Real Solution Formulas of Cubic and Quartic Equations Applied to Generate Dynamic Diagrams with Inequality Constraints

Ting Zhao, Dongming Wang, Hoon Hong, Philippe Aubry

HAL Id: hal-00683596

https://hal.sorbonne-universite.fr/hal-00683596v1

Submitted on 29 Mar 2012

Real Solution Formulas of Cubic and Quartic Equations Applied to Generate Dynamic Diagrams with Inequality Constraints

Ting Zhao[†]
SKLSDE – LMIB – School of Mathematics
and Systems Science
Beihang University
Beijing 100191, China
zhaoting522@163.com

Dongming Wang
[†]Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie – CNRS
4 place Jussieu – BP 169
75252 Paris cedex 05, France
dongming.wang@lip6.fr

Hoon Hong
Department of Mathematics
North Carolina State University
Box 8205, Raleigh
NC 27695, USA
hong@ncsu.edu

Philippe Aubry
Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie – CNRS
4 place Jussieu – BP 169
75252 Paris cedex 05, France
philippe.aubry@lip6.fr

ABSTRACT

The approach of solving geometric constraints involving inequalities proposed by Hong and others uses triangular decomposition, solution formulas, and quantifier elimination. We 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 and quartic equations are replaced by newly introduced real solution formulas with inequality constraints. Several examples are presented to illustrate the enhanced approach and to demonstrate the advantages and effectiveness of the new solution formulas. An implementation of the enhanced approach in Java with interface to Epsilon and QEPCAD for automated generation of dynamic diagrams is outlined and some experimental data are provided.

Keywords

Geometric constraint solving, real solution formula, quantifier elimination, dynamic diagram, inequality constraint

1. INTRODUCTION

This paper presents 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. 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 the geometric 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 visualize the geometric configurations corresponding to the real solutions of the system for changing values of the parameters.

A large number of software systems are available for producing static and dynamic diagrams interactively or (semi-)automatically. We name 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 for solving the involved parametric semi-algebraic systems based, e.g., on symbolic or numeric computation, or both combined. There are also highly efficient preprocessing methods, developed mostly in the computer aided design community, which transform geometric

---

[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/

specifications into easy-to-draw[9] geometric constructions using graph analysis and rule-based reasoning. We refer to [7] for an account of such methods and approaches. 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.

The problem of solving real geometric constraints involving inequalities was addressed first by Hong and his coauthors in [9]. 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 [15]).

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 [9, 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 this paper, 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 involving no division by small numbers are recalled in Section 2 and the process of dynamic diagram generation using the enhanced HLLW approach is sketched and illustrated by an example in Section 3. We present the results of our studies on three geometric configurations, the configuration of Apollonius circles and the dynamic diagrams for Feuerbach's and Thébault-Taylor's theorems, in Section 4 to demonstrate the advantages and effectiveness of the new solution formulas and the enhanced approach. In Section 5, implementation of the enhanced approach in Java with interface to Epsilon and QEPCAD for automated generation of dynamic diagrams is discussed and some experimental data are provided. The paper is concluded with a few remarks in Section 6.

---

[9]e.g., with ruler and compass or other basic drawing tools [5]

2. REAL SOLUTION FORMULAS OF CUBIC AND QUARTIC EQUATIONS WITH CONSTRAINTS

In this section, we recall the real convention of square and cubic roots and the Lagrange-type real solution formulas of cubic equations and quartic equations with constraints introduced in [19, 10]. These formulas will be compared with the Ferro-Cardano-type solution formulas used in [9].

DEFINITION 1 (REAL CONVENTION [19]). *The* real convention *chooses the square root $\sqrt[2]{x}$ and cubic root $\sqrt[3]{x}$ of x so that*

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

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

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. The following result is proved in [19].

THEOREM 1 (REAL SOLUTION FORMULAS: CUBIC [19]). *Let*

$$f(x) = x^3 + a_2 x^2 + a_1 x + a_0 \in \mathbb{R}[x]$$

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 $x \in \mathbb{R}$,

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

where

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

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

$$x_3 = (-a_2 + \omega^2 c_1 + \omega^1 c_2)/3, \quad s = \sqrt[2]{-3\,p_1},$$

$$\omega = \mathrm{e}^{i\frac{2\pi}{3}} = -\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}i,$$

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

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

and

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

$$\Phi_1(x) := [f' > 0 \wedge f'' > 0] \vee [f' = 0 \wedge f'' \ge 0],$$

$$\Phi_2(x) := [f' \le 0] \qquad\quad \vee [f'' = 0],$$

$$\Phi_3(x) := [f' > 0 \wedge f'' < 0] \vee [f' = 0 \wedge f'' \le 0].$$

The reader may compare the above formulas with those in [9, Prop. 3]. The formulas in the following theorem, parallel to Theorem 1, are somewhat more complicated. A complete proof of the result is quite involved. The details are given in the paper [10] under preparation.

THEOREM 2 (REAL SOLUTION FORMULAS: QUARTIC).
Let

$$f(x) = x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 \in \mathbb{R}[x]$$

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 $x \in \mathbb{R}$,

$$[f(x) = 0 \wedge \Gamma(x)] \iff$$
$$[x = x_1 \wedge \Gamma_1] \vee [x = x_2 \wedge \Gamma_2] \vee [x = x_3 \wedge \Gamma_3] \vee [x = x_4 \wedge \Gamma_4],$$

where

$$x_1 = (-a_3 + k_1 + k_2 + \sigma_2 k_3)/4,$$
$$x_2 = (-a_3 + k_1 - k_2 - \sigma_2 k_3)/4,$$
$$x_3 = (-a_3 - k_1 + k_2 - \sigma_2 k_3)/4,$$
$$x_4 = (-a_3 - k_1 - k_2 + \sigma_2 k_3)/4,$$
$$k_1 = \sqrt{(p_3 - 4\,\omega^2 t_1 - 4\,\omega^1 t_2)/3},$$
$$k_2 = \sqrt{(p_3 - 4\,\omega^0 t_1 - 4\,\omega^0 t_2)/3},$$
$$k_3 = \sqrt{(p_3 - 4\,\omega^1 t_1 - 4\,\omega^2 t_2)/3},$$
$$t_1 = w^{-\sigma_1}\sqrt[3]{(p_2 + 3\,s)/2}, \quad t_2 = w^{+\sigma_1}\sqrt[3]{(p_2 - 3\,s)/2},$$
$$s = \sqrt{-3\,p_1}, \quad \omega = e^{i\frac{2\pi}{3}} = -\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}i,$$
$$\sigma_1 = \begin{cases} 0 & \text{if } p_1 < 0, \\ \text{sign}(p_2) & \text{if } p_1 \geq 0, \end{cases}$$
$$\sigma_2 = \begin{cases} +1 & \text{if } (r \leq 1 \wedge p_4 \leq 0) \vee (r > 1 \wedge p_4 > 0), \\ -1 & \text{if } (r \leq 1 \wedge p_4 > 0) \vee (r > 1 \wedge p_4 \leq 0), \end{cases}$$
$$r = \text{number of distinct real roots of } f \in \{0,1,2,3,4\},$$
$$\begin{aligned} p_1 = {}& 18\,a_3^3 a_1 a_0 + 256\,a_0^3 - 27a_1^4 - 6\,a_3^2 a_1^2 a_0 - 192\,a_3 a_1 a_0^2 \\ & + 18\,a_3 a_1^3 a_2 + 144\,a_2 a_3^2 a_0^2 + a_2^2 a_3^2 a_1^2 - 4\,a_2^3 a_3^2 a_0 \\ & + 144\,a_0 a_1^2 a_2 - 4\,a_3^3 a_1^3 - 128\,a_2^2 a_0^2 + 16\,a_2^4 a_0 - 4\,a_2^3 a_1^2 \\ & - 27\,a_3^4 a_0^2 - 80\,a_3 a_1 a_2^2 a_0, \end{aligned}$$
$$p_2 = 27\,a_3^2 a_0 + 9\,a_3 a_2 a_1 - 2\,a_2^3 + 72\,a_2 a_0 - 27\,a_1^2,$$
$$p_3 = 3\,a_3^2 - 8\,a_2, \quad p_4 = 4\,a_2 a_3 - 8\,a_1 - a_3^3,$$

and

$$\Gamma_j := (\exists\, x \in \mathbb{R})\,[f(x) = 0 \wedge \Gamma(x) \wedge \Phi_j(x)], \quad j = 1,2,3,4,$$
$$\begin{aligned} \Phi_1(x) := {}& [p_1 \geq 0 \wedge f' \geq 0 \wedge (f'' \geq 0 \wedge f''' \geq 0)] \vee \\ & [p_1 < 0 \wedge p_4 > \wedge f' > 0], \end{aligned}$$
$$\begin{aligned} \Phi_2(x) := {}& [p_1 \geq 0 \wedge f' \leq 0 \wedge (f'' \leq 0 \vee f''' \geq 0)] \vee \\ & [p_1 < 0 \wedge p_4 \leq 0 \wedge f' > 0], \end{aligned}$$
$$\begin{aligned} \Phi_3(x) := {}& [p_1 \geq 0 \wedge f' \geq 0 \wedge (f'' \leq 0 \vee f''' \leq 0)] \vee \\ & [p_1 < 0 \wedge p_4 > 0 \wedge f' < 0], \end{aligned}$$
$$\begin{aligned} \Phi_4(x) := {}& [p_1 \geq 0 \wedge f' \leq 0 \wedge (f'' \geq 0 \wedge f''' \leq 0)] \vee \\ & [p_1 < 0 \wedge p_4 \leq 0 \wedge f' < 0]. \end{aligned}$$

Note that the number r of distinct real roots of f can be determined by using the complete discrimination system [17], which is a set of explicit expressions in the coefficients of f.

The real solution formulas given in the above two theorems 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 [9] 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 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 1 and 2 above are computationally better in terms of numeric stability and geometric continuity for dynamic diagram animation. The example of Apollonius circles in Section 4 will illustrate the practical advantages of using these new formulas over the previous ones.

The formulas in Theorems 1 and 2 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 nonconstant 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.

3. GENERATING DYNAMIC DIAGRAMS WITH INEQUALITY CONSTRAINTS

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 $x_1, \ldots, x_n$ the coordinates of points and other geometric entities involved in the objects of $\mathcal{O}$; then the set $\mathcal{C}$ of constraints together with the conditions to exclude some degenerate cases may be expressed as a semi-algebraic system. The geometric constraint solving problem is then reduced to solving this semi-algebraic system.

Using the HLLW approach enhanced with the real solution formulas in Theorems 1 and 2 to solve parametric semi-algebraic systems, the process for automated generation of dynamic diagrams involving inequality constraints consists of the following steps (see [9] and also [13]).

1. Assign coordinates to the points involved in the geometric objects and introduced other variables if neces-

sary, so that the geometric constraints are expressed as a semi-algebraic system of equalities and inequalities.

2. Decompose the semi-algebraic system 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 $x_1, \ldots, x_n$ into parameters and dependent variables.

4. Randomly choose a set of allowable real numeric values for the parameters satisfying the constraints and compute the values of the dependent variables according to 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.

The dynamic animation of a drawn static 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 1. 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:[10]

$A(x_3, 0)$, $B(u_1, x_4)$, $O(x_1, 0)$, $O_1(0, 0)$, $T_1(1, 0)$,
$O_2(0, 1)$, $T_2(u_2, x_2)$.

Then the geometric constraints may be expressed as the following semi-algebraic system of equalities and inequalities:

$$\begin{cases} F_1 = -x_1 x_2 + x_1 - u_2 = 0, \\ F_2 = -x_2^2 + 2\,u_2 x_1 - 2\,x_1 - u_2^2 + 1 = 0, \\ F_3 = -x_4^2 + 2\,u_1 x_1 - 2\,x_1 - u_1^2 + 1 = 0, \\ F_4 = x_4^2 + x_1 x_3 - u_1 x_3 - u_1 x_1 + u_1^2 = 0, \\ G_1 = x_1 - 1 > 0, \quad G_2 = x_2 > 0, \quad G_3 = x_2 - 1 < 0. \end{cases}$$

The correspondence between the geometric constraints and the polynomial equalities and inequalities is shown below:

$F_1 = 0 \iff O, O_2, T_2$ are collinear; $F_4 = 0 \iff OB \perp AB$;
$F_2 = 0 \iff |OT_1| = |OT_2|$; $F_3 = 0 \iff |OT_1| = |OB|$;
$G_1 > 0 \iff$ circles O, O_1 are externally tangent;
$G_2 > 0, G_3 < 0 \iff$ circles O, O_2 are externally tangent.

---

[10]Note that coordinates may also be assigned automatically, see, e.g., the function `Coordinate` in GEOTHER [15].
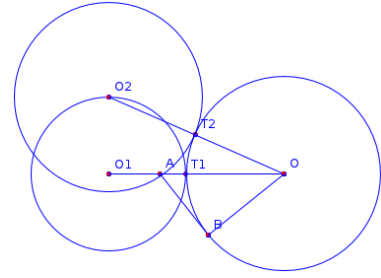

Figure 1: External Tangency

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 triangular set

$$\begin{bmatrix} -2\,u_2 x_1^3 + 2\,x_1^3 + u_2^2 x_1^2 - 2\,u_2 x_1 + u_2^2, \\ x_1 x_2 - x_1 + u_2, \\ x_1 x_3 - u_1 x_3 + u_1 x_1 - 2\,x_1 + 1, \\ x_4^2 - 2\,u_1 x_1 + 2\,x_1 + u_1^2 - 1 \end{bmatrix}.$$

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

$\Delta_1 : \quad u_1 - 1 > 0 \wedge Q_1 - 27 < 0 \wedge Q_2 < 0 \wedge Q_3 > 0$
$\qquad \wedge \{u_2 - u_1 > 0 \vee [Q_1 > 0 \wedge Q_4 \leq 0]$
$\qquad\qquad \vee [Q_1 < 0 \wedge Q_5 < 0] \vee [Q_5 > 0 \wedge Q_4 \leq 0]\};$

$\Delta_2 : \quad u_1 - 1 \geq 0 \wedge Q_1 - 27 \leq 0 \wedge Q_2 \leq 0 \wedge Q_3 < 0$
$\qquad \wedge \{2\,u_1^2 + 2\,u_1^2 + 5\,u_1 - 11 < 0 \vee Q_5 > 0$
$\qquad\qquad \vee [Q_1 > 0 \wedge Q_5 = 0] \vee [Q_5 < 0 \wedge Q_4 \leq 0]\},$

$Q_1 = u_1^3 - u_1^2 - u_1 - 4, \quad Q_2 = u_2^3 + 2\,u_2^2 + 11\,u_2 - 16,$
$Q_3 = u_2^4 + 36\,u_2^2 - 90\,u_2 + 54,$
$Q_4 = u_1^2 u_2^2 + 2\,u_1 u_2^2 + 5\,u_2^2 - u_1^3 u_2 - 3\,u_1^2 u_2 - 7\,u_1 u_2 - 5\,u_2$
$\qquad + u_1^3 + 3\,u_1^2 + 3\,u_1 + 1,$
$Q_5 = u_1^2 u_2^2 + u_2^2 - 2\,u_1^3 u_2 - 2\,u_1 u_2 + 2\,u_1^3,$

and

$$X_{11} = \frac{u_2^2}{6\,(u_2 - 1)} + \frac{\omega^1 c_1}{3} + \frac{\omega^2 c_2}{3}, \quad X_2 = 1 - \frac{u_2}{x_1},$$

$$X_{12} = \frac{u_2^2}{6\,(u_2 - 1)} + \frac{\omega^2 c_1}{3} + \frac{\omega^1 c_2}{3}, \quad X_3 = \frac{2\,x_1 - x_1 u_1 - 1}{x_1 - u_1},$$

$$X_4 = \sqrt{-2\,x_1 + 2\,x_1 u_1 - u_1^2 + 1},$$

$$p_1 = \frac{16\,u_2^3 - u_2^6 - 2\,u_2^5 - 11\,u_2^4}{4\,(u_2 - 1)^2}, \; p_2 = \frac{54\,u_2^4 - 125\,u_2^3 + 72\,u_2^2}{4\,(u_2 - 1)^3}$$

with c_1, c_2, ω the same as in Theorem 1 of Section 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 1 can be drawn and animated efficiently.

4. SOME GEOMETRIC CONFIGURATIONS

In this section, we report our studies on three well-known geometric configurations, the configuration of Apollonius cir-

cles 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.

4.1 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., [5, 9, 11, 12]). 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 [9] 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 [9]. Following the HLLW approach as in [9] but using the new solution formulas instead of the old ones, we can compute six weak SRRs, of which two are listed as follows:

$\Delta_1(u_1, u_2, u_3),\ x_1 = X_{11},\ x_2 = X_2,\ x_3 = X_3,\ x_4 = X_4;$

$\Delta_2(u_1, u_2, u_3),\ x_1 = X_{12},\ x_2 = X_2,\ x_3 = X_3,\ x_4 = X_4,$

where Δ_1 and Δ_2 are formulas composed of inequality constraints in the parameters u_1, u_2 and u_3 only, X_2, X_3, X_4 are solutions of simple linear and quadratic equations,

$$X_{11} = \frac{u_1^2}{6\,(u_1 - 1)} + \frac{\omega^1 c_1}{3} + \frac{\omega^2 c_2}{3},$$

$$X_{12} = \frac{u_1^2}{6\,(u_1 - 1)} + \frac{\omega^2 c_1}{3} + \frac{\omega^1 c_2}{3},$$

$$p_1 = \frac{16\,u_1^3 - u_1^6 - 2\,u_1^5 - 11\,u_1^4}{4\,(u_1 - 1)^2},\ p_2 = \frac{54\,u_1^4 - 125\,u_1^3 + 72\,u_1^2}{4\,(u_1 - 1)^3},$$

and c_1, c_2, ω are the same as in Theorem 1 of Section 2.

The representation corresponding to X_{11} given in [9] is

$$X'_{11} = \frac{\sqrt[R3]{\delta}}{6\,(u_1 - 1)} + \frac{u_1^4 - 12\,u_1^2 + 12\,u_1}{6\,(u_1 - 1)\,\sqrt[R3]{\delta}} + \frac{u_1^4}{6\,(u_1 - 1)},$$

where

$\delta = d + 6\sqrt{3D}u_1(u_1 - 1),\quad d = u_1^6 + 36\,u_1^4 - 90\,u_1^3 + 54\,u_1^2,$

$D = u_1^6 + 8\,u_1^4 - 36\,u_1^3 + 43\,u_1^2 - 16\,u_1.$

X'_{11} may encounter the unstable case "0/0" 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 "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 Δ_1 and Δ_2 ensure
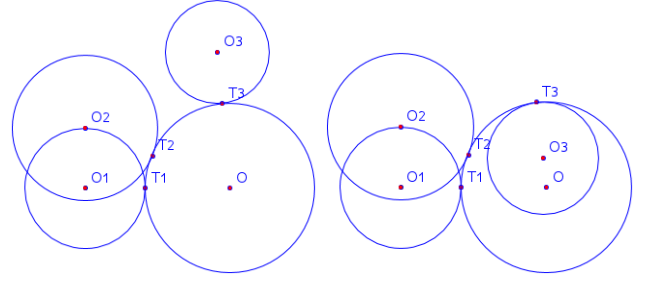

Figure 2: Apollonius Circles

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 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.

4.2 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 [3] 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 3 be assigned as follows:

$A(0, 0),\quad B(1, 0),\quad C(u_1, 1),\quad I(x_1, x_2),\quad D(x_1, 0),$

$A_1(x_3, x_4),\quad A_2(x_5, x_6),\quad A_3(x_7, 0),\quad N(x_8, x_9).$

The geometric constraints for the configuration may be specified by

$\tan\angle CBI = \tan\angle IBA,\quad \tan\angle CAI = \tan\angle IAB,$

A_1 is the midpoint of BC, A_2 is the midpoint of AC,

A_3 is the midpoint of AB, $|NA_3| = |NA_2|,$

$|NA_3| = |NA_1|,$ I is inside $\triangle ABC$.

This specification may be translated into a semi-algebraic system of 9 polynomial equalities and 3 inequalities. The semi-algebraic system may then be decomposed by using the enhanced HLLW approach into the following two weak SRRs:

$\Delta_i(u_1),\quad x_1 = X_{1i}, x_2 = X_2, \ldots, x_9 = X_9,\quad i = 1, 2,$

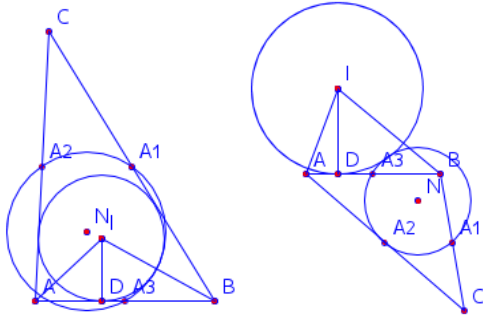**Figure 3: Feuerbach Configuration**

where Δ_i are composed of inequality constraints in u_1 only, $X_2, \ldots, X_9$ are simple rational expressions,

$$X_{11} = (2 + k_1 - k_2 - \sigma_2 k_3)/4,$$
$$X_{12} = (2 - k_1 + k_2 - \sigma_2 k_3)/4,$$
$$\sigma_2 = \begin{cases} +1 & \text{if } r \le 1, \\ -1 & \text{if } r > 1, \end{cases}$$

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

$$p_1 = 4\,u_1^{10} - 20\,u_1^9 + 57\,u_1^8 - 108\,u_1^7 + 158\,u_1^6 - 180\,u_1^5$$
$$+ 165\,u_1^4 - 116\,u_1^3 + 64\,u_1^2 - 24\,u_1 + 4,$$
$$p_2 = 2\,u_1^6 - 6\,u_1^5 - 3\,u_1^4 + 16\,u_1^3 - 27\,u_1^2 + 18\,u_1,$$
$$p_3 = 8\,u_1^2 - 8\,u_1 + 12,$$

k_1, k_2, k_3 are the same as in Theorem 2 of Section 2, and T (the first polynomial in the only triangular set) is quartic in x_1 with u_1 as parameter.[11]

The dynamic diagram generated automatically using the above weak SRRs is shown in Figure 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 $\triangle ABC$".

4.3 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., [14, pp. 130–131], [18], 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 non-trivial. We have tried the enhanced HLLW approach in a brute-force way without suc-

---

[11]The number r of distinct real roots of T can be determined by using the discrimination system: first produce the discrimination sequence $[D_1, \ldots, D_4] = [1, p_3, p_5, p_1]$ of T 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 $r = l - 2\,v$ [17].

cess. 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:

$A(1, 1)$, $B(0, 0)$, $C(u_1, 0)$, $D(1, 0)$, $O(x_1, x_2)$,

$O_1(x_3, x_4)$, $E(x_3, 0)$, $F(1, x_4)$, $G(x_5, x_6)$, $O_2(x_7, x_8)$,

$M(x_7, 0)$, $H(1, x_8)$, $J(x_9, x_{10})$, $I(x_{11}, x_{12})$, $K(x_{11}, 0)$.

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.
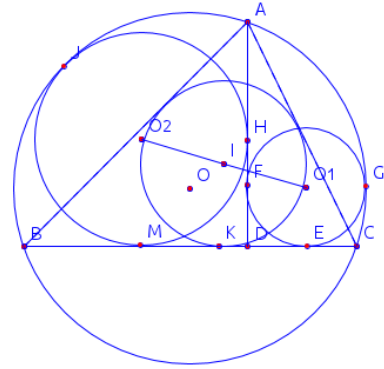

Figure 4: Thébault-Taylor Configuration

Subproblem 1. Construct circle O_1 with the following specification of geometric constraints:

O is the circumcenter of $\triangle ABC$,
O, O_1, and G are collinear,
$|O_1 E| = |O_1 F|$, $|O_1 E| = |O_1 G|$, $|OB| = |OG|$,
O_1 is inside the rectangle with AC as diagonal.

The corresponding semi-algebraic system may be decomposed into one weak SRR

$$\Delta_1(u_1), \; x_1 = X_{11}, \ldots, x_6 = X_{16}$$

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.

Subproblem 2. Construct circle O_2 with the following specification:

O is the circumcenter of $\triangle ABC$,
O, O_2, and J are collinear,
$|O_2 M| = |O_2 H|$, $|O_2 M| = |O_2 J|$, $|OB| = |OJ|$,
O_2 is inside the rectangle with AB as diagonal.

The corresponding semi-algebraic system may be decomposed into three weak SRRs

$$\Theta_j(u_1), \; x_7 = X_{j7}, \ldots, x_{10} = X_{j10}, \quad j = 1, 2, 3,$$

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.

Subproblem 3. Construct circle I with the following bspecification:

$$\tan \angle CBI = \tan \angle IBA, \quad \tan \angle ACI = \tan \angle ICB,$$

I is inside $\triangle ABC$.

The corresponding semi-algebraic system may be decomposed into two weak SRRs:

$$\Lambda_k(u_1), \ x_{11} = X_{k11}, \ x_{12} = X_{k12}, \quad k = 1, 2.$$

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 1 for the computing time). The dynamic diagram shown in Figure 4 has been generated automatically, making use of the computed SRRs.

5. IMPLEMENTATION AND EMPIRICAL DATA

In this section, we discuss our implementation of the process of dynamic diagram generation using both the original and the enhanced HLLW approach, point out the functionalities of the current version of our implemented program, and present some empirical data in table form to show the performance of the program for generating the diagrams of 10 theorems in plane Euclidean geometry.

We have implemented the process of diagram generation sketched in Section 3 in Java with interface to the Epsilon library [14] in Maple and the QEPCAD program [1] 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 (see Figure 5) 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. This diagram generation program is independent of GEOTHER and the previous implementation of the HLLW approach therein. It is still under development and will be made available at http://geo.cc4cm.org.

The input specification of the geometric constraints is parsed and translated in Java into a 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 are well implemented. Our choice of Maple is mainly due to the fact that the Epsilon library [14] containing an implementation of several efficient triangular decomposition methods has been developed in the system. In our Java program, a piece of Maple code is generated and Maple is invoked to execute the code. In Maple, the set of (equality) polynomials is decomposed by using one of the Epsilon functions into irreducible triangular sets and QEPCAD is then invoked to eliminate quantifiers from the real solution formulas of equations of the polynomials in the triangular sets with inequality constraints. The computed SRRs are written into a file for the Java program to read in.

QEPCAD [1] is a powerful C program for quantifier elimination based on partial cylindrical algebraic decomposition [2]. 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
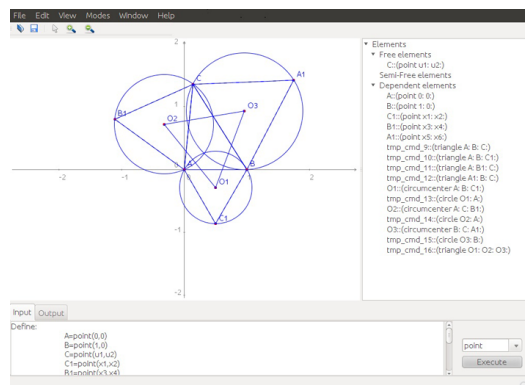

Figure 5: Graphic User Interface

elimination, including REDLOG [16, 4], QEQUAD [8], and SturmHabicht [6]. From some example tests, we have observed that QEPCAD takes less computing time and outputs simpler formulas in general. For the 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.

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, draw a static diagram according to the values of the variables, and update 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.

Table 1 shows the number of parameters, the number of dependent variables, the highest degree of equalities, and the number of inequality constraints for 10 semi-algebraic systems expressing the hypotheses of 10 well-known geometric theorems and the times for triangular decomposition, quantifier elimination, and the whole process of computing the SRRs of each semi-algebraic system. 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 and each call costs less than one second. From Table 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.

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.

Table 1: Data and Timings for 10 Examples

Theorem	Par No	Dep No	Max Deg	Ineq No	TD Time	QE Time	Time
Apollonius	4	4	3	3	0.228	82.761	84.927
Butterfly	4	7	2	1	0.085	42.530	44.100
Feuerbach	1	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

Par: Parameter; Dep: Dependant Variable; Ineq: Inequality; TD: Triangular Decomposition;
QE: Quantifier Elimination

6. CONCLUSION AND DISCUSSIONS

We have shown that the HLLW approach [9] used in dynamic diagram generation for computing SRRs of semi-algebraic systems can be enhanced by incorporating the Lagrange-type real solution formulas of cubic and quartic equations with inequality constraints introduced recently in [19, 10]. 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 process of automated generation of dynamic diagrams based on the enhanced HLLW approach has been sketched and implemented in Java with interface to Epsilon and QEPCAD. Our Java program 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.

Problems that remain for further investigation include (1) improving the efficiency of quantifier elimination by taking into account the structure of the quantified subformulas in the real solution formulas with inequality constraints, (2) simplifying the real solution formulas of quartic equations with or without constraints, (3) studying the numeric stability of solution formulas and the geometric continuity of dynamic diagrams when some denominators are close to zero, and (4) solving 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.

7. ACKNOWLEDGEMENTS

8. REFERENCES

[1] C. W. Brown and H. Hong. QEPCAD — quantifier elimination by partial cylindrical algebraic decomposition. 2011. http://www.cs.usna.edu/~qepcad/B/QEPCAD.html.

[2] G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *J. Symb. Comput.*, 12:299–328, 1991.

[3] R. Dixon. *Mathographics*. New York, Dover, 1991.

[4] 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.

[5] X.-S. Gao, K. Jiang, and C.-C. Zhu. Geometric constraint solving with conics and linkages. *Comput. Aided Design*, 34(6):421–433, 2002.

[6] 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.

[7] C. M. Hoffmann and R. Joan-Arinyo. A brief on constraint solving. *Comput. Aided Design Appl.*, 2(5):655–663, 2005.

[8] H. Hong. Quantifier elimination for formulas constrained by quadratic equations via slope resultants. *Comput. J.*, 36(5):440–449, 1993.

[9] 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.

[10] H. Hong, D. Wang, and T. Zhao. Solution formulas for quartic equations without or with constraints. 2011, in preparation.

[11] 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.

[12] R. H. Lewis and S. Bridgett. Conic tangency equations and Apollonius problems in biochemistry and pharmacology. *Math. Comput. Simul.*, 61(2):101–114, 2003.

[13] 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.

[14] D. Wang. *Elimination Practice: Software Tools and Applications*. Imperial College Press, London, 2004.

[15] 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.

[16] 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.

[17] L. Yang, X.-R. Hou, and Z.-B. Zeng. A complete discrimination system for polynomials. *Sci. China (Ser. E)*, 39(6):628–646, 1996.

[18] Z. Ye, S.-C. Chou, and X.-S. 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.

[19] T. Zhao, D. Wang, and H. Hong. Solution formulas for cubic equations without or with constraints. *J. Symb. Comput.*, 46(8):904–918, 2011.