) = \mathrm{e}^{i\frac{\pm 4}{6}\pi}(r_1 - r_2) + \mathrm{e}^{i\frac{\pm 6}{6}\pi}(r_2 - r_3); \\ q'_1 &= \omega^0(r_1 - r_2) - \omega^1(r_2 - r_3) = \mathrm{e}^{i\frac{\pm 0}{6}\pi}(r_1 - r_2) + \mathrm{e}^{i\frac{-2}{6}\pi}(r_2 - r_3), \\ q'_2 &= \omega^1(r_1 - r_2) - \omega^2(r_2 - r_3) = \mathrm{e}^{i\frac{\pm 4}{6}\pi}(r_1 - r_2) + \mathrm{e}^{i\frac{\pm 2}{6}\pi}(r_2 - r_3), \\ q'_3 &= \omega^2(r_1 - r_2) - \omega^0(r_2 - r_3) = \mathrm{e}^{i\frac{-4}{6}\pi}(r_1 - r_2) + \mathrm{e}^{i\frac{\pm 6}{6}\pi}(r_2 - r_3). \end{aligned}$$

Now we prove the lemma for every configuration of the solutions.

(1) $p_1 > 0$ and $p_2 > 0$. In this case, $f = 0$ has three real solutions $r_3 < r_2 < r_1$ and

$r_2 - r_3 < r_1 - r_2$. Thus

$$\begin{aligned}
\frac{0}{6}\pi < \arg q_1 < \frac{+\frac{0}{6}\pi + \frac{2}{6}\pi}{2} &= +\frac{1}{6}\pi, & -\frac{1}{6}\pi = \frac{-\frac{2}{6}\pi + \frac{0}{6}\pi}{2} < \arg q_1' < +\frac{0}{6}\pi, \\
-\frac{4}{6}\pi < \arg q_2 < \frac{-\frac{4}{6}\pi - \frac{2}{6}\pi}{2} &= -\frac{3}{6}\pi, & +\frac{3}{6}\pi = \frac{+\frac{4}{6}\pi + \frac{2}{6}\pi}{2} < \arg q_2' < +\frac{4}{6}\pi, \\
+\frac{4}{6}\pi < \arg q_3 < \frac{+\frac{4}{6}\pi + \frac{6}{6}\pi}{2} &= +\frac{5}{6}\pi; & -\frac{5}{6}\pi = \frac{-\frac{4}{6}\pi - \frac{6}{6}\pi}{2} < \arg q_3' < -\frac{4}{6}\pi.
\end{aligned}$$

Since $\operatorname{Re} q = \operatorname{Re} q' = p_2/2$ (where $\operatorname{Re} q$ denotes the real part of q), we have $|\arg q| < \pi/2$, $|\arg q'| < \pi/2$. Therefore $0 \le |\arg \sqrt[3]{q}| < \pi/6$, $0 \le |\arg \sqrt[3]{q'}| < \pi/6$. Hence $\sqrt[3]{q} = q_1$, $\sqrt[3]{q'} = q_1'$. So we have $c_1 = q_1,\ c_2 = q_1'$.

(2) $p_1 > 0$ and $p_2 = 0$. In this case, $f = 0$ has three real solutions $r_3 < r_2 < r_1$ and $r_2 - r_3 = r_1 - r_2$. Thus

$$\begin{aligned}
\arg q_1 = \frac{+\frac{0}{6}\pi + \frac{2}{6}\pi}{2} &= +\frac{1}{6}\pi, & \arg q_1' = \frac{+\frac{0}{6}\pi - \frac{2}{6}\pi}{2} &= -\frac{1}{6}\pi, \\
\arg q_2 = \frac{-\frac{4}{6}\pi - \frac{2}{6}\pi}{2} &= -\frac{3}{6}\pi, & \arg q_2' = \frac{+\frac{2}{6}\pi + \frac{4}{6}\pi}{2} &= +\frac{3}{6}\pi, \\
\arg q_3 = \frac{+\frac{4}{6}\pi + \frac{6}{6}\pi}{2} &= +\frac{5}{6}\pi; & \arg q_3' = \frac{-\frac{4}{6}\pi - \frac{6}{6}\pi}{2} &= -\frac{5}{6}\pi.
\end{aligned}$$

Since $\operatorname{Re} q = \operatorname{Re} q' = p_2/2 = 0$, we have $|\arg q| = \pi/2$, $|\arg q'| = \pi/2$. Therefore $|\arg \sqrt[3]{q}| = \pi/2$, $|\arg \sqrt[3]{q'}| = \pi/2$. Hence $\sqrt[3]{q} = q_2$, $\sqrt[3]{q'} = q_2'$. So we have $c_1 = q_2,\ c_2 = q_2'$.

(3) $p_1 > 0$ and $p_2 < 0$. In this case, $f = 0$ has three real solutions $r_3 < r_2 < r_1$ and $r_1 - r_2 < r_2 - r_3$. Thus

$$\begin{aligned}
+\frac{1}{6}\pi = \frac{+\frac{0}{6}\pi + \frac{2}{6}\pi}{2} < \arg q_1 &< +\frac{2}{6}\pi, & -\frac{2}{6}\pi < \arg q_1' < \frac{+\frac{0}{6}\pi - \frac{2}{6}\pi}{2} = -\frac{1}{6}\pi, \\
-\frac{3}{6}\pi = \frac{-\frac{4}{6}\pi - \frac{2}{6}\pi}{2} < \arg q_2 &< -\frac{2}{6}\pi, & +\frac{2}{6}\pi < \arg q_2' < \frac{+\frac{2}{6}\pi + \frac{4}{6}\pi}{2} = +\frac{3}{6}\pi, \\
+\frac{5}{6}\pi = \frac{+\frac{4}{6}\pi + \frac{6}{6}\pi}{2} < \arg q_3 &< +\frac{6}{6}\pi; & -\frac{6}{6}\pi < \arg q_3' < \frac{-\frac{4}{6}\pi - \frac{6}{6}\pi}{2} = -\frac{5}{6}\pi.
\end{aligned}$$

Since $\operatorname{Re} q = \operatorname{Re} q' = p_2/2 < 0$, we have $|\arg q| > \pi/2$, $|\arg q'| > \pi/2$. Therefore $5\,\pi/6 < |\arg \sqrt[3]{q}| < \pi$, $5\,\pi/6 < |\arg \sqrt[3]{q'}| < \pi$. Hence $\sqrt[3]{q} = q_3$, $\sqrt[3]{q'} = q_3'$. So we have $c_1 = q_3,\ c_2 = q_3'$.

(4) $p_1 = 0$ and $p_2 > 0$. In this case, $f = 0$ has a simple real solution r_1 and a double real solution $r_2 = r_3$ such that $r_3 = r_2 < r_1$. Thus

$$\arg q_1 = +\frac{0}{6}\pi, \quad \arg q_2 = -\frac{4}{6}\pi, \quad \arg q_3 = +\frac{4}{6}\pi;$$
$$\arg q_1' = +\frac{0}{6}\pi, \quad \arg q_2' = +\frac{4}{6}\pi, \quad \arg q_3' = -\frac{4}{6}\pi.$$

Since $q = q' = p_2/2 > 0$, we have $\arg q = 0$, $\arg q' = 0$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = 0$. Hence $\sqrt[3]{q} = q_1$, $\sqrt[3]{q'} = q_1'$. So we have $c_1 = q_1,\ c_2 = q_1'$.

(5) $p_1 = 0$ and $p_2 = 0$. In this case, $f = 0$ has a triple real solution $r_3 = r_2 = r_1$. It follows that $q_1 = q_2 = q_3 = 0$, $q_1' = q_2' = q_3' = 0$. Since $q = q' = p_2/2 = 0$, we have $\arg q = 0$, $\arg q' = 0$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = 0$. Hence we can choose $\sqrt[3]{q} = q_2$, $\sqrt[3]{q'} = q_2'$. So we have $c_1 = q_2,\ c_2 = q_2'$.

(6) $p_1 = 0$ and $p_2 < 0$. In this case, $f = 0$ has a simple real solution r_3 and a double real solution $r_1 = r_2$ such that $r_3 < r_2 = r_1$. Thus

$$\arg q_1 = +\frac{2}{6}\pi, \quad \arg q_2 = -\frac{2}{6}\pi, \quad \arg q_3 = +\frac{6}{6}\pi;$$
$$\arg q_1' = -\frac{2}{6}\pi, \quad \arg q_2' = +\frac{2}{6}\pi, \quad \arg q_3' = +\frac{6}{6}\pi.$$

Since $q = q' = p_2/2 < 0$, we have $\arg q = \pi$, $\arg q' = \pi$. Therefore $\arg \sqrt[3]{q} = \pi$, $\arg \sqrt[3]{q'} = \pi$. Hence $\sqrt[3]{q} = q_3$, $\sqrt[3]{q'} = q_3'$. So we have $c_1 = q_3,\ c_2 = q_3'$.

(7) $p_1 < 0$ and $p_2 > 0$. In this case, $f = 0$ has a real solution r_1 and a pair of complex conjugates $r_3 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $r_1 > \alpha$ and $\beta > 0$. Simple calculation gives

$$q_1 = \omega^0(r_1 - \alpha + \sqrt{3}\,\beta), \quad q_1' = \omega^0(r_1 - \alpha - \sqrt{3}\,\beta),$$
$$q_2 = \omega^2(r_1 - \alpha + \sqrt{3}\,\beta), \quad q_2' = \omega^2(r_1 - \alpha - \sqrt{3}\,\beta),$$
$$q_3 = \omega^1(r_1 - \alpha + \sqrt{3}\,\beta); \quad q_3' = \omega^1(r_1 - \alpha - \sqrt{3}\,\beta).$$

Note that

$$q = (r_1 - \alpha + \sqrt{3}\,\beta)^3 > 0, \quad q' = (r_1 - \alpha - \sqrt{3}\,\beta)^3.$$

We consider the three subcases.

(a) $r_1 - \alpha - \sqrt{3}\,\beta > 0$. In this case,

$$\arg q_1 = +\frac{0}{3}\pi, \quad \arg q_2 = -\frac{2}{3}\pi, \quad \arg q_3 = +\frac{2}{3}\pi;$$
$$\arg q_1' = +\frac{0}{3}\pi, \quad \arg q_2' = +\frac{2}{3}\pi, \quad \arg q_3' = -\frac{2}{3}\pi.$$

Since $q > 0$, $q' > 0$, we have $\arg q = 0$, $\arg q' = 0$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = 0$. Hence $\sqrt[3]{q} = q_1$, $\sqrt[3]{q'} = q_1'$.

(b) $r_1 - \alpha - \sqrt{3}\,\beta = 0$. In this case, $q_1 = q_2 = q_3 = 0$, $q_1' = q_2' = q_3' = 0$ and thus

$$\arg q_1 = 0, \quad \arg q_2 = 0, \quad \arg q_3 = 0;$$
$$\arg q_1' = 0, \quad \arg q_2' = 0, \quad \arg q_3' = 0.$$

Since $q = q' = 0$, we have $\arg q = 0$, $\arg q' = 0$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = 0$. Hence we can choose $\sqrt[3]{q} = q_1$, $\sqrt[3]{q'} = q_1'$.

(c) $r_1 - \alpha - \sqrt{3}\,\beta < 0$. In this case,

$$\arg q_1 = +\frac{0}{3}\pi, \quad \arg q_2 = -\frac{2}{3}\pi, \quad \arg q_3 = +\frac{2}{3}\pi;$$
$$\arg q_1' = +\frac{3}{3}\pi, \quad \arg q_2' = -\frac{1}{3}\pi, \quad \arg q_3' = +\frac{1}{3}\pi.$$

Since $q > 0$, $q' < 0$, we have $\arg q = 0$, $\arg q' = \pi$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = \pi$. Hence $\sqrt[3]{q} = q_1$, $\sqrt[3]{q'} = q_1'$.

So we have $c_1 = q_1$, $c_2 = q_1'$.

(8) $p_1 < 0$ and $p_2 = 0$. In this case, $f = 0$ has a real solution r_2 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 = \alpha$ and $\beta > 0$. Simple calculation gives

$$q_1 = \omega^1\sqrt{3}\,\beta, \qquad q_2 = \omega^0\sqrt{3}\,\beta, \qquad q_3 = \omega^2\sqrt{3}\,\beta;$$
$$q_1' = -\omega^2\sqrt{3}\,\beta, \quad q_2' = -\omega^0\sqrt{3}\,\beta, \quad q_3' = -\omega^1\sqrt{3}\,\beta.$$

Thus

$$\arg q_1 = +\frac{2}{3}\pi, \quad \arg q_2 = +\frac{0}{3}\pi, \quad \arg q_3 = -\frac{2}{3}\pi;$$
$$\arg q_1' = +\frac{1}{3}\pi, \quad \arg q_2' = +\frac{3}{3}\pi, \quad \arg q_3' = -\frac{1}{3}\pi.$$

Since $q = 3s/2 > 0$, $q' = -3s/2 < 0$, we have $\arg q = 0$, $\arg q' = \pi$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = \pi$. Hence $\sqrt[3]{q} = q_2$, $\sqrt[3]{q'} = q_2'$. So we have $c_1 = q_2$, $c_2 = q_2'$.

(9) $p_1 < 0$ and $p_2 < 0$. In this case, $f = 0$ has a real solution r_3 and a pair of complex conjugates $r_2 = \alpha + i\beta$ and $r_1 = \alpha - i\beta$ such that $r_3 < \alpha$ and $\beta > 0$. Simple calculation gives

$$\begin{aligned} q_1 &= \omega^2(r_3 - \alpha + \sqrt{3}\,\beta), \quad & q_1' &= \omega^2(r_3 - \alpha - \sqrt{3}\,\beta), \\ q_2 &= \omega^1(r_3 - \alpha + \sqrt{3}\,\beta), \quad & q_2' &= \omega^1(r_3 - \alpha - \sqrt{3}\,\beta), \\ q_3 &= \omega^0(r_3 - \alpha + \sqrt{3}\,\beta); \quad & q_3' &= \omega^0(r_3 - \alpha - \sqrt{3}\,\beta). \end{aligned}$$

Note that

$$q = (r_3 - \alpha + \sqrt{3}\,\beta)^3, \quad q' = (r_3 - \alpha - \sqrt{3}\,\beta)^3 < 0.$$

We consider the three subcases.

(a) $r_3 - \alpha + \sqrt{3}\,\beta > 0$. In this case,

$$\begin{aligned} \arg q_1 &= -\frac{2}{3}\pi, \quad & \arg q_2 &= +\frac{2}{3}\pi, \quad & \arg q_3 &= +\frac{0}{3}\pi; \\ \arg q_1' &= -\frac{1}{3}\pi, \quad & \arg q_2' &= +\frac{1}{3}\pi, \quad & \arg q_3' &= +\frac{3}{3}\pi. \end{aligned}$$

Since $q > 0$, $q' < 0$, we have $\arg q = 0$, $\arg q' = \pi$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = \pi$. Hence $\sqrt[3]{q} = q_3$, $\sqrt[3]{q'} = q_3'$.

(b) $r_3 - \alpha + \sqrt{3}\,\beta = 0$. In this case, $q_1 = q_2 = q_3 = 0$, $q_1' = q_2' = q_3' = 0$ and thus

$$\begin{aligned} \arg q_1 &= 0, \quad & \arg q_2 &= 0, \quad & \arg q_3 &= 0; \\ \arg q_1' &= -\frac{1}{3}\pi, \quad & \arg q_2' &= +\frac{1}{3}\pi, \quad & \arg q_3' &= +\frac{3}{3}\pi. \end{aligned}$$

Since $q = 0$, $q' < 0$, we have $\arg q = 0$, $\arg q' = \pi$. Therefore $\arg \sqrt[3]{q} = 0$, $\arg \sqrt[3]{q'} = \pi$. Hence we can choose $\sqrt[3]{q} = q_3$, $\sqrt[3]{q'} = q_3'$.

(c) $r_3 - \alpha + \sqrt{3}\,\beta < 0$. In this case,

$$\begin{aligned} \arg q_1 &= +\frac{1}{3}\pi, \quad & \arg q_2 &= -\frac{1}{3}\pi, \quad & \arg q_3 &= +\frac{3}{3}\pi; \\ \arg q_1' &= -\frac{1}{3}\pi, \quad & \arg q_2' &= +\frac{1}{3}\pi, \quad & \arg q_3' &= +\frac{3}{3}\pi. \end{aligned}$$

Since $q < 0$, $q' < 0$, we have $\arg q = \pi$, $\arg q' = \pi$. Therefore $\arg \sqrt[3]{q} = \pi$, $\arg \sqrt[3]{q'} = \pi$. Hence $\sqrt[3]{q} = q_3$, $\sqrt[3]{q'} = q'_3$.

So we have $c_1 = q_3,\ c_2 = q'_3$. □

Lemma 3.4.3. The solution x_2 is always real.

Proof. We use the results and the notations in the proof of Lemma 3.4.2.

(1) $p_1 > 0$ and $p_2 > 0$. In this case, we have $c_1 = q_1,\ c_2 = q'_1$. Substituting c_1 and c_2 into x_2 in Lagrange's formula and simplifying the resulting expressions using $\omega^3 = 1$ and $\omega^0 + \omega^1 + \omega^2 = 0$, we see that

$$x_2 = \frac{3\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_2 + (\omega^0 + \omega^1 + \omega^2)\,r_3}{3} = r_1.$$

(2) $p_1 > 0$ and $p_2 = 0$. In this case $c_1 = q_2,\ c_2 = q'_2$. Similar calculation yields $x_2 = r_2$.

(3) $p_1 > 0$ and $p_2 < 0$. In this case $c_1 = q_3,\ c_2 = q_3$. Similar calculation yields $x_2 = r_3$.

(4) $p_1 = 0$ and $p_2 > 0$. In this case $c_1 = q_1,\ c_2 = q'_1$. Similar calculation yields $x_2 = r_1$.

(5) $p_1 = 0$ and $p_2 = 0$. In this case $c_1 = q_2,\ c_2 = q'_2$. Similar calculation yields $x_2 = r_2$.

(6) $p_1 = 0$ and $p_2 < 0$. In this case $c_1 = q_3,\ c_2 = q'_3$. Similar calculation yields $x_2 = r_3$.

(7) $p_1 < 0$ and $p_2 > 0$. In this case $c_1 = q_1,\ c_2 = q'_1$. Similar calculation yields $x_2 = r_1$.

(8) $p_1 < 0$ and $p_2 = 0$. In this case $c_1 = q_2,\ c_2 = q'_2$. Similar calculation yields $x_2 = r_2$.

(9) $p_1 < 0$ and $p_2 < 0$. In this case $c_1 = q_3,\ c_2 = q_3'$. Similar calculation yields $x_2 = r_3$.

It is clear that $x_2 = r_1$ when $p_2 > 0$; $x_2 = r_2$ when $p_2 = 0$; $x_2 = r_3$ when $p_2 < 0$. According to the configurations in Figure 3.2, we see immediately that x_2 is always real. □

Proof of Theorem 3.3.3. Recalling Lemma 3.4.2, we consider the following three cases.

(1) $c_1 = \omega^0 r_1 + \omega^1 r_2 + \omega^2 r_3 \ \wedge\ c_2 = \omega^0 r_1 + \omega^2 r_2 + \omega^1 r_3'$.

Substituting c_1 and c_2 into x_k and simplifying the resulting expressions using $\omega^3 = 1$ and $\omega^0 + \omega^1 + \omega^2 = 0$, we see that

$$\begin{aligned}
x_1 &= \frac{3\,r_3 + (\omega^0 + \omega^1 + \omega^2)\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_2}{3} = r_3,\\
x_2 &= \frac{3\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_2 + (\omega^0 + \omega^1 + \omega^2)\,r_3}{3} = r_1,\\
x_3 &= \frac{3\,r_2 + (\omega^0 + \omega^1 + \omega^2)\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_2}{3} = r_2.
\end{aligned}$$

(2) $c_1 = \omega^2 r_1 + \omega^0 r_2 + \omega^1 r_3 \ \wedge\ c_2 = \omega^1 r_1 + \omega^0 r_2 + \omega^2 r_3'$.

Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(3) $c_1 = \omega^1 r_1 + \omega^2 r_2 + \omega^0 r_3 \ \wedge\ c_2 = \omega^2 r_1 + \omega^1 r_2 + \omega^0 r_3'$.

Similar calculation yields $x_1 = r_2,\ x_2 = r_3,\ x_3 = r_1$.

From Lemma 3.4.3, x_2 is always real. □

3.5 Formulas for Cubic Equations with Constraints

In Section 3.3, we have introduced a correct convention for choosing the square and cubic roots. Using this convention and Lagrange's formula, we present real solution formulas for the general real-coefficient cubic equation under equality and inequality constraints. Constraints naturally arise in applications such as geometric constraint solving [63, 131]. The representations of the real solutions coupled with real constraints are achieved by combining Thom's lemma [7, p. 50] and the complex-solution formulas.

Let $\wedge$, $\vee$, $\Rightarrow$, and $\neg$ stand for the logical connectives "and", "or", "imply", and "not" respectively. Denote by $\mathbb{R}$ the field of real numbers and $\mathbb{R}[x]$ the ring of polynomials in x with real coefficients. We have the following result.

Theorem 3.5.1. *Let $f(x) = x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{R}[x]$ and $\Gamma(x)$ be a formula composed by $\wedge$, $\vee$, $\Rightarrow$, and $\neg$ of polynomial equality and inequality relations in x, the coefficients of $f(x)$, and other parameters. Then for all $x \in \mathbb{R}$,*

$$[f(x) = 0 \wedge \Gamma(x)] \iff [x = x_1 \wedge \Gamma_1] \vee [x = x_2 \wedge \Gamma_2] \vee [x = x_3 \wedge \Gamma_3],$$

where

$$\begin{aligned} x_1 &= (-a_2 + \omega^{(1-\sigma)}c_1 + \omega^{(2+\sigma)}c_2)/3, \\ x_2 &= (-a_2 + \omega^{(0-\sigma)}c_1 + \omega^{(0+\sigma)}c_2)/3, \\ x_3 &= (-a_2 + \omega^{(2-\sigma)}c_1 + \omega^{(1+\sigma)}c_2)/3, \\ \sigma &= \operatorname{sign}(p_2), \end{aligned}$$

and

$$\Gamma_j := (\exists\, x \in \mathbb{R})\,[f(x) = 0 \wedge \Gamma(x) \wedge \Phi_j(x)], \quad j = 1, 2, 3,$$

$$\begin{aligned} \Phi_1(x) &:= [f'(x) > 0 \wedge f''(x) > 0] \;\vee\; [f'(x) = 0 \wedge f''(x) \geq 0], \\ \Phi_2(x) &:= [f'(x) \leq 0] \;\vee\; [f''(x) = 0], \\ \Phi_3(x) &:= [f'(x) > 0 \wedge f''(x) < 0] \;\vee\; [f'(x) = 0 \wedge f''(x) \leq 0]. \end{aligned}$$

Here c_1, c_2, p_2, ω are the same as in Lagrange's formula given in Theorem 2.1.5 of Section 2.1.3.

Proof. Will be given in the next section. □

Remark 3.5.2. Note that the above formula is slightly different from the Lagrange formula (in Theorem 2.1.5), in that the exponents for ω are adjusted depending on the sign of p_2. This adjustment is essential for the correctness of the theorem.

Remark 3.5.3. It turns out (and will be shown in the proof of the theorem) that the three complex solutions of f satisfy

$$\operatorname{Re} x_3 \leq \operatorname{Re} x_2 \leq \operatorname{Re} x_1.$$

Remark 3.5.4. The real constraints in the formulas are given as three existentially quantified subformulas Γ_j. If needed, one could eliminate the existential quantifier using, e.g., the method based on partial cylindrical algebraic decomposition [32]. However, if $\Gamma(x)$ is restricted to a combination of polynomial equalities and inequalities of degree ≤ 3 in x, one could use the alternative approach of [136] that provides explicit symbolic real solutions of cubic equations. Such solutions can be efficiently substituted in real side conditions at practically low price of the linear and quadratic real quantifier elimination [134, 137] in REDLOG [39].

Example 3.5.5. We illustrate Theorem 3.5.1 using a simple example. Let

$$\begin{aligned} f(x) &:= x^3 - ax + 1 \\ \Gamma(x) &:= -1/2 \leq x \leq 1/2 \end{aligned}$$

where a is a parameter. Direct calculations, using the formula in Theorem 3.5.1, yield

$$\begin{aligned} p_1 &= a_2^2 a_1^2 + 18\, a_2 a_1 a_0 - 4\, a_1^3 - 27\, a_0^2 - 4\, a_2^3 a_0 = 4\, a^3 - 27, \\ p_2 &= 9\, a_2 a_1 - 27\, a_0 - 2\, a_2^3 = -27, \\ s &= \sqrt[2]{-3\, p_1} = \sqrt[2]{81 - 12\, a^3}, \\ c_1 &= \sqrt[3]{(p_2 + 3\, s)/2} = \sqrt[3]{(-27 + 3\sqrt[2]{81 - 12\, a^3})/2}, \\ c_2 &= \sqrt[3]{(p_2 - 3\, s)/2} = \sqrt[3]{(-27 - 3\sqrt[2]{81 - 12\, a^3})/2}, \\ \sigma &= \operatorname{sign}(p_2) = -1, \\ x_1 &= (-a_2 + \omega^{(1-\sigma)} c_1 + \omega^{(2+\sigma)} c_2)/3 = (\omega^2 c_1 + \omega^1 c_2)/3, \\ x_2 &= (-a_2 + \omega^{(0-\sigma)} c_1 + \omega^{(0+\sigma)} c_2)/3 = (\omega^1 c_1 + \omega^2 c_2)/3, \\ x_3 &= (-a_2 + \omega^{(2-\sigma)} c_1 + \omega^{(1+\sigma)} c_2)/3 = (\omega^0 c_1 + \omega^0 c_2)/3, \end{aligned}$$

and

$$\Gamma_j := (\exists\, x \in \mathbb{R})\, [x^3 - ax + 1 = 0 \wedge -1/2 \leq x \leq 1/2 \wedge \Phi_j(x)], \quad j = 1, 2, 3,$$

$$\Phi_1(x) := [\, 3\,x^2 - a > 0 \wedge 6\,x > 0\,] \vee [\, 3\,x^2 - a = 0 \wedge 6\,x \geq 0\,],$$

$$\Phi_2(x) := [\, 3\,x^2 - a \leq 0\,] \qquad \vee [\, 6\,x = 0\,],$$

$$\Phi_3(x) := [\, 3\,x^2 - a > 0 \wedge 6\,x < 0\,] \vee [\, 3\,x^2 - a = 0 \wedge 6\,x \leq 0\,].$$

Using the real quantifier elimination procedure QEPCAD [15, 32] to eliminate the existential quantifiers in the above formula, we obtain the following quantifier-free formulas equivalent to Γ_j:

$$\Gamma_1 \iff false,$$

$$\Gamma_2 \iff 4\,a - 9 \geq 0,$$

$$\Gamma_3 \iff 4\,a + 7 \leq 0.$$

Hence we finally obtain

$$[x^3 - ax + 1 = 0 \wedge -1/2 \leq x \leq 1/2] \iff [x = x_2 \wedge 4\,a - 9 \geq 0] \vee [x = x_3 \wedge 4\,a + 7 \leq 0].$$

We can also use the real quantifier elimination function in REDLOG [39] to obtain the following quantifier-free formulas equivalent to Γ_j:

$$\Gamma_1 \iff false,$$

$$\Gamma_2 \iff 4\,a^3 - 27 > 0 \wedge 4\,a - 9 \geq 0,$$

$$\Gamma_3 \iff 4\,a^3 - 27 < 0 \wedge 4\,a + 7 \leq 0.$$

Simplifying the above formulas, we get the same result as using QEPCAD. □

3.6 Proof of the Correctness of the Formulas

In this section, we prove Theorem 3.5.1 stated in the previous section. The proof will be divided into the following two lemmas. The proof of each lemma will be further divided into cases depending on the solution indexing in Figure 3.2.

Lemma 3.6.1. $x_1 = r_1$, $x_2 = r_2$, and $x_3 = r_3$.

Proof. We use the results and the same q_i, q_i' from Lemma 3.4.2.

(1) $p_1 > 0$ and $p_2 > 0$. In this case, we have $c_1 = q_1,\ c_2 = q_1',\ \sigma = +1$. Substituting c_1 and c_2 into u_k in Theorem 3.5.1 and simplifying the resulting expressions using $\omega^3 = 1$ and $\omega^0 + \omega^1 + \omega^2 = 0$, we see that

$$x_1 = \frac{3\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_2}{3} = r_1,$$
$$x_2 = \frac{3\,r_2 + (\omega^0 + \omega^1 + \omega^2)\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_3}{3} = r_2,$$
$$x_3 = \frac{3\,r_3 + (\omega^0 + \omega^1 + \omega^2)\,r_1 + (\omega^0 + \omega^1 + \omega^2)\,r_2}{3} = r_3.$$

(2) $p_1 > 0$ and $p_2 = 0$. In this case $c_1 = q_2,\ c_2 = q_2',\ \sigma = 0$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(3) $p_1 > 0$ and $p_2 < 0$. In this case $c_1 = q_3,\ c_2 = q_3',\ \sigma = -1$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(4) $p_1 = 0$ and $p_2 > 0$. In this case $c_1 = q_1,\ c_2 = q_1',\ \sigma = +1$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(5) $p_1 = 0$ and $p_2 = 0$. In this case $c_1 = q_2,\ c_2 = q_2',\ \sigma = 0$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(6) $p_1 = 0$ and $p_2 < 0$. In this case $c_1 = q_3,\ c_2 = q_3',\ \sigma = -1$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(7) $p_1 < 0$ and $p_2 > 0$. In this case $c_1 = q_1,\ c_2 = q_1',\ \sigma = +1$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(8) $p_1 < 0$ and $p_2 = 0$. In this case $c_1 = q_2,\ c_2 = q_2',\ \sigma = 0$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$.

(9) $p_1 < 0$ and $p_2 < 0$. In this case $c_1 = q_3,\ c_2 = q_3',\ \sigma = -1$. Similar calculation yields $x_1 = r_1,\ x_2 = r_2,\ x_3 = r_3$. □

The indexing of solutions in Figure 3.2 also permits us to establish the following lemma using the idea underlying Thom's lemma [7, p. 50]: each real r_k is uniquely determined by the signs of the derivatives of f at r_k.

Lemma 3.6.2. $\Gamma_j \iff r_j \in \mathbb{R} \wedge \Gamma(r_j)$.

Proof. Note that

$$\Gamma_j := (\exists\, z \in \mathbb{R})\,[f(z) = 0 \wedge \Gamma(z) \wedge \Phi_j(z)] \iff \bigvee_{k=1}^{3} r_k \in \mathbb{R} \wedge \Gamma(r_k) \wedge \Phi_j(r_k).$$

We need to determine $\Phi_j(r_k)$. For this, observe that

$$f'(r_1) = (r_1 - r_2)(r_1 - r_3), \quad f''(r_1) = 2\,(r_1 - r_2) + 2\,(r_1 - r_3),$$
$$f'(r_2) = (r_2 - r_1)(r_2 - r_3), \quad f''(r_2) = 2\,(r_2 - r_1) + 2\,(r_2 - r_3),$$
$$f'(r_3) = (r_1 - r_3)(r_2 - r_3), \quad f''(r_3) = 2\,(r_3 - r_1) + 2\,(r_3 - r_2).$$

For each configuration of the solutions, we can determine the signs of the derivatives of f at r_k, as in Table 3.1 (where the blanks are non-real). From the signs of the derivatives, it is easy to obtain the truth values of Φ_j as in Table 3.2 (where the blanks are false).

Table 3.1: Signs of Derivatives of Cubic Polynomial f

p_1	+	+	+	0	0	0	−	−	−
p_2	+	0	−	+	0	−	+	0	−
$f'(r_1)$	+	+	+	+	0	0	+	0	
$f''(r_1)$	+	+	+	+	0	+	+		
$f'(r_2)$	−	−	−	0	0	0		+	
$f''(r_2)$	−	0	+	−	0	+		0	
$f'(r_3)$	+	+	+	0	0	+		0	+
$f''(r_3)$	−	−	−	−	0	−			−

Table 3.2: Truth Values of Φ_j of Solutions of Cubic Equation

p_1	+	+	+	0	0	0	−	−	−
p_2	+	0	−	+	0	−	+	0	−
$\Phi_1(r_1)$	true	true	true	true	true	true	true		
$\Phi_1(r_2)$					true	true			
$\Phi_1(r_3)$					true				
$\Phi_2(r_1)$					true	true			
$\Phi_2(r_2)$	true	true	true	true	true	true		true	
$\Phi_2(r_3)$				true	true				
$\Phi_3(r_1)$					true				
$\Phi_3(r_2)$				true	true				
$\Phi_3(r_3)$	true	true	true	true	true	true			true

From Table 3.2, we see immediately that

$$\Gamma_j \iff \bigvee_{k=1}^{3} r_k \in \mathbb{R} \wedge \Gamma(r_k) \wedge \Phi_j(r_k) \iff r_j \in \mathbb{R} \wedge \Gamma(r_j).$$

Therefore the lemma is proved. □

Proof of Theorem 3.5.1. Let $x \in \mathbb{R}$. By Lemmas 3.6.1 and 3.6.2, we have

$$\begin{aligned} f(x) = 0 \wedge \Gamma(x) &\iff (x = r_1 \vee x = r_2 \vee x = r_3) \wedge \Gamma(x) \\ &\iff [x = r_1 \wedge r_1 \in \mathbb{R} \wedge \Gamma(r_1)] \vee \\ &\qquad [x = r_2 \wedge r_2 \in \mathbb{R} \wedge \Gamma(r_2)] \vee \\ &\qquad [x = r_3 \wedge r_3 \in \mathbb{R} \wedge \Gamma(r_3)] \\ &\iff [x = x_1 \wedge \Gamma_1] \vee [x = x_2 \wedge \Gamma_2] \vee [x = x_3 \wedge \Gamma_3]. \end{aligned}$$

The theorem is proved. □

Chapter 4

Solution Formulas for Quartic Equations Without or With Constraints

In this chapter, we generalize the solution formulas from the cubic to the quartic case. We adjust the Lagrange-type solution formulas for quartic equations and prove that the real convention can also yield correct interpretations of these adjusted formulas. In particular, using the real convention, we present the real solution formulas for the general real-coefficient quartic polynomial equations with equality and inequality constraints. The extension of the solution formulas from the cubic to the quartic case is not straightforward and needs some quite sophisticated techniques including the theories underlying Sturm-Habicht sequence and discrimination systems. Most of the material in this chapter is based on the joint paper [64] with Hong and Wang.

4.1 Adjusted Solution Formula for Quartic Equations

A number of problems in computer graphics are reduced to finding approximate real solutions of quartic equations. Quartics are the highest degree polynomials which can be solved by the method of radicals according to the well known Abel/Galois theory. The solution of a quartic equation requires the solution of a subsidiary cubic equation. Therefore the standard convention for the square and cubic roots can not always yield correct interpretations for the general quartic equations with real coefficients. Naturally, we think about whether the real convention represented in Section 3.3 is correct for the quartic formulas. After many investigations and experiments, we find the adjusted quartic solution formula (presented in Theorem 4.1.1) which is quite different from the Lagrange's quartic formula (in Theorem 2.1.6), and prove the correctness of the real convention for this formula. The adjustment is essential for obtaining the real solution formulas for the general real-coefficient quartic equation under equality and inequality constraints provided in Theorem 4.4.1 in Section 4.4. The extension of the solution formulas from the cubic to the quartic case is not straightforward and needs some quite sophisticated techniques including the theories underlying Sturm-Habicht sequence and discrimination systems [54, 90, 147, 148].

Theorem 4.1.1 (Quartic Formula). *Let $f(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{R}[x]$, the formula for the four solutions x_1, x_2, x_3, x_4 of the equation $f(x) = 0$ is,*

$$
\begin{aligned}
x_1 &= (-a_3 + k_1 + k_2 + k_3)/4, & k_1 &= \sqrt{(p_3 - 4\,\omega^2 t_1 - 4\,\omega^1 t_2)/3},\\
x_2 &= (-a_3 + k_1 - k_2 - k_3)/4, & k_2 &= \sqrt{(p_3 - 4\,\omega^0 t_1 - 4\,\omega^0 t_2)/3},\\
x_3 &= (-a_3 - k_1 + k_2 - k_3)/4, & k_3 &= \sigma_2\sqrt{(p_3 - 4\,\omega^1 t_1 - 4\,\omega^2 t_2)/3},\\
x_4 &= (-a_3 - k_1 - k_2 + k_3)/4, & &\\
t_1 &= w^{-\sigma_1}\sqrt[3]{(p_2 + 3\,s)/2}, & s &= \sqrt{-3\,p_1},\\
t_2 &= w^{+\sigma_1}\sqrt[3]{(p_2 - 3\,s)/2}, & \omega &= \mathrm{e}^{i\frac{2\pi}{3}} = -\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}i,
\end{aligned}
$$

$$
\sigma_1 = \begin{cases} 0 & \text{if } p_1 < 0,\\ \mathrm{sign}(p_2) & \text{if } p_1 \geq 0, \end{cases}
$$

$$
\sigma_2 = \begin{cases} +1 & \text{if } (r \leq 1 \wedge p_4 \leq 0) \vee (r > 1 \wedge p_4 > 0),\\ -1 & \text{if } (r \leq 1 \wedge p_4 > 0) \vee (r > 1 \wedge p_4 \leq 0), \end{cases}
$$

r = *number of distinct real roots of* $f \in \{0, 1, 2, 3, 4\}$,

$$
\begin{aligned}
p_1 = {} & 18\,a_3^3 a_1 a_2 a_0 + 256\,a_0^3 - 27 a_1^4 - 6\,a_3^2 a_1^2 a_0 - 192\,a_3 a_1 a_0^2 + 18\,a_3 a_1^3 a_2\\
& + 144\,a_2 a_3^2 a_0^2 + a_2^2 a_3^2 a_1^2 - 4\,a_2^3 a_3^2 a_0 + 144\,a_0 a_1^2 a_2 - 4\,a_3^3 a_1^3 - 128\,a_2^2 a_0^2\\
& + 16\,a_2^4 a_0 - 4\,a_2^3 a_1^2 - 27\,a_3^4 a_0^2 - 80\,a_3 a_1 a_2^2 a_0,\\
p_2 = {} & -27\,a_3^2 a_0 + 9\,a_3 a_2 a_1 - 2\,a_2^3 + 72\,a_2 a_0 - 27\,a_1^2,\\
p_3 = {} & 3\,a_3^2 - 8\,a_2, \qquad p_4 = 4\,a_2 a_3 - 8\,a_1 - a_3^3.
\end{aligned}
$$

The above formulas under the real convention yields correct solutions for all quartic polynomial equations with real coefficients.

Proof. Will be given in Section 4.3. □

Note that the number r of distinct real roots of f can be determined by using the discrimination system presented in Section 2.2: first produce the discrimination sequence $[D_1, \ldots, D_4] = [1, p_3, p_5, p_1]$ of f with real coefficients, where

$$
p_5 = 16\,a_2 a_0 a_4^2 - 18\,a_1^2 a_4^2 - 4\,a_4 a_2^3 + 14\,a_1 a_2 a_3 a_4 - 6\,a_0 a_4 a_3^2 + a_2^2 a_3^2 - 3\,a_1 a_3^3
$$

and p_1, p_3 are the same as in Theorem 4.1.1, and then construct the sign list and the revised sign list of sequence. The number r can be computed by the number of sign changes of the revised sign list and the number of non-vanishing members in the revised sign list (see Theorem 2.2.5).

In the following section, configuration indexing of the solutions for the quartic equations is presented. In Section 4.3, we prove that the real convention represented in Section 3.3 can also yield correct interpretations of the adjusted solution formula in Theorem 4.1.1 for all quartic polynomials with real coefficients. In Section 4.4, using the real convention, we present real solution formulas for the general real-coefficient quartic equation under equality and inequality constraints and prove its correctness.

4.2 Configuration of the Solutions of Quartic Equations

Let f be an arbitrary (monic) quartic polynomial with real coefficients. Let r_1, r_2, r_3, r_4 be the four (complex) solutions of $f = 0$. Using the well known relations

$$\begin{aligned}
a_3 &= -r_1 - r_2 - r_3 - r_4,\\
a_2 &= r_1r_2 + r_3r_1 + r_3r_2 + r_4r_1 + r_4r_2 + r_4r_3,\\
a_1 &= -r_1r_2r_3 - r_4r_1r_2 - r_4r_3r_1 - r_4r_3r_2,\\
a_0 &= r_1r_2r_3r_4,
\end{aligned}$$

we can rewrite p_1, p_2 and p_4 as

$$\begin{aligned}
p_1 &= (r_1 - r_2)^2(r_1 - r_3)^2(r_1 - r_4)^2(r_2 - r_3)^2(r_2 - r_4)^2(r_3 - r_4)^2,\\
p_2 &= (r_1r_2 + r_1r_3 + r_2r_4 + r_3r_4 - 2\,r_1r_4 - 2\,r_2r_3)\cdot\\
&\quad (r_1r_2 + r_1r_4 + r_2r_3 + r_3r_4 - 2\,r_1r_3 - 2\,r_2r_4)\cdot\\
&\quad (r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 - 2\,r_1r_2 - 2\,r_3r_4),\\
p_4 &= (r_1 + r_2 - r_3 - r_4)(r_1 + r_3 - r_2 - r_4)(r_1 + r_4 - r_2 - r_3).
\end{aligned}$$

It is easy to verify that the signs of p_1, p_2, p_4, and the number r of distinct real roots of f determine the "configuration" of the solutions r_1, r_2, r_3 and r_4, as shown in Figure 4.1 and Figure 4.2. Each rectangle denotes a complex plane, in which the horizontal line is the real axis with left-to-right direction. A small disk stands for a simple solution, a bigger disk for a double solution, a second biggest disk for a triple solution, and the biggest disk for a quadruple solution. We have also indexed the solutions so that we can refer to them later on. Note that some configurations in the indexing are peculiar (causing solutions jump discontinuously) but they are essential.

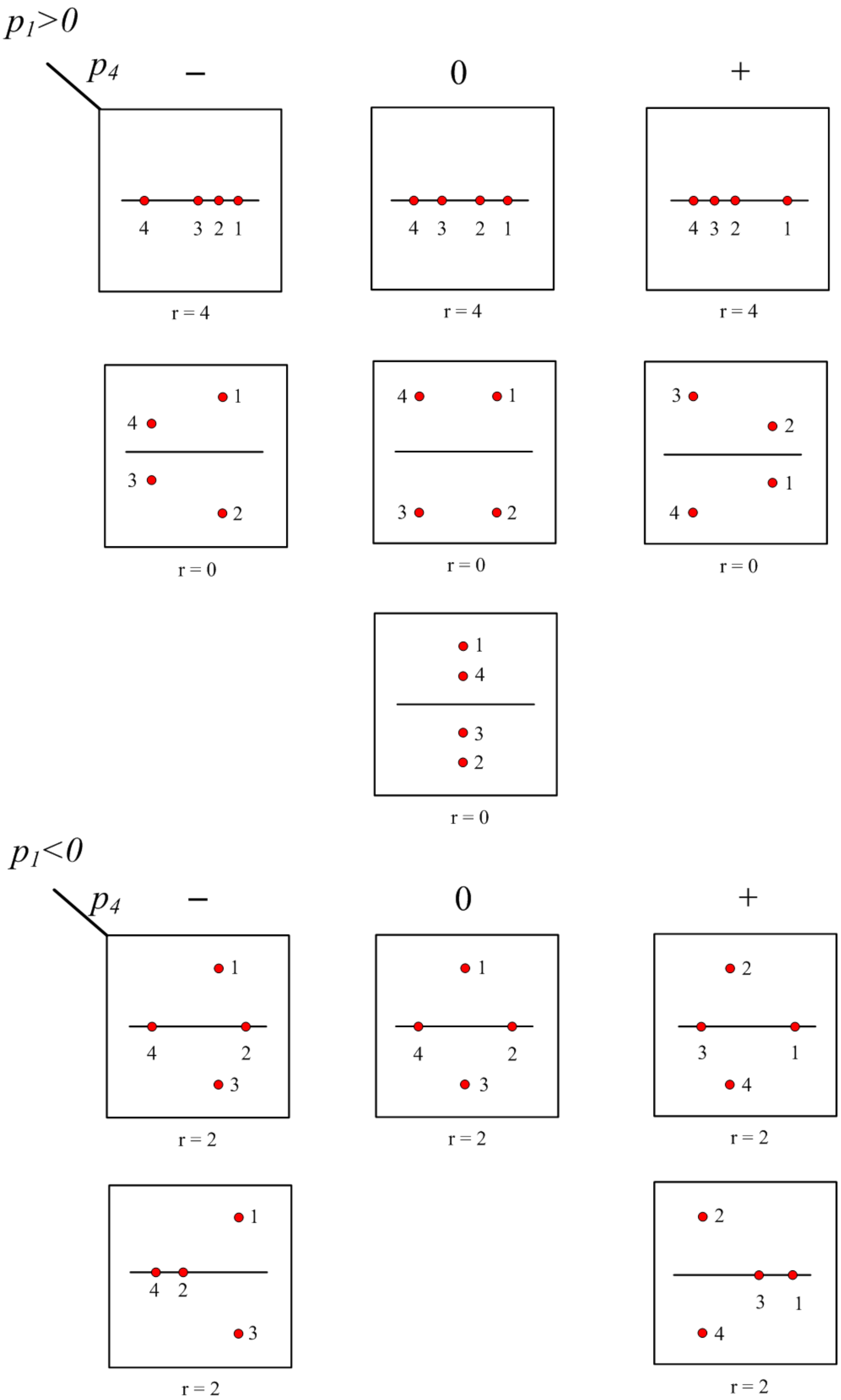

Figure 4.1: Solution Indexing for Quartic Equation ($p_1 \neq 0$)

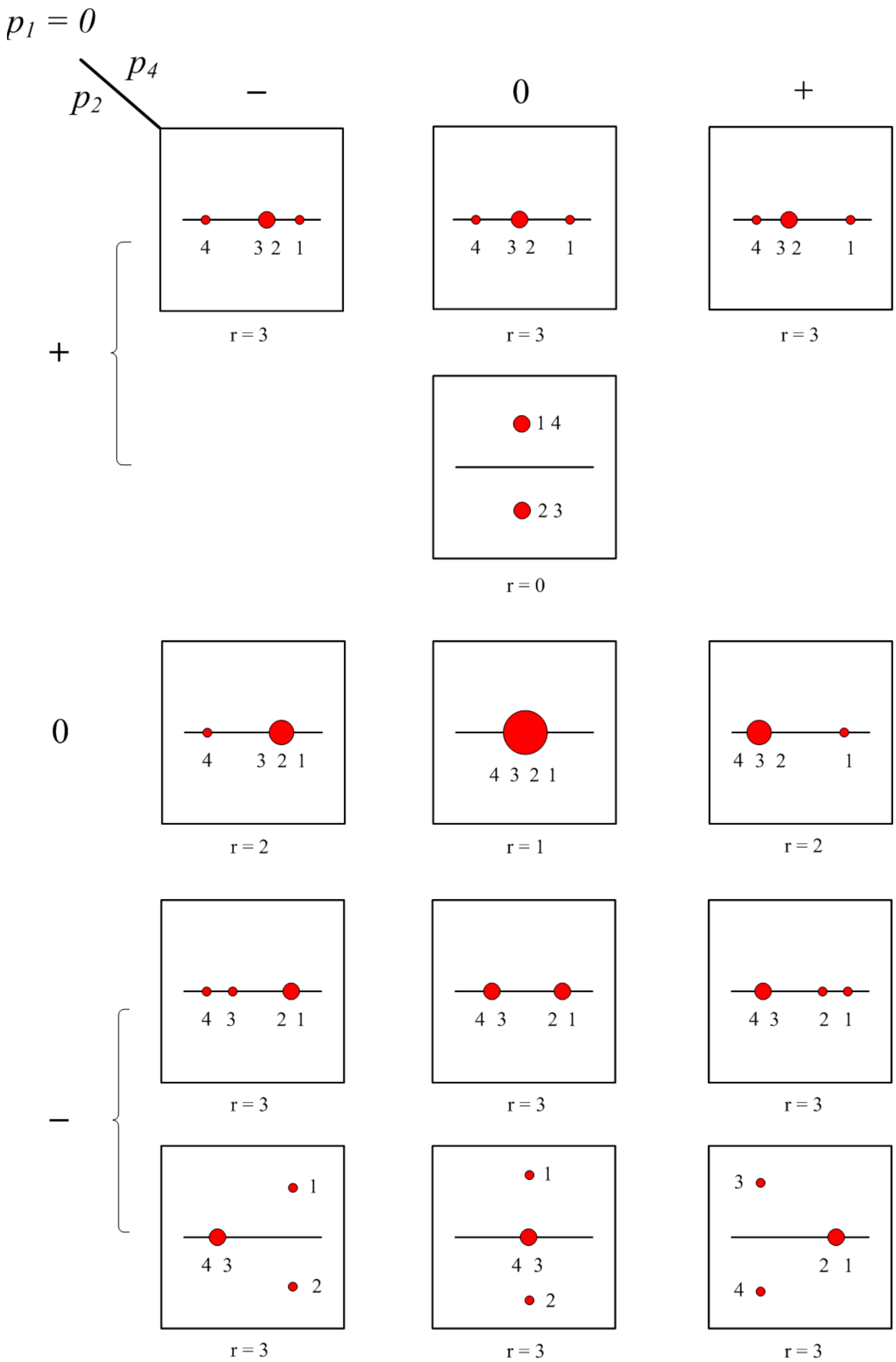

Figure 4.2: Solution Indexing for Quartic Equation ($p_1 = 0$)

4.3 Proof of the Correctness of the Real Convention

In this section, we prove Theorem 4.1.1 stated in Section 4.1. The proof proceeds by rewriting, in terms of the solutions r_1, r_2, r_3 and r_4, the expressions for s, t_1, t_2, k_1, k_2, k_3 and x_1, x_2, x_3, x_4 in the adjusted quartic formula, taking radicals according to the real convention. It is split into the following several lemmas.

Lemma 4.3.1. $s = i\sqrt{3}\,(r_1-r_2)(r_1-r_3)(r_1-r_4)(r_2-r_3)(r_2-r_4)(r_3-r_4)$.

Proof. Let $q = -3\,p_1$. Then we obviously have

$$q = \Big[i\sqrt{3}\,(r_1-r_2)(r_1-r_3)(r_1-r_4)(r_2-r_3)(r_2-r_4)(r_3-r_4)\Big]^2.$$

Hence $\sqrt{q}$ is one of the following:

$$q_1 = +i\sqrt{3}\,(r_1-r_2)(r_1-r_3)(r_1-r_4)(r_2-r_3)(r_2-r_4)(r_3-r_4),$$
$$q_2 = -i\sqrt{3}\,(r_1-r_2)(r_1-r_3)(r_1-r_4)(r_2-r_3)(r_2-r_4)(r_3-r_4).$$

We proceed to show that $s = q_1$ in every configuration of the solutions in Figure 4.1 and Figure 4.2.

(1) $p_1 > 0$ and $r = 4$. In this case, $f = 0$ has four real solutions indexed as $r_4 < r_3 < r_2 < r_1$. Note that

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(2) $p_1 > 0$, $p_4 > 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha - i\beta$, $r_2 = \alpha + i\beta$, $r_3 = \mu + i\nu$ and $r_4 = \mu - i\nu$ such that $\alpha > \mu$, $\beta < \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = +4\sqrt{3}\,\beta\nu\left[(\alpha-\mu)^2+(\beta+\nu)^2\right]\left[(\alpha-\mu)^2+(\beta-\nu)^2\right] i,$$
$$q_2 = -4\sqrt{3}\,\beta\nu\left[(\alpha-\mu)^2+(\beta+\nu)^2\right]\left[(\alpha-\mu)^2+(\beta-\nu)^2\right] i.$$

Then

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(3) $p_1 > 0$, $p_4 \leq 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha \geq \mu$, $\beta \geq \nu$, $\beta > 0$ and $\nu > 0$. Note that $\alpha = \mu$ and $\beta = \nu$ can not appear simultaneously. Simple calculation shows that

$$q_1 = +4\sqrt{3}\,\beta\nu\left[(\alpha-\mu)^2 + (\beta+\nu)^2\right]\left[(\alpha-\mu)^2 + (\beta-\nu)^2\right] i,$$
$$q_2 = -4\sqrt{3}\,\beta\nu\left[(\alpha-\mu)^2 + (\beta+\nu)^2\right]\left[(\alpha-\mu)^2 + (\beta-\nu)^2\right] i.$$

Then

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(4) $p_1 = 0$. In this case, $f = 0$ has multiple solutions. It follows that $q_1 = q_2 = 0$ and

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(5) $p_1 < 0$ and $p_4 > 0$. In this case, $f = 0$ has two real solutions r_1, r_3 and a pair of complex conjugates $r_2 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $r_1 > r_3$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +2\sqrt{3}\,\beta(r_1 - r_3)\left[(r_1-\alpha)^2 + \beta^2\right]\left[(r_3-\alpha)^2 + \beta^2\right] > 0,$$
$$q_2 = -2\sqrt{3}\,\beta(r_1 - r_3)\left[(r_1-\alpha)^2 + \beta^2\right]\left[(r_3-\alpha)^2 + \beta^2\right] < 0.$$

Then

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$.

(6) $p_1 < 0$ and $p_4 \leq 0$. In this case, $f = 0$ has two real solutions r_2, r_4 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 > r_4$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +2\sqrt{3}\,\beta(r_2 - r_4)\left[(r_2 - \alpha)^2 + \beta^2\right]\left[(r_4 - \alpha)^2 + \beta^2\right] > 0,$$

$$q_2 = -2\sqrt{3}\,\beta(r_2 - r_4)\left[(r_2 - \alpha)^2 + \beta^2\right]\left[(r_4 - \alpha)^2 + \beta^2\right] < 0.$$

Then

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $s = q_1$. □

Lemma 4.3.2. $t_1 = \omega^2(r_1 + r_4)(r_2 + r_3) + \omega^0(r_1 + r_3)(r_2 + r_4) + \omega^1(r_1 + r_2)(r_3 + r_4)$.

Proof. Let $q = (p_2 + 3\,s)/2$. Recalling Lemma 4.3.1, substitution and factorization yield

$$q = \left[\omega^0(r_1 + r_4)(r_2 + r_3) + \omega^1(r_1 + r_3)(r_2 + r_4) + \omega^2(r_1 + r_2)(r_3 + r_4)\right]^3.$$

Let

$$y_1 = (r_1 + r_4)(r_2 + r_3),$$

$$y_2 = (r_1 + r_3)(r_2 + r_4),$$

$$y_3 = (r_1 + r_2)(r_3 + r_4),$$

and note that

$$y_1 - y_2 = (r_1 - r_2)(r_3 - r_4),$$

$$y_1 - y_3 = (r_1 - r_3)(r_2 - r_4),$$

$$y_2 - y_3 = (r_1 - r_4)(r_2 - r_3).$$

Then we have

$$q = (\omega^0 y_1 + \omega^1 y_2 + \omega^2 y_3)^3.$$

Hence $\sqrt[3]{q}$ is one of the following:

$$q_1 = \omega^0 y_1 + \omega^1 y_2 + \omega^2 y_3,$$
$$q_2 = \omega^2 y_1 + \omega^0 y_2 + \omega^1 y_3,$$
$$q_3 = \omega^1 y_1 + \omega^2 y_2 + \omega^0 y_3.$$

Let

$$g(x) = (x - y_1)(x - y_2)(x - y_3).$$

Note that the cubic polynomial $g(x)$ is the *resolvent cubic* of the quartic polynomial $f(x)$. The coefficients of $g(x)$ can be found fairly easily using the reduction algorithm for symmetric polynomials, which yields

$$g(x) = x^3 - 2\,a_2 x^2 + (a_2^2 - 4\,a_0 + a_1 a_3)x + (a_1^2 + a_0 a_3^2 - a_1 a_2 a_3),$$

and p_1, p_2 of the cubic polynomial $g(x)$ with respect to the coefficients of $g(x)$ as shown in Theorem 2.1.5 are just the same ones of the quartic polynomial $f(x)$ in Theorem 4.1.1.

Now we prove that $t_1 = q_2$ for every configuration of the solutions.

(1) $p_1 > 0$. In this case, $f = 0$ has four simple real solutions or two pairs of complex conjugates. We consider the following three subcases, and prove that y_1, y_2, y_3 are real and can be indexed as $y_3 < y_2 < y_1$.

- $r = 4$. In this case, four simple real solutions are indexed as $r_4 < r_3 < r_2 < r_1$. Hence $y_3 < y_2 < y_1$.
- $r = 0$ and $p_4 > 0$. In this case, two pairs of complex conjugates are $r_1 = \alpha - i\beta$, $r_2 = \alpha + i\beta$, $r_3 = \mu + i\nu$ and $r_4 = \mu - i\nu$ such that $\alpha > \mu$, $\beta < \nu$,

$\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$y_1 = (\alpha + \mu)^2 + (\beta + \nu)^2,$$
$$y_2 = (\alpha + \mu)^2 + (\beta - \nu)^2,$$
$$y_3 = 4\,\alpha\mu,$$

and

$$y_1 - y_2 = 4\,\beta\nu > 0,$$
$$y_1 - y_3 = (\alpha - \mu)^2 + (\beta + \nu)^2 > 0,$$
$$y_2 - y_3 = (\alpha - \mu)^2 + (\beta - \nu)^2 > 0.$$

Hence $y_3 < y_2 < y_1$.

- $r = 0$ and $p_4 \leq 0$. In this case, two pairs of complex conjugates are $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha \geq \mu$, $\beta \geq \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$y_1 = (\alpha + \mu)^2 + (\beta + \nu)^2,$$
$$y_2 = (\alpha + \mu)^2 + (\beta - \nu)^2,$$
$$y_3 = 4\,\alpha\mu,$$

and

$$y_1 - y_2 = 4\,\beta\nu > 0,$$
$$y_1 - y_3 = (\alpha - \mu)^2 + (\beta + \nu)^2 > 0,$$
$$y_2 - y_3 = (\alpha - \mu)^2 + (\beta - \nu)^2 > 0.$$

Hence $y_3 < y_2 < y_1$.

So $g(x) = 0$ has three simple real solutions indexed as $y_3 < y_2 < y_1$. According to the configuration for the cubic solutions in Figure 3.2, we consider the following three subcases according to the sign of p_2, and prove that $t_1 = q_2$ in each subcase.

(a) $p_2 > 0$. In this case, $y_2 - y_3 < y_1 - y_2$. According to the results of Case (1) in the proof of Lemma 3.4.2 in Section 3.4, we have $\sqrt[3]{q} = q_1$. Since $p_1 > 0$ and $p_2 > 0$, then $\sigma_1 = +1$ and so $t_1 = \omega^{-1} q_1 = q_2$.

(b) $p_2 = 0$. In this case, $y_2 - y_3 = y_1 - y_2$. According to the results of Case (2) in the proof of Lemma 3.4.2 in Section 3.4, we have $\sqrt[3]{q} = q_2$. Since $p_1 > 0$ and $p_2 = 0$, then $\sigma_1 = 0$ and so $t_1 = \omega^{-0} q_2 = q_2$.

(c) $p_2 < 0$. In this case, $y_1 - y_2 > y_2 - y_3$. According to the results of Case (3) in the proof of Lemma 3.4.2 in Section 3.4, we have $\sqrt[3]{q} = q_3$. Since $p_1 > 0$ and $p_2 < 0$, then $\sigma_1 = -1$ and so $t_1 = \omega^{-(-1)} q_3 = q_2$.

Thus we have $t_1 = q_2$.

(2) $p_1 = 0$, $p_2 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_4 and a double real solution $r_2 = r_3$ such that $r_4 < r_3 = r_2 < r_1$. Hence $y_3 = y_2 < y_1$. According to the results of Case (4) in the proof of Lemma 3.4.2 in Section 3.4, we have $\sqrt[3]{q} = q_1$. Since $p_1 = 0$ and $p_2 > 0$, then $\sigma_1 = +1$ and so $t_1 = \omega^{-1} q_1 = q_2$.

(3) $p_1 = 0$, $p_2 > 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha = \mu$, $\beta = \nu$ and $\beta > 0$. Simple calculation gives

$$y_1 = 4\,\alpha^2 + 4\,\beta^2, \quad y_2 = 4\,\alpha^2, \quad y_3 = 4\,\alpha^2.$$

Then we have $y_3 = y_2 < y_1$. According to the results of Case (4) in the proof of Lemma 3.4.2 in Section 3.4, we have $\sqrt[3]{q} = q_1$. Since $p_1 = 0$ and $p_2 > 0$, then $\sigma_1 = +1$ and so $t_1 = \omega^{-1} q_1 = q_2$.

(4) $p_1 = 0$ and $p_2 = 0$. In this case, $f = 0$ has a triple real solution and a simple real solution or a quadruple real solution. We consider the following three subcases, and prove that $y_3 = y_2 = y_1$ in each subcase.

(a) $p_4 > 0$. In this case, $f = 0$ has a simple real solution r_1 and a triple real solution $r_2 = r_3 = r_4$ such that $r_4 = r_3 = r_2 < r_1$. Hence $y_3 = y_2 = y_1$.

(b) $p_4 = 0$. In this case, $f = 0$ has a quadruple real solution $r_4 = r_3 = r_2 = r_1$. Hence $y_3 = y_2 = y_1$.

(c) $p_4 < 0$. In this case, $f = 0$ has a simple real solution r_4 and a triple real solution $r_1 = r_2 = r_3$ such that $r_4 < r_3 = r_2 = r_1$. Hence $y_3 = y_2 = y_1$.

Then $g = 0$ has a triple solution $y_3 = y_2 = y_1$. According to the results of Case (5) in the proof of Lemma 3.4.2 in Section 3.4, we have $\sqrt[3]{q} = q_2$. Since $p_1 = 0$ and $p_2 = 0$, then $\sigma_1 = 0$ and so $t_1 = \omega^{-0} q_2 = q_2$.

(5) $p_1 = 0$ and $p_2 < 0$. In this case, there are six subcases according to the number r of distinct real solutions and the sign of p_4. We prove that $y_3 < y_2 = y_1$ in each subcase.

(a) $p_4 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_2 and a double real solution $r_3 = r_4$ such that $r_4 = r_3 < r_2 < r_1$. Hence $y_3 = y_2 = y_1$.

(b) $p_4 > 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_2 = r_1$ and a pair of complex conjugates $r_3 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $\alpha < r_1 = r_2$ and $\beta > 0$. Simple calculation gives

$$y_1 = (\alpha + r_2)^2 + \beta^2,$$
$$y_2 = (\alpha + r_2)^2 + \beta^2,$$
$$y_3 = 4\,\alpha r_2.$$

Hence $y_3 < y_2 = y_1$.

(c) $p_4 = 0$ and $r = 2$. In this case, $f = 0$ has two double real solutions $r_1 = r_2$ and $r_3 = r_4$ such that $r_4 = r_3 < r_2 = r_1$. Hence $y_3 < y_2 = y_1$.

(d) $p_4 = 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $\alpha = r_3 = r_4$ and $\beta > 0$. Simple calculation gives

$$y_1 = 4\,\alpha^2 + \beta^2, \quad y_2 = 4\,\alpha^2 + \beta^2, \quad y_3 = 4\,\alpha^2.$$

Hence $y_3 < y_2 = y_1$.

(e) $p_4 < 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_3, r_4 and a double real solution $r_1 = r_2$ such that $r_4 < r_3 < r_2 = r_1$. Hence $y_3 < y_2 = y_1$.

(f) $p_4 < 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $r_4 = r_3 < \alpha$ and $\beta > 0$. Simple calculation gives

$$\begin{aligned} y_1 &= (\alpha + r_4)^2 + \beta^2, \\ y_2 &= (\alpha + r_4)^2 + \beta^2, \\ y_3 &= 4\,\alpha r_4. \end{aligned}$$

Hence $y_3 < y_2 = y_1$.

Then $g = 0$ has a simple real solution y_3 and a double real solution $y_1 = y_2$ such that $y_3 < y_2 = y_1$. According to the results of Case (6) in the proof of Lemma 3.4.2 in Section 3.4, we have $\sqrt[3]{q} = q_3$. Since $p_1 = 0$ and $p_2 < 0$, then $\sigma_1 = -1$ and so $t_1 = \omega^{-(-1)} q_3 = q_2$.

(6) $p_1 < 0$. In this case, $f = 0$ has two simple real solutions and a pair of complex conjugates, and $g = 0$ has a simple real solution and a pair of complex conjugates. Note that the indexing of the cubic solutions y_1, y_2 and y_3 is not the same as in Figure 3.2 for the configurations of $p_1 < 0$. We show the new indexing for the three cubic solutions in Figure 4.3 for the case $p_1 < 0$. From here to the end

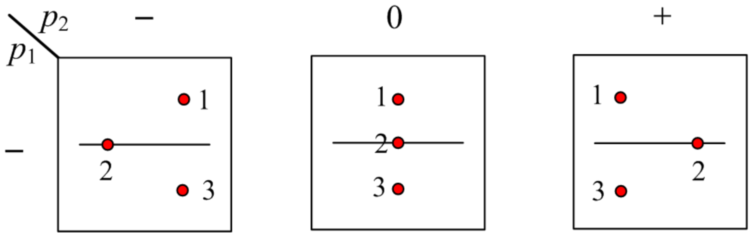

Figure 4.3: Another Solution Indexing for Cubic Equation for Case $p_1 < 0$

of the proof of this lemma, we use this new indexing for the cubic solutions for the condition $p_1 < 0$. Now we prove that $\sqrt[3]{q} = q_2$ in each of the following two subcases, according to the sign of p_4.

(a) $p_4 > 0$. In this case, $f = 0$ has two simple real solution r_1, r_3 and a pair of complex conjugates $r_2 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $r_1 > r_3$ and $\beta > 0$. Simple calculation gives

$$y_1 = (\alpha r_1 + \alpha r_3 + r_1 r_3 + \alpha^2 + \beta^2) + (r_1 - r_3)\beta i,$$

$$y_2 = 2\,\alpha(r_1 + r_3),$$

$$y_3 = (\alpha r_1 + \alpha r_3 + r_1 r_3 + \alpha^2 + \beta^2) - (r_1 - r_3)\beta i.$$

Let

$$\zeta = \alpha r_1 + \alpha r_3 + r_1 r_3 + \alpha^2 + \beta^2, \quad \delta = (r_1 - r_3)\beta.$$

Then $y_1 = \zeta + i\delta$ and $y_3 = \zeta - i\delta$ with $\delta > 0$. We will show that $\sqrt[3]{q} = q_2$ in each of the following three subcases.

(i) $p_2 > 0$. In this case, we have $\zeta < y_2$. Simple calculation gives

$$q_1 = \omega^1(y_2 - \zeta + \sqrt{3}\,\delta),$$

$$q_2 = \omega^0(y_2 - \zeta + \sqrt{3}\,\delta),$$

$$q_3 = \omega^2(y_2 - \zeta + \sqrt{3}\,\delta).$$

Thus

$$\arg q_1 = +\frac{2}{3}\pi, \quad \arg q_2 = +\frac{0}{3}\pi, \quad \arg q_3 = -\frac{2}{3}\pi.$$

Since $q = (p_2 + 3\,s)/2 > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

(ii) $p_2 = 0$. In this case, we have $\zeta = y_2$. Simple calculation gives

$$q_1 = \omega^1 \sqrt{3}\,\delta, \quad q_2 = \omega^0 \sqrt{3}\,\delta, \quad q_3 = \omega^2 \sqrt{3}\,\delta.$$

Thus

$$\arg q_1 = +\frac{2}{3}\pi, \quad \arg q_2 = +\frac{0}{3}\pi, \quad \arg q_3 = -\frac{2}{3}\pi.$$

Since $q = 3\,s/2 > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

(iii) $p_2 < 0$. In this case, we have $\zeta > y_2$. Simple calculation gives

$$q_1 = \omega^1(y_2 - \zeta + \sqrt{3}\,\delta),$$
$$q_2 = \omega^0(y_2 - \zeta + \sqrt{3}\,\delta),$$
$$q_3 = \omega^2(y_2 - \zeta + \sqrt{3}\,\delta).$$

Note also that

$$q = (y_2 - \zeta + \sqrt{3}\,\delta)^3.$$

We prove that $\sqrt[3]{q} = q_2$ in each of the following three subcases.

- $y_2 - \zeta + \sqrt{3}\,\delta > 0$. In this case,

  $$\arg q_1 = +\frac{2}{3}\pi, \quad \arg q_2 = +\frac{0}{3}\pi, \quad \arg q_3 = -\frac{2}{3}\pi.$$

  Since $q > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

- $y_2 - \zeta + \sqrt{3}\,\delta = 0$. In this case, $q_1 = q_2 = q_3 = 0$ and thus

  $$\arg q_1 = 0, \quad \arg q_2 = 0, \quad \arg q_3 = 0.$$

Since $q = 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

- $y_2 - \zeta + \sqrt{3}\,\delta < 0$. In this case,

$$\arg q_1 = -\frac{1}{3}\pi, \quad \arg q_2 = +\frac{3}{3}\pi, \quad \arg q_3 = +\frac{1}{3}\pi.$$

Since $q < 0$, we have $\arg q = \pi$. Therefore $\arg \sqrt[3]{q} = \pi$. Hence $\sqrt[3]{q} = q_2$.

(b) $p_4 \leq 0$. In this case, $f = 0$ has two simple real solution r_2, r_4 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 > r_4$ and $\beta > 0$. Simple calculation gives

$$y_1 = (\alpha r_2 + \alpha r_4 + r_2 r_4 + \alpha^2 + \beta^2) + (r_2 - r_4)\beta i,$$

$$y_2 = 2\,\alpha(r_2 + r_4),$$

$$y_3 = (\alpha r_2 + \alpha r_4 + r_2 r_4 + \alpha^2 + \beta^2) - (r_2 - r_4)\beta i.$$

Let

$$\zeta = \alpha r_2 + \alpha r_4 + r_2 r_4 + \alpha^2 + \beta^2, \quad \delta = (r_2 - r_4)\beta.$$

Then we have $y_1 = \zeta + i\delta$ and $y_3 = \zeta - i\delta$ with $\delta > 0$. Now we prove that $\sqrt[3]{q} = q_2$ in each of the following three subcases.

(i) $p_2 > 0$. In this case, we have $\zeta < y_2$. Simple calculation gives

$$q_1 = \omega^1(y_2 - \zeta + \sqrt{3}\,\delta),$$

$$q_2 = \omega^0(y_2 - \zeta + \sqrt{3}\,\delta),$$

$$q_3 = \omega^2(y_2 - \zeta + \sqrt{3}\,\delta).$$

Thus

$$\arg q_1 = +\frac{2}{3}\pi, \quad \arg q_2 = +\frac{0}{3}\pi, \quad \arg q_3 = -\frac{2}{3}\pi.$$

Since $q = (p_2 + 3\,s)/2 > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

(ii) $p_2 = 0$. In this case, we have $\zeta = y_2$. Simple calculation gives

$$q_1 = \omega^1 \sqrt{3}\,\delta, \quad q_2 = \omega^0 \sqrt{3}\,\delta, \quad q_3 = \omega^2 \sqrt{3}\,\delta.$$

Thus

$$\arg q_1 = +\frac{2}{3}\pi, \quad \arg q_2 = +\frac{0}{3}\pi, \quad \arg q_3 = -\frac{2}{3}\pi.$$

Since $q = 3\,s/2 > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

(iii) $p_2 < 0$. In this case, we have $\zeta > y_2$. Simple calculation gives

$$q_1 = \omega^1 (y_2 - \zeta + \sqrt{3}\,\delta),$$
$$q_2 = \omega^0 (y_2 - \zeta + \sqrt{3}\,\delta),$$
$$q_3 = \omega^2 (y_2 - \zeta + \sqrt{3}\,\delta).$$

Note also that

$$q = (y_2 - \zeta + \sqrt{3}\,\delta)^3.$$

We prove that $\sqrt[3]{q} = q_2$ in each of the following three subcases.

- $y_2 - \zeta + \sqrt{3}\,\delta > 0$. In this case,

  $$\arg q_1 = +\frac{2}{3}\pi, \quad \arg q_2 = +\frac{0}{3}\pi, \quad \arg q_3 = -\frac{2}{3}\pi.$$

  Since $q > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

- $y_2 - \zeta + \sqrt{3}\,\delta = 0$. In this case, $q_1 = q_2 = q_3 = 0$ and thus

  $$\arg q_1 = 0, \quad \arg q_2 = 0, \quad \arg q_3 = 0.$$

  Since $q = 0$, we have $\arg q = 0$. Therefore $\arg \sqrt[3]{q} = 0$. Hence $\sqrt[3]{q} = q_2$.

- $y_2 - \zeta + \sqrt{3}\,\delta < 0$. In this case,

$$\arg q_1 = -\frac{1}{3}\pi, \quad \arg q_2 = +\frac{3}{3}\pi, \quad \arg q_3 = +\frac{1}{3}\pi.$$

  Since $q < 0$, we have $\arg q = \pi$. Therefore $\arg \sqrt[3]{q} = \pi$. Hence $\sqrt[3]{q} = q_2$.

Since $p_1 < 0$, then $\sigma_1 = 0$ and so $t_1 = \omega^{-0} q_2 = q_2$.

Thus the lemma is proved. □

Lemma 4.3.3. $t_2 = \omega^1(r_1 + r_4)(r_2 + r_3) + \omega^0(r_1 + r_3)(r_2 + r_4) + \omega^2(r_1 + r_2)(r_3 + r_4)$.

Proof. The proof is essentially the same as that the previous Lemma 4.3.2. One only has to swap the indices 2 and 4. □

Lemma 4.3.4. $k_1 = r_1 + r_2 - r_3 - r_4$.

Proof. Let $q = (p_3 - 4\,\omega^2 t_1 - 4\,\omega^1 t_2)/3$. Recalling Lemma 4.3.2 and Lemma 4.3.3, substitution and factorization yield

$$q = (r_1 + r_2 - r_3 - r_4)^2.$$

Hence $\sqrt{q}$ is one of the following:

$$q_1 = +(r_1 + r_2 - r_3 - r_4), \quad q_2 = -(r_1 + r_2 - r_3 - r_4).$$

We proceed to show that $k_1 = q_1$ in every configuration of the solutions in Figure 4.1 and Figure 4.2.

(1) $p_1 > 0$ and $r = 4$. In this case, $f = 0$ has four simple real solutions indexed as $r_4 < r_3 < r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

  Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(2) $p_1 > 0$, $p_4 > 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha - i\beta$, $r_2 = \alpha + i\beta$, $r_3 = \mu + i\nu$ and $r_4 = \mu - i\nu$ such that $\alpha > \mu$, $\beta < \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = +2(\alpha - \mu) > 0, \quad q_2 = -2(\alpha - \mu) < 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Since $q = 4(\alpha - \mu)^2 > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt{q} = 0$. Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(3) $p_1 > 0$, $p_4 \leq 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha \geq \mu$, $\beta \geq \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = +2(\alpha - \mu) \geq 0, \quad q_2 = -2(\alpha - \mu) \leq 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi \text{ or } 0.$$

Since $q = 4(\alpha - \mu)^2 > 0$, we have $\arg q = 0$. Therefore $\arg \sqrt{q} = 0$. Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(4) $p_1 = 0$, $p_2 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_4 and a double real solution $r_2 = r_3$ such that $r_4 < r_3 = r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(5) $p_1 = 0$, $p_2 > 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha = \mu$, $\beta = \nu$ and $\beta > 0$. Simple calculation gives $q_1 = q_2 = 0$. Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(6) $p_1 = 0$, $p_2 = 0$ and $p_4 > 0$. In this case, $f = 0$ has a simple real solution r_1 and a triple real solution $r_2 = r_3 = r_4$ such that $r_4 = r_3 = r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(7) $p_1 = 0$, $p_2 = 0$ and $p_4 = 0$. In this case, $f = 0$ has a quadruple real solution $r_4 = r_3 = r_2 = r_1$. It follows that $q_1 = q_2 = 0$ and

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(8) $p_1 = 0$, $p_2 = 0$ and $p_4 < 0$. In this case, $f = 0$ has a simple real solution r_4 and a triple real solution $r_1 = r_2 = r_3$ such that $r_4 < r_3 = r_2 = r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(9) $p_1 = 0$ and $p_2 < 0$. In this case, there are six subcases according to the number r of distinct real solutions and the sign of p_4. Now we prove that $k_1 = q_1$ in each subcase.

(a) $p_4 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_2 and a double real solution $r_3 = r_4$ such that $r_4 = r_3 < r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(b) $p_4 > 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_2 = r_1$ and a pair of complex conjugates $r_3 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $\alpha < r_1 = r_2$ and $\beta > 0$. Simple calculation gives

$$q_1 = +(r_1 + r_2 - 2\,\alpha) > 0, \quad q_2 = -(r_1 + r_2 - 2\,\alpha) < 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(c) $p_4 = 0$ and $r = 2$. In this case, $f = 0$ has two double real solutions $r_1 = r_2$ and $r_3 = r_4$ such that $r_4 = r_3 < r_2 = r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(d) $p_4 = 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $\alpha = r_3 = r_4$ and $\beta > 0$. Simple calculation gives $q_1 = q_2 = 0$. Then

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(e) $p_4 < 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_3, r_4 and a double real solution $r_1 = r_2$ such that $r_4 < r_3 < r_2 = r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(f) $p_4 < 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $r_4 = r_3 < \alpha$ and $\beta > 0$. Simple calculation gives

$$q_1 = +(2\,\alpha - r_3 - r_4) > 0, \quad q_2 = -(2\,\alpha - r_3 - r_4) < 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(10) $p_1 < 0$ and $p_4 > 0$. In this case, $f = 0$ has two simple real solutions r_1, r_3 and a pair of complex conjugates $r_2 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $r_1 > r_3$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +(r_1 - r_3) + 2\,\beta i, \quad q_2 = -(r_1 - r_3) - 2\,\beta i.$$

Therefore

$$0 < \arg q_1 < +\frac{\pi}{2}, \quad -\pi < \arg q_2 < -\frac{\pi}{2}.$$

Since $\operatorname{Im} q = 4\,\beta(r_1 - r_3) > 0$ (where $\operatorname{Im} q$ denotes the imaginary part of q), we have $0 < \arg q < \pi$. Therefore $0 < \arg \sqrt{q} < \pi/2$. Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$.

(11) $p_1 < 0$ and $p_4 \leq 0$. In this case, $f = 0$ has two real solutions r_2, r_4 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 > r_4$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +(r_2 - r_4) + 2\,\beta i, \quad q_2 = -(r_2 - r_4) - 2\,\beta i.$$

Therefore

$$0 < \arg q_1 < +\frac{\pi}{2}, \quad -\pi < \arg q_2 < -\frac{\pi}{2}.$$

Since $\operatorname{Im} q = 4\,\beta(r_2 - r_4) > 0$, we have $0 < \arg q < \pi$. Therefore $0 < \arg \sqrt{q} < \pi/2$. Hence $\sqrt{q} = q_1$. Thus $k_1 = q_1$. □

Lemma 4.3.5. $k_2 = r_1 - r_2 + r_3 - r_4$.

Proof. Let $q = (p_3 - 4\,\omega^0 t_1 - 4\,\omega^0 t_2)/3$. Recalling Lemma 4.3.2 and Lemma 4.3.3, substitution and factorization yield

$$q = (r_1 - r_2 + r_3 - r_4)^2.$$

Hence $\sqrt{q}$ is one of the following:

$$q_1 = +(r_1 - r_2 + r_3 - r_4), \quad q_2 = -(r_1 - r_2 + r_3 - r_4).$$

We proceed to show that $k_2 = q_1$ in every configuration of the solutions in Figure 4.1 and Figure 4.2.

(1) $p_1 > 0$ and $r = 4$. In this case, $f = 0$ has four simple real solutions indexed as $r_4 < r_3 < r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(2) $p_1 > 0$, $p_4 > 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha - i\beta$, $r_2 = \alpha + i\beta$, $r_3 = \mu + i\nu$ and $r_4 = \mu - i\nu$ such that $\alpha > \mu$, $\beta < \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = +2\,(\nu - \beta)i, \quad q_2 = -2\,(\nu - \beta)i.$$

Therefore

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,(\nu - \beta)^2 < 0$, we have $\arg q = \pi$. Therefore $\arg \sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(3) $p_1 > 0$, $p_4 \leq 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha \geq \mu$, $\beta \geq \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = +2\,(\beta - \nu)i, \quad q_2 = -2\,(\beta - \nu)i.$$

If $\beta > \nu$, then

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,(\beta - \nu)^2 < 0$, we have $\arg q = \pi$. Therefore $\arg \sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

If $\beta = \nu$, then $q_1 = q_2 = 0$ and $\arg q_1 = \arg q_2 = 0$. Since $q = 0$, we have $\arg q = 0$. Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(4) $p_1 = 0$, $p_2 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_4 and a double real solution $r_2 = r_3$ such that $r_4 < r_3 = r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(5) $p_1 = 0$, $p_2 > 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha = \mu$, $\beta = \nu$ and $\beta > 0$. Simple calculation gives $q_1 = q_2 = 0$. Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(6) $p_1 = 0$, $p_2 = 0$ and $p_4 > 0$. In this case, $f = 0$ has a simple real solution r_1 and a triple real solution $r_2 = r_3 = r_4$ such that $r_4 = r_3 = r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(7) $p_1 = 0$, $p_2 = 0$ and $p_4 = 0$. In this case, $f = 0$ has a quadruple real solution $r_4 = r_3 = r_2 = r_1$. It follows that $q_1 = q_2 = 0$ and

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(8) $p_1 = 0$, $p_2 = 0$ and $p_4 < 0$. In this case, $f = 0$ has a simple real solution r_4 and a triple real solution $r_1 = r_2 = r_3$ such that $r_4 < r_3 = r_2 = r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(9) $p_1 = 0$ and $p_2 < 0$. In this case, there are six subcases according to the number r of distinct real solutions and the sign of p_4. Now we prove that $k_2 = q_1$ in each subcase.

(a) $p_4 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_2 and a double real solution $r_3 = r_4$ such that $r_4 = r_3 < r_2 < r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(b) $p_4 > 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_2 = r_1$ and a pair of complex conjugates $r_3 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $\alpha < r_1 = r_2$ and $\beta > 0$. Simple calculation gives

$$q_1 = +2\,\beta i, \quad q_2 = -2\,\beta i.$$

Then

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,\beta^2 < 0$, we have $\arg q = \pi$. Therefore $\arg \sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(c) $p_4 = 0$ and $r = 2$. In this case, $f = 0$ has two double real solutions $r_1 = r_2$ and $r_3 = r_4$ such that $r_4 = r_3 < r_2 = r_1$. It follows that $q = q_1 = q_2 = 0$ and then

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(d) $p_4 = 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $\alpha = r_3 = r_4$ and $\beta > 0$. Simple calculation gives

$$q_1 = +2\,\beta i, \quad q_2 = -2\,\beta i.$$

Then

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,\beta^2 < 0$, we have $\arg q = \pi$. Therefore $\arg \sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(e) $p_4 < 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_3, r_4 and a double real solution $r_1 = r_2$ such that $r_4 < r_3 < r_2 = r_1$. Note that

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(f) $p_4 < 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $r_4 = r_3 < \alpha$ and $\beta > 0$. Simple calculation gives

$$q_1 = +2\,\beta i, \quad q_2 = -2\,\beta i.$$

Then

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,\beta^2 < 0$, we have $\arg q = \pi$. Therefore $\arg \sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(10) $p_1 < 0$ and $p_4 > 0$. In this case, $f = 0$ has two simple real solutions r_1, r_3 and a pair of complex conjugates $r_2 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $r_1 > r_3$ and $\beta > 0$. Simple calculation gives

$$q_1 = +(r_1 + r_3 - 2\,\alpha), \quad q_2 = -(r_1 + r_3 - 2\,\alpha).$$

We prove that $k_2 = q_1$ in each of the following subcases.

(a) $r_3 < \alpha < r_1$ and $\alpha - r_3 < r_1 - \alpha$. In this case,

$$q_1 > 0, \quad q_2 < 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(b) $\alpha < r_3 < r_1$. In this case,

$$q_1 > 0, \quad q_2 < 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(11) $p_1 < 0$ and $p_4 = 0$. In this case, $f = 0$ has two simple real solutions r_2, r_4 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 > \alpha > r_4$, $2\,\alpha = r_2 + r_4$ and $\beta > 0$. Simple calculation gives $q = q_1 = q_2 = 0$. Thus

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(12) $p_1 < 0$ and $p_4 < 0$. In this case, $f = 0$ has two simple real solutions r_2, r_4 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 > r_4$ and $\beta > 0$. Simple calculation gives

$$q_1 = +(2\,\alpha - r_2 - r_4), \quad q_2 = -(2\,\alpha - r_2 - r_4).$$

We prove that $k_2 = q_1$ in each of the following subcases.

(a) $r_4 < \alpha < r_2$ and $\alpha - r_4 > r_2 - \alpha$. In this case,

$$q_1 > 0, \quad q_2 < 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$.

(b) $r_4 < r_2 < \alpha$. In this case,

$$q_1 > 0, \quad q_2 < 0.$$

Therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Thus $k_2 = q_1$. □

Lemma 4.3.6. $k_3 = r_1 - r_2 - r_3 + r_4$.

Proof. Let $q = (p_3 - 4\,\omega^1 t_1 - 4\,\omega^2 t_2)/3$. Recalling Lemma 4.3.2 and Lemma 4.3.3, substitution and factorization yield

$$q = (r_1 - r_2 + r_3 - r_4)^2.$$

Hence $\sqrt{q}$ is one of the following:

$$q_1 = +[(r_1 - r_2) - (r_3 - r_4)], \quad q_2 = -[(r_1 - r_2) - (r_3 - r_4)].$$

We proceed to show that $k_3 = q_1$ in every configuration of the solutions in Figure 4.1 and Figure 4.2.

(1) $p_1 > 0$, $p_4 > 0$ and $r = 4$. In this case, $f = 0$ has four simple real solutions $r_4 < r_3 < r_2 < r_1$ and $r_3 - r_4 < r_1 - r_2$. Thus $q_1 > 0$ and $q_2 < 0$, and therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Since $r > 1$ and $p_4 > 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(2) $p_1 > 0$, $p_4 = 0$ and $r = 4$. In this case, $f = 0$ has four simple real solutions $r_4 < r_3 < r_2 < r_1$ and $r_3 - r_4 = r_1 - r_2$. It follows that $q = q_1 = q_2 = 0$, and therefore

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \leq 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(3) $p_1 > 0$, $p_4 < 0$ and $r = 4$. In this case, $f = 0$ has four simple real solutions $r_4 < r_3 < r_2 < r_1$ and $r_3 - r_4 > r_1 - r_2$. Thus $q_1 < 0$ and $q_2 > 0$, and therefore

$$\arg q_1 = \pi, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \leq 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(4) $p_1 > 0$, $p_4 > 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha - i\beta$, $r_2 = \alpha + i\beta$, $r_3 = \mu + i\nu$ and $r_4 = \mu - i\nu$ such that $\alpha > \mu$, $\beta < \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = -2\,(\beta + \nu)i, \quad q_2 = +2\,(\beta + \nu)i.$$

Therefore

$$\arg q_1 = -\frac{\pi}{2}, \quad \arg q_2 = +\frac{\pi}{2}.$$

Since $q = -4\,(\beta + \nu)^2 < 0$, we have $\arg q = \pi$. Therefore $\arg\sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_2$. Since $r \leq 1$ and $p_4 > 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(5) $p_1 > 0$, $p_4 \leq 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha \geq \mu$, $\beta \geq \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = +2\,(\beta + \nu)i, \quad q_2 = -2\,(\beta + \nu)i.$$

Therefore

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,(\beta + \nu)^2 < 0$, we have $\arg q = \pi$. Therefore $\arg\sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Since $r \leq 1$ and $p_4 \leq 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(6) $p_1 = 0$, $p_2 > 0$, $q_4 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_4 and a double real solution $r_2 = r_3$ such that $r_4 < r_3 = r_2 < r_1$

and $r_3 - r_4 < r_1 - r_2$. Thus $q_1 > 0$ and $q_2 < 0$, and therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Since $r > 1$ and $p_4 > 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(7) $p_1 = 0$, $p_2 > 0$, $q_4 = 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_4 and a double real solution $r_2 = r_3$ such that $r_4 < r_3 = r_2 < r_1$ and $r_3 - r_4 = r_1 - r_2$. It follows that $q = q_1 = q_2 = 0$, and therefore

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \leq 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(8) $p_1 = 0$, $p_2 > 0$, $q_4 = 0$ and $r = 0$. In this case, $f = 0$ has two pairs of complex conjugates $r_1 = \alpha + i\beta$, $r_2 = \alpha - i\beta$, $r_3 = \mu - i\nu$ and $r_4 = \mu + i\nu$ such that $\alpha = \mu$, $\beta = \nu$, $\beta > 0$ and $\nu > 0$. Simple calculation shows that

$$q_1 = +4\,\beta i, \quad q_2 = -4\,\beta i.$$

Therefore

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -16\,\beta^2 < 0$, we have $\arg q = \pi$. Therefore $\arg\sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Since $r \leq 1$ and $p_4 \leq 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(9) $p_1 = 0$, $p_2 > 0$, $q_4 < 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_4 and a double real solution $r_2 = r_3$ such that $r_4 < r_3 = r_2 < r_1$ and $r_3 - r_4 > r_1 - r_2$. Thus $q_1 < 0$ and $q_2 > 0$, and therefore

$$\arg q_1 = \pi, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \leq 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(10) $p_1 = 0$, $p_2 = 0$, $p_4 > 0$ and $r = 2$. In this case, $f = 0$ has a simple real solution r_1 and a triple real solution $r_2 = r_3 = r_4$ such that $r_4 = r_3 = r_2 < r_1$. Thus $q_1 > 0$ and $q_2 < 0$, and therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Since $r > 1$ and $p_4 > 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(11) $p_1 = 0$, $p_2 = 0$, $p_4 = 0$ and $r = 1$. In this case, $f = 0$ has a quadruple real solution $r_4 = r_3 = r_2 = r_1$. It follows that $q_1 = q_2 = 0$ and

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_1$. Since $r \le 1$ and $p_4 \le 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(12) $p_1 = 0$, $p_2 = 0$, $p_4 < 0$ and $r = 2$. In this case, $f = 0$ has a simple real solution r_4 and a triple real solution $r_1 = r_2 = r_3$ such that $r_4 < r_3 = r_2 = r_1$. Thus $q_1 < 0$ and $q_2 > 0$, and therefore

$$\arg q_1 = \pi, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \le 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_1 = q_2$.

(13) $p_1 = 0$ and $p_2 < 0$. In this case, there are six subcases according to the number r of distinct real solutions and the sign of p_4. Now we prove that $k_3 = q_1$ in each subcase.

(a) $p_4 > 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_1, r_2 and a double real solution $r_3 = r_4$ such that $r_4 = r_3 < r_2 < r_1$. Thus $q_1 > 0$ and $q_2 < 0$, and therefore

$$\arg q_1 = 0, \quad \arg q_2 = \pi.$$

Hence $\sqrt{q} = q_1$. Since $r > 1$ and $p_4 > 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(b) $p_4 > 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_2 = r_1$ and a pair of complex conjugates $r_3 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $\alpha < r_1 = r_2$ and $\beta > 0$. Simple calculation shows that

$$q_1 = -2\,\beta i, \quad q_2 = +2\,\beta i.$$

Therefore

$$\arg q_1 = -\frac{\pi}{2}, \quad \arg q_2 = +\frac{\pi}{2}.$$

Since $q = -4\,\beta^2 < 0$, we have $\arg q = \pi$. Therefore $\arg\sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_2$. Since $r \le 1$ and $p_4 > 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(c) $p_4 = 0$ and $r = 2$. In this case, $f = 0$ has two double real solutions $r_1 = r_2$ and $r_3 = r_4$ such that $r_4 = r_3 < r_2 = r_1$. It follows that $q = q_1 = q_2 = 0$, and then

$$\arg q_1 = 0, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \le 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(d) $p_4 = 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $\alpha = r_3 = r_4$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +2\,\beta i, \quad q_2 = -2\,\beta i.$$

Therefore

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,\beta^2 < 0$, we have $\arg q = \pi$. Therefore $\arg\sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Since $r \le 1$ and $p_4 \le 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(e) $p_4 < 0$ and $r = 3$. In this case, $f = 0$ has two simple real solutions r_3, r_4 and a double real solution $r_1 = r_2$ such that $r_4 < r_3 < r_2 = r_1$. Thus $q_1 < 0$ and $q_2 > 0$, and therefore

$$\arg q_1 = \pi, \quad \arg q_2 = 0.$$

Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \leq 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$.

(f) $p_4 < 0$ and $r = 1$. In this case, $f = 0$ has a double real solution $r_3 = r_4$ and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$ such that $r_4 = r_3 < \alpha$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +2\,\beta i, \quad q_2 = -2\,\beta i.$$

Therefore

$$\arg q_1 = +\frac{\pi}{2}, \quad \arg q_2 = -\frac{\pi}{2}.$$

Since $q = -4\,\beta^2 < 0$, we have $\arg q = \pi$. Therefore $\arg\sqrt{q} = \pi/2$. Hence $\sqrt{q} = q_1$. Since $r \leq 1$ and $p_4 \leq 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(14) $p_1 < 0$, $p_4 > 0$ and $r = 2$. In this case, $f = 0$ has two real solutions r_1, r_3 and a pair of complex conjugates $r_2 = \alpha + i\beta$ and $r_4 = \alpha - i\beta$ such that $r_1 > r_3$ and $\beta > 0$. Simple calculation shows that

$$q_1 = +(r_1 - r_3) - 2\,\beta i, \quad q_2 = -(r_1 - r_3) + 2\,\beta i.$$

Therefore

$$-\frac{\pi}{2} < \arg q_1 < 0, \quad +\frac{\pi}{2} < \arg q_2 < +\pi.$$

Since $\operatorname{Im} q = -4\,\beta(r_1 - r_3) < 0$, we have $-\pi < \arg q < 0$. Therefore $-\pi/2 < \arg\sqrt{q} < 0$. Hence $\sqrt{q} = q_1$. Since $r > 1$ and $p_4 > 0$, we have $\sigma_2 = +1$ and so $k_3 = (+1)q_1 = q_1$.

(15) $p_1 < 0$, $p_4 \leq 0$ and $r = 2$. In this case, $f = 0$ has two real solutions r_2, r_4 and a pair of complex conjugates $r_1 = \alpha + i\beta$ and $r_3 = \alpha - i\beta$ such that $r_2 > r_4$ and $\beta > 0$. Simple calculation shows that

$$q_1 = -(r_2 - r_4) + 2\,\beta i, \quad q_2 = +(r_2 - r_4) - 2\,\beta i.$$

Therefore

$$+\frac{\pi}{2} < \arg q_1 < +\pi, \quad -\frac{\pi}{2} < \arg q_2 < 0.$$

Since $\operatorname{Im} q = -4\,\beta(r_2 - r_4) < 0$, we have $-\pi < \arg q < 0$. Therefore $-\pi/2 < \arg\sqrt{q} < 0$. Hence $\sqrt{q} = q_2$. Since $r > 1$ and $p_4 \leq 0$, we have $\sigma_2 = -1$ and so $k_3 = (-1)q_2 = q_1$. □

Proof of Theorem 4.1.1. Recalling the previous lemmas, substituting $k_1 = r_1 + r_2 - r_3 - r_4$, $k_2 = r_1 - r_2 + r_3 - r_4$ and $k_3 = r_1 - r_2 - r_3 + r_4$ into x_n, we see that

$$\begin{aligned}
x_1 &= \frac{(r_1 + r_2 + r_3 + r_4) + k_1 + k_2 + k_3}{4} = r_1,\\
x_2 &= \frac{(r_1 + r_2 + r_3 + r_4) + k_1 - k_2 - k_3}{4} = r_2,\\
x_3 &= \frac{(r_1 + r_2 + r_3 + r_4) - k_1 + k_2 - k_3}{4} = r_3,\\
x_4 &= \frac{(r_1 + r_2 + r_3 + r_4) - k_1 - k_2 + k_3}{4} = r_4.
\end{aligned}$$

Thus the theorem is proved. □

4.4 Formulas for Quartic Equations with Constraints

Using the real convention and quartic formula introduced in Section 4.1, we present the real solution formulas for the general real-coefficient quartic equations under equality and inequality constraints. Constraints naturally arise in applications such as geometric constraint solving [63, 131]. The representation of the real solutions coupled with real constraints are achieved by combining Thom's lemma [7, p. 50], the signs of p_1 and p_4 and the complex-solution formulas.

Theorem 4.4.1. *Let* $f(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \in \mathbb{R}[x]$ *and* $\Gamma(x)$ *be a formula composed by* $\wedge$, $\vee$, $\Rightarrow$, *and* $\neg$ *of polynomial equality and inequality relations in* x, *the coefficients of* $f(x)$, *and other parameters. Then for all* $x \in \mathbb{R}$,

$$[f(x) = 0 \wedge \Gamma(x)] \iff [x = x_1 \wedge \Gamma_1] \vee [x = x_2 \wedge \Gamma_2] \vee [x = x_3 \wedge \Gamma_3] \vee [x = x_4 \wedge \Gamma_4],$$

where

$$\begin{aligned}
\Gamma_j &:= (\exists\, x \in \mathbb{R})\, [f(x) = 0 \wedge \Gamma(x) \wedge \Phi_j(x)], \quad j = 1, 2, 3, 4,\\
\Phi_1(x) &:= [p_1 \geq 0 \wedge f'(x) \geq 0 \wedge (f''(x) \geq 0 \wedge f'''(x) \geq 0)]\\
&\quad \vee [p_1 < 0 \wedge p_4 > 0 \wedge f'(x) > 0],\\
\Phi_2(x) &:= [p_1 \geq 0 \wedge f'(x) \leq 0 \wedge (f''(x) \leq 0 \vee f'''(x) \geq 0)]\\
&\quad \vee [p_1 < 0 \wedge p_4 \leq 0 \wedge f'(x) > 0],\\
\Phi_3(x) &:= [p_1 \geq 0 \wedge f'(x) \geq 0 \wedge (f''(x) \leq 0 \vee f'''(x) \leq 0)]\\
&\quad \vee [p_1 < 0 \wedge p_4 > 0 \wedge f'(x) < 0],\\
\Phi_4(x) &:= [p_1 \geq 0 \wedge f'(x) \leq 0 \wedge (f''(x) \geq 0 \wedge f'''(x) \leq 0)]\\
&\quad \vee [p_1 < 0 \wedge p_4 \leq 0 \wedge f'(x) < 0].
\end{aligned}$$

Here x_1, x_2, x_3, x_4 *are the same as in Theorem 4.1.1 given in Section 4.1.*

Proof. Note that

$$\Gamma_j := (\exists\, z \in \mathbb{R})\, [f(z) = 0 \wedge \Gamma(z) \wedge \Phi_j(z)] \iff \bigvee_{n=1}^{4} r_n \in \mathbb{R} \wedge \Gamma(r_n) \wedge \Phi_j(r_n).$$

We need to determine $\Phi_j(r_k)$. For this, observe that

$$\begin{aligned}
f'(r_1) &= (r_1 - r_2)(r_1 - r_3)(r_1 - r_4),\\
f''(r_1) &= 2\,(r_1 - r_2)(r_1 - r_3) + 2\,(r_1 - r_2)(r_1 - r_4) + 2\,(r_1 - r_3)(r_1 - r_4),\\
f'''(r_1) &= 4\,(r_1 - r_2) + 4\,(r_1 - r_3) + 4\,(r_1 - r_4),\\
f'(r_2) &= (r_2 - r_1)(r_2 - r_3)(r_2 - r_4),\\
f''(r_2) &= 2\,(r_2 - r_3)(r_2 - r_4) + 2\,(r_2 - r_1)(r_2 - r_4) + 2\,(r_2 - r_1)(r_2 - r_3),\\
f'''(r_2) &= 4\,(r_2 - r_1) + 4\,(r_2 - r_3) + 4\,(r_2 - r_4),
\end{aligned}$$

$$\begin{aligned}
f'(r_3) &= (r_3-r_1)(r_3-r_2)(r_3-r_4),\\
f''(r_3) &= 2\,(r_3-r_2)(r_3-r_4)+2\,(r_3-r_1)(r_3-r_4)+2\,(r_3-r_1)(r_3-r_2),\\
f'''(r_3) &= 4\,(r_3-r_1)+4\,(r_3-r_2)+4\,(r_3-r_4).
\end{aligned}$$

For each configuration of the solutions, we can determine the signs of the derivatives of f at r_n, as in Table 4.1 and Table 4.2 (where the blanks are non-real, and the case marked with $*$ has three possibilities: $+$, $-$ or 0). From the signs of the derivatives, it is easy to obtain the truth values of Φ_j as in Table 4.3 and Table 4.4 (where the blanks are false).

From Table 4.3 and Table 4.4, we see immediately that

$$\Gamma_j \iff \bigvee_{k=1}^{4} r_k \in \mathbb{R} \wedge \Gamma(r_k) \wedge \Phi_j(r_k) \iff r_j \in \mathbb{R} \wedge \Gamma(r_j).$$

Let $x \in \mathbb{R}$. Then by Theorem 4.1.1, we have

$$\begin{aligned}
f(x)=0 \wedge \Gamma(x) &\iff (x=r_1 \vee x=r_2 \vee x=r_3 \vee x=r_4) \wedge \Gamma(x)\\
&\iff [x=r_1 \wedge r_1 \in \mathbb{R} \wedge \Gamma(r_1)] \vee\\
&\qquad [x=r_2 \wedge r_2 \in \mathbb{R} \wedge \Gamma(r_2)] \vee\\
&\qquad [x=r_3 \wedge r_3 \in \mathbb{R} \wedge \Gamma(r_3)] \vee\\
&\qquad [x=r_4 \wedge r_4 \in \mathbb{R} \wedge \Gamma(r_4)]\\
&\iff [x=x_1 \wedge \Gamma_1] \vee [x=x_2 \wedge \Gamma_2] \vee [x=x_3 \wedge \Gamma_3] \vee [x=x_4 \wedge \Gamma_4].
\end{aligned}$$

The theorem is proved. □

Table 4.1: Signs of Derivatives of Quartic Polynomial f (I)

p_1	+	+	+			+	+	+	+	+	+			−	−	−	−	−	−
p_2	+	0	−			+	0	−	+	0	−			+	0	−	+	0	−
p_4	−	−	−			0	0	0	+	+	+			−	−	−	0	0	0
$f'(r_1)$	+	+	+			+	+	+	+	+	+								
$f''(r_1)$	+	+	+			+	+	+	+	+	+								
$f'''(r_1)$	+	+	+			+	+	+	+	+	+								
$f'(r_2)$	−	−	−			−	−	−	−	−	−	−	−	+	+	+	+	+	+
$f''(r_2)$	−	−	$*$			−	−	$*$	−	−	+	0	−	+	+	+	+	+	+
$f'''(r_2)$	+	+	+			+	+	+	$*$	$*$	+	+	$*$	+	+	+	+	+	+
$f'(r_3)$	+	+	+	+	+	+	+	+	+	+	+								
$f''(r_3)$	−	−	+	0	−	−	−	$*$	−	−	$*$								
$f'''(r_3)$	$*$	$*$	−	−	$*$	−	−	−	−	−	−								
$f'(r_4)$	−	−	−			−	−	−	−	−	−			−	−	−	−	−	−
$f''(r_4)$	+	+	+			+	+	+	+	+	+			+	+	$*$	+	+	+
$f'''(r_4)$	−	−	−			−	−	−	−	−	−			−	−	$*$	−	−	−

Table 4.2: Signs of Derivatives of Quartic Polynomial f (II)

p_1	−	−	−	0	0	0				0	0	0		0	0	0			
p_2	+	0	−	+	0	−				+	0	−		+	0	−			
p_4	+	+	+	−	−	−				0	0	0		+	+	+			
$f'(r_1)$	+	+	+	+	0	0				+	0	0		+	+	+	0		
$f''(r_1)$	+	+	$*$	+	0	+				+	0	+		+	+	+	+		
$f'''(r_1)$	+	+	$*$	+	+	+				+	0	+		+	+	+	+		
$f'(r_2)$				0	0	0				0	0	0		0	0	−	−	−	0
$f''(r_2)$				−	0	+				−	0	+		−	0	+	0	−	+
$f'''(r_2)$				−	+	+				0	0	+		+	−	+	0	−	+
$f'(r_3)$	−	−	−	0	0	+	+	+	0	0	0	0	0	0	0	0			
$f''(r_3)$	+	+	+	−	0	+	0	−	+	−	0	+	+	−	0	+			
$f'''(r_3)$	−	−	−	−	+	−	−	$*$	−	0	0	−	0	+	−	−			
$f'(r_4)$				−	−	−	0			−	0	0	0	−	0	0			
$f''(r_4)$				+	+	+	+			+	0	+	+	+	0	+			
$f'''(r_4)$				−	−	−	−			−	0	−	0	−	−	−			

Table 4.3: Truth Values of Φ_j of Solutions of Quartic Equation (I)

p_1	+	+	+	+	+	+	+	+	+	−	−	−	−	−	−
p_2	+	0	−	+	0	−	+	0	−	+	0	−	+	0	−
p_4	−	−	−	0	0	0	+	+	+	−	−	−	0	0	0
$\Phi_1(r_1)$	true	true	true	true	true	true	true	true	true						
$\Phi_1(r_2)$															
$\Phi_1(r_3)$															
$\Phi_1(r_4)$															
$\Phi_2(r_1)$															
$\Phi_2(r_2)$	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true
$\Phi_2(r_3)$															
$\Phi_2(r_4)$															
$\Phi_3(r_1)$															
$\Phi_3(r_2)$															
$\Phi_3(r_3)$	true	true	true	true	true	true	true	true	true						
$\Phi_3(r_4)$															
$\Phi_4(r_1)$															
$\Phi_4(r_2)$															
$\Phi_4(r_3)$															
$\Phi_4(r_4)$	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true

Table 4.4: Truth Values of Φ_j of Solutions of Quartic Equation (II)

p_1	−	−	−	0	0	0	0	0	0	0	0	0
p_2	+	0	−	+	0	−	+	0	−	+	0	−
p_4	+	+	+	−	−	−	0	0	0	+	+	+
$\Phi_1(r_1)$	true	true	true	true	true	true	true	true	true	true	true	true
$\Phi_1(r_2)$					true	true		true	true			
$\Phi_1(r_3)$					true			true				
$\Phi_1(r_4)$								true				
$\Phi_2(r_1)$					true	true		true	true			
$\Phi_2(r_2)$				true	true	true	true	true	true	true	true	true
$\Phi_2(r_3)$				true	true		true	true		true	true	
$\Phi_2(r_4)$								true			true	
$\Phi_3(r_1)$					true			true				
$\Phi_3(r_2)$				true	true		true	true		true	true	
$\Phi_3(r_3)$	true	true	true	true	true	true	true	true	true	true	true	true
$\Phi_3(r_4)$								true	true		true	true
$\Phi_4(r_1)$								true				
$\Phi_4(r_2)$								true			true	
$\Phi_4(r_3)$								true	true		true	true
$\Phi_4(r_4)$				true	true	true	true	true	true	true	true	true

Chapter 5

Generating Dynamic Diagrams with Inequality Constraints

The approach of solving geometric constraints involving inequalities proposed by Hong and others uses triangular decomposition, solution formulas, and quantifier elimination. In this chapter, we recall the HLLW approach and show that for generating dynamic diagrams automatically the performance of this approach can be enhanced, in terms of stability of numeric computation and quality of generated diagrams, when the used solution formulas of cubic equations are replaced by the real solution formulas with equality and inequality constraints introduced in Chapter 3, and also in terms of adding the quartic case introduced in Chapter 4. Several examples are presented to illustrate the enhanced approach and demonstrate the advantages and effectiveness of the new solution formulas. Most of the material in this chapter is taken from the joint paper [155] with Aubry, Hong and Wang.

5.1 Introduction and Motivation

Dynamic geometric constraint solving is required in modern geometric engineering, in particular for dynamically producing diagrams of given geometric objects satisfying given geometric constraint relations. It has been studied extensively in the area of computer aided geometric design and modeling (see the recent survey [66] and references therein), which leads to several approaches based on graph analysis, algebraic computation, and logic reasoning. These approaches have been developed mainly for solving geometric constraints that may be expressed algebraically as equalities. There is little work on solving geometric constraints involving inequalities as well as equalities. This may be mainly due to the well known inherent practical difficulties of dealing with general inequality constraints. There are methods (such as cylindrical algebraic decomposition and others) that can handle, in principle, arbitrarily general equality/inequality constraints, but the practical complexity is prohibitive for even moderate size of problems. Therefore it is imperative to develop effective methods that can directly handle geometric constraints involving inequalities.

The problem of solving real geometric constraints involving inequalities was addressed first by Hong and his coauthors in [63]. The proposed approach allows one to handle geometric constraints involving order relations such as "inside," "external," and "between." We follow this approach (referred to as the HLLW approach) and place our emphasis on dealing with inequality constraints and on automating the entire process of dynamic diagram generation (as done in GEOTHER [131]).

The HLLW approach is applicable for solving such geometric constraint problems that may be transformed into triangular semi-algebraic systems of equalities of degree less than 5 and inequalities of arbitrary degree with parameters. The degree restriction is posed because of its use of solution formulas of equations by radicals. For any constraint problem expressed as a semi-algebraic system, the approach works by first decomposing the set of equality constraints into finitely many irreducible triangular sets. Then, for each triangular set with inequality constraints, the space of parameters is decomposed into finitely many domains by means of real quantifier elimination, such that associated with each domain there is a set of explicit expressions of the dependent variables in terms of the parameters (and the previous dependent variables) with radicals. In this way, the semi-algebraic system is decomposed into finitely many (weak) *solution representations by radicals* (or SRRs for short, see [63, p. 184]) such that the set of real solutions of the system is equal to the union of the sets of real solutions given by the (weak) SRRs. For any given values of the parameters, if they satisfy the parameter constraints in some (weak) SRRs, then the values of the dependent variables may be easily computed by direct evaluation of the corresponding explicit expressions.

In the following, we enhance the HLLW approach by replacing the Ferro–Cardano-type solution formulas of cubic equations used there with newly introduced Lagrange-type real solution formulas with inequality constraints and by incorporating new real solution formulas of quartic equations with inequality constraints. We will recall the HLLW approach and enhance it with the new real solution formulas involving no division by small numbers in Section 5.2 and Section 5.3. The process of dynamic diagram generation using the enhanced HLLW approach is sketched and illustrated by an example in Section 5.4. We present the results of our studies on three geometric configurations, the configuration of Apollonius circles and the dynamic diagrams for Feuerbachs and Thébault–Taylors theorems, in Section 5.5 to demonstrate the advantages and effectiveness of the new solution formulas and the enhanced approach.

5.2 Representing Real Solutions of Semi-algebraic Systems by Radicals

Consider the following parametric semi-algebraic system of equations and inequalities

$$\begin{cases} F_1(x_1, \ldots, x_n) = 0, \\ \qquad \cdots\cdots \\ F_s(x_1, \ldots, x_n) = 0, \\ G_1(x_1, \ldots, x_n) \lessgtr 0, \\ \qquad \cdots\cdots \\ G_t(x_1, \ldots, x_n) \lessgtr 0, \end{cases} \tag{5.1}$$

where $F_1, \ldots, F_s, G_1, \ldots, G_t$ are polynomials in $x_1, \ldots, x_n$ with rational coefficients and $\lessgtr$ may take any of the inequality operators $<$, $\leq$, $>$, $\geq$, $\neq$. We wish to represent the real solutions of (5.1) by means of explicit formulae with radicals. This is not possible in general (as it is well known from Abel/Galois theory that the solutions of polynomial equations of degree greater than 4 in general cannot be expressed in terms of radicals and field operations), so our objective is to compute such representations for polynomials of low degree. Once such representations are available, they can be efficiently instantiated repeatedly for dynamic update of diagrams.

Let $u = (u_1, \ldots, u_d)$ be a subset of the variables $x_1, \ldots, x_n$. Denote by $y_1, \ldots, y_r$ all the other variables x_i not in u. Note that $u_1, \ldots, u_d, y_1, \ldots, y_r$ is a permutation of $x_1, \ldots, x_n$ (so $d + r = n$ and $\{u_1, \ldots, u_d, y_1, \ldots, y_r\} = \{x_1, \ldots, x_n\}$). We call u parameters (or parametric variables) and $y_1, \ldots, y_r$ dependents (or dependent variables).

Definition 5.2.1. Let $\Gamma(u)$ be a quantifier-free formula composed of equality and inequality relations in the parameters u, and h_j a rational expressions of u with radicals for $1 \leq j \leq r$. We call

$$\Gamma(u),\ y_1 = h_1(u),\ y_2 = h_2(u),\ \ldots,\ y_r = h_r(u) \tag{5.2}$$

a *solution representation by radicals (SRR)* in u.

We want to decompose the semi-algebraic system (5.1) into finitely many SRRs of the form (5.2) such that the set of real solutions of (5.1) is equal to the union of the sets of real solutions given by the SRRs.

To compute SRRs, we first decompose the set of polynomials $F_1, \ldots, F_s$ into (irreducible) triangular sets by using the method of characteristic sets or other methods [128, 130]. Each triangular set $\mathbb{T}$ may be written in the form

$$\mathbb{T} = [T_1(u, y_1), T_2(u, y_1, y_2), \ldots, T_r(u, y_1, \ldots, y_r)], \tag{5.3}$$

where $u, y_1, \ldots, y_r$ is a permutation of $x_1, \ldots, x_n$ as above. Then the problem is reduced to considering the following set of constraints for every triangular set

$$T_1(u, y_1) = 0, \ \ldots, \ T_r(u, y_1, \ldots, y_r) = 0, \ \Gamma_r(u, y_1, \ldots, y_r), \tag{5.4}$$

where $\Gamma_r(u, y_1, \ldots, y_r) := G_1 \lessgtr 0, \ldots, G_t \lessgtr 0, I_1 \neq 0, \ldots, I_r \neq 0$ and where I_i is the initial of T_i.

For any given value $\bar{u}$ of u, one can solve the equations $T_1 = 0, \ldots, T_r = 0$ successively for $y_1, \ldots, y_r$ and then verify which solutions satisfy the formula Γ_r in the *t*riangular representation (5.4). This simple approach works theoretically but has two drawbacks for finding real solutions. First, it is possible that real solutions are found for $y_1, \ldots, y_k$ $(1 \leq k < r)$ but there is no real solution for y_{k+1}. In this case, the computation of the real solutions for $y_1, \ldots, y_k$ is waste. Second, if a found real solution of $T_1 = 0, \ldots, T_r = 0$ does not satisfy Γ_r, then the computation of this solution is also waste. How to avoid or reduce such waste? In what follows we explain how to do so by eliminating the variables $y_r, \ldots, y_1$ successively from the inequality constraints in Γ_r using $T_r = 0, \ldots, T_1 = 0$ respectively.

Consider first the constraints

$$T_r(u, y_1, \ldots, y_r) = 0, \ \Gamma_r(u, y_1, \ldots, y_r), \tag{5.5}$$

In the next section, we will show how to decompose (5.5) into finitely many explicit root representations of the form

$$y_r = h_r^{(j)}(u, y_1, \ldots, y_{r-1}), \ \Gamma_{r-1}^{(j)}, \tag{5.6}$$

such that $\Gamma_{r-1}^{(j)}$ does not contain the variable y_r. Then we can deal with the constraints

$$T_{r-1}(u, y_1, \ldots, y_{r-1}) = 0, \ \Gamma_{r-1}^{(j)}, \tag{5.7}$$

similarly for each j. Continuing this way, we will be able to decompose (5.4) into finitely many explicit root representations

$$\Gamma_0^{(i)}(u), \ y_1 = h_1^{(i)}(u), y_2 = h_2^{(i)}(u, y_1), \ldots, y_r = h_r^{(i)}(u, y_1, \ldots, y_{r-1}), \tag{5.8}$$

with each $\Gamma_0^{(i)}(u)$ a conjunction of disjunctions of equality and inequality relations in u and each $h_j^{(i)}$ a rational expression of $u, y_1, \ldots, y_{j-1}$ with radicals. In other words, the space of parameters u is decomposed into finitely many domains D_i defined by $\Gamma_0^{(i)}(u)$, such that for any given values $\bar{u}$ of u, if $\bar{u} \in D_i$, then the values of the dependent variables $y_1, \ldots, y_r$ are

$$\bar{y_1} = h_1^{(i)}(\bar{u}), \bar{y_2} = h_2^{(i)}(\bar{u}, \bar{y_1}), \ldots, \bar{y_r} = h_r^{(i)}(\bar{u}, \bar{y_1}, \ldots, \bar{y_{r-1}}).$$

Note that a domain D_i may be disconnected and two domains may be joined.

We may substitute

$$y_1 = h_1^{(i)}(u), \ldots, y_j = h_j^{(i)}(u, y_1, \ldots, y_{j-1}), \tag{5.9}$$

into $y_{j+1} = h_{j+1}^{(i)}(u, y_1, \ldots, y_j)$, so that (5.8) become SRRs. In practice, we may wish to keep the form (5.8) because the substitution of (5.9) into $h_{j+1}^{(i)}$ may increase the size of the expression considerably. We call the representation (5.8) a *w*eak SRR. The method of decomposing a triangular representation of the form (5.4) into SRRs uses root formulae (as long as the degree of each T_i in y_i is less than 5) and real quantifier elimination [32]. From this method, we can easily construct a decomposition of (5.1) into SRRs as desired.

We say that a polynomial $P(z)$ is composed of polynomials $P_1(z), \ldots, P_k(z)$ if $P(z) = P_1(P_2(\cdots P_k(z) \cdots))$. The method sketched above allows us to establish the following main result.

Theorem 5.2.2. *For any semi-algebraic system (5.1) in $x_1, \ldots, x_n$, if the polynomials in the triangular sets obtained from all the irreducible triangular decompositions of the involved sets of polynomial equations are composed of polynomials of degree less* *than 5 with respect to their leading variables, then one can decompose (5.1) into q SRRs*

$$\Gamma^{(i)}(u^{(i)}),\ y_1^{(i)} = h_1^{(i)}(u^{(i)}),\, y_2^{(i)} = h_2^{(i)}(u^{(i)}),\, \ldots,\, y_{r_i}^{(i)} = h_{r_i}^{(i)}(u^{(i)})$$

$(1 \leq i \leq q)$ *such that the set of real solutions of (5.1) is equal to*

$$\bigcup_{i=1}^{q}\{(\bar{u}^{(i)}, h_1^{(i)}(\bar{u}^{(i)}),\, \ldots,\, h_{r_i}^{(i)}(\bar{u}^{(i)}))\,|\,\Gamma^{(i)}(\bar{u}^{(i)})\},$$

where $u^{(i)}, y_1^{(i)}, \ldots, y_{r_i}^{(i)}$ *is a permutation of* $x_1, \ldots, x_n$ *for each* i *and the union of zero sets is carried out after permuting* $u^{(i)}, y_1^{(i)}, \ldots, y_{r_i}^{(i)}$ *back to* $x_1, \ldots, x_n$.

The method explained above and indicated in Theorem 5.2.2 has both theoretical and practical interests because the explicit SRRs computed may provide an efficient way for the computation of real solutions of (5.1). We will demonstrate how it can be applied effectively to solving dynamic geometric constraints involving inequalities in Section 5.4. For dynamic animation, the computation of real solutions has to be performed in real time and thus should be kept as inexpensive as possible, while more expensive symbolic precomputation is acceptable.

5.3 Solution Formulas of Univariate Equations with Inequality Constraints

In this section, we explain how to eliminate the variable y_r from the inequality constraints in $\Gamma_r(u, y_1, \ldots, y_{r-1}, y_r)$ as in (5.4) by using $T_r(u, y_1, \ldots, y_{r-1}, y_r) = 0$ as in (5.3), when the degree of T_r in y_r is small. For notational convenience, we will write x for y_r, $f(x)$ for $T_r(u, y_1, y_1, \ldots, y_{r-1}, y_r)$, and $\Gamma(x)$ for $\Gamma_r(u, y_1, \ldots, y_{r-1}, y_r)$. The linear and quadratic cases are the same as in [63].

Theorem 5.3.1 (Linear Case)**.** *Let* $f(x) = x + a_0 \in \mathbb{R}[x]$ *and* $\Gamma(x)$ *be a formula composed by* $\wedge$, $\vee$, $\Rightarrow$, *and* $\neg$ *of polynomial equality and inequality relations in* x, *the coefficients of* $f(x)$, *and other parameters. Then for all* $x \in \mathbb{R}$,

$$[f(x) = 0 \wedge \Gamma(x)] \iff [x = -a_0 \wedge \Gamma(-a_0)].$$

Proof. $f(x) = 0$ is equivalent to $x = -a_0$. The proof immediately follows. □

Theorem 5.3.2 (Quadratic Case)**.** *Let* $f(x) = x^2 + a_1 x + a_0 \in \mathbb{R}[x]$ *and* $\Gamma(x)$ *be a formula composed by* $\wedge$, $\vee$, $\Rightarrow$, *and* $\neg$ *of polynomial equality and inequality relations in* x, *the coefficients of* $f(x)$, *and other parameters. Then for all* $x \in \mathbb{R}$,

$$[f(x) = 0 \wedge \Gamma(x)] \iff [x = -a_1 + \sqrt{p_1} \wedge \Gamma^+] \vee [x = -a_1 - \sqrt{p_1} \wedge \Gamma^-],$$

where

$$p_1 = a_1^2 - 4\,a_0,$$

and

$$\Gamma^+ := (\exists x)[\Gamma(x) \wedge f(x) = 0 \wedge f'(x) \geq 0],$$
$$\Gamma^- := (\exists x)[\Gamma(x) \wedge f(x) = 0 \wedge f'(x) \leq 0].$$

Proof. $f(x) = 0$ is equivalent to

$$x = -a_1 + \sqrt{p_1} \vee x = -a_1 - \sqrt{p_1}.$$

Now we make the key observation

$$x = -a_1 + \sqrt{p_1} \iff f(x) = 0 \wedge f'(x) \geq 0,$$
$$x = -a_1 - \sqrt{p_1} \iff f(x) = 0 \wedge f'(x) \leq 0.$$

The proof immediately follows. □

The real solution formulas with constraints for the cubic equations and quartic equations were presented in Theorem 3.5.1 and Theorem 4.4.1 respectively.

The real solution formulas for cubic and quartic equations avoid division by small numbers and thus the numerically unstable case near "0/0" (i.e., both the numerator and the denominator are close to zero). On the contrary, the conventional solution formulas used in [63] for $f(x) = x^3 + a_2 x^2 + a_1 x + a_0$ involve quotients of the form $(3\,a_1 - a_2^2)/\gamma$, where γ is the third root of $36\,a_2 a_1 - 108\,a_0 - 8\,a_2^3 + 12\sqrt{-3\,p_1}$ and p_1 is the discriminant of f (as in Theorem 3.5.1). It may be easily verified that, when a_1, a_2 and a_3 contain a common factor depending on the parameters, this expression leads to the near "0/0" case for some values of the parameters.

For geometric constraint solving as well as other applications, one needs to solve polynomial equations with gradually changing coefficients, for which a solution formula with divisions may encounter near "0/0" and thus may result in significant numeric errors. Therefore, the solution formulas given in Theorems 3.5.1 and 4.4.1 above are computationally better in terms of numeric stability and geometric continuity for dynamic diagram animation. The example of Apollonius circles in Section 5.5 will illustrate the practical advantages of using these new formulas over the previous ones.

The formulas in Theorems 3.5.1 and 4.4.1 are presented for normalized f (which has leading coefficient 1). The polynomials in triangular sets may have initials (i.e., leading coefficients with respect to leading variables) which are not constants but depend on the parameters. Such initials may appear in the denominators of the solution formulas. They cannot be zero because in the process of triangular decomposition, inequation constraints are added to rule out the case in which the initials vanish. However, the appearance of non-constant initials in the denominators of the formulas cannot be avoided and may cause problems when the initials are very close to zero. This can also be seen from the example of Apollonius circles.

In order to use the above theorems, we need to eliminate the existential quantifier from the formula. This can be done by using any real quantifier elimination procedure (such as QEPCAD [15, 32], REDLOG [40, 136], QEQUAD [60], or SturmHabicht [53, 54]).

5.4 Generation of Dynamic Diagrams

Now we present the main algorithmic steps for automatically generating dynamic diagrams with inequality constraints based on the enhanced HLLW approach and give an example to illustrate the approach used in the process of diagram generation.

Let $\mathcal{O}$ be a set of geometric objects in a geometric space (e.g., Euclidean plane or space) and $\mathcal{C}$ a set of geometric constraints among the objects in $\mathcal{O}$. The problem of dynamic diagram generation is to decide whether the objects in $\mathcal{O}$ can be placed in the space such that the constraints in $\mathcal{C}$ are all satisfied; if so, construct, for any given assignment of allowable values to the parameters, one or several static diagrams that satisfy the constraints. We use inequality constraints to describe order relations of the objects such as "between," "internal," and "outside" and use inequality or inequation constraints to rule out degenerate cases in which diagrams cannot be properly constructed. For dynamic animation, parametric values are given successively with small changes, static diagrams are constructed accordingly, and their motion may then be shown on screen.

Denote by $x_1, ..., x_n$ the coordinates of points and other geometric entities involved in the objects of $\mathcal{O}$; then the set of $\mathcal{C}$ of constraints together with the conditions to exclude some degenerate cases may be expressed as a semi-algebraic system. The geometric constraint solving problem then reduced to solve this semi-algebraic system.

Sometimes, constraints may be specified as quantified formulae. In this case, we may eliminate the quantifiers to obtain quantifier-free formulae using known methods such as PCAD (partial cylindrical algebraic decomposition) [32]. So we can assume that the geometric constraint solving problem under consideration may be formulated algebraically in the form (5.1).

By the method presented in the previous two sections, we can decompose the semi-algebraic system (5.1) into finitely many weak SRRs

$$\begin{aligned} \Gamma^{(i)}(u^{(i)}), \quad y_1^{(i)} &= h_1^{(i)}(u^{(i)}), \\ y_2^{(i)} &= h_2^{(i)}(u^{(i)}, y_1^{(i)}), \\ &\cdots \\ y_{r_i}^{(i)} &= h_{r_i}^{(i)}(u^{(i)}, y_1^{(i)}, \ldots, y_{r-1}^{(i)}), \end{aligned} \tag{5.10}$$

where $u^{(i)}, y_1^{(i)}, \ldots, y_{r_i}^{(i)}$ is a permutation of $x_1, \ldots, x_n$ for each i. System (5.1) has a real solution if and only if there exist an i and a set $\bar{u}^{(i)}$ of real values of $u^{(i)}$ such that $\Gamma^{(i)}(\bar{u}^{(i)})$ holds true. In case there is no parameter, we will either end up with the conclusion that (5.1) has no real solution, or find the radical expressions of all the real solutions, yielding finitely many diagrams.

If there are infinitely many real values $\bar{u}$ of u such that $\Gamma_i(\bar{u}_i)$ holds, then the diagrams are dynamic. In this case, from each weak SRR and the identification of parameters and dependents we can determine which points in the geometric objects are free, semi-free, or dependent points. If the geometric constraint solving problem is well formulated, then all the SRRs should have the same set of parameters. We may assume that this is the case.

To generate a dynamic diagram, we implement the weak SRRs into the drawing program. For initialization, a random set $\bar{u}$ of real values for u is chosen so that some $\Gamma_i(\bar{u}_i)$ holds. Then the real values of u (corresponding to the free or semi-free points) change continuously by the user, for example, using mouse dragging. For any chosen values $\bar{u}$ of u, the program verifies whether some $\Gamma_i(\bar{u}_i)$ holds. If no such Γ_i exists, then the values $\bar{u}$ are not allowed and the diagram remains unmoved. Otherwise, $\Gamma_i(\bar{u}_i)$ holds true for some i. In the case, the corresponding values for the dependent variables

$$\bar{y}_1^{(i)} = h_1^{(i)}(\bar{u}^{(i)}), \bar{y}_2^{(i)} = h_2^{(i)}(\bar{u}^{(i)}, \bar{y}_1^{(i)}), \ldots, \bar{y}_{r_i}^{(i)} = h_{r_i}^{(i)}(\bar{u}^{(i)}, \bar{y}_1^{(i)}, \ldots, \bar{y}_{r-1}^{(i)})$$

are computed. The diagram is then redrawn according to these values that determine the new locations of the geometric objects in the diagram.

Using the HLLW approach enhanced with the real solution formulas in Theorems 3.5.1 and 4.4.1 to solve parametric semi-algebraic systems, the process for automated generation of dynamic diagrams involving inequality constraints consists of the following steps (see [63] and [129]).

1. Assign coordinates to the points involved in the geometric objects and introduced other variables if necessary; so that the geometric constraints are expressed as a semi-algebraic system of equalities and inequalities of the form in (5.1).

2. Decompose the system (5.1) into finitely many weak SRRs using the enhanced HLLW approach.

3. Determine the free and semi-free points according to the identification of the variables $x_1, ..., x_n$ into parameters and dependents.

4. Randomly choose a set of real numerical values for the parametric variables satisfying the constraints and compute the values of the dependent variables from the corresponding weak SRRs.

5. Check whether all the points with the current values of coordinates are within the given window range and no two of them are too close. If not, then go back to Step 4.

6. Draw the geometric objects and label the points with the current values of variables in the window(s) to obtain one or several static diagrams.

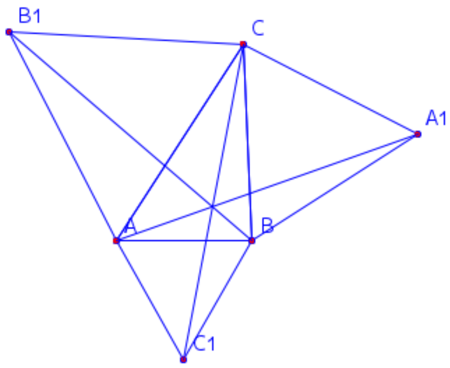

Figure 5.1: Steiner Configuration

The animation of the drawn diagram may be implemented by the following two additional steps.

7. Update the values of the free coordinates of the free or semi-free points being moved with mouse dragging and recompute the values of the dependent variables according to the corresponding weak SRRs.
8. Redraw the geometric objects and relabel the points with the updated values of variables in the window of the static diagram.

The following example illustrates the enhanced HLLW approach used in the process of diagram generation with our preliminary implementation.

Example 5.4.1 (Steiner Theorem)**.** Given an arbitrary triangle ABC, draw three equilateral triangle ABC_1, ACB_1, and BCA_1 all outward or all inward. The well-known Steiner theorem claims that the three lines AA_1, BB_1, and CC_1 are concurrent.

To ensure that the triangles are drawn all outward, we need to use inequalities. Without loss if generality, let the coordinates of the points be assigned as follows:[1]

$$A(0,0),\ B(1,0),\ C(u_1,u_2),\ C_1(x_1,x_2),\ B_1(x_3,x_4),\ A_1(x_5,x_6).$$

Then the geometric constraints may be expressed as the following equalities and

[1]Note that coordinates may also be assigned automatically, see, e.g., the function `Coordinate` in GEOTHER [131].

inequalities:

$$\begin{cases} F_1 = 2, x_1 - 1 = 0, & (|AC_1| = |BC_1|) \\ F_2 = x_2^2 + x_1^2 - 1 = 0, & (|AC_1| = |AB|) \\ F_3 = x_4^2 + x_3^2 - u_2^2 - u_1^2 = 0, & (|AB_1| = |AC|) \\ F_4 = 2\,u_2x_4 + 2\,u_1x_3 - u_2^2 - u_1^2 = 0, & (|AB_1| = |CB_1|) \\ F_5 = x_6^2 + x_5^2 - 2\,x_5 - u_2^2 - u_1^2 + 2\,u_1 = 0, & (|BA_1| = |BC|) \\ F_6 = 2\,u_2x_6 + 2\,u_1x_5 - 2\,x_5 - u_2^2 - u_1^2 + 1 = 0, & (|BA_1| = |CA_1|) \\ G_1 = u_1u_2x_6 - u_2x_6 - u_2^2x_5 + u_2^2 < 0, & (BCA_1 \text{ outward}) \\ G_2 = -u_2u_1x_4 + u_2^2x_3 < 0, & (ACB_1 \text{ outward}) \\ G_3 = u_2x_2 < 0. & (ABC_1 \text{ outward}) \end{cases}$$

Assume that C is a free point, so that u_1, u_2 are free parameters. The set of polynomials $\{F_1, \ldots, F_6\}$ may be decomposed over $\mathbb{Q}(u_1, u_2)$ into four triangular sets. However, three of them turn out to be inconsistent with the inequality constraints. So we get only one triangular set $[T_1, \ldots, T_6]$

$$T_1 = 2\,x_1 - 1, \quad T_2 = 4\,x_2^2 - 3, \quad T_3 = 2\,x_3 - 2\,u_2x_2 + u_1,$$
$$T_4 = 2\,x_4 - 2\,u_1x_2 - u_2, \quad T_5 = 2\,x_5 - 2\,u_2x_2 - u_1 - 1,$$
$$T_6 = 2\,x_6 - 2\,u_1x_2 + 2\,x_2 + u_2.$$

From the triangular set, we can obtain the following weal SRRs:

$$\begin{aligned} u_2 < 0, \quad & x_1 = \frac{1}{2},\ x_2 = \frac{\sqrt{3}}{2},\ x_3 = \frac{1}{2}\,u_1 + u_2x_2,\ x_4 = \frac{1}{2}\,u_2 - u_1x_2, \\ & x_5 = \frac{1}{2}\,u_1 - u_2x_2 + \frac{1}{2},\ x_6 = \frac{1}{2}\,u_2 - x_2 + u_1x_2; \\ u_2 > 0, \quad & x_1 = \frac{1}{2},\ x_2 = -\frac{\sqrt{3}}{2},\ x_3 = \frac{1}{2}\,u_1 + u_2x_2,\ x_4 = \frac{1}{2}\,u_2 - u_1x_2, \\ & x_5 = \frac{1}{2}\,u_1 - u_2x_2 + \frac{1}{2},\ x_6 = \frac{1}{2}\,u_2 - x_2 + u_1x_2. \end{aligned}$$

With these representations, the dynamic diagram as shown in Figure 5.1 can be drawn and animated efficiently.

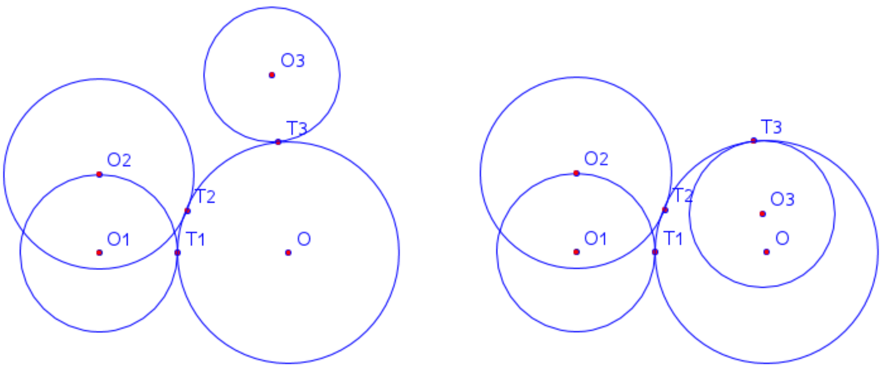

Figure 5.2: Apollonius Circles

5.5 Examples of Geometric Configurations

In this section, we report our studies on three well-known geometric configurations, the configuration of Apollonius circles and the dynamic diagrams for Feuerbach's and Thébault-Taylor's theorems. These configurations are used as examples to show the advantages of the new real solution formulas of cubic and quartic equations with inequality constraints and to demonstrate the effectiveness of the enhanced HLLW approach. The configurations are formulated unambiguously with inequality constraints and thus cannot be handled by other existing methods and software packages designed for solving constraints involving equalities only.

5.5.1 Apollonius Circles

The problem is to construct circles that are simultaneously tangent to three given circles and it has applications in geometric modeling, biochemistry, and pharmacology. The construction and drawing of Apollonius circles have been studied extensively (see, e.g., [50, 63, 71, 87]). However, most of the studies of the problem are based on algebraic formulations using equalities only, so internal contact and external contact of circles cannot be clearly distinguished. For many applications, what is of real interest is actually the case of external contact. The use of inequality constraints allows one to exclude the case of internal contact of circles.

The configuration of Apollonius circles has already been studied in [63] using the HLLW approach. Here we revisit the configuration in order to compare some of the solution representations in the SRRs computed using the Ferro–Cardano-type solution formulas and the new Lagrange-type solution formulas. For this purpose, let us use the same assignment of coordinates and the same algebraic formulation as in [63].

If the coordinates are chosen as

$$O_1(0,0),\ T_1(1,0),\ O_2(0,1),\ T_2(u_1,x_2),\ O_3(u_3,x_4),\ T_3(u_2,x_3),\ O(x_1,0),$$

then the constraints for the four circles to tangent externally may be expressed algebraically as

$$\begin{cases} F_1 = x_1x_2 - x_1 + u_1 = 0, & (O,\ O_2,\ T_2 \text{ collinear})\\ F_2 = -x_2^2 + 2\,u_1x_1 - 2\,x_1 + u_2^2 + 1 = 0, & (T_1,\ T_2 \text{ on } O)\\ F_3 = -x_3^2 + 2\,u_2x_1 - 2\,x_1 - u_2^2 + 1 = 0, & (T_1,\ T_3 \text{ on } O)\\ F_4 = -x_1x_4 + u_2x_4 - x_1x_3 - u_3x_3 = 0, & (O,\ O_2,\ T_2 \text{ collinear})\\ G_1 = (\exists\lambda_1)[\lambda_1 > 0 \wedge \lambda_1 < 1 \wedge \lambda_1x_1 = 1], & (O,\ O_1 \text{ ex-tangent})\\ G_2 = (\exists\lambda_2)[\lambda_2 > 0 \wedge \lambda_2 < 1 \wedge \lambda_2x_1 = u_1 & (O,\ O_2 \text{ ex-tangent})\\ \qquad \wedge -\lambda_2 + 1 = x_2], & \\ G_3 = (\exists\lambda_3)[\lambda_3 > 0 \wedge \lambda_3 < 1 \wedge (1-\lambda_3)x_4 = x_3 & (O,\ O_3 \text{ ex-tangent})\\ \qquad \wedge \lambda_3x_1 + (1-\lambda_3)u_3 = u_2]. & \end{cases}$$

First, we decompose the quantified variables $\lambda_1,\ \lambda_2,\ \lambda_3$ to get a semi-algebraic system and then compute an irreducible triangular decomposition of $\{F_1, ..., F_4\}$ over $\mathbb{Q}(u_1,u_2,u_3)$ with respect to variable order $x_1 \prec \cdots \prec x_4$. The decomposition consist of only one triangluar set $[T_1, ...T_4]$ with

$$\begin{aligned} T_1 &= -2\,u_1x_1^3 + 2\,x_1^3 + u_2^2x_1^2 - 2\,u_1x_1 + u_1^2,\\ T_2 &= x_1x_2 - x_1 + u_1,\\ T_3 &= x_3^2 - 2\,u_2x_1 - 2\,x_1 + u_2^2 - 1,\\ T_4 &= x_1x_4 - u_2x_4 - x_1x_3 + u_3x_3. \end{aligned}$$

Following the HLLW approach as in [63] but using the new solution formulas instead of the old ones, we can compute six weak SRRs, of which two are listed as follows:

$$\begin{aligned} &\Delta_1(u_1,u_2,u_3),\ x_1 = X_{11},\ x_2 = X_2,\ x_3 = X_3,\ x_4 = X_4;\\ &\Delta_2(u_1,u_2,u_3),\ x_1 = X_{12},\ x_2 = X_2,\ x_3 = X_3,\ x_4 = X_4, \end{aligned}$$

where

$$\Delta_1(u_1,u_2,u_3):\quad u_1-1>0\wedge u_1^3-12\,u_1+12>0\wedge u_1^4+36\,u_1^2-90\,u_1+54>0 \wedge u_2-1\ge 0\wedge[[Q_1\ge 0\wedge Q_2<0\wedge Q_4>0\wedge u_3-u_2>0]\vee[Q_1 \ge 0\wedge Q_3<0\wedge u_3-u_2>0]\vee[Q_3>0\wedge Q_4<0\wedge u_3-u_2<0] \vee[Q_1<0\wedge Q_2>0\wedge Q_4<0\wedge u_3-u_2<0]];$$

$$\Delta_2(u_1,u_2,u_3):\quad u_1^3+2\,u_1^2+11\,u_1-16\le 0\wedge u_1^4+36\,u_1^2-90\,u_1+54<0\wedge u_2- 1\ge 0\wedge[[Q_1\ge 0\wedge Q_2\le 0\wedge Q_5>0\ \wedge u_3-u_2>0]\vee[Q_3<0\wedge Q_4>0\wedge u_3-u_2>0]\vee[Q_3>0\wedge Q_5<0\wedge u_3-u_2<0]\vee[Q_4 <0\wedge u_3-u_2<0]],$$

$$Q_1=u_1u_2^3-u_2^3-u_1^2+u_2^2+3\,u_1u_2^2-3\,u_2^2-2\,u_1^2u_2+7\,u_1u_2-3\,u_2-5\,u_1^2+5\,u_1-1,$$
$$Q_2=3\,u_1u_2^2-3\,u_2^2-2\,u_1^2u_2+6\,u_1u_2-6\,u_2-2\,u_1^2+7\,u_1-3,$$
$$Q_3=2u_1u_2^3-2\,u_2^3-u_1^2u_2^2+2\,u_1u_2-u_1^2,\qquad Q_4=6\,u_1u_2-6\,u_2-u_1^2,$$
$$Q_5=3\,u_1u_2^2-3\,u_2^2-u_1^2u_2+u_1,$$

and

$$X_{11}=\frac{u_1^2}{6\,(u_1-1)}+\frac{\omega^1c_1}{3}+\frac{\omega^2c_2}{3},\qquad X_{12}=\frac{u_1^2}{6\,(u_1-1)}+\frac{\omega^2c_1}{3}+\frac{\omega^1c_2}{3},$$
$$X_2=1-\frac{u_1}{x_1},\qquad X_3=\sqrt{2\,u_2x_1-2\,x_1-u_2^2+1},\qquad X_4=\frac{x_1x_3-u_3x_3}{x_1-u_2},$$
$$p_1=\frac{-u_1^6-2\,u_1^5-11\,u_1^4+16\,u_1^3}{4\,(u_1-1)^2},\qquad p_2=\frac{54\,u_1^4-125\,u_1^3+72\,u_1^2}{4\,(u_1-1)^3},$$

and c_1, c_2, ω are the same as in Theorem 2.1.5 of Section 2.1.3.

The representation corresponding to X_{11} given in [63] is

$$X_{11}'=\frac{\sqrt[3]{\delta}^{R}}{6\,(u_1-1)}+\frac{u_1^4-12\,u_1^2+12\,u_1}{6\,(u_1-1)\,\sqrt[3]{\delta}^{R}}+\frac{u_1^4}{6\,(u_1-1)},$$

where

$$\delta=d+6\sqrt{3D}u_1(u_1-1),\qquad d=u_1^6+36\,u_1^4-90\,u_1^3+54\,u_1^2,$$
$$D=u_1^6+8\,u_1^4-36\,u_1^3+43\,u_1^2-16\,u_1.$$

X_{11}' may encounter the unstable "near 0/0" case when $u_1=0$ or $u_1\approx 1.115749397$. When the diagram is animated by gradually changing values of the parameters, it will be out of shape (e.g., some of the geometric objects disappear) if u_1 is near 0 or 1.115749397. However, $X_{11}=0$ when $u_1=0$, and $X_{11}\approx 1.068695857$ when

$u_1 \approx 1.115749397$, so "near 0/0" does not occur. Note that the denominator $u_1 - 1$, which is the initial of the first polynomial in the triangular set, cannot be avoided. Although the conditions $\Delta_1(u_1)$ and $\Delta_2(u_1)$ ensure that $u_1 - 1$ does not vanish, X_{11} will be very large when u_1 is close to 1. We have also considered other cases of circle contacts, such as three internal contacts, one external and two internal contacts, and one internal and two external contacts, by changing the inequality constraints. Similar phenomena have been observed.

To further highlight the role of inequality constraints, let us look at the two diagrams, one with three external contacts and the other with two external contacts and one internal contact of circles, shown in Figure 5.2. With inequality constraints, we can generate any of the two diagrams with the specified numbers of external and internal circle contacts and during dynamic animation of the generated diagram the numbers of circle contacts will remain unchanged. Without inequality constraints, one does not know *a priori* which of the two diagrams will be generated and during dynamic animation two externally tangent circles may become internally tangent, and vice versa.

5.5.2 Feuerbach Theorem

The circle passing through the feet of the three altitudes of an arbitrary triangle also passes through other six meaningful points and thus is called the *nine-point circle* of the triangle. Feuerbach's theorem [37] states that the nine-point circle of any triangle is tangent internally to the inscribed circle and externally to the three escribed circles. We want to draw part of this configuration: a dynamic diagram for the nine-point circle (with center N) to be tangent to the inscribed circle (with center I). To specify the incenter I as an intersection point of angular bisectors, we need to ensure that the bisectors are all internal, which cannot be done by using polynomial equalities only. We can do so by using inequalities.

Let the coordinates of the points as shown in Figure 5.3 be assigned as follows:

$$A(0,0),\quad B(1,0),\quad C(u_1,1),\quad I(x_1,x_2),\quad D(x_1,0),$$
$$A_1(x_3,x_4),\quad A_2(x_5,x_6),\quad A_3(x_7,0),\quad N(x_8,x_9).$$

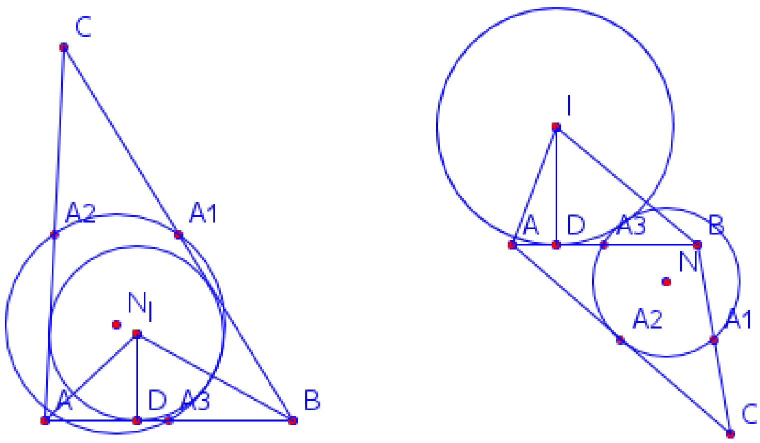

Figure 5.3: Feuerbach Configuration

The geometric constraints for the configuration may be specified as

$$\begin{cases} F_1 = x_1^2 + (2\,x_2 - 2\,u_1 x_2 - 2)x_1 + 2\,u_1 x_2 & (\tan\angle CBI = \tan\angle ABI) \\ \qquad - x_2^2 - 2\,x_2 + 1 = 0, & \\ F_2 = x_2^2 + 2\,u_1 x_1 x_2 - x_1^2 = 0, & (\tan\angle CAI = \tan\angle BAI) \\ F_3 = 2\,x_3 - u_1 - 1 = 0, & (A_1 \text{ midpoint of } BC) \\ F_4 = 2\,x_4 - 1 = 0, & \\ F_5 = 2\,x_5 - u_1 = 0, & (A_2 \text{ midpoint of } AC) \\ F_6 = 2\,x_6 - 1 = 0, & \\ F_7 = 2\,x_7 - 1 = 0, & (A_3 \text{ midpoint of } AB) \\ F_8 = 2\,x_5 x_8 + x_7^2 - 2\,x_7 x_8 - x_5^2 + 2\,x_6 x_9 - x_6^2 = 0, & (|NA_3| = |NA_2|) \\ F_9 = 2\,x_3 x_8 + x_7^2 - 2\,x_7 x_8 - x_3^2 + 2\,x_4 x_9 - x_4^2 = 0, & (|NA_3| = |NA_1|) \\ G_1 = -x_1 + u_2 x_2 - x_2 + 1 > 0, & (I \text{ inside } \triangle ABC) \\ G_2 = x_1 - u_2 x_2 > 0, & \\ G_3 = x_2 > 0. & \end{cases}$$

First we compute an irreducible triangular decomposition of $\{F_1, \ldots, F_9\}$ over $\mathbb{Q}(u_1)$ with respect to variable ordering $x_1 \prec \ldots \prec x_9$. The decomposition consists of only one triangular set $[T_1, \ldots, T_9]$ with

$$T_1 = 4\,x_1^4 - 8\,x_1^3 - (4\,u_1^2 - 4\,u_1)x_1^2 + (4\,u_1^2 + 4)x_1 - 1,$$
$$T_2 = (2\,u_1 + 2\,x_1 - 2)x_2 - 2\,x_1 + 1,$$
$$T_3 = 2\,x_3 - u_1 - 1, \qquad T_4 = 2\,x_4 - 1,$$

$$T_5 = 2\,x_5 - u_1, \qquad T_6 = 2\,x_6 - 1,$$
$$T_7 = 2\,x_7 - 1, \qquad T_8 = 4\,x_8 - 2\,x_3 - u_1,$$
$$T_9 = 4\,x_9 - 4\,x_8 + 4\,x_8 u_1 - u_1^2.$$

Then the semi-algebraic system may then be decomposed by using the enhanced HLLW approach into the following two weak SRRs:

$$\begin{aligned}
2\,u_1 - 1 < 0, \quad & x_1 = X_{11},\, x_2 = \frac{2\,x_1 - 1}{2\,(u_1 + x_1 - 1)},\, x_3 = \frac{1}{2} + \frac{u_1}{2}, x_4 = \frac{1}{2},\, x_5 = \frac{u_1}{2},\, x_6 = \frac{1}{2},\\
& x_7 = \frac{1}{2},\, x_8 = \frac{u_1}{4} + \frac{x_3}{2},\, x_9 = x_8 - x_8 u_1 + \frac{u_1^2}{4};\\
2\,u_1 - 1 > 0, \quad & x_1 = X_{12},\, x_2 = \frac{2\,x_1 - 1}{2\,(u_1 + x_1 - 1)},\, x_3 = \frac{1}{2} + \frac{u_1}{2}, x_4 = \frac{1}{2},\, x_5 = \frac{u_1}{2},\, x_6 = \frac{1}{2},\\
& x_7 = \frac{1}{2},\, x_8 = \frac{u_1}{4} + \frac{x_3}{2},\, x_9 = x_8 - x_8 u_1 + \frac{u_1^2}{4},
\end{aligned}$$

where

$$\begin{aligned}
& X_{11} = (2 + k_1 - k_2 - k_3)/4, \qquad X_{12} = (2 - k_1 + k_2 - k_3)/4,\\
& \sigma_2 = \begin{cases} +1 & \text{if } r \le 1,\\ -1 & \text{if } r > 1, \end{cases}\\
& r = \text{number of distinct real roots of } T_1 \in \{0, 1, 2, 3, 4\},\\
& p_1 = 4\,u_1^{10} - 20\,u_1^9 + 57\,u_1^8 - 108\,u_1^7 + 158\,u_1^6 - 180\,u_1^5 + 165\,u_1^4 - 116\,u_1^3\\
& \qquad + 64\,u_1^2 - 24\,u_1 + 4,\\
& p_2 = 2\,u_1^6 - 6\,u_1^5 - 3\,u_1^4 + 16\,u_1^3 - 27\,u_1^2 + 18\,u_1,\\
& p_3 = 8\,u_1^2 - 8\,u_1 + 12,
\end{aligned}$$

k_1, k_2, k_3 are the same as in Theorem 4.1.1 of Section 4.1, and T_1 (the first polynomial in the only triangular set) is quartic in x_1 with u_1 as parameter.[2]

The dynamic diagram generated automatically using the above weak SRRs is shown in Figure 5.3, on the left-hand side; the diagram shown on the right-hand side is generated by changing the order relation to "I is outside $\triangle ABC$". And without inequality constraints, the two diagrams will both appear during dynamic animation.

[2]The number r of distinct real solutions of T_1 can be determined by using the discrimination system (see Section 2.2): first produce the discrimination sequence $[D_1, \ldots, D_4] = [1, p_3, p_5, p_1]$ of T_1 with parametric coefficients, where $p_5 = 2\,u_1^6 - 6\,u_1^5 + 13\,u_1^4 - 16\,u_1^3 + 17\,u_1^2 - 10\,u_1 + 6$, and then for any given value of u_1, construct the sign list and the revised sign list of sequence. If the number of sign changes of the revised sign list is v and the number of non-vanishing members in the revised sign list is l, then $r = l - 2\,v$ [147].

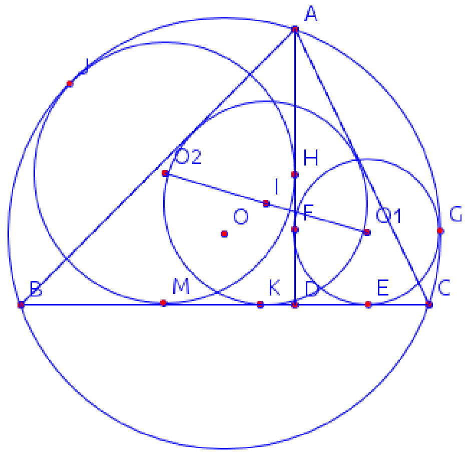

Figure 5.4: Thébault–Taylor Configuration

5.5.3 Thébault–Taylor Theorem

The Thébault conjecture of 1938, confirmed by K. B. Taylor in 1983, has an interesting history itself and in the recent evolvement of automated theorem proving in geometry (see, e.g., [130, pp. 130–131], [152], and references therein). Its configurations, if specified unambiguously with order relations, present a challenge for dynamic diagram generation. Given a triangle ABC and any point D on BC, one version of the problem is to draw the inscribed circle and the circumcircle of the triangle and two additional circles, each tangent to AD, BC, and the circumcircle, such that their centers and the incenter of the triangular are collinear. The Thébault–Taylor theorem implies that this configuration is always drawable.

An unambiguous algebraic formulation of the problem requires polynomial inequalities and computing the SRRs of the semi-algebraic system is nontrivial. We have tried the enhanced HLLW approach in a brute-force way without success. Here we consider a special case of the problem by assuming that D is the perpendicular foot of the line AD to BC. The coordinates of points are assigned as follows:

$$A(1,1),\quad B(0,0),\quad C(u_1,0),\quad D(1,0),\quad O(x_1,x_2),$$
$$O_1(x_3,x_4),\quad E(x_3,0),\quad F(1,x_4),\quad G(x_5,x_6),\quad O_2(x_7,x_8),$$
$$M(x_7,0),\quad H(1,x_8),\quad J(x_9,x_{10}),\quad I(x_{11},x_{12}),\quad K(x_{11},0).$$

Even though the problem is considerably simplified in the special case, the computations involved in the HLLW approach still cannot be completed within one hour. The corresponding quantified formulas contain large polynomials in several variables, which makes quantifier elimination difficult. Therefore, we try to split the problem into subproblems of construction, following the basic idea of the graph-analysis approach. For our special problem, it is easy to observe that the constructions of the three circles I, O_1, and O_2 are completely separate, so we can divide the problem into three subproblems. For each subproblem, it is not very difficult to compute the SRRs of the corresponding semi-algebraic system using the enhanced HLLW approach.

Subproblem 1. Construct circle O_1 with the specification of geometric constraints that can be expressed algebraically as:

$$\begin{cases} F_1 = 2\,x_1u_1 - u_1^2 = 0, & (O \text{ circumcenter of } \triangle ABC) \\ F_2 = 2\,x_2 + 2\,x_1 - 2 = 0, & \\ F_3 = -x_4x_5 + x_4x_1 + x_2x_5 + x_3x_6 & (O,\ O_1,\ G \text{ collinear}) \\ -x_2x_3 - x_1x_6 = 0, & \\ F_4 = x_4^2 - x_3^2 + 2\,x_3 - 1 = 0, & (|O_1E| = |O_1F|) \\ F_5 = -x_3^2 + 2\,x_3x_5 - x_5^2 + 2\,x_4x_6 - x_6^2 = 0, & (|O_1E| = |O_1G|) \\ F_6 = 2\,x_1x_5 - x_5^2 + 2\,x_2x_6 - x_6^2 = 0, & (|OB| = |OG|) \\ G_1 = x_4 \geq 0, \quad G_2 = x_4 - 1 \leq 0, & (O_1 \text{ inside the rectangle} \\ G_3 = x_3 - 1 \geq 0, \quad G_4 = x_3 - u_1 \leq 0. & \text{ with } AC \text{ as diagonal}) \end{cases}$$

The set $\{F_1, \ldots, F_6\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2)$ into two irreducible triangular sets. However, one of them turns out to be inconsistent with the inequality constraints. So we get only one triangular set $[T_1, \ldots, T_6]$, where

$$\begin{aligned} &T_1 = -2\,x_1 + u_1, \qquad T_2 = 2\,x_2 + u_1 - 2, \\ &T_3 = x_3^4 + 4\,u_1^2x_3 - u_1^2 - 2\,u_1^2x_3^2 + 2\,u_1x_3^2 - 4\,x_3^3 - 4\,x_3u_1 + 4\,x_3^2, \\ &T_4 = x_4 - x_3 + 1, \\ &T_5 = u_1^2x_5 - 2\,u_1x_4^2 + 2\,x_2u_1x_4 + u_1x_3^2 - 4\,u_1x_5x_3 - 4\,x_3x_2^2 + 4\,x_5x_3^2 \\ &\qquad + 4\,x_5x_4^2 + 4\,x_2^2x_5 - 2\,x_3^3 - 8\,x_2x_5x_4 + 4\,x_3x_2x_4, \\ &T_6 = 2\,u_2x_6 - 2\,x_5 + 2\,x_5u_1 - u_1^2 - u_2^2 + 1. \end{aligned}$$

Then the corresponding semi-algebraic system may be decomposed into one weak SRR

$$\Delta_1(u_1),\ x_1 = \frac{u_1}{2},\ x_2 = -\frac{u_1}{2} + 1,\ x_3 = X_{13},\ x_4 = x_3 - 1,\ x_5 = X_5,\ x_6 = X_6$$

with $\Delta_1(u_1)$ holding for infinitely many values of u_1 and five other weak SRRs holding for only finitely many values of u_1, where

$$\Delta_1(u_1):\; u_1-1\geq 0\wedge[7\,u_1^2-32\,u_1+32\neq 0\vee u_1-2<0],$$

and

$$\begin{aligned}
&X_5=\frac{2\,u_1x_4^2-2\,x_2u_1x_4-u_1x_3^2+4\,x_3x_2^2-4\,x_3x_2x_4+2\,x_3^3}{4\,x_3^2-4\,u_1x_3+u_1^2+4\,x_4^2-8\,x_4x_2+4\,x_2^2},\\
&X_6=\frac{2\,x_5x_2+u_1x_4-2\,x_3x_2-2\,x_5x_4}{u_1-2\,x_3},\\
&X_{13}=(4-k_1+k_2-k_3)/4,\\
&\sigma_2=\begin{cases}+1 & \text{if } r\leq 1,\\ -1 & \text{if } r>1,\end{cases}\\
&r=\text{number of distinct real roots of } T_3\in\{0,1,2,3,4\},\\
&p_1=256\,u_1^{10}-1536\,u_1^9+4352\,u_1^8-7168\,u_1^7+7168\,u_1^6-4096\,u_1^5+1024\,u_1^4,\\
&p_2=16\,u_1^6-48\,u_1^5-48\,u_1^4+320\,u_1^3-480\,u_1^2+384\,u_1-128,\\
&p_3=16\,u_1^2-16\,u_1+16,
\end{aligned}$$

k_1, k_2, k_3 are the same as in Theorem 4.1.1 of Section 4.1, and r can be determined by the discrimination system of T_3 (see Section 2.2).

Subproblem 2. Construct circle O_2 with the specification of geometric constraints that can be expressed algebraically as:

$$\begin{cases}
F_1=2\,x_1u_1-u_1^2=0, & (O \text{ circumcenter of } \triangle ABC)\\
F_2=2\,x_2+2\,x_1-2=0, & \\
F_7=-x_8x_9+x_8x_1+x_2x_9+x_7x_{10} & (O,\ O_2,\ J \text{ collinear})\\
-x_7x_2-x_1x_{10}=0, & \\
F_8=x_8^2-x_7^2+2\,x_7-1=0, & (|O_2M|=|O_2H||)\\
F_9=-x_7^2+2\,x_7x_9-x_9^2+2\,x_8x_{10}-x_{10}^2=0, & (|O_2M|=|O_2J|)\\
F_{10}=2\,x_1x_9-x_9^2+2\,x_2x_{10}-x_{10}^2=0, & (|OB|=|OJ|)\\
G_5=x_8\geq 0,\quad G_6=x_8-1\leq 0, & (O_2 \text{ inside the rectangle}\\
G_7=x_7-1\leq 0,\quad G_8=x_7\geq 0. & \text{with } AB \text{ as diagonal})
\end{cases}$$

Note that here F_1, F_2 are the same as in Subproblem 1, we just use them to eliminate x_1, x_2 from the constraints, but do not change the value of x_1, x_2.

The set $\{F_1, F_2, F_7, \ldots, F_{10}\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2)$ into two irreducible triangular sets. However, one of them turns out to be inconsistent with the inequality constraints. So we get only one triangular set $[T_1, T_2, T_7, \ldots, T_{10}]$, where

$$
\begin{aligned}
&T_1 = -2\,x_1 + u_1, \qquad T_2 = 2\,x_2 + u_1 - 2,\\
&T_7 = x_7^4 - 4\,u_1x_7^3 - u_1^2 + 2\,u_1^2x_7^2 - 2\,u_1x_7^2 + 4\,x_7^3 + 4\,x_7u_1 - 4\,x_7^2,\\
&T_8 = x_8 + x_7 - 1,\\
&T_9 = u_1^2x_9 - 2\,x_8^2u_1 - 4\,u_1x_7x_9 + u_1x_7^2 + 2\,x_8u_1x_2 - 8\,x_8x_9x_2 + 4\,x_2^2x_9\\
&\qquad - 4\,x_2^2x_7 + 4\,x_8^2x_9 + 4\,x_7^2x_9 - 2\,x_7^3 + 4\,x_8x_7x_2,\\
&T_{10} = 2\,x_8x_9 - x_8u_1 - 2\,x_2x_9 - 2\,x_7x_{10} + 2\,x_7x_2 + u_1x_{10}.
\end{aligned}
$$

Then the corresponding semi-algebraic system may be decomposed into three weak SRRs

$$\Theta_j(u_1),\ x_7 = X_{j7}, x_8 = -x_7 + 1, x_9 = X_9, x_{10} = X_{10}, \quad j = 1, 2, 3,$$

with $\Theta_j(u_1)$ holding for infinitely many values of u_1 and four other weak SRRs holding for only finitely many values of u_1, where

$$
\begin{aligned}
\Theta_1(u_1):&\quad u_1 - 1 \le 0;\\
\Theta_2(u_1):&\quad u_1 - 1 \le 0 \wedge u_1 > 0;\\
\Theta_3(u_1):&\quad u_1 > 0,
\end{aligned}
$$

and

$$
\begin{aligned}
&X_9 = \frac{2\,x_8^2u_1 - 2\,x_8u_1x_2 + 2\,x_7^3 - u_1x_7^2 + 4\,x_2^2x_7 - 4\,x_8x_7x_2}{4\,x_7^2 - 4\,x_7u_1 + u_1^2 + 4\,x_8^2 - 8\,x_8x_2 + 4\,x_2^2},\\
&X_{10} = \frac{x_8u_1 - 2\,x_8x_9 + 2\,x_2x_9 - 2\,x_7x_2}{u_1 - 2\,x_7},\\
&X_{17} = (4\,u_1 - 4 + k_1 + k_2 + k_3)/4,\\
&X_{27} = (4\,u_1 - 4 + k_1 - k_2 - k_3)/4,\\
&X_{37} = (4\,u_1 - 4 - k_1 + k_2 - k_3)/4,\\
&r = \text{number of distinct real roots of } T_7 \in \{0, 1, 2, 3, 4\},\\
&p_1 = 256\,u_1^{10} - 1536\,u_1^9 + 4352\,u_1^8 - 7168\,u_1^7 + 7168\,u_1^6 - 4096\,u_1^5 + 1024\,u_1^4,\\
&p_2 = -16\,u_1^6 + 48\,u_1^5 + 48\,u_1^4 - 320\,u_1^3 + 480\,u_1^2 - 384\,u_1 + 128,\\
&p_3 = 32\,u_1^2 - 80\,u_1 + 80,\\
&p_4 = 32\,u_1^3 - 128\,u_1^2 + 192\,u_1 - 128,
\end{aligned}
$$

$k_1,\ k_2,\ k_3, \sigma_2$ are the same as in Theorem 4.1.1 of Section 4.1, and r can be determined by the discrimination system of T_7 (see Section 2.2).

Subproblem 3. Construct circle I with the specification of geometric constraints that can be expressed algebraically as:

$$\begin{cases} F_{11} = x_{12}^2 - x_{11}^2 + 2\,x_{11}x_{12} = 0, & (\tan\angle CBI = \tan\angle IBA) \\ F_{12} = x_{11}^2 - 2\,x_{11}u_1 + u_1^2 - 2\,x_{11}x_{12} + 2\,x_{12}u_1 & (\tan\angle ACI = \tan\angle ICB) \\ \qquad + 2\,u_1x_{12}x_{11} - 2\,x_{12}u_1^2 - x_{12}^2 = 0, & \\ G_9 = u_1x_{11} - u_1x_{12} > 0, & (I \text{ inside } \triangle ABC) \\ G_{10} = u_1^2 + u_1x_{12} - u_1x_{11} - x_{12}u_1^2 > 0, & \\ G_{11} = u1^2x_{12} > 0. & \end{cases}$$

The set $\{F_1, F_2, F_7, \ldots, F_{10}\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2)$ into two irreducible triangular sets. However, one of them turns out to be inconsistent with the inequality constraints. So we get only one triangular set $[T_1, T_2, T_7, \ldots, T_{10}]$, where

$$\begin{aligned} T_{11} &= 4\,x_{11}^4 + 4\,u_1^2x_{11}^2 + 4\,u_1x_{11}^2 - 4\,x_{11}u_1^2 - 8\,x_{11}^3u_1 - 8\,x_{11}^2 + 8\,x_{11}u_1 - u_1^2, \\ T_{12} &= -u_1 + 2\,x_{12}u_1 + 2\,x_{11} - 2\,x_{12} - 2\,x_{11}x_{12}. \end{aligned}$$

Then the corresponding semi-algebraic system may be decomposed into two weak SRRs:

$$\begin{array}{ll} u_1 - 2 < 0 \wedge u_1 \neq 0, & x_{11} = X_{111},\ x_{12} = \dfrac{-2\,x_{11} + u_1}{-2\,x_{11} - 2 + 2\,u_1}; \\ u_1 - 2 > 0, & x_{11} = X_{211},\ x_{12} = \dfrac{-2\,x_{11} + u_1}{-2\,x_{11} - 2 + 2\,u_1}, \end{array}$$

where

$$\begin{aligned} & X_{111} = (2\,u_1 + k_1 - k_2 - k_3)/4, \quad X_{211} = (2\,u_1 - k_1 + k_2 - k_3)/4, \\ & \sigma_2 = \begin{cases} +1 & \text{if } r \leq 1, \\ -1 & \text{if } r > 1, \end{cases} \\ & r = \text{number of distinct real roots of } T_{11} \in \{0,1,2,3,4\}, \\ & p_1 = 4\,u_1^8 - 32\,u_1^7 + 112\,u_1^6 - 224\,u_1^5 + 272\,u_1^4 - 192\,u_1^3 + 64\,u_1^2, \\ & p_2 = -2\,u_1^6 + 12\,u_1^5 - 30\,u_1^4 + 40\,u_1^3 - 12\,u_1^2 - 24\,u_1 + 16, \\ & p_3 = 4\,u_1^2 - 8\,u_1 + 16, \end{aligned}$$

$k_1,\ k_2,\ k_3$ are the same as in Theorem 4.1.1 of Section 4.1, and r can be determined by the discrimination system of T_{11} (see Section 2.2).

For all the three subproblems, solutions formulas of quartic equations with inequality constraints need be used for computing the weak SRRs. Combining these weak SRRs, we obtain $1 \times 3 \times 2 = 6$ weak SRRs for the original problem in the special case (see Table 6.1 for the computing time). The dynamic diagram shown in Figure 5.4 has been generated automatically, making use of the computed SRRs.

Chapter 6

GeoDraw: A Software Package for Automated Generation of Dynamic Diagrams

The original and enhanced HLLW approach for solving dynamic geometric constraints involving equalities and inequalities has been explained in the previous chapters. We have developed a software package GeoDraw for drawing dynamic diagrams, which implements the process of generating dynamic diagrams automatically by using the enhanced HLLW approach. In this chapter, we discuss the design and implementation of GeoDraw, point out the functionalities and features of the current version, provide several examples to show the usage of GeoDraw, and present some empirical data in table form to show the performance of the package for generating the diagrams of 15 theorems in plane Euclidean geometry. Most of the material in this chapter is based on the paper [153].

6.1 Introduction and Motivation

Dynamic diagrams have features of interaction, intelligence and real time. By observing the animation of dynamic diagrams, we can learn the relations of the involved geometric objects more directly or even find out new geometry theorems, thus bring learning and study of geometry into a new era. There are many dynamic geometry software systems, which can be divided into two types: constructive type and declarative type. Most of the systems can draw configurations of constructive type, that is, each geometric object is constructed from the previously constructed objects. For example, Cinderella, GCLC, GeoGebra and Super Sketchpad, these systems allow the users to draw geometric diagrams interactively. The others can draw configurations of declarative type, which is solving the corresponding semi-algebraic system with respect to the geometric relations among the objects and generating the diagrams according to the geometric elements with certain coordinate values, such as GEOTHER, Geometry Expressions and JGEX. The approaches for solving dynamic geometric constraints in these software systems are based, e.g., on numeric or symbolic computation, or both combined. Note that most of the approaches are proposed to deal with geometric constraints involving equalities only, despite that inequality constraints occur very frequently in real-world problems of geometry, such as internal or external tangent of the circles, internal or external bisection of angles, etc. This may be mainly due to the fact that dealing with inequalities would greatly increase computation difficulty and complexity, and cost much computation time and space. Therefore, it is meaningful to develop effective methods that can deal with geometric constraints involving inequalities.

The original and enhanced HLLW approach introduced in the previous chapter can directly tackle geometric constraints involving inequalities as well as equalities. We have developed a software package GeoDraw for drawing dynamic diagrams automatically from the predicate specification of a given set of geometric relations among a set of points. This package implements the enhanced HLLW approach, which is independent of GEOTHER and the previous implementation of the HLLW approach therein. However, GEOTHER only provides some demos for generating diagrams with equalities of degree less than or equal to 3 and inequalities. GeoDraw is more general and can deal with problems of equalities of degree 4.

6.2 Design and Implementation

In this section, we describe the design principles of GeoDraw and the concrete implementation of generating dynamic diagrams with inequality constraints.

6.2.1 Design Principles

According to the original and enhanced HLLW approach explained in the preceding chapters, the whole process of generating dynamic diagrams automatically is a combination of symbolic and numerical computations, and how to handle the relations between these two parts directly influences the efficiency of the diagram generation.

We bring in symbolic computation in order to tackle geometric constraints involving inequalities as well as equalities, and also to generate more complicated diagrams that the dynamic geometry software systems for drawing diagrams constructively

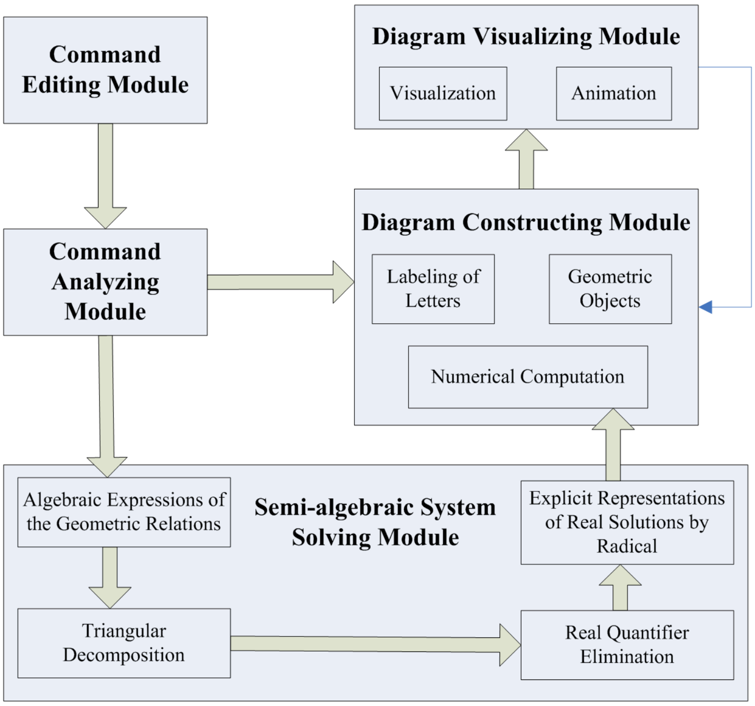

Figure 6.1: Architecture of GeoDraw

(or interactively), which is only based on numerical computation, can not handle . The automated incorporation of non-trivial symbolic expressions computed to generate diagrams can be very time-consuming, especially when the polynomials are of high degree with many parameters. However, in a specific dynamic diagram drawing problem, not all geometric objects are involved in inequalities. Thus we can enhance drawing efficiency by dividing the set of geometric objects into two sets: the ones appearing in inequalities, and the others. Hence the number of geometric objects involving symbolic computation would be reduced to the least.

Geometric objects involving inequalities can be further split into several completely separated components when the objects in each component are relatively independent, so that variables and constrains in each component could be decreased (see Example 6.5.4). The purpose of doing these is to reduce the difficulty and complexity of symbolic computation. The software package GeoDraw can draw configurations of constructive type based on numerical computation and configurations of declarative type based on the combination of both symbolic and numerical computations, and allows the diagrams to be generated with several independent components and enables users to choose drawing types flexibly to improve the efficiency.

GeoDraw includes three layers and five modules. Three layers refer to user, control and operating layers and five modules include command editing module, diagram visualizing module, command analyzing module, diagram constructing module and semi-algebraic system solving module (see Figure 6.1).

The user layer, which includes command editing module and diagram visualizing module, provides a graphical interface for users and accepts operations by users. The control layer, which includes command analyzing module and diagram constructing module, analyzes the drawing commands from the users, determines to use the constructive type or the declarative type to draw the diagram, performs some simple numerical computations with the received expressions, constructs the diagram with the geometric objects, communicates with the user layer in real time and reconstructs the diagram according to the new data provided by the users. The operating layer, which is an optional part decided by the analyzing module, computes explicit representations of real solutions of the corresponding parametric semi-algebraic system by radicals based on symbolic computation.

6.2.2 Implementation Issues

GeoDraw is implemented in Java with interface to the Epsilon library [130] in Maple and the QEPCAD program [15] in C. The main part of the program is written in Java, one of the most convenient programming languages for graphics, animation, web, and interface implementation. It provides a graphic user interface which allows the user to input the predicate specification of the geometric constraints of a diagram and to animate and fine-tune the generated diagram by mouse click and dragging. According to the inputs of the users, the system will determine the type for drawing diagrams automatically.

When drawing configurations of constructive type, the real values of the coordinates of the points in geometric objects are computed successively according to the geometric relations among these objects by only numerical computation, and then construct the diagram step by step in accordance with the values of the coordinates.

When drawing configurations of declarative type, which implements the process of diagram generation sketched in Chapter 5, the input specification of the geometric constraints is parsed and translated in Java into a parametric semi-algebraic system of equalities and inequalities. The decomposition of the set of (equality) polynomials into irreducible triangular sets requires extensive polynomial operations, including factorization over successive algebraic extension fields, and thus cannot be easily implemented from scratch in Java. Therefore, we decided to use an external computer algebra system in which basic polynomial operations can be well implemented. Our choice of Maple is mainly due to the fact that the Epsilon library [130], containing an implementation of several efficient triangular decomposition methods, has been developed in the system.

Epsilon library is a unique and rich library for triangular-set-based polynomial elimination, and no other similar library is available publicly. Its functions for zero decomposition are all Gröbner free, and the CharSets package is still one of the most complete, comprehensive, and widely available implementations of the characteristic set method.

In our Java program, a piece of Maple code is generated and then Maple is invoked to execute the code. In Maple, the set of (equality) polynomials is decomposed into irreducible triangular sets by using one of the Epsilon functions and QEPCAD is then invoked to eliminate quantifiers from the real solution formulas of polynomial equations in the triangular sets with inequality constraints. The computed SRRs are written into a file for the Java program to read in.

QEPCAD [15] is a powerful C program for quantifier elimination based on partial cylindrical algebraic decomposition [32]. It has been widely used to deal with decision and computation problems involving inequalities over the reals. We have also considered other programs for real quantifier elimination, including REDLOG [40, 136], QEQUAD [60], and SturmHabicht [53]. From some example tests, we have observed that QEPCAD takes less computing time and outputs simpler formulas in general. For Apollonius circles problem, REDLOG cannot finish the required quantifier elimination within one hour, while QEPCAD can do it in 83 seconds on the same computer (see Section 5.5.1 and Table 6.1).

The main Java program parses the SRRs read from the file, chooses a set of real values satisfying the conditions in the SRRs for the parameters, computes the corresponding values of the dependent variables according to the SRRs, draws a static diagram according to the values of the variables, and updates the diagram with new values of the variables, when the values of the parameters are changed (e.g., by mouse dragging), by recomputing the values of the dependent variables according to the SRRs.

With the interfaces between Java, Maple, and QEPCAD, the entire process of dynamic diagram generation is fully automated. For the involved symbolic and numeric computation, there are two main stages: preprocessing and updating. The preprocessing stage, carried out only once for each problem, computes symbolically the SRRs of the semi-algebraic system. Usually, it involves heavy algebraic computations and is time-consuming. The updating stage, which may be carried out repeatedly, evaluates numerically the values of the dependant variables for changing values of the parameters according to the (same) SRRs. Updating can be repeated reliably and efficiently for real-time dynamic animation because the involved numeric evaluations of the SRRs are computationally fast.

6.3 Functionalities and Features

GeoDraw is the geometry software that can generate dynamic diagrams with both equality and inequality constraints. The drawn diagram may be animated and fine-tuned by mouse click and dragging with the given geometric relations maintained. GeoDraw can be used to draw configurations of both constructive type and declarative type. The whole process of drawing dynamic diagrams making use of symbolic triangular decomposition, symbolic real quantifier elimination with parsing, numerical computation, graphic drawing, and letter labeling in Java is completely automated.

GeoDraw provides a graphic user interface (see Figure 6.2) which allows the user to do many operations, such as inputting the predicate specification of the geometric constraints of a diagram, checking information of the geometric objects, animating the drawn diagram with mouse, etc. The graphical interface in Figure 6.2 includes four windows, with the following functions.

1. Input window (bottom left), in which users can input the drawing commands. When the constructive type is adopted, input the commands in order to construct the diagram directly (see Example 6.5.1); when the declarative type is adopted, use key words of `Define`, `Constrain` and `Display`. The commands in the `Define`

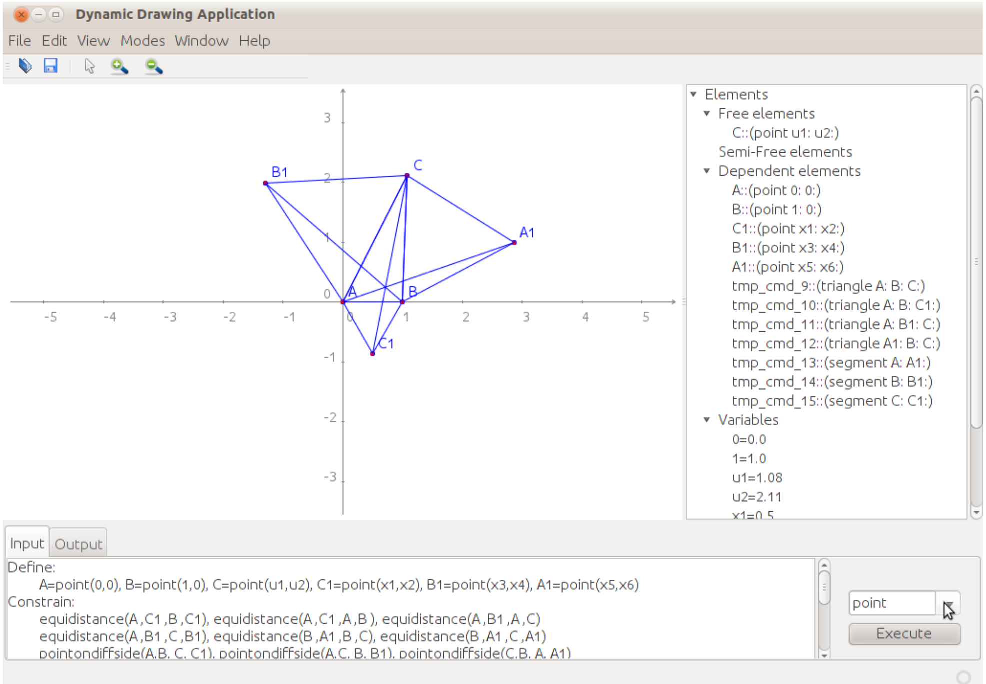

Figure 6.2: Graphic User Interface

part are used to assign the coordinates of the points; in the `Constrain` part are used to describe the geometric relations among the points; and in the `Display` part are used to draw the configurations of the diagrams. Note that in `Display` part, predicates for geometric relations of constructive type can also be carried out and users can add the geometric objects that do not need symbolic computation.

2. Command Prompt Window (bottom right), which shows all commands (i.e. predicate specification) carried out by the software. Users can click the commands directly and add them into the input window.

3. Output Window (top left), which shows the drawn dynamic diagram. When the arrow button in the toolbar is clicked, the free points are shown in green, the semi-free points in yellow, and the dependent points in red. One can drag any of the green and yellow points, and the diagram changes accordingly in real time, satisfying the given geometric relations. Users can also choose the background with axis or not.

4. Information Window (top right), which shows free, semi-free and dependent elements in the drawn diagram, and the current real values of all parameters and

variables. When the diagram changes, the value would alter accordingly.

GeoDraw also implements some common functions, such as zoom in, zoom out, save, open, etc. One more function is remarkable which is implemented for the feature that the symbolic computation is very complicated and time-consuming during the process of dynamic diagram generation: the import and export function. Users can export a diagram generation instance as a XML file with script and solution formulas according to the corresponding semi-algebraic system, and import it to generate the diagram again without repetitive symbolic computation.

Features that distinguish GeoDraw from other dynamic diagram drawing software systems are

1. Capability of generating dynamic diagrams with inequality constraints;
2. Combination of drawings for configurations of constructive and declarative types, flexible choice of drawing types for enhancing drawing efficiency;
3. Combination of symbolic and numerical computations, capable of complex dynamic diagram generation by invoking computer algebra systems for symbolic computation.

6.4 Predicate Specifications

We have implemented a number of predicates, and each of them contains a standard set of information entries. As introduced in Section 6.2.1, GeoDraw can do drawings for configurations of both constructive type and declarative type. The information entries of the predicates in these two types are different.

For the predicates for geometric relations of constructive type, which can be used with key word `Display` or without any key word, the information entries include: English meaning and geometric information. Consider, for instance, the predicate `circumcenter(A,B,C)`. Then the corresponding entries are: `compute and draw the circumcenter of the triangle` ABC, where A, B and C are the existing points constructed in the previous steps. These information entries allow the program to compute the coordinates of the points by numerical computation and draw geometric objects like points, lines and circles passing through certain points.

For the predicates for geometric relations of declarative type, which should be used with the key word `Constrain`, the information entries include: English meaning and algebraic expression (when the points are assigned coordinates). Consider, for instance, the predicate `collinear(A,B,C)`. Then the corresponding entries are: `the three points` A, B `and` C `are collinear`,

$$A_xB_y + B_xC_y + C_xA_y - A_xC_y - B_xA_y - C_xB_y,$$

where (A_x, A_y), (B_x, B_y), (C_x, C_y) are the coordinates of A, B, C respectively. These information entries allow the program to translate the predicate specification of a geometric problem into algebraic expressions (equalities and inequalities) automatically.

The assignment of coordinates can be done in `Define` part using the function `point` (see the Example 6.5.2 for Steiner Theorem) by the user. It is also recommend that the user could identify the parametric and dependent variables from the specification, for example, u_i for the parameters and x_j for the dependents, where i, j are natural numbers.

We list the predicates for geometric relations of constructive type in Appendix A, and for geometric relations of declarative type in Appendix B. The sets of predicates are not exhaustive and may be easily enlarged; it is easy to implement a new predicate and add it to the different sets according to the information entries of the predicate.

6.5 Examples and Analysis

GeoDraw can be used to draw configurations of constructive and declarative type. Note that, sometimes, we can combine these two types for drawing diagrams more efficiently. For the declarative type, the generation of a diagram can also be divided into several completely separated components in the `Constrain` part as explained above. Therefore, in fact, there are four typical ways to draw diagrams: the constructive way, the declarative way, the combination of the both, and the way of splitting the generation of a diagram into several components. According to the input of the users, the system will choose the corresponding way to construct the diagram automatically. Several examples are given below to illustrate how to draw diagrams through these four ways by using the different predicate specifications as input. The computations

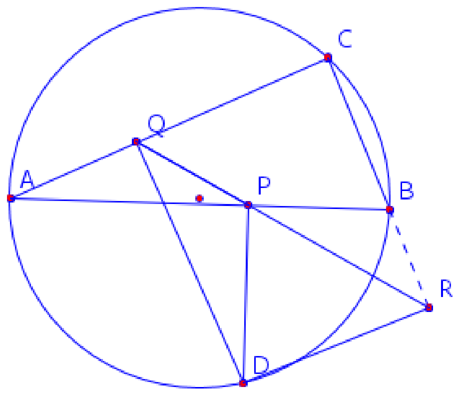

Figure 6.3: Simson Configuration

in the examples were done on an x86_64 under Linux 2.6.18 and the times are given in CPU seconds.

Example 6.5.1 (Simson Theorem)**.** If the points A, B and C are arbitrary, the point D is on the circumcircle of the triangle ABC, P is the perpendicular foot of the line DP to the line AB, Q is the perpendicular foot of the line DQ to the line AC, and R is the perpendicular foot of the line DR to the line BC, then the three points P, Q and R are collinear.

The geometric relations in this theorem can be algebraically expressed as equalities only. It is also not difficult to construct the diagram step by step according to the geometric constraints among the objects. The dynamic diagram as shown in Figure 6.3 can be drawn automatically in the constructive way with the following predicate specification as input

```
A=point(), B=point(), C=point(), D=pointon(circle(A,B,C)),
perpline(segment(A,C),D,Q), perpline(segment(A,B),D,P),
perpline(segment(B,C),D,R), segment(Q,P), segment(Q,R)
```

where A, B, C are three arbitrary points in the plane. Note that, in general, the diagram is not uniquely determined: some points may be free or semi-free. Therefore, the diagram drawn is actually an (randomly yet adequately chosen) instance.

The diagram can be animated and fine-tuned by mouse click and dragging with the given equalities of the geometric relations maintained.

Example 6.5.2 (Steiner Theorem)**.** See the description and inside process of symbolic computation of Steiner Theorem in Example 5.4.1 of Section 5.4.

To ensure that the triangles are drawn all outward, we need inequalities to represent the geometric constraints in this theorem. Therefore, we choose the declarative way to draw the diagram shown in Figure 5.1 of Section 5.4 with the following predicate specification as input

```
Define:
  A=point(0,0), B=point(1,0), C=point(u1,u2),
  C1=point(x1,x2), B1=point(x3,x4), A1=point(x5,x6)
Constrain:
  equidistance(A,C1,B,C1), equidistance(A,C1,A,B),
  equidistance(A,B1,A,C), equidistance(A,B1,C,B1),
  equidistance(B,A1,B,C), equidistance(B,A1,C,A1),
  pointondiffside(A,B,C,C1), pointondiffside(A,C,B,B1),
  pointondiffside(C,B,A,A1)
Display:
  triangle(A,B,C), triangle(A,B,C1), triangle(A,B1,C),
  triangle(A1,B,C), segment(A,A1), segment(B,B1),
  segment(C,C1)
```

where C is a free point and the predicate `equidistance` specifies a geometric relation that can be translated algebraically as an equality, and `pointondiffside` as an inequality. The diagram can be animated by mouse click and dragging with the given equalities and inequalities of the geometric relations maintained.

Example 6.5.3 (Napoleon Theorem)**.** Draw an arbitrary triangle ABC, three equilateral triangles ABC_1, ACB_1, and BCA_1 all inward or outward from $\triangle ABC$, and the circumcenters O_1, O_2, and O_3 of $\triangle ABC_1$, $\triangle ACB_1$, and $\triangle BCA_1$ form another equilateral triangle.

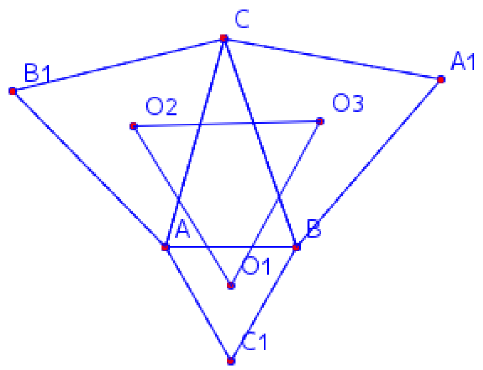

Figure 6.4: Napoleon Configuration

Note that the first part of the theorem is just the same as Steiner Theorem in Example 6.5.2 where the geometric constraints involve inequalities. However, in the second part, the coordinates of the three circumcenters can be directly computed with respect to the real values of the other points by simple numerical computation. Therefore, we choose the combination of constructive and declarative way to draw the diagram shown in Figure 6.4 with the following predicate specification by replacing the `Display` part in Example 6.5.2, and keeping the `Define` and `Constrain` parts unchanged.

```
Display:
  triangle(A,B,C), triangle(A,B,C1), triangle(A,B1,C),
  triangle(A1,B,C), O1=circumcenter(A,B,C1),
  O2=circumcenter(A,C,B1), O3=circumcenter(B,C,A1),
  triangle(O1,O2,O3)
```

We can also draw the diagram by using the declarative way only with three more predicates adding to the `Constrain` part, i.e. add six equalities to the corresponding semi-algebraic system of Steiner Theorem, which will increase the complexity of the symbolic computation and reduce the efficiency. By using the combination way, we can generate the diagram in 28.701 seconds, while the declarative way can do it in 58.613 seconds on the same computer.

The process of solving the corresponding semi-algebraic system by the declarative way only is represented in Example C.2 of Appendix C, while the process by the combination way is the same as Steiner Theorem shown in Example 5.4.1.

Example 6.5.4 (Thébault–Taylor Theorem)**.** See the description and inside process of symbolic computation of Thébault–Taylor Theorem in Section 5.5.3.

As explained before, we split the problem into three completely separated subproblems of construction for simplification, following the basic idea of the graph-analysis approach. For each subproblem, it needs inequalities to express the geometric relations among the involved points. The diagram shown in Figure 5.4 of Section 5.5.3 can be drawn with the following predicate specification as input

```
Define:
  A=point(1,1), B=point(0,0), C=point(u1,0),
  D=point(1,0), O=point(x1,x2), O1=point(x3,x4),
  E=point(x3,0), F=point(1,x4), G=point(x5,x6),
  O2=point(x7,x8), M=point(x7,0),H=point(1,x8),
  J=point(x9,x10), I=point(x11,x12), K=point(x11,0)
Constrain:
  equidistance(E,O1,F,O1), equidistance(E,O1,G,O1),
  equidistance(B,O,G,O), circumcenter(A,B,C,O),
  collinear(O,O1,G), inrectangle(A,C,O1)
Constrain:
  equidistance(M,O2,H,O2), equidistance(M,O2,J,O2),
  equidistance(B,O,J,O), circumcenter(A,B,C,O),
  collinear(O,O2,J), inrectangle(A,B,O2)
Constrain:
  equiangle(C,B,I,A,B,I), equiangle(A,C,I,B,C,I),
  pointonsameside(A,B,C,I), pointonsameside(A,C,B,I),
  pointonsameside(C,B,A,I)
Display:
  circle(O,A),circle(O1,E), circle(O2,M),
  triangle(A,B,C), segment(A, D), segment(O1, O2)
```

where each `Constrain` part can be parsed as a semi-algebraic system with equalities and inequalities. The diagram can be animated by mouse click and dragging the semi-free point C. The generation of the diagram takes 67.943 seconds. However, if we combine these three parts into one, the time for generating the diagram will be more than 20000 seconds.

Table 6.1: Data and Timings for 15 Examples

Theorem	Para No	Dep No	Max Deg	Ineq No	TD Time	QE Time	Time
Apollonius	3	4	3	3	0.228	82.761	84.927
Butterfly	4	7	2	1	0.085	42.530	44.100
Feuerbach	3	9	4	3	0.100	12.580	14.780
Morley	2	6	2	9	0.711	>20000	>20000
Napoleon	2	12	2	3	0.154	57.657	58.613
Pappus	6	2	1	6	0.081	1.516	2.397
M. Paterson	3	7	1	3	0.133	5.252	6.185
Steiner	2	6	2	3	0.132	27.863	28.701
Steiner Square	3	5	1	3	0.126	39.763	40.690
Thébault–Taylor	1	12	4	11	0.988	61.997	67.943
Example C.7	2	4	3	3	0.485	3132.425	3133.737
Example C.8	3	4	1	1	0.071	4.722	4.803
Example C.9	3	5	2	3	0.134	11.312	11.617
Example C.10	2	8	2	4	0.152	10.368	10.601

Para: Parameter; Dep: Dependant Variable; Ineq: Inequality

More examples are tested, and Table 6.1 reports the number of parameters, the number of dependent variables, the highest degree of equalities, and the number of inequality constraints for 15 semi-algebraic systems expressing the hypotheses of 15 well-known geometric theorems. It presents the respective timings for triangular decomposition, quantifier elimination, and the whole process of computing the SRRs of each semi-algebraic system. The details of these examples are given in Section 5.4, Section 5.5, and Appendix C. All the computations were performed on an x86_64 under Linux 2.6.18 and the times are given in CPU seconds. The quantifier elimination time includes not only the time of computation with QEPCAD, but also the time for invoking QEPCAD and translating the formulas into the QEPCAD input format. The latter depends on how many times QEPCAD is called in the problem, with each call costing less than one second. From Table 6.1, one sees that quantifier elimination using QEPCAD is the most time-consuming step of the (enhanced) HLLW approach. This is not a surprise because it is known that inequality constraints are difficult to handle.

Chapter 7

Conclusion and Future Work

In this thesis, we have studied the problem of generating dynamic diagrams with inequality constraints as well as equality constraints. The generation of dynamic diagrams amounts to producing a segment of program that implements the process of solving the corresponding parametric semi-algebraic system and visualizes the geometric configurations corresponding to the real solutions of the system for changing values of the parameters. The problem of solving real geometric constraint involving inequalities was addressed first by Hoon and coauthors. The proposed approach (referred to HLLW approach) is applicable for solving such geometric constraint problem that may be transformed into triangular semi-algebraic systems of equalities of degree less than 5 and inequalities of arbitrary degree with parameters. We have shown that for generating dynamic diagram automatically the performance of the HLLW approach can be enhanced in terms of stability of numeric computation and quality of generated diagrams. The used Ferro–Cardano-type solution formulas of cubic and quartic equations therein involve divisions, which may encounter a numerically unstable case (i.e., "near 0/0" case), when both the numerator and the denominator are close to zero.

In order to enhance the HLLW approach, we are concerned with the real solution formulas for cubic and quartic polynomial equations with real coefficients. A real convention (for square and cubic roots) which provides correct interpretations of the Lagrange formula for all cubic polynomial equations with real coefficients has been provided. Using this convention, we have also presented real solution formulas for the general cubic equations with real coefficients under equality and inequality constraints.

We have generalized the solution formulas from the cubic to the quartic equations. With a lot of investigations and experiments, we have adjusted the Lagrange-type solution formula for the real-coefficient quartic equations. The real convention has been proved that it can also provide correct interpretations of the adjusted formula for all quartic polynomial equations with real coefficients. Using this convention, we have also presented real solution formulas in radicals for the general quartic equation with real coefficients under equality and inequality constraints. The extension of the solution formulas from the cubic to the quartic case is not straightforward and needs some quite sophisticated techniques including the theories underlying Sturm-Habicht sequence and discrimination systems.

The real convention we found is a uniform convention that does not depend on the coefficients of the polynomials and that can always yield correct interpretations of the Lagrange-type formulas for all cubic and quartic polynomial equations with real coefficients. The new real solution formulas with constraints we given avoid divisions by small numbers and thus the numerically unstable “near 0/0” case. They are computationally better in terms of numeric stability and geometric continuity for dynamic diagram animation.

The HLLW approach has been enhanced by replacing the Ferro–Cardano-type solution formulas of cubic equations used there with newly introduced Lagrange-type real solution formulas with inequality constraints and by incorporating new real solution formulas of quartic equations with inequality constraints. The enhanced approach has been illustrated and the advantages and effectiveness of the new solution formulas have been demonstrated by several examples with experimental data.

The enhanced approach consists of two stages: *preprocessing* and *updating*. During the preprocessing stage (which is carried out only once), we compute, symbolically, explicit representations (radical expressions in parameters) of the solutions of the semi-algebraic system representing the geometric constraints. During the updating stage (which is carried out repeatedly), we evaluate, numerically, the radical expressions. Once the preprocessing has been done, the repeated updating can be carried out reliably and efficiently since it only involves evaluating radical expressions, yielding correct and dynamic (real-time) animation.

We have developed a software package GeoDraw for drawing dynamic diagrams automatically based on the predicate specification of a given set of geometric relations involving inequality constraints. This package has implemented the process of generating dynamic diagrams by using the enhanced HLLW approach, and can be used to draw configurations of constructive type based on numerical computation, and configurations of declarative type based on a combination of both symbolic and numeric computations. GeoDraw allows the diagrams to be generated with several independent components and enables users to choose drawing types flexibly to improve the efficiency. The system includes three layers and five modules. Three layers refer to user, control and operating layers and five modules include command editing module, diagram visualizing module, command analyzing module, diagram constructing module and semi-algebraic system solving module. The whole process of dynamic diagram generation combining triangular decomposition in Maple, real quantifier elimination in QEPCAD with parsing, numerical computation, graphic drawing, and letter labeling in Java is completely automated. GeoDraw allows the user to input the geometric constraints of a dynamic diagram with predicate specification and generates the diagram automatically for the user to animate and fine-tune using mouse click and dragging.

To end the summary of our work, we highlight the following main contributions of this thesis:

- A real convention which provides correct interpretations of the Lagrange-type formulas for all cubic and quartic polynomial equations with real coefficients;
- Real solution formulas without divisions by small numbers for the general cubic and quartic equations under equality and inequality constraints which are given as some existentially quantified subformulas;
- An approach for automatically generating dynamic diagrams, which is distinct from other existing approaches by its capabilities of dealing with inequality constraints and ensuring numeric stability in diagram animation;
- A software package GeoDraw for drawing dynamic diagrams automatically, based on the predicate specification of a given set of geometric relations involving inequality constraints.

Although the preprocessing stage in the approach carried out only once, it can be very time-consuming when the involved polynomials are of high degree with many parameters. Hence an important challenge for future work is to improve the speed of the preprocessing stage, in particular, real quantifier elimination. It is well known that general real quantifier elimination is intrinsically difficult. However, it seems that the formulas arising in the context of geometric constraints are not arbitrary, but have certain special structures. Therefore, one should investigate how to exploit those structures in order to develop specialized and thus more efficient real quantifier elimination methods, and solve more difficult parametric systems of geometric constraints to generate dynamic diagrams specified with order relations, e.g., for Morley's and Thébaut–Taylor's theorems without ambiguity. It is also meaningful to study the numeric stability of solution formulas and the geometric continuity of dynamic diagrams when some denominators are close to zero.

In addition, for solving polynomial equations, the following questions still remain for future investigation.

(1) Whether there is a convention that yields correct solutions for all cubic and quartic polynomial equations with *complex* coefficients.

(2) Whether Theorem 3.5.1 and the result in [136] can be combined to obtain a more efficient formulation. This insightful question was raised by an anonymous referee who also suggested that there should be a strong connection between the solutions u_i in the second part of the present paper and the symbolic solutions γ_i and real types of polynomials in [136]. We have investigated the issues and indeed there is a strong connection. However, we are not yet able to combine them into a better formulation due to various technical subtleties. We agree that it is worthwhile to pursue this as future work.

(3) How to simplify the result in Theorem 4.4.1. The real quartic solutions coupled with real constraints which are not only the derivatives but also two polynomials with respect to the coefficients. According to Thom's lemma, there exists representations of the constraints only with derivatives.

(4) How to generalize the solution formulas from the quartic to the case of degree more than 4. Though there are no general solution formulas in radicals for the polynomial equations of degree more than 4, there might be other simple representations of the solutions, such as trigonometric functions.

Bibliography

[1] B. Aldefeld. Variation of geometric based on a geometric-reasoning method. *Comput. Aided Design*, 20(3):117–126, 1988.

[2] E. L. Allgower and K. Georg. Continuation and path following. *Acta Numer.*, 7:1–64, 1993.

[3] H. Anai and H. Yanami. SyNRAC: A Maple-package for solving real algebraic constraints. In *Proceedings of the International Conference on Computational Science*, LNCS 2657, pages 828–837. Springer, Berlin Heidelberg, 2003.

[4] N. Anning. Discussions: A cubic equation of Newton's. *Amer. Math. Monthly*, 33(4):211–212, 1926.

[5] P. Aubry, D. Lazard, and M. Moreno Maza. On the theories of triangular sets. *J. Symb. Comput.*, 28:105–124, 1999.

[6] P. Aubry and M. Moreno Maza. Triangular sets for solving polynomial systems: a comparative implementation of four methods. *J. Symb. Comput.*, 28(1–2):125–154, 1999.

[7] S. Basu, R. Pollack, and M. F. Roy. *Algorithms in Real Algebraic Geometry.* Springer-Verlag, New York, 2006.

[8] E. T. Bell. *The Development of Mathematics.* McGraw–Hill, New York, 1940.

[9] A. H. Borning. The programming language aspects of ThingLab, a constrained oriented simulation laboratory. *ACM Trans. Progr. Lang. Syst.*, 3(4):353–387, 1981.

[10] A. H. Borning, R. Anderson, and B. Freeman-Benson. Indigo: A local propagation algorithm for inequality constraints. In *Proceedings of the 1996 ACM Symposium on User Interface Software and Technology*, pages 129–136, Seattle, Washington, USA, 1996. ACM Press.

[11] E. Bortolotti. *R Bombelli L'Algebra, Books I–V.* Milan, 1966.

[12] C. A. Boyer. *A History of Mathematics.* Wiley, New York, 1968.

[13] C. W. Brown. Simple CAD construction and its applications. *J. Symb. Comput.*, 31(5):521–547, 2001.

[14] C. W. Brown. Fast simplifications for Tarski formulas. In *Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*, pages 63–70. ACM Press, New York, 2009.

[15] C. W. Brown and H. Hong. QEPCAD — quantifier elimination by partial cylindrical algebraic decomposition. 2011. `http://www.cs.usna.edu/~qepcad/B/QEPCAD.html`.

[16] C. W. Brown and A. Strzeboński. Black-box/white-box simplification and applications to quantifier elimination. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 69–76. ACM Press, New York, 2010.

[17] B. D. Brüderlin. Using geometric rewrite rules for solving geometric problems symbolically. *Theoret. Comput. Sci.*, 116(2):291–303, 1993.

[18] H. Brunotte. On cubic CNS polynomials with three real roots. *Acta Sci. Math. (Szeged)*, 70(3-4):495–504, 2004.

[19] S. A. Buchanan and A. de Penington. Constraint definition system: a computer-algebra based approach to solve geoemtric constraint problems. *Comput. Aided Design*, 25(12):741–750, 1993.

[20] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem Nulldimensionalen Polynomideal.* Ph.D. Thesis, Mathematical Institute, University of Innsbruck, Austria, 1965.

[21] B. Buchberger. A theoretical basis for the reduction of polynomials to canonical forms. *ACM SIGSAM Bulletin*, 10(3):19–29, 1976.

[22] B. Buchberger. A criterion for detecting unnecessary reductions in the construction of Gröbner bases. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, pages 3–21, 1979.

[23] B. Buchberger. A critical-pair/completion algorithm for finitely generated ideals in rings. In *Proceedings of the Symposium on Logic and Machines: Decision Problems and Complexity*, pages 137–161. Springer-Verlag, London, 1984.

[24] B. Buchberger. *Multidimensional Systems Theory.* Reidel Publishing Company, 1985.

[25] B. Buchberger. Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements in the residue class ring modulo a zero dimensional polynomial ideal. *J. Symb. Comput.*, 41(3–4), 2006.

[26] J. F. Canny and I. Z. Emiris. Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symb. Comput.*, 20(2):117–149, 1995.

[27] G. Cardano. *Ars Magna.* Dover Publications, New York, 1993.

[28] C. Chen, J. H. Davenport, J. P. May, M. Moreno Maza, B. Xia, and R. Xiao. Triangular decomposition of semi-algebraic systems. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 187–194. ACM Press, New York, 2010.

[29] S. Chou. An introduction to Wu's method for mechanical theorem proving in geometry. *J. Automat. Reason.*, 4(3):237–267, 1988.

[30] S. Chou and X. Gao. Ritt-Wu's decomposition algorithm and geometry theorem proving. In *Proceedings of 10th International Conference on Automated Deduction*, volume 449, pages 207–220. Springer, 1990.

[31] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proceedings of the 2nd Conference on Automata Theory and Formal Languages*, pages 134–183. Springer-Verlag, Berlin Heidelberg, 1975.

[32] G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *J. Symb. Comput.*, 12(3):299–328, 1991.

[33] S. Corvez and F. Rouillier. Using computer algebra tools to classify serial manipulators. In *Automated Deduction in Geometry, Proceedings of the 4th International Workshop*, volume 2930, 2004.

[34] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Springer, USA, 2005.

[35] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra* (Third Edition). Springer, New York, 2007.

[36] C. D'Andrea and I. Z. Emiris. Hybrid sparse resultant matrices for bivariate systems. *J. Symb. Comput.*, 2002.

[37] R. Dixon. *Mathographics.* New York, Dover, 1991.

[38] M. Djorić and P. Janičić. Constructions, instructions, interactions. *Teach. Math. Appl.*, 23(2):69–88, 2004.

[39] A. Dolzmann and T. Sturm. Simplification of quantifier-free formulas over ordered fields. *J. Symb. Comput.*, 24(2):209–231, 1997.

[40] A. Dolzmann, T. Sturm, and V. Weispfenning. A new approach for automatic theorem proving in real geometry. *J. Automat. Reason.*, 21(3):357–380, 1998.

[41] C. Durand. *Symbolic and Numerical Techniques for Constraint Solving.* PhD Thesis, Computer Science, Purdue University, 1998.

[42] I. Z. Emiris. *Sparse Elimination and Applications in Kinematics.* Ph.D. Thesis, UC Berkeley, USA, 1994.

[43] E. B. Escott. Cubic congruences with three real roots. *Ann. Math.*, 11(2):86–92, 1910.

[44] H. Eves. *An Introduction to the History of Mathematics*. Saunders College Publishing, Philadelphia, PA, 1983.

[45] J. C. Faugère. A new efficient algorithm for computing Gröbner basis (F4). *J. Pure Appl. Algebra*, 139(1–3):61–88, 1999.

[46] J. C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 75–83, New York, 2002. ACM Press.

[47] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner basis by change of ordering. *J. Symb. Comput.*, 16(4):329–344, 1993.

[48] I. Fudos and C. M. Hoffmann. Correctness proof of a geometric constraint solver. *Internat. J. Comput. Geom. Appl.*, 6(4):405–420, 1996.

[49] I. Fudos and C. M. Hoffmann. A graph-constructive approach to solving systems of geometric constraints. *ACM Trans. Graph.*, 16(2):179–216, 1997.

[50] X. Gao, K. Jiang, and C. Zhu. Geometric constraint solving with conics and linkages. *Comput. Aided Design*, 34(6):421–433, 2002.

[51] X. Gao, Y. Luo, and C. Yuan. A characteristic set method for difference polynomial systems. *J. Symb. Comput.*, 44(3):242–260, 2009.

[52] X. Gao and D. Wang. *Mathematics Mechanization and Applications*. Academic Press, London, 2000.

[53] L. González-Vega. A combinatorial algorithm solving some quantifier elimination problems. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 300–316, Springer-Verlag, Wien New York, 1996.

[54] L. González-Vega, H. Lombardi, T. Recio, and M. F. Roy. Sturm-habicht sequence. In G. H. Gonnet, editor, *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation*, pages 136–146, ACM Press, New York, 1989.

[55] L. Guilbeau. The history of the solution of the cubic equation. *Math. News Lett.*, 5(4):8–12, 1930.

[56] A. Heydon and G. Nelson. *The Jnno-2 constraint-based drawing editor*. 1994.

[57] R. C. Hillyard and I. C. Braid. Characterizing non-ideal shapes in terms of dimensions and tolerances. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, pages 234–238, New York, USA.

[58] H. Hironaka. Resolution of singularities of an algebraic variety over a field of characteristic zero: I, II. *Ann. Math.*, 79(1–2):109–326, 1964.

[59] C. M. Hoffmann and R. Joan-Arinyo. A brief on constraint solving. *Comput. Aided Design Appl.*, 5(2):655–663, 2005.

[60] H. Hong. Quantifier elimination for formulas constrained by quadratic equations via slope resultants. *Comput. J.*, 36(5):440–449, 1993.

[61] H. Hong. Groebner basis under composition II. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 79–85, Zurich, 1996. ACM Press.

[62] H. Hong. Groebner basis under composition I. *J. Symb. Comput.*, 25(5):643–663, 1998.

[63] H. Hong, L. Li, T. Liang, and D. Wang. Solving dynamic geometric constraints involving inequalities. In J. Calmet, T. Ida, and D. Wang, editors, *Artificial Intelligence and Symbolic Computation*, volume 4120 of *LNAI*, pages 181–195, Springer-Verlag, Berlin Heidelberg, 2006.

[64] H. Hong, D. Wang, and T. Zhao. Solution formulas for quartic equations without or with constraints. 2011. In preparation.

[65] N. Jackiw. *The Geometer's Sketchpad v 4.0.* Key Curriculum Press, Berkeley, 2002.

[66] R. Joan-Arinyo and C. M. Hoffmann. A brief on constraint solving. 2005. `http://www.cs.purdue.edu/homes/~cmh/distribution/papers/Constraints/ThailandFull.pdf`.

[67] R. Joan-Arinyo and A. Soto-Riera. A correct rule-based geoemtric constraint solver. *Comput. Graph.*, 21(5):599–609, 1997.

[68] R. Joan-Arinyo and A. Soto-Riera. A ruler-and-compass geometric constraint solver. In M. Pratt, R. Sriram, and M. Wozny, editors, *Product Modeling for Computer Integrated Design and Manufacture*, pages 384–393, Chapman and Hall, London, 1997.

[69] L. W. Johnson and R. D. Riess. *Numerical Analysis.* Addison-Wesley, 1982.

[70] M. Kalkbrener. A generalized Euclidean algorithm for computing triangular representations of algebraic varieties. *J. Symb. Comput.*, 15(2):143–167, 1993.

[71] D. Kim, D. S. Kim, and K. Sugihara. Apollonius tenth problem via radius adjustment and Mobius transformations. *Comput. Aided Design*, 38(1):14–21, 2006.

[72] K. Kondo. PIGMOD: Parametric and interactive geoemtric modeller for mechanical design. *Comput. Aided Design*, 22(10):633–644, 1990.

[73] K. Kondo. Algebraic method for manipulation of dimensional relationships in geometric models. *Comput. Aided Design*, 24(3):141–147, 1992.

[74] U. Kortenkamp. *Foundations of Dynamic Geometry*. PhD thesis, ETH Zürich, 1999.

[75] G. Kramer. *Solving Geometric Constraints Systems*. MIT Press, 1992.

[76] G. A. Kramer. Using degrees of freedom analysis to solve geometric constraint systems. In J. Rossignac and J. Turner, editors, *Symposium on Solid Modeling Foundations and CAD/CAM Applications*, pages 371–378, Austin, TX, 1991. ACM Press.

[77] G. A. Kramer. A geometric constraint engine. *Artif. Intell.*, 58(1-3):327–360, 1992.

[78] G. Kwaiter, V. Gaildrat, and R. Caubet. Interactive constraint system for solid modeling objects. In C. Hoffmann and J. Rossignac, editors, *Fourth Symposium on Solid Modeling and Applications*, pages 265–270. ACM SIGGRAPH, 1997.

[79] J. L. Lagrange. Réflexions sur la résolution algébrique des équations. In *Nouveaux Mémoires de l'Académie royale des Sciences et Belles-Lettres de Berlin*, pages 205–421, Oeuvres complètes, tome 3, 1770.

[80] H. Lamure and D. Michelucci. Solving geometric constraints by homotopy. In C. Hoffmann and J. Rossignac, editors, *Third Symposium on Solid Modeling and Applications*, pages 17–19, Utah USA, 1995. ACM Press.

[81] D. Lazard. Résolution des systèmes d'équations algébriques. *Theor. Comput. Sci.*, 15:77–110, 1981.

[82] D. Lazard. Gröbner-bases, Gaussian elimination and resolution of systems of algebraic equations. In *Proceedings of EUROCAL 1983*, pages 146–156, London, 1983.

[83] D. Lazard. A new method for solving algebraic systems of positive dimension. *Discrete Appl. Math.*, 33(1-3):147–160, 1991.

[84] D. Lazard. Solving zero-dimensional algebraic systems. *J. Symb. Comput.*, 13(2):117–132, 1992.

[85] D. Lazard and F. Rouillier. Solving parametric polynomial systems. *J. Symb. Comput.*, 42(6):636–667, 2007.

[86] J. Y. Lee and K. Kim. Geometric reasoning for knowledge-based parametric design using graph representations. *Comput. Aided Design*, 28(10):831–841, 1996.

[87] R. H. Lewis and S. Bridgett. Conic tangency equations and Apollonius problems in biochemistry and pharmacology. *Math. Comput. Simul.*, 61(2):101–114, 2003.

[88] T. Y. Li. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numer.*, 6:399–436, 1997.

[89] T. Y. Li. Solving polynomial systems by polyhedral homotopoies. *Taiwan. J. Math.*, 3(3):251–279, 1999.

[90] S. Liang and J. Zhang. A complete discrimination system for polynomials with complex coefficients and its automatic generation. *Sci. China (Ser. E)*, 42(2):113–128, 1998.

[91] R. Light and D. Gossard. Modification of geometric models through variational geometry. *Comput. Aided Design*, 14:209–214, 1982.

[92] V. C. Lin, D. Gossard, and R. A. Light. Variational geometry in computer-aided design. *ACM Comput. Graph.*, 15(3):171–177, 1981.

[93] J. Liu, Z. Liu, and M. Wang. The term orderings which are compatible with composition II. *J. Symb. Comput.*, 35(2):153–168, 2003.

[94] F. S. Macaulay. *The Algebraic Theory of Modular Systems.* Stechert-Hafner Service Agency, New York London, 1964.

[95] B. Mishra. *Algorithmic Algebra.* Springer, New York, 1993.

[96] H. M. Möller and B. Buchberger. The construction of multivariate polynomials with preassigned zeros. In *Proceedings of EUROCAM 1982*, pages 24–31, Marseille, 1982.

[97] H. M. Möller and H. J. Stetter. Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. *Numer. Math.*, 70:311–329, 1995.

[98] H. M. Möller and R. Tenberg. Multivariate polynomial system solving using intersections of eigenspaces. *J. Symb. Comput.*, 32:513–531, 2001.

[99] J. Nathan. *Basic Algebra.* Dover, 2009.

[100] G. Nelson. Juno, a constraint-based graphics system. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, pages 235–224, San Francisco, July 22-26, 1985.

[101] O. Neugebauer. *The Exact Science in Antiquity, 2nd ed.* Brown University Press, Porvidence, RI, 1957.

[102] M. Noro and K. Yokoyama. A modular method to compute the rational univariate representation of zero-dimensional ideals. *J. Symb. Comput.*, 28(1–2):243–264, 1999.

[103] J. J. O'Connor and E. F. Robertson. Rafael Bombelli. In *MacTutor History of Mathematics Archive*, Unversity of St. Andrew, Soctland, 2010.

[104] J. C. Owen. Algebraic solution for geoemtry from dimensional constraints. In J. Rossignac and J. Turner, editors, *Symposium on Solid Modeling Foundations and CAD/CAM Applications*, pages 397–407, Austin, TX, 1991. ACM Press.

[105] W. Pan and D. Wang. Uniform Gröbner bases for ideals generated by polynomials with parametric exponents. In *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*, pages 269–276, Genova, 2006.

[106] M. P. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing.* Springer Verlag, 2002.

[107] M. Riccardi. Solution of cubic and quartic equations. *Formalized Math.*, 17(2):117–122, 2009.

[108] J. Richter-Gebert and U. Kortenkamp. *The Interactive Geometry Software Cinderella.* Springer, Berlin, 1999.

[109] J. F. Ritt. *Differential Equations from the Algebraic Standpoint.* American Mathematical Society, New York, 1932.

[110] J. F. Ritt. *Differential Algebra.* American Mathematical Society, New York, 1950.

[111] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Appl. Algebr. Eng. Comm.*, 9(5):433–461, 1999.

[112] O. W. Salomons, F. van Slooten, F. van Houten, and H. Kals. Conceptual graphs in constraint based re-design. In C. Hoffmann and J. Rossignac, editors, *Third Symposium on Solid Modeling and Applications*, pages 17–19, Salt Lake City, Utah USA, 1995. ACM Press.

[113] A. I. Shirshov. Certain algorithmic problems for lie algebras. *ACM SIGSAM Bulletin*, 33(2):3–6, June 1999.

[114] D. E. Smith. *History of Mathematics (Vol. 2).* Dover Publications, New York, 2003.

[115] W. Sohrt and B. D. Brüderlin. Interaction with constraints in 3D modeling. *Inter. J. Comput. Geom. Appl.*, 1(4):405–425, 1991.

[116] L. Solano and P. Brunet. Constructive constraint-based model for parametric CAD systems. *Comput. Aided Design*, 26(8):614–621, 1994.

[117] A. Strzeboński. Computation with semialgebraic sets represented by cylindrical algebraic formulas. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 61–68. ACM Press, New York, 2010.

[118] G. Sunde. A CAD system with declarative specification of shape. In *Eurographics Workshop on Intelligent CAD Systems*, pages 90–105, 1987.

[119] I. Sutherland. Sketchpad, a man-machine graphhical communication system. In *Proceedings of the Spring Joint Computation Conference*, pages 329–345. IFIPS, 1963.

[120] A. Tarski. *A Decision Method for Elementary Algebra and Geometry* (Second Edition). University of California Press, Berkeley Los Angeles, 1951.

[121] B. van der Waerden. *Modern Algebra.* F. Ungar Publishing Co., New York, USA, 1950.

[122] A. Verroust, F. Schonek, and D. Roller. Rule-oriented method for parameterized computer-aided design. *Comput. Aided Design*, 24(10):531–540, 1992.

[123] D. Wang. A new theorem discovered by computer prover. *J. Geom.*, 36(1–2):173–182, 1989.

[124] D. Wang. An elimination method for polynomial systems. *J. Symb. Comput.*, 16(2):83–114, 1993.

[125] D. Wang. GEOTHER: a geometry theorem prover. In *Proceedings of CADE 1996*, 1996.

[126] D. Wang. Decomposing polynomial systems into simple systems. *J. Symb. Comput.*, 25(3):295–314, 1998.

[127] D. Wang. Computing triangular systems and regular systems. *J. Symb. Comput.*, 30(2):221–236, 2000.

[128] D. Wang. *Elimination Methods.* Springer, Wien, New York, 2001.

[129] D. Wang. Automated generation of diagrams with Maple and Java. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 277–287, Springer-Verlag, Berlin Heidelberg, 2003.

[130] D. Wang. *Elimination Practice: Software Tools and Applications.* Imperial College Press, London, 2004.

[131] D. Wang. GEOTHER 1.1: Handling and proving geometric theorems automatically. In F. Winkler, editor, *Automated Deduction in Geometry*, volume 2930 of *LNAI*, pages 194–215, Springer-Verlag, Berlin Heidelberg, 2004.

[132] M. Wang and Z. Liu. Remarks on Gröbner basis for ideals under composition. In *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*, pages 237–244, London, Ontario, Canada, 2001. ACM Press.

[133] V. Weispfenning. A new approach to quantifier elimination for real algebra. In B. F. Caviness and J. R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition.*

[134] V. Weispfenning. The complexicity of linear problems in fields. *J. Symb. Comput.*, 5(1–2):3–27, 1988.

[135] V. Weispfenning. Comprehensive Gröbner bases. *J. Symb. Comput.*, 14(1):1–29, July 1992.

[136] V. Weispfenning. Quantifier elimination for real algebra — the cubic case. In *Proceedings of the 1994 International Symposium on Symbolic and Algebraic Computation*, pages 258–263, ACM Press, New York, 1994.

[137] V. Weispfenning. Quantifier elimination for real algebra — the quadratic case and beyond. *Appl. Algebra Eng. Commun. Comput.*, 8(2):85–101, 1997.

[138] F. Winkler. A p-adic approch to the computation of Gröbner bases. *J. Symb. Comput.*, 6(2–3):287–304, 1988.

[139] F. Winkler, B. Buchberger, F. Lichtenberger, and H. Rolletschek. An algorithm for constructing canonical bases of polynomial ideals. *ACM Trans. Math. Softw.*, 11(1):66–78, 1985.

[140] W.-T. Wu. Basic principles of mechanical theorem proving in elementary geometries. *J. Automat. Reason.*, 2(3):221–252, 1986.

[141] W.-T. Wu. On zeros of algebraic equations — an application of Ritt principle. *Kexue Tongbao*, 31(1):1–5, 1986.

[142] W.-T. Wu. A zero structure theorem for polynomial equations-solving. *Math. Mech. Res. Preprints*, 1:2–12, 1987.

[143] W.-T. Wu. Mechanical theorem proving in geometries. In B. Buchberger and G. E. Collins, editors, *Texts and Monographs in Symbolic Computations*, Springer-Verlag, 1994.

[144] W.-T. Wu. *Mathematics Mechanization*. Science Press/Kluwer Academic, Beijing, 2000.

[145] Y. Yamaguchi and F. Kimura. A constraint modeling system for variational geometry. In J. U. Turner, M. J. Wozny, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 147–156, Elservier North Holland, 1990.

[146] L. Yang, X. Hou, and Z. Zeng. An alternative algorithm for determining the number of real roots of a polynomial. In *Proceedings of the International Workshop on Logic and Software Engineering*, pages 151–161, Would Scientific, Singapore, 1996.

[147] L. Yang, X. Hou, and Z. Zeng. A complete discrimination system for polynomials. *Sci. China (Ser. E)*, 39(6):628–646, 1996.

[148] L. Yang and B. Xia. Explicit criterion to determine the number of positive roots of a polynomial. *MM Res. Preprints*, (15):134–145, 1995.

[149] L. Yang and J. Zhang. Searching dependency between algebraic equations: An algorithm applied to automated reasoning. In J. Johnson, S. McKee, and A. Vella, editors, *Artificial Intelligence in Mathematics*, pages 147–156, Oxford, 1994. Oxford University Press.

[150] L. Yang, J. Zhang, and X. Hou. *Non-Linear Equation Systems and Automated Theorem Proving (in Chinese)*. Shanghai Press of Science Technology and Education, Shanghai, 1996.

[151] C. Yap. *Fundamental Problems in Algorithmic Algebra*. Oxford University Press, USA, 1999.

[152] Z. Ye, S. Chou, and X. Gao. Visually dynamic presentation of proofs in plane geometry — part 1. basic features and the manual input method. *J. Automat. Reason.*, 45(3):213–241, 2010.

[153] T. Zhao. Software geodraw for generating dynamic diagrams with inequality constraints (in Chinese). *Comput. Appl. Softw.*, 2012. Accepted.

[154] T. Zhao, D. Wang, and H. Hong. Solution formulas for cubic equations without or with constraints. *J. Symb. Comput.*, 46(8):904–918, 2011.

[155] T. Zhao, D. Wang, H. Hong, and P. Aubry. Real solution formulas of cubic and quartic equations applied to generate dynamic diagrams with inequality constraints. In *Proceeding of the 27th Symposium On Applied Computing*, Italy, 2012. ACM Press.

[156] M. Zhou and F. Winkler. Computing difference-differencial dimension polynomials by relative Gröbner bases in difference-differencial modules. *J. Symb. Comput.*, 43(10):726–745, 2008.

Appendix A

Predicates for Geometric Relations of Constructive Type in GeoDraw

Predicate	Instance	Meaning
angle(3)	angle(A,B,C)	Generate an angle ABC.
bisectorline(2)	bisectorline(A,B)	Generate the perpendicular bisector of the line AB.
centroid(3)	centroid(A,B,C)	Generate the centroid of the triangle ABC.
circle(2-3)	circle(A,B,C)/circle(A,B) /circle(A,r)	Generate a circle with three points, or with the center A and through B, or with the center A and radius r.
circumcenter(3)	circumcenter (A,B,C)	Generate the circumcenter of the triangular ABC.
fixedpoint(2)	fixedpoint(x,y)	Generate a point with the fixed coordinates.
inbesector(3)	inbesector(A,B,C)	Generate a random point on the inner bisector of the angle ABC.
intersection(2)	intersection(line,line) intersection(circle,circle) intersection(line,circle)	Generate the points intersected by two lines, two circles or a line and a circle.
line(2)	line(A,B)/line(A,r)	Generate a line by two points, or by a point and the slope.
midpoint(2)	midpoint(A,B)	Generate the mid-point of the point A and the point B.
onperpbesector (3)	onperpbesctor(A,B)	Generate a random point on the perpendicular bisector of the line AB.

orthocenter(3)	orthocenter(A,B,C)	Generate the orthocenter of the triangle ABC.
paraline(2)	paraline(line,A)	Generate a line which is parallel to the given line and through the given point A.
perpfoot(2)	perpfoot(line,line)	Generate the perpendicular foot of two lines.
perpline(2)	perpline(line,A)	Generate a line which is perpendicular to the given line and through the given point A.
point(2)	point(x,y)	Generate a free point with the given initial value.
pointin(1)	pointin(circle)	Generate a point inside the circle.
pointon(1)	pointon(line/segment /circle)	Generate a random point on a line, a segment or a circle.
pointondiffside (2)	pointondiffside(line,A)	Generate a point which is on the different side of the given line with the point A.
pointonsameside (2)	pointonsameside (line,A)	Generate a point which is on the same side of the given line as the point A.
pointout(1)	pointout(circle)	Generate a point outside the circle.
reflection(2-3)	reflection(A,B,C) /reflection(line,C)	Generate the reflection point of the point C with respect to the line AB.
rotate(3)	rotate(A,B,n)	Rotate the point B with respect to the point A anti-clockwise by degree n.
segment(2)	segment(A,B)	Generate a segment by two points.

tangentline(2)	tangentline(circle,A)	Generate the tangent line of the circle through the point A (A is on or out the circle).
tangentpoint(2)	tangentpoint(circle,circle) /tangentpoint(line,circle)	Generate the tangent point of two circles, or a line and a circle.
triangle(3)	triangle(A,B,C)	Generate a triangle ABC.

Appendix B

Predicates for Geometric Relations of Declarative Type in GeoDraw

Predicate	Instance	Meaning
area(3)	area(A,B,C)	Compute the area of the triangle ABC.
centroid(4)	centroid(A,B,C,D)	The point D is the centroid of the triangle ABC.
circumcenter(4)	circumcenter (A,B,C,D)	The point D is the circumcenter of the triangular ABC.
cocircular(4)	cocircular(A,B,C,D)	The four points A, B, C and D are cocircular.
collinear(3)	collinear(A,B,C)	The three points A, B and C are collinear.
concurrent(6)	concurrent (A,B,C,D,E,F)	The three lines AB, CD and EF are concurrent.
equal(2)	equal(A,B)	The point A is equal to the point B.
equiangle(6-8)	equiangle (A,B,C,D,E,F,2,3)	Two times the angle ABC is equal to three times the angle DEF.
equidistance(4)	equidistance (A,B,C,D)	The distance between the point B and the point A is equal to the distance between the point C and the point D.
extangent(4)	extangent(A,B,C,D)	The circle centred at A and passing through B is external tangent to the circle centred at C and passing through D.

inbesector(4)	inbesector(A,B,C,D)	The point D is on the inner bisector of the angle ABC.
intersection(5)	intersection(A,B,C,D,E)	The two lines AB and CD intersect at the point E.
inrectangle(3)	inrectangle(A,B,C)	The point C is inside the rectangle with AB as diagonal.
midpoint(3)	midpoint(A,B,C)	The point C is the mid-point of A and B.
oncirc(3)	oncirc(A,B,C)	The point C is on the circle centred at A and passing through B.
oncircle(4)	oncircle(A,B,C,D)	The point D is on the circumcircle of the triangle ABC.
online(3)	online(A,B,C)	The point C is on the line AB.
onperpbesector (3)	onperpbesctor(A,B,C)	The point C is on the perpendicular bisector of the line AB.
orthocenter(4)	orthocenter(A,B,C,D)	The point D is the orthocenter of the triangle ABC.
parallel(4)	paraline(A,B,C,D)	The line AB is parallel to the line CD.
perpendicular (4)	perpendicular(A,B,C,D)	The line AB is perpendicular to the line CD.
perpfoot(5)	perpfoot(A,B,C,D,E)	The point E is the perpendicular foot of the line AB to the line CD.
pointondiffside (4)	pointondiffside (A,B,C,D)	The points C and D are on the different side of the line AB.
pointonsameside (4)	pointonsameside (A,B,C,D)	The points C and D are on the same side of the line AB.

reflection(4)	reflection(A,B,C,D)	The point D is the reflection point of the point C with respect to the line AB.
sdistance(2)	sdistance(A,B)	The square of distance between the two points A and B.

Appendix C
Geometric Configurations

In addition to the examples given in the main text, this appendix lists 10 examples with details used for experimental test.

Example C.1 (M. Paterson) Three similar isosceles triangles, A_1BC, AB_1C, and ABC_1, are erected internally on the three respective sides, BC, CA, AB, of a triangle ABC, then AA_1, BB_1 and CC_1 are concurrent.

To ensure that the triangles A_1BC, AB_1C, and ABC_1 are drawn all inward, we need to use inequality constraints. Without loss of generality, let the coordinates of the points be assigned as follows:

$$A(0,0),\ B(u_4,0),\ C(u_2,u_3),\ C_1(x_1,u_1),\ B_1(x_3,x_2),\ A_1(x_5,x_4).$$

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases} F_1 = 2\,u_4x_1 - u_4^2 = 0, & (|AC_1| = |BC_1|) \\ F_2 = -u_1x_2u_3 - u_1u_2x_3 + u_1u_2u_4 + u_1u_4x_3 - u_1u_4^2 & \\ \qquad - x_1u_3x_3 + x_1u_4u_3 + x_1u_2x_2 - x_1u_4x_2 = 0, & (\tan\angle BAC_1 = \tan\angle CBA_1) \\ F_3 = -2\,u_4x_3 + u_4^2 + 2\,u_2x_3 - u_2^2 + 2\,x_2u_3 - u_3^2 = 0, & (|A_1B| = |A_1C|) \\ F_4 = -u_1x_4u_3 + u_1u_3^2 - u_1u_2x_5 + u_1u_2^2 & \\ \qquad - x_1u_3x_5 + x_1u_2x_4 = 0, & (\tan\angle BAC_1 = \tan\angle ACB_1) \\ F_5 = 2\,u_2x_5 - u_2^2 + 2\,x_4u_3 - u_3^2 = 0, & (|AB_1| = |CB_1|) \\ G_1 = -u_3x_2u_4^2 - u_3^2x_3u_4 + u_4^2u_3^2 + u_2x_2u_3u_4 > 0, & (BCA_1 \text{ inward}) \\ G_2 = u_4u_3(u_3x_5 - u_2x_4) > 0, & (ACB_1 \text{ inward}) \\ G_3 = u_3u_1u_4^2 > 0. & (ABC_1 \text{ inward}) \end{cases}$$

Assume that B, C_1 are semi-free points and C is a free point, so that $u_1,\ldots,u_4$ are free parameters. The set of polynomials $\{F_1,\ldots,F_5\}$ may be decomposed over

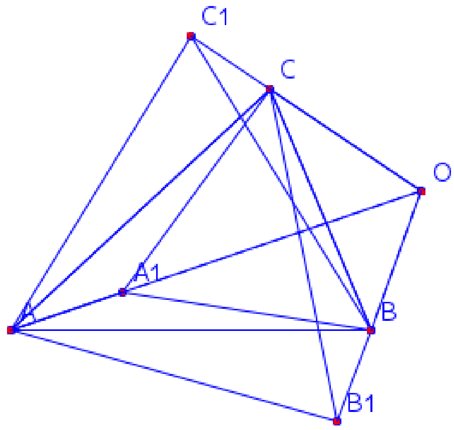

Figure 1: M. Paterson Configuration

$\mathbb{Q}(u_1,\ldots,u_4)$ into one triangular set $[T_1,\ldots,T_5]$ with

$$
\begin{aligned}
T_1 &= 2\,x_1 - u_4,\\
T_2 &= 2\,u_1u_2 - 2\,u_4u_1 + u_4u_3 - 2\,u_4x_2,\\
T_3 &= -2\,u_1x_2u_3 - 2\,u_1u_2x_3 + 2\,u_1u_2u_4 + 2u_1u_4x_3 - 2\,u_1u_4^2 - u_4u_3x_3\\
&\quad + u_4^2u_3 + u_4u_2x_2 - u_4^2x_2,\\
T_4 &= 2\,x_4 - 2\,u_1x_2 - u_2,\\
T_5 &= 2\,x_5 - 2\,u_2x_2 - u_1 - 1.
\end{aligned}
$$

From this triangular set, using the solution formulas of linear equations with inequality constraints, and eliminating the quantifiers, we obtain the following weak SRR:

$$
\begin{aligned}
\Delta(u_1,\ldots,u_4),\quad x_1 &= \frac{u_4}{2},\quad x_2 = \frac{-2\,u_4u_1 + 2\,u_1u_2 + u_4u_3}{2\,u_4},\\
x_3 &= \frac{-2\,u_1u_4^2 + u_4^2u_3 + 2\,u_4u_1u_2 - u_4^2x_2 - 2\,u_1x_2u_3 + u_2x_2u_4}{-2\,u_4u_1 + 2\,u_1u_2 + u_4u_3},\\
x_4 &= \frac{2\,u_1u_2 - u_4u_3}{2\,u_4},\quad x_5 = \frac{u_2x_4u_4 - 2\,u_1x_4u_3 + 2\,u_1u_3^2 + 2\,u_1u_2^2}{2\,u_1u_2 + u_4u_3},
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta(u_1,\ldots,u_4):\ & u_4 \neq 0 \wedge 2\,u_2u_1 + u_4u_3 \neq 0 \wedge 2\,u_2u_1 - 2\,u_4u_1 + u_4u_3 \neq 0\\
& \wedge[(u_3 > 0 \wedge u_1 > 0) \vee (u_3 < 0 \wedge u_1 < 0)].
\end{aligned}
$$

With these solution representations, the values of x_i may be easily computed for any given real values of $u_1,\ldots,u_4$ and the corresponding dynamic diagram as shown in

Figure 1 can be drawn and animated efficiently.

Example C.2 (Napoleon Theorem) Draw an arbitrary triangle ABC, three equilateral triangles ABC_1, ACB_1, and BCA_1 all inward to or all outward from $\triangle ABC$, and the circumcircles of $\triangle ABC_1$, $\triangle ACB_1$, and $\triangle BCA_1$. Connect the centers of the three circles.

The Napoleon theorem states that the three circumcenters O_1, O_2, and O_3 form an equilateral triangle. To express the order relation that $\triangle ABC_1$, $\triangle ACB_1$, and $\triangle BCA_1$ are all inward to or all outward from $\triangle ABC$, we need to use inequalities.

Without loss of generality, let the coordinates of the points be assigned as follows:

$$A(0,0),\ B(1,0),\ C(u_1,u_2),\ O_1(x_1,x_2),\ O_2(x_3,x_4),\ O_3(x_5,x_6),$$
$$C_1(x_7,x_8),\ B_1(x_9,x_{10}),\ A_1(x_{11},x_{12}).$$

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases}
F_1 = 2\,x_1 - 1 = 0, & (O_1 \text{ circumcenter of } \triangle ABC_1)\\
F_2 = 2\,x_1x_7 - 2\,x_1 - x_7^2 + 2\,x_2x_8 - x_8^2 + 1 = 0, & \\
F_3 = -2\,x_3u_1 + u_1^2 - 2\,u_2x_4 + u_2^2 = 0, & (O_2 \text{ circumcenter of } \triangle ACB_1)\\
F_4 = 2\,x_3x_9 - 2\,x_3u_1 - x_9^2 + u_1^2 + 2\,x_4x_{10} - 2\,u_2x_4 - x_{10}^2 + u_2^2 = 0, & \\
F_5 = 2\,x_5 - 1 - 2\,x_5u_1 + u_1^2 - 2\,u_2x_6 + u_2^2 = 0, & (O_3 \text{ circumcenter of } \triangle BCA_1)\\
F_6 = 2\,x_5x_{11} - 2\,x_5u_1 - x_{11}^2 + u_1^2 + 2\,x_6x_{12} - 2\,u_2x_6 - x_{12}^2 + u_2^2 = 0, & \\
F_7 = 2\,x_7 - 1 = 0, & (|AC_1| = |BC_1|)\\
F_8 = x_7^2 + x_8^2 - 1 = 0, & (|AC_1| = |AB|)\\
F_9 = x_9^2 + x_{10}^2 - u_2^2 - u_1^2 = 0, & (|AB_1| = |AC|)\\
F_{10} = 2\,u_2x_{10} + 2\,u_1x_9 - u_2^2 - u_1^2 = 0, & (|AB_1| = |CB_1|)\\
F_{11} = x_{12}^2 + x_{11}^2 - 2\,x_{11} - u_2^2 - u_1^2 + 2\,u_1 = 0, & (|BA_1| = |BC|)\\
F_{12} = 2\,u_2x_{12} + 2\,u_1x_{11} - 2\,x_{11} - u_1^2 - u_2^2 + 1 = 0 & (|BA_1| = |CA_1|)\\
G_1 = u_1u_2x_{12} - u_2x_{12} - u_2^2x_{11} + u_2^2 < 0, & (BCA_1 \text{ outward})\\
G_2 = -u_1u_2x_{10} + u_2^2x_9 < 0, & (ACB_1 \text{ outward})\\
G_3 = u_2x_8 < 0. & (ABC_1 \text{ outward})
\end{cases}$$

Assume that C is a free point, so that u_1, u_2 are free parameters. The set $\{F_1, \ldots, F_{12}\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2)$ into four irreducible triangular sets. However, three of them turn out to be inconsistent with the inequality constraints. So we get only one triangular set $[T_1, \ldots, T_{12}]$, where

$$T_1 = 2\,x_1 - 1, \qquad T_2 = 12\,x_2^2 - 1, \qquad T_3 = 2\,x_3 - u_1 - 2\,u_2x_2,$$
$$T_4 = 2\,u_2x_4 + 2\,x_3u_1 - u_1^2 - u_2^2, \qquad T_5 = 2\,x_5 + 2\,u_2x_2 - u_1 - 1,$$
$$T_6 = 2\,u_2x_6 - 2\,x_5 + 2\,x_5u_1 - u_1^2 - u_2^2 + 1, \qquad T_7 = 2\,x_7 - 1,$$
$$T_8 = 4\,x_2x_8 - 1, \qquad T_9 = (2\,u_2x_3 - 2\,u_1x_4)x_9 - u_2^2x_4 - 2\,u_2x_3u_1 + u_1^2x_4,$$
$$T_{10} = 2\,x_{10}u_2 + 2\,u_1x_9 - u_1^2 - u_2^2,$$
$$T_{11} = (2\,u_2x_5 - 2\,u_2 - 2\,u_1x_6 + 2\,x_6)x_{11} - 2\,u_2x_5u_1 - x_6 + u_1^2x_6 - u_2^2x_6 + 2\,u_1u_2,$$
$$T_{12} = 2\,u_2x_{12} - 2\,x_{11} + 2\,u_1x_{11} - u_1^2 - u_2^2 + 1.$$

From this triangular set, using the solution formulas of linear and quadratic equations with inequality constraints, and eliminating the quantifiers, we obtain the following weak SRRs:

$$\begin{aligned} u_2 < 0,\ \ & x_1 = \frac{1}{2},\ x_2 = \frac{\sqrt{3}}{6},\ x_3 = X_3,\ x_4 = X_4,\ x_5 = X_5,\ x_6 = X_6,\ x_7 = \frac{1}{2},\\ & x_8 = \frac{1}{4\,x_2},\ x_9 = X_9,\ x_{10} = X_{10},\ x_{11} = X_{11},\ x_{12} = X_{12};\\ u_2 > 0,\ \ & x_1 = \frac{1}{2},\ x_2 = -\frac{\sqrt{3}}{6},\ x_3 = X_3,\ x_4 = X_4,\ x_5 = X_5,\ x_6 = X_6,\ x_7 = \frac{1}{2},\\ & x_8 = \frac{1}{4\,x_2},\ x_9 = X_9,\ x_{10} = X_{10},\ x_{11} = X_{11},\ x_{12} = X_{12}, \end{aligned}$$

where

$$X_3 = \frac{1}{2}u_1 + u_2x_2, \qquad X_4 = \frac{-2\,u_1x_3 + u_2^2 + u_1^2}{2\,u_2},$$
$$X_5 = \frac{1}{2}u_1 - u_2x_2 + \frac{1}{2}, \qquad X_6 = \frac{2\,x_5 - 2\,u_1x_5 + u_1^2 + u_2^2 - 1}{2\,u_2},$$
$$X_9 = \frac{-u_2^2x_4 - 2\,u_2x_3u_1 + u_1^2x_4}{2(-u_2x_3 + u_1x_4)}, \qquad X_{10} = \frac{-2\,u_1x_9 + u_1^2 + u_2^2}{2\,u_2},$$
$$X_{11} = \frac{-u_2^2x_6 - x_6 + 2\,u_1u_2 - 2\,u_2x_5u_1 + u_1^2x_6}{2(u_2 - u_2x_5 + u_1x_6 - x_6)},$$
$$X_{12} = \frac{2\,x_{11} + u_1^2 - 2\,u_1x_{11} + u_2^2 - 1}{2\,u_2}.$$

With these solution representations, the values of x_i may be easily computed for any given real values of u_1, u_2 with $u_2 \neq 0$ and the corresponding dynamic diagram as shown in Figure 6.4 can be drawn and animated efficiently.

Example C.3 (Square Steiner) On the three sides AB, AC and BC of triangle ABC, three squares $ABIH$, $ACDE$ and $BCFG$ are drawn outward.

To ensure that the squares $ABIH$, $ACDE$ and $BCFG$ are drawn all outward, we need to use inequality constraints. Without loss of generality, let the coordinates of the points be assigned as follows:

$$A(u_1, 0),\ B(u_2, u_3),\ C(0,0),\ D(0, x_1),\ E(u_1, -u_1),\ F(x_2, x_3),\ H(x_4, x_5).$$

Note that here we do not assign the coordinates of points G and I. The reason is that once the real values of the points A, B, C, F and H's coordinates are given, it is easy to compute the coordinates of the reflection points G and I by numerical computation only. Therefore, we do not involve the computation of the coordinates of G and I in the symbolic computation. Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases} F_1 = u_2^2 + u_3^2 - x_2^2 - x_3^2 = 0, & (|BC| = |CF|) \\ F_2 = x_1^2 - u_1^2 = 0, & (|CD| = |AC|) \\ F_3 = -x_2 u_2 - x_3 u_3 = 0, & (CF \perp BC) \\ F_4 = -2\,u_1 u_2 + u_2^2 + u_3^2 + 2\,u_1 x_4 - x_4^2 - x_5^2 = 0, & (|AB| = |AH|) \\ F_5 = u_2 x_4 - u_1 u_2 - u_1 x_4 + u_1^2 + u_3 x_5 = 0, & (AB \perp AH) \\ G_1 = u_1 x_5 u_3 u_2 - u_1^2 x_5 u_3 - u_1 x_4 u_3^2 + u_1^2 u_3^2 < 0, & (ABIH \text{ outward}) \\ G_2 = u_3 x_1 u_1^2 < 0, & (ACDE \text{ outward}) \\ G_3 = -u_1 x_3 u_3 u_2 + u_1 x_2 u_3^2 < 0. & (BCFG \text{ outward}) \end{cases}$$

Assume that A, E are semi-free points and B is a free point, so that $u_1,\ u_2,\ u_3$ are free parameters. The set of polynomials $\{F_1, \ldots, F_5\}$ may be decomposed over $\mathbb{Q}(u_1, u_2, u_3)$ into eight triangular sets. From these triangular sets, using the solution formulas of linear and quadratic equations with inequalty constraints, and eliminating

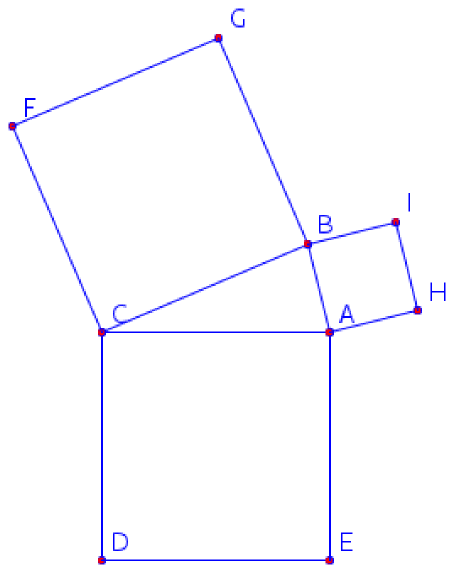

Figure 2: Square Steiner Configuration

the quantifiers, we obtain the following two weak SRRs:

$$\Delta_1(u_1,u_2,u_3),\;\; x_1=u_1,\;\; x_2=u_3,\;\; x_3=-u_2,\;\; x_4=-u_3+u_1,$$
$$x_5=\frac{-u_2x_4+u_1u_2+u_1x_4-u_1^2}{u_3};$$
$$\Delta_2(u_1,u_2,u_3),\;\; x_1=-u_1,\;\; x_2=-u_3,\;\; x_3=u_2,\;\; x_4=u_3+u_1,$$
$$x_5=\frac{-u_2x_4+u_1u_2+u_1x_4-u_1^2}{u_3},$$

where

$$\Delta_1(u_1,u_2,u_3):\;\; (u1>0\wedge u3<0)\vee(u1<0\wedge u3>0);$$
$$\Delta_2(u_1,u_2,u_3):\;\; (u1>0\wedge u3>0)\vee(u1<0\wedge u3<0).$$

With these solution representations, the values of x_i may be easily computed for any given real values of u_1, u_2, u_3. The coordinates of G and I can be directly computed with the values of u_j and x_i, and the corresponding dynamic diagram as shown in Figure 2 can be drawn and animated efficiently.

Example C.4 (Butterfly Theorem) Given a circle O and A, B, C, D four points on the circle. E is the intersection of line AC and line BD. Through E draw a line perpendicular to OE, meeting AD at F and BC at G. The theorem states that $FE \equiv GE$.

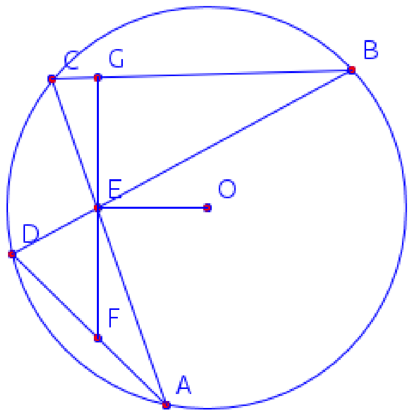

Figure 3: Butterfly Configuration

To express the order relation of the points and make the diagram like a butterfly, we need to use inequalities. Without loss of generality, let the coordinates of the points be assigned as follows:

$$E(0,0),\ O(u_1,0),\ A(u_2,u_3),\ B(x_1,u_4),\ C(x_3,x_2),\ D(x_5,x_4),\ F(0,x_6),\ G(0,x_6).$$

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases} F_1 = u_2^2 - 2\,u_2u_1 + u_3^2 - x_1^2 + 2\,x_1u_1 - u_4^2 = 0, & (|OA| = |OB|) \\ F_2 = u_2^2 - 2\,u_2u_1 + u_3^2 - x_3^2 + 2\,x_3u_1 - x_2^2 = 0, & (|OA| = |OC|) \\ F_3 = u_2^2 - 2\,u_2u_1 + u_3^2 - x_5^2 + 2\,x_5u_1 - x_4^2 = 0, & (|OA| = |OD|) \\ F_4 = u_3x_3 - u_2x_2 = 0, & (A,\ C,\ E \text{ collinear}) \\ F_5 = u_4x_5 - x_1x_4 = 0, & (B,\ D,\ E \text{ collinear}) \\ F_6 = x_6x_5 - u_3x_5 + u_2x_4 - u_2x_6 = 0, & (A,\ D,\ F \text{ collinear}) \\ F_7 = x_7x_3 - u_4x_3 + x_1x_2 - x_1x_7 = 0, & (B,\ C,\ G \text{ collinear}) \\ G_1 = u_3x_2u_1^2 < 0, & (A,\ B,\ C,\ D \text{ ordered}) \\ G_2 = u_4x_4u_1^2 < 0. & \end{cases}$$

Assume that A is a free point and B, O are semi-free points, so that $u_1,\ldots,u_4$ are free parameters. The set $\{F_1,\ldots,F_7\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1\ldots,u_4)$ into four irreducible triangular sets. From these triangular sets, using the solution formulas of linear and quadratic equations with inequalty constraints, and eliminating the quantifiers, we obtain the following two weak SRRs:

$$\Delta_1(u_1 \ldots, u_4),\ x_1 = u_1 + X_1,\ x_2 = X_2, \ldots, x_7 = X_7;$$
$$\Delta_2(u_1 \ldots, u_4),\ x_1 = u_1 - X_1,\ x_2 = X_2, \ldots, x_7 = X_7,$$

where

$$\begin{aligned}
\Delta_1(u_1 \ldots, u_4):\ & u_1 \neq 0 \wedge u_3 \neq 0 \wedge u_4 \neq 0 \wedge u_3^2 + u_2^2 - 2\,u_1u_2 > 0 \wedge Q_1 \leq 0 \\
& \wedge [Q_2 > 0 \vee (u_4 > 0 \wedge Q_3 < 0) \vee\ (u_3 < 0 \wedge Q_4 > 0) \\
& \quad \vee (Q_4 > 0 \wedge Q_3 > 0) \vee (u_4 > 0 \wedge Q_4 < 0) \\
& \quad \vee (u_4 < 0 \wedge Q_3 > 0) \vee (Q_3 < 0 \wedge Q_4 < 0)]; \\
\Delta_2(u_1 \ldots, u_4):\ & u_1 \neq 0 \wedge u_3 \neq 0 \wedge u_4 \neq 0 \wedge u_3^2 + u_2^2 - 2\,u_1u_2 > 0 \wedge Q_1 \leq 0 \\
& \wedge [Q_2 < 0 \vee (u_4 > 0 \wedge Q_3 < 0) \vee\ (u_3 < 0 \wedge Q_4 > 0) \\
& \quad \vee (Q_4 > 0 \wedge Q_3 > 0) \vee (u_4 > 0 \wedge Q_4 < 0) \\
& \quad \vee (u_4 < 0 \wedge Q_3 > 0) \vee (Q_3 < 0 \wedge Q_4 < 0)],
\end{aligned}$$

$$\begin{aligned}
& Q_1 = u_4^2 - u_3^2 - u_2^2 + 2\,u_1u_2 - u_1^2, \quad Q_2 = u_2u_3^2 + u_1u_3^2 + u_2^3 - u_1u_2^2, \\
& Q_3 = u_3^2u_4 + u_2^2u_4 - u_3^3 - u_2^2u_3 + 2\,u_1u_2u_3, \\
& Q_4 = u_3^2u_4 + u_2^2u_4 + u_3^3 + u_2^2u_3 - 2\,u_1u_2u_3,
\end{aligned}$$

and

$$\begin{aligned}
& X_1 = \sqrt{u_1^2 + u_2^2 - 2\,u_2u_1 + u_3^2 - u_4^2}, \quad X_2 = \frac{u_3(-u_2^2 + 2\,u_2u_1 - u_3^2)}{u_3^2 + u_2^2}, \\
& X_3 = \frac{u_2x_2}{u_3}, \quad X_4 = \frac{u4(u_2^2 - 2\,u_2u_1 + u_3^2)}{-u_2^2 + 2\,u_2u_1 - u_3^2 - 2\,x_1u_1}, \quad X_5 = \frac{x_1x_4}{u_4}, \\
& X_6 = \frac{-u_3x_5 + u_2x_4}{-x_5 + u_2}, \quad X_7 = \frac{-u_4x_3 + x_1x_2}{-x_3 + x_1}.
\end{aligned}$$

With these solution representations, the values of x_i may be easily computed for any given real values of $u_1, \ldots, u_4$ that satisfy the conditions and the corresponding dynamic diagram as shown in Figure 3 can be drawn and animated efficiently.

Example C.5 (Pappus Theorem) Let three points A, B, C be incident to a single straight line, where B is between A and C, and another three points D, E, F incident

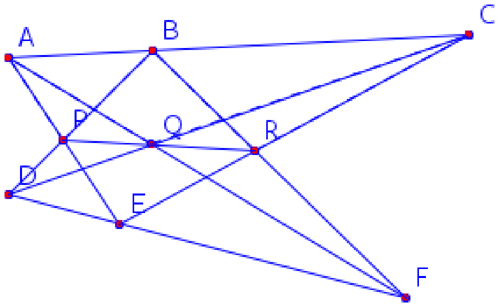

Figure 4: Pappus Configuration

to anther straight line, where E is between D and F. P is the intersection of line AE and line BD. Q is the intersection of line AF and line CD. R is the intersection of line BF and line CE. Then P, Q, R are collinear.

To express the order relation of the points, we need to use inequalities. Without loss of generality and to give a concise presentation of the results, let the coordinates of the points be assigned as follows:

$$A(0,1),\ B(x_1,u_5),\ C(u_1,u_2),\ D(0,0),\ E(x_2,u_6),\ F(u_3,u_4).$$

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases} F_1 = -u_5u_1 + u_1 + x_1u_2 - x_1 = 0, & (A,\ B,\ C \text{ collinear}) \\ F_2 = -u_6u_3 + x_2u_4 = 0, & (D,\ E,\ F \text{ collinear}) \\ G_1 = u_1 > 0 \wedge x_1 > 0 \wedge x_1 < u_1, & (B \text{ between } A \text{ and } C) \\ G_2 = u_3 > 0 \wedge x_2 > 0 \wedge x_2 < u_3. & (E \text{ between } D \text{ and } F) \end{cases}$$

Assume that C, F are free points and B, E are semi-free points, so that $u_1, \ldots, u_6$ are free parameters. The set $\{F_1, F_2\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, \ldots, u_6)$ into one irreducible triangular set $[T_1, T_2]$, where

$$T_1 = u_5u_1 - u_1 - x_1u_2 + x_1,$$
$$T_2 = u_6u_3 - x_2u_4.$$

From this triangular set, using the solution formulas and eliminating the quantifiers, we obtain one weak SRRs.

$$\Delta(u_1, \ldots, u_6),\ x_1 = \frac{u_1(u_5 - 1)}{u_2 - 1},\ x_2 = \frac{u_6u_3}{u_4},$$

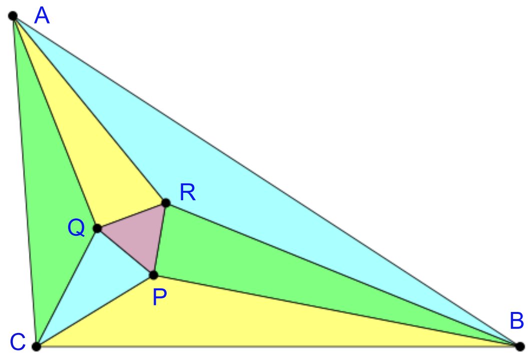

Figure 5: Morley Configuration

where

$$\begin{aligned}\Delta(u_1,\ldots,u_6):\ & u_1>0\wedge u_2-1\neq 0\wedge u_3>0\wedge u_4\neq 0\\ & \wedge[(u_5-1>0\wedge u_5-u_2<0\wedge u_6>0\wedge u_6-u_4<0)\\ & \vee(u_5-1>0\wedge u_5-u_2<0\wedge u_6<0\wedge u_6-u_4>0)\\ & \vee(u_5-1<0\wedge u_5-u_2>0\wedge u_6>0\wedge u_6-u_4<0)\\ & \vee(u_5-1<0\wedge u_5-u_2>0\wedge u_6<0\wedge u_6-u_4>0)].\end{aligned}$$

With these solution representations, the values of x_i may be easily computed for any given real values of $u_1,\ldots,u_6$ that satisfy the conditions and the corresponding dynamic diagram as shown in Figure 4 can be drawn and animated efficiently.

Example C.6 (Morley Theorem [128, 140]) The neighboring internal trisectors of the three angles of an arbitrary triangle ABC intersect to three points P, Q, R, which form an equilateral triangle (see Figure 5 which is taken from Wikipedia).

To ensure that the trisectors are all internal, we need to use inequalities. Without loss of generality and to give a concise presentation of the results, let the coordinates of the points be assigned as follows as:

$$A(x_2,x_1),\ B(u_1,0),\ C(u_2,0),\ P(0,1),\ Q(x_6,x_5),\ R(x_4,x_3).$$

We use the formulation of the theorem from [128], where the hypothesis consist of

$$\begin{aligned}&\angle ABC=3\angle PBC,\ \angle ACB=3\angle PCB,\ \angle CAB=3\angle RAB,\\&\angle ABR=3\angle PBC,\ \angle ACQ=3\angle PCB,\ \angle BAR=3\angle QAC.\end{aligned}$$

We add the following inequality hypothesis to ensure that the trisectors are all internal

$$P \text{ inside } \triangle ABC, \quad Q \text{ inside } \triangle ABC, \quad R \text{ inside } \triangle ABC.$$

The hypothesis can be expressed as 6 polynomial equations and 9 inequality constraints, with the highest degree of 3. The set of hypothesis-polynomials can be decomposed over $\mathbb{Q}(u_1, u_2)$ into two irreducible triangular sets. For each triangular set with the inequality constraints, we want to compute the SRRs. However, the computations do not finished within 20000 seconds during the real quantifier elimination part.

Example C.7 Given two circles O_1 and O_2 and an arbitrary line AB, construct another circle O which is tangent to line AB at point B and tangent to circles O_1 and O_2, both externally, at points T_1 and T_2 respectively.

To express the order relation of external tangency, we need to use inequalities. Without loss of generality and to give a concise presentation of the results, let the coordinates of the points be assigned as follows:

$$A(x_3, 0),\ B(u_1, x_4),\ O(x_1, 0),\ O_1(0, 0),\ T_1(1, 0),\ O_2(0, 1),\ T_2(u_2, x_2).$$

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases} F_1 = -x_1x_2 + x_1 - u_2 = 0, & (O,\ O_2,\ T_2 \text{ collinear}) \\ F_2 = -x_2^2 + 2\,u_2x_1 - 2\,x_1 - u_2^2 + 1 = 0, & (|OT_1| = |OT_2|) \\ F_3 = -x_4^2 + 2\,u_1x_1 - 2\,x_1 - u_1^2 + 1 = 0, & (|OT_1| = |OB|) \\ F_4 = x_4^2 + x_1x_3 - u_1x_3 - u_1x_1 + u_1^2 = 0, & (OB \perp AB) \\ G_1 = x_1 - 1 > 0, & (O,\ O_1 \text{ ex-tangent}) \\ G_2 = x_2 > 0, \quad G_3 = x_2 - 1 < 0. & (O,\ O_2 \text{ ex-tangent}) \end{cases}$$

Assume that B and T_2 are semi-free points, so that u_1, u_2 are free parameters. The set $\{F_1, \ldots, F_4\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2)$ into one irreducible

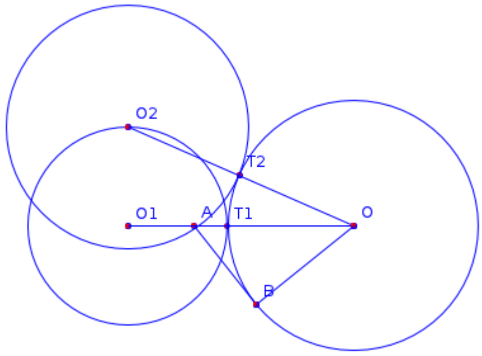

Figure 6: External Tangency

triangular set $[T_1, \ldots, T_4]$, where

$$T_1 = -2\,u_2x_1^3 + 2\,x_1^3 + u_2^2x_1^2 - 2\,u_2x_1 + u_2^2,$$
$$T_2 = x_1x_2 - x_1 + u_2,$$
$$T_3 = x_1x_3 - u_1x_3 + u_1x_1 - 2\,x_1 + 1,$$
$$T_4 = x_4^2 - 2\,u_1x_1 + 2\,x_1 + u_1^2 - 1.$$

From this triangular set, using the solution formulas and eliminating the quantifiers, we obtain six weak SRRs. We list two of them below:

$$\Delta_1(u_1, u_2),\ x_1 = X_{11},\ x_2 = X_2,\ x_3 = X_3,\ x_4 = X_4;$$
$$\Delta_2(u_1, u_2),\ x_1 = X_{12},\ x_2 = X_2,\ x_3 = X_3,\ x_4 = X_4,$$

where

$$\begin{aligned}\Delta_1(u_1,u_2):\ & u_1 - 1 > 0 \wedge Q_1 - 27 < 0 \wedge Q_2 < 0 \wedge Q_3 > 0 \\ & \wedge \{u_2 - u_1 > 0 \vee [Q_1 > 0 \wedge Q_4 \le 0] \vee [Q_1 < 0 \wedge Q_5 < 0] \\ & \quad \vee [Q_5 > 0 \wedge Q_4 \le 0]\}; \\ \Delta_2(u_1,u_2):\ & u_1 - 1 \ge 0 \wedge Q_1 - 27 \le 0 \wedge Q_2 \le 0 \wedge Q_3 < 0 \\ & \wedge \{Q_6 < 0 \vee Q_5 > 0 \vee [Q_1 > 0 \wedge Q_5 = 0] \vee [Q_5 < 0 \wedge Q_4 \le 0]\},\end{aligned}$$

$$Q_1 = u_1^3 - u_1^2 - u_1 - 4, \quad Q_2 = u_2^3 + 2\,u_2^2 + 11\,u_2 - 16, \quad Q_3 = u_2^4 + 36\,u_2^2 - 90\,u_2 + 54,$$
$$Q_4 = u_1^2u_2^2 + 2\,u_1u_2^2 + 5\,u_2^2 - u_1^3u_2 - 3\,u_1^2u_2 - 7\,u_1u_2 - 5\,u_2 + u_1^3 + 3\,u_1^2 + 3\,u_1 + 1,$$
$$Q_5 = u_1^2u_2^2 + u_2^2 - 2\,u_1^3u_2 - 2\,u_1u_2 + 2\,u_1^3, \quad Q_6 = 2\,u_1^2 + 2\,u_1^2 + 5\,u_1 - 11,$$

and

$$X_{11} = \frac{u_2^2}{6\,(u_2-1)} + \frac{\omega^1 c_1}{3} + \frac{\omega^2 c_2}{3}, \quad X_{12} = \frac{u_2^2}{6\,(u_2-1)} + \frac{\omega^2 c_1}{3} + \frac{\omega^1 c_2}{3},$$
$$X_2 = 1 - \frac{u_2}{x_1}, \quad X_3 = \frac{2\,x_1 - x_1 u_1 - 1}{x_1 - u_1}, \quad X_4 = \sqrt{-2\,x_1 + 2\,x_1 u_1 - u_1^2 + 1},$$
$$p_1 = \frac{16\,u_2^3 - u_2^6 - 2\,u_2^5 - 11\,u_2^4}{4\,(u_2-1)^2}, \; p_2 = \frac{54\,u_2^4 - 125\,u_2^3 + 72\,u_2^2}{4\,(u_2-1)^3}$$

with c_1, c_2, ω the same as in Theorem 2.1.5 of Section 2.1.3.

With these solution representations, the values of x_i may be easily computed for any given real values of u_1, u_2 that satisfy the conditions and the corresponding dynamic diagram as shown in Figure 6 can be drawn and animated efficiently.

Example C.8 In a triangle ABC, let p and q be the radii of two circles through A, touching side BC at B and C, respectively. Let R be the radius of the circumcircle of the triangle ABC. Then $pq = R^2$.

To avoid the degenerate cases, we need to use inequalities. Without loss of generality and to give a concise presentation of the results, let the coordinates of the points be assigned as follows:

$$B(0,0),\; C(u_1,0),\; A(u_2,u_3),\; O(x_2,x_1),\; Q(0,x_3),\; P(u_1,x_4).$$

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases} F_1 = -2\,x_2 u_1 + u_1^2 = 0, & (|OB| = |OC|) \\ F_2 = -2\,x_2 u_2 + u_2^2 - 2\,x_1 u_3 + u_3^2 = 0, & (|OB| = |OA|) \\ F_3 = u_2^2 - 2\,x_3 u_3 + u_3^2 = 0, & (|QB| = |QA|) \\ F_4 = u_2^2 - 2\,x_3 u_3 + u_3^2 = 0, & (|PA| = |PC|) \\ G_1 = u_3 u_1 \neq 0, & (A,\ B,\ C \text{ not collinear}) \end{cases}$$

Assume that A is a free point and C is a semi-free point, so that u_1, u_2, u_3 are free parameters. The set $\{F_1, \ldots, F_4\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2, u_3$

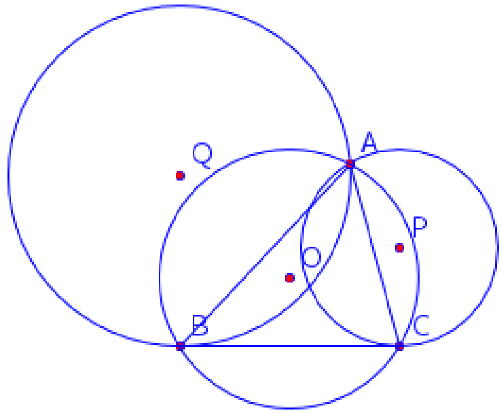

Figure 7: Configuration of Example C.8

into one irreducible triangular set $[T_1, \ldots, T_4]$, where

$$T_1 = u_1 u_2 - u_2^2 + 2\, x_1 u_3 - u_3^2,$$
$$T_2 = -2\, x_2 + u_1,$$
$$T_3 = u_2^2 - 2\, x_3 u_3 + u_3^2,$$
$$T_4 = u_1^2 - 2\, u_1 u_2 + u_2^2 - 2\, x_4 u_3 + u_3^2.$$

From this triangular set, using the solution formulas and eliminating the quantifiers, we obtain one weak SRRs.

$$u_1 \neq 0 \wedge u_3 \neq 0, \;\; x_1 = \frac{u_2^2 - u_1 u_2 + u_3^2}{2\, u_3}, \;\; x_2 = \frac{1}{2\, u_1}, \;\; x_3 = \frac{u_2^2 + u_3^2}{2\, u_3},$$
$$x_4 = \frac{u_1^2 - 2\, u_1 u_2 + u_2^2 + u_3^2}{2\, u_3}.$$

With these solution representations, the values of x_i may be easily computed for any given real values of u_1, u_2, u_3 that satisfy the conditions and the corresponding dynamic diagram as shown in Figure 7 can be drawn and animated efficiently.

Example C.9 Let incircle (with center I) of triangle ABC touch the side BC at X and let A_1 be the midpoint of this side. Then the line $A_1 I$ (extended) bisectors AX

To specify the incenter I as an intersection point of angular bisectors, we need to ensure that the bisectors are all internal, which need to use inequalities. Without loss of generality and to give a concise presentation of the results, let the coordinates of the points be assigned as follows:

$$B(0,0), \;\; C(u_1,0), \;\; I(u_2,u_3), \;\; A(x_2,x_1), \;\; A_1(0,x_3), \;\; X(u_2,0), \;\; O(x_5,x_4).$$

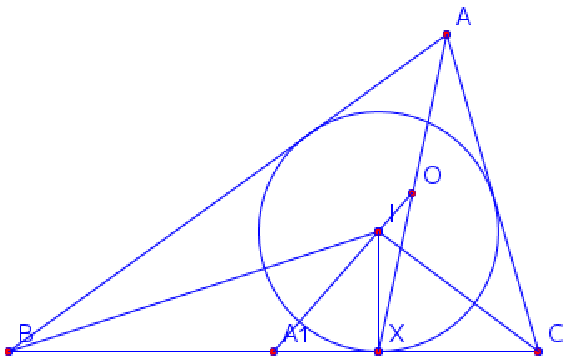

Figure 8: Configuration of Example C.9

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$
\begin{cases}
F_1 = -x_1u_3^2 - 2\,u_3u_2x_2 + 2\,u_3u_1u_2 + 2\,u_3u_1x_2 & (\tan\angle BCI = \tan\angle ACI) \\
\qquad - 2\,u_3u_1^2 + u_2^2x_1 - 2\,u_2x_1u_1 + u_1^2x_1 = 0, & \\
F_2 = x_1u_3^2 + 2\,u_3u_2x_2 - u_2^2x_1 = 0, & (\tan\angle CBI = \tan\angle ABI) \\
F_3 = 2\,x_3 - u_1 = 0, & (A_1 \text{ midpoint of } B \text{ and } C) \\
F_4 = x_4u_2 + u_3x_3 - x_3x_4 - x_5u_3 = 0, & (O,\ A_1,\ I \text{ collinear}) \\
F_5 = -u_2x_1 + x_1x_5 + x_4u_2 - x_2x_4 = 0, & (O,\ A,\ X \text{ collinear}) \\
G_1 = u_3x_1u_1^2 > 0, & (I \text{ inside} \triangle ABC) \\
G_2 = u_2u_1x_1^2 - u_3u_1x_1x_2 > 0, & \\
G_3 = -u_2u_1x_1^2 + u_3u_1x_1x_2 - u_3x_1u_1^2 + x_1^2u_1^2 > 0. &
\end{cases}
$$

Assume that I is a free point and C is a semi-free point, so that u_1, u_2, u_3 are free parameters. The set $\{F_1, \ldots, F_5\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2, u_3)$ into one irreducible triangular set $[T_1, \ldots, T_5]$, where

$$
\begin{aligned}
T_1 &= 2\,u_3u_1u_2 - u_2x_1u_1 + x_1u_3^2 + u_2^2x_1 - 2\,u_3u_2^2,\\
T_2 &= x_1u_3^2 + 2\,u_3u_2x_2 - 2\,u_3u_1u_2 - 2\,u_3u_1x_2 + 2\,u_3u_1^2 - u_2^2x_1 + 2\,u_2x_1u_1 - u_1^2x_1,\\
T_3 &= -2\,x_3 + u_1,\\
T_4 &= -2\,u_3u_2x_1 + 2\,u_3x_4u_2 - 2\,u_3x_2x_4 + 2\,x_1x_4u_2 + x_1u_1u_3 - x_1u_1x_4,\\
T_5 &= 2\,x_4u_2 + u_1u_3 - u_1x_4 - 2\,x_5u_3.
\end{aligned}
$$

From this triangular set, using the solution formulas and eliminating the quantifiers, we obtain one weak SRRs.

$$\Delta(u_1,u_2,u_3),\ x_1=X_1,\ x_2=X_2,\ x_3=\frac{u_1}{2},\ x_4=X_4,\ x_5=X_5.$$

where

$$\begin{aligned}\Delta(u_1,u_2,u_3):\ & u_1\neq 0\wedge u_2-u_1\neq 0\wedge 2\,u_2-u_1\neq 0\wedge u_3\neq 0\wedge u_3^2+u_2^2-u_1u_2\neq 0\\ & \wedge[u_3^2+u_2^2-u_1u_2<0\vee(u_1>0\wedge u_2-u_1>0)\vee(u_1<0\wedge u_2-u_1<0)],\end{aligned}$$

and

$$\begin{aligned}X_1&=\frac{2\,u_3u_2(-u_2+u_1)}{-u_3^2-u_2^2+u_2u_1},\\ X_2&=\frac{x_1u_3^2-2\,u_3u_1u_2+2\,u_3u_1^2-u_2^2x_1+2\,u_2x_1u_1-u_1^2x_1}{2\,u_3(-u_2+u_1)},\\ X_4&=\frac{x_1u_3(-2\,u_2+u_1)}{-2\,u_3u_2+2\,u_3x_2-2\,u_2x_1+x_1u_1},\quad X_5=\frac{2\,u_2x_4+u_3u_1-u_1x_4}{2\,u_3}.\end{aligned}$$

With these solution representations, the values of x_i may be easily computed for any given real values of u_1,u_2,u_3 that satisfy the conditions and the corresponding dynamic diagram as shown in Figure 8 can be drawn and animated efficiently.

Example C.10 (Kosnita Theorem) Let point O is the circumcenter of an arbitrary triangle ABC. Points O_1, O_2, O_3 are the circumcenters of the triangles ABO, ACO and BCO. Then lines CO_1, BO_2 and AO_3 are concurrent.

To avoid degenerate cases, we need to use inequalities. Without loss of generality and to give a concise presentation of the results, let the coordinates of the points be assigned as follows:

$$A(0,0),\ B(1,0),\ C(u_1,u_2),\ O(x_1,x_2),\ O_1(x_3,x_4),\ O_2(x_5,x_6),\ O_3(x_7,x_8).$$

Then the geometric constraints may be expressed as the following semi-algebraic

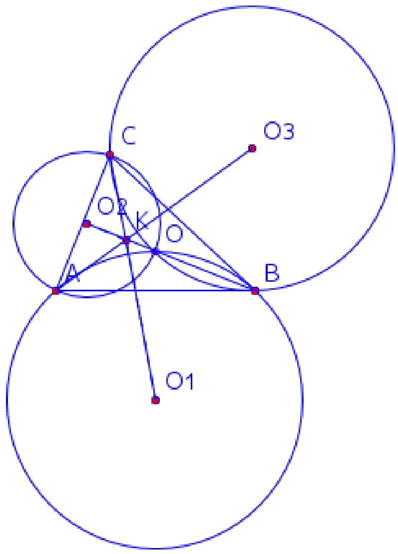

Figure 9: Kosnita Configuration

system of equalities and inequalities:

$$\begin{cases} F_1 = 1 - 2\,x_1 = 0, & (O \text{ circumcenter of } \triangle ABC) \\ F_2 = 2\,x_1u_1 - 2\,x_1 - u_1^2 + 1 + 2\,x_2u_2 - u_2^2 = 0, & \\ F_3 = 1 - 2\,x_3 = 0, & (O_1 \text{ circumcenter of } \triangle ABO) \\ F_4 = 2\,x_3x_1 - 2\,x_3 - x_1^2 + 1 + 2\,x_4x_2 - x_2^2 = 0, & \\ F_5 = -2\,x_5u_1 + u_1^2 - 2\,u_2x_6 + u_2^2 = 0, & (O_2 \text{ circumcenter of } \triangle ACO) \\ F_6 = 2\,x_5x_1 - 2\,x_5u_1 - x_1^2 + u_1^2 + 2\,x_6x_2 - 2\,u_2x_6 - x_2^2 + u_2^2 = 0, & \\ F_7 = 2\,x_7 - 1 - 2\,x_7u_1 + u_1^2 - 2\,u_2x_8 + u_2^2 = 0, & (O_3 \text{ circumcenter of } \triangle BCO) \\ F_8 = 2\,x_7x_1 - 2\,x_7u_1 - x_1^2 + u_1^2 + 2\,x_8x_2 - 2\,u_2x_8 - x_2^2 + u_2^2 = 0, & \\ G_1 = u_2 \neq 0, & (A,\ B,\ C \text{ not collinear}) \\ G_2 = x_4 \neq 0, & (A,\ B,\ O_1 \text{ not collinear}) \\ G_3 = -u_2x_5 + u_1x_6 \neq 0, & (A,\ C,\ O_2 \text{ not collinear}) \\ G_4 = -u_2x_7 + u_2 + u_1x_8 - x_8 \neq 0. & (B,\ C,\ O_3 \text{ not collinear}) \end{cases}$$

Assume that C is a free point, so that u_1, u_2 are free parameters. The set $\{F_1, \ldots, F_8\}$ of polynomials may be decomposed over $\mathbb{Q}(u_1, u_2)$ into one irreducible triangular set

$[T_1, \ldots, T_8]$, where

$$T_1 = -1 + 2\,x_1, \quad T_2 = -u_1 + u_1^2 - 2\,x_2 u_2 + u_2^2, \quad T_3 = -1 + 2\,x_3,$$
$$T_4 = -1 - 8\,x_4 x_2 + 4\,x_2^2, \quad T_5 = 4\,u_2 x_5 - u_2 - 4\,u_2 x_2^2 - 8\,x_5 u_1 x_2 + 4\,u_1^2 x_2 + 4\,u_2^2 x_2,$$
$$T_6 = -2\,x_5 u_1 + u_1^2 - 2\,u_2 x_6 + u_2^2,$$
$$T_7 = -4\,x_7 u_2 + 3\,u_2 - 4\,u_2 x_2^2 + 8\,x_7 x_2 - 4\,x_2 - 8\,x_7 u_1 x_2 + 4\,u_1^2 x_2 + 4\,u_2^2 x_2,$$
$$T_8 = 2\,x_7 - 1 - 2\,x_7 u_1 + u_1^2 - 2\,u_2 x_8 + u_2^2.$$

From this triangular set, using the solution formulas and eliminating the quantifiers, we obtain one weak SRRs.

$$\Delta(u_1, u_2),\ x_1 = \frac{1}{2},\ x_2 = X_2,\ x_3 = \frac{1}{2},\ x_4 = X_4, \ldots, x_8 = X_8.$$

where

$$\begin{aligned}\Delta(u_1, u_2):\ & u_1 \neq 0 \wedge u_1 - 1 \neq 0 \wedge u_2 \neq 0 \wedge u_2^2 + u_1^2 - u_1 \neq 0 \wedge u_2 + u_1 \neq 0 \\ & \wedge u_2 - u_1 + 1 \neq 0 \wedge u_2^2 + u_2 + u_1^2 - u_1 \neq 0 \wedge u_2 - u_1 \neq 0 \\ & \wedge u_2 + u_1 - 1 \neq 0 \wedge u_2^2 - u_2 + u_1^2 - u_1 \neq 0,\end{aligned}$$

and

$$X_2 = \frac{-u1 + u1^2 + u2^2}{2\,u_2}, \quad X_4 = \frac{-1 + 4\,x_2^2}{8\,x_2}, \quad X_5 = \frac{4\,u_1^2 x_2 - u_2 - 4\,u_2 x_2^2 + 4\,u_2^2 x_2}{4(-u_2 + 2\,u_1 x_2)},$$
$$X_6 = \frac{-2\,u_1 x_5 + u_1^2 + u_2^2}{2\,u_2}, \quad X_7 = \frac{-4\,x_2 + 3\,u_2 - 4\,u_2 x_2^2 + 4\,u_2^2 x_2 + 4\,u_1^2 x_2}{4(u_2 - 2\,x_2 + 2\,u_1 x_2)}$$
$$X_8 = \frac{2\,x_7 - 1 - 2\,x_7 u_1 + u_1^2 + u_2^2}{2\,u_2}$$

With these solution representations, the values of x_i may be easily computed for any given real values of u_1, u_2 that satisfy the conditions and the corresponding dynamic diagram as shown in Figure 9 can be drawn and animated efficiently.

Publications

[1] **Ting Zhao**, Dongming Wang and Hoon Hong. Solution Formulas for Cubic Equations Without or With Constraints. *J. Symb. Comput.* 46(8): 904-918, 2011.

[2] **Ting Zhao**, Dongming Wang, Hoon Hong and Philippe Aubry. Real Solution Formulas of Cubic and Quartic Equations Applied to Generate Dynamic Diagrams with Inequality Constraints. *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. Riva del Garda, Italy, ACM Press, 94-101, 2012.

[3] **Ting Zhao**. Software GeoDraw for Generating Dynamic Diagrams with Inequality Constraints. *Comput. Appl. Softw.* (in Chinese), 2012. Accepted.

[4] Xiaoyu Chen, Tielin Liang, Dongming Wang and **Ting Zhao**. Towards a Dynamic Environment for Geometry Research and Education. *Proceedings of the 5th Asian Workshop on Foundations of Software*. Xiamen, China, 153-156, 2007.

[5] Xiaoyu Chen, Dongming Wang and **Ting Zhao**. GeoText: an Intelligent Dynamic Geometry Textbook[A]. Submitted to ISSAC 2012.

[6] Hoon Hong, Dongming Wang and **Ting Zhao**. Solution Formulas for Quartic Equations Without or With Constraints. In preparation, 2012.

Curriculum Vitae

Name	Ting Zhao	**Date of Birth**	May 22, 1982
Gender	Female	**Place of Birth**	Zhejiang, China
E-mail	zhaoting522@163.com		

EDUCATION

Ph.D. co-supervision student:

- Beihang University, Beijing, China, September 2005 - present
- LIP 6 - Université Paris 6, Paris, France, September 2007 - present

B.S. in Applied Mathematics: Beihang University, Beijing, China, July 2005

Doulbe B.A. in English: Beihang University, Beijing, China, July 2005

ACADEMIC EXPERIENCE

Presentation at

- the 27th Annual ACM Symposium on Applied Computing held in Riva del Garda, Italy, March 26 – 30, 2012;
- the 2nd NCSU-China Symbolic Computation Collaboration Workshop held in Hangzhou, China, March 5 – 9, 2007.

Participation at

- the 4th International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS 2011) held in Beijing, China, June 21 – 24, 2011;
- the International Conference on Mathematics Mechanization - In honor of professor Wen-Tsun Wu's nineties birthday held in Beijing, China, May 11 – 13, 2009;
- the 2nd International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS 2007) held in Paris, France, December 5 – 7, 2007;

- the 8th International Conference on Artificial Intelligence and Symbolic Computation (AISC 2006) held in Beijing, China, September 20 – 22, 2006;
- the 1st International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS 2006) held in Beijing, China, July 24 – 26, 2006;
- the 2nd Summer School in Symbolic Computation (SSSC 2006) held in Beijing, China, July 15-22, 2006;
- the International Seminar on Symbolic Computation in Education (SCE 2006) held in Beijing, China, April 12 – 14, 2006.

OTHER EXPERIENCE

- Obtained financial support from the China Scholarship Council for the study in University Paris 6, France, October 2007 – May 2009.
- Visit the Korea Institute for Advanced Study (KIAS) in Seoul, Korea, October 25 – November 6, 2008.
- Obtained the grants to attend the 3rd RISC/SCIEnce Training School in Symbolic Computation in the Castle of Hagenberg, Austria, July 7 – 20, 2008.
- Visit the symbolic computation group at North Carolina State University in the USA, May 20 – June 1, 2007.