Solving Geometric Constraint Systems

II. A Symbolic Approach and Decision of Rc-constructibility[2]

Xiao-Shan Gao
Institute of Systems Science
Academia Sinica
Beijing 100080, P.R. China
e-mail: xgao@mmrc.iss.ac.cn

Shang-Ching Chou
Department of Computer Science
The Wichita State University
Wichita, KS 67208, USA
e-mail: chou@cs.twsu.edu

Abstract. This paper reports a geometric constraint solving approach based on symbolic computation. With this approach, we can compute robust numerical solutions for a set of equations and give complete methods of deciding whether the constraints are independent and whether a constraint system is over-constraint. Based on symbolic computation, we also give a decision procedure for the problem of deciding whether a constrainted diagram can be constructed with ruler and compass (rc-constructibility).

Keywords. Geometric constraint system solving, symbolic computation, decision of ruler and compass construction, intelligent CAD.

1 Introduction

In Part I of this paper [10], we use a construction approach to solve geometric constraint systems. The advantage of this approach and all the other construction approaches [11, 15, 20] is that once a solution is found, the remaining steps of drawing this diagram become easy. Unfortunately, construction approaches are generally not complete, that is, the methods can not find drawing procedures for many diagrams that can be drawn with ruler and compass. In this paper, we study a complete approach of solving geometric constraint systems.

The *symbolic approach* is quite similar to the numerical approach [16, 19]. In the symbolic approach, we also first transform the geometric constraints into algebraic equations by assigning each point its coordinates. After that, instead of using numerical methods to solve the algebraic equation set, we will use general symbolic methods such as Wu-Ritt's decomposition algorithm [22] or the Gröbner basis method [1] to change the equation set to new forms which are easy to solve. Finally, we solve these new equations numerically. The most often used easy-solving equation form is the *triangular form* (see Section 2 for precise definition). In some of the numerical approaches [16, 19], the equations are also simplified, while triangulation could be considered an ultimate simplification. We should mention that a triangular system of equations is not always more robustly solved than the original system. Problems might occur if the polynomials in the triangular sets are very large or numerical error propagation is out of control. But as shown by this paper, triangularization does simplify the original equations in many cases.

In this paper, we use Wu-Ritt's decomposition algorithm [22] to solve geometric constraint systems. We discuss how to compute robust numerical solutions for a set of equations and how to decide whether the constraints are independent and whether a constraint system is over-constraint. Our program based on Wu-Ritt's decomposition method is quite efficient. All of the 134 constraint systems reported in [10] can be solved within seconds. For more complicated problems like the regular pentagon problem (Example 2.1), the program can give a much clearer picture than the numerical approach and can lead to efficient and robust solutions. Although many difficult problems can be solved with this approach, in the worst case Wu-Ritt's method has exponential complexity [7]. A combination of the construction and the symbolic approaches may provide a more practical method.

Related to the symbolic approach, Hoffmann used the Gr"obner basis method to solve some special constraint systems [11]. Kondo used the Groebner basis method to test whether two dimensional constraints are independent, and if not, to find the relation between them [14]. In [3], Wu-Ritt's method and the Gr"obner basis method were used to find relations among any geometry quantities.

By transforming the constraint system into algebraic form, we can give a decision procedure for the problem of deciding whether a diagram can be constructed with ruler and compass (rc-constructibility). The problem of rc-constructibility is reduced to decide whether the roots of a univariate equation of degree $2^k (k > 1)$ and with rational expressions in parameters as coefficients can be solved in terms of square roots. Our decision procedure is quite efficient for the special case of quartic equation, while in the general case it is an exponential algorithm. Fortunately, most geometric problems give equations of degree either two or four. For instance, 95% of the 134 problems reported in [10] give equations of degree two or four.

The rest of the paper is organized as follows. In Section 2, we describe how to solve constraint systems with Wu-Ritt's decomposition algorithm. In Section 3, we give the decision procedure for the rc-constructibility.

2 Solving Geometric Constraint Systems with Wu-Ritt's Decomposition Algorithm

2.1. Basic Results of Wu-Ritt's Decomposition Algorithm. We only introduce the results needed in this paper. For more discussions about the method, see [22]. Let $\mathbf{Q}$ be the field of rational numbers, and $\mathbf{Q}[x_1, ..., x_n]$ or $\mathbf{Q}[X]$ the set of polynomials in variables $x_1, ..., x_n$. For $P \in \mathbf{Q}[X]-\mathbf{Q}$, we can write $P = c_d x_p^d + ... + c_1 x_p + c_0$, where $c_i \in B[x_1, ..., x_{p-1}]$, $p > 0$, and $c_d \neq 0$. We call p the *class*, c_d the *initial*, and x_p the *leading variable* of P, or

$class(P) = p$, $init(P) = c_d$, and $lv(P) = x_p$.

A sequence of polynomials $ASC = A_1, ..., A_p$ in $\mathbf{Q}[X]$ is said to be in *triangular form*, if either $p = 1$ and $A_1 \neq 0$ or $0 < class(A_i) < class(A_j)$ for $1 \leq i < j$.

Let PS be a polynomial set in $\mathbf{Q}[X]$. For an extension field E of $\mathbf{Q}$, let

$$Zero(PS) = \{a \in E^n \mid \forall P \in PS, P(a) = 0\}$$

For a polynomial set PS and a polynomial D in $\mathbf{Q}[X]$, let $Zero(PS/D) = Zero(PS) - Zero(D)$.

For a polynomial set PS in $\mathbf{Q}[X]$, *Wu-Ritt's zero decomposition algorithm* can be used to represent the zero set of PS as the union of the zero sets of polynomial sets in triangular form. More precisely, we have [22]

$$Zero(PS) = \bigcup_{1 \leq i \leq k} Zero(TS_i/I_i) \tag{2.1}$$

where each TS_i is a polynomial set in triangular form and I_i is the production of the initials of the polynomials in TS_i.

Let $TS = \{P_1, ..., P_p\}$ be a polynomial set in triangular form. We rename the leading variables of P_i as y_i and the remaining variables as $u_1, ..., u_q (p + q = n)$. Then TS can be written as

$$P_1(U, y_1), P_2(U, y_1, y_2), \cdots, P_p(U, y_1, y_2, ..., y_p) \tag{2.2}$$

where $U = u_1, ..., u_q$.

In Section 3, we need the following result [9]. For each triangular set TS of the form (2.2), let I be the production of the initials of its polynomials. Then we can find a new variable w such that $Zero(TS/I)$ can be obtained as follows.

$$R(U, w) = 0, y_i = M_i(U, w)/N_i(U, w), (i = 1, ..., p) \tag{2.3}$$

where R, M_i, and N_i are polynomials in $\mathbf{Q}[U, w]$. In other words, the solutions of a polynomial set in triangular form can be reduced to the roots of a univariate polynomial equation.

The solution of a set of polynomial equations in triangular form is easy to compute. Take TS in (2.2) as an example. For a set of numerical values for the u_i, $P_1 = 0$ becomes a univariate equation. We can find the solutions for y_1 from this univariate equation. Next, by substituting the numerical solution for y_1 into equation P_2, $P_2 = 0$ becomes a univariate equation. We now solve y_2 from this univariate equation, and so on. We can use Newton's iteration method to solve univariate polynomial equations. To find all the real roots of a univariate equation, we can first use the algorithm reported in [5] to separate the roots of the equations and then use Newton's method to compute each of the root.

2.2. A Non-trivial Example. Let us first give an example.

Example 2.1 (Regular Pentagon) *Draw a regular pentagon $ABCDE$ whose edge is r.*

Let $A = (0,0), B = (r,0), C = (x_2, y_2), D = (x_3, y_3), E = (x_4, y_4)$. Since there are five points in the diagram, we need $2*5-3=7$ constraints to determine the shape of the diagram. The constraint $|AB| = r$ is already satisfied automatically. We need six other constraints:

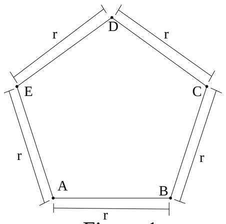

Figure 1

$$
\begin{array}{ll}
|AB| = |AE|: & h_1 = y_4^2 + x_4^2 - r^2 = 0 \\
|AB| = |BC|: & h_2 = y_2^2 + x_2^2 - 2rx_2 = 0 \\
|AB| = |CD|: & h_3 = y_3^2 - 2y_2y_3 + x_3^2 - 2x_2x_3 + y_2^2 + x_2^2 - r^2 = 0 \\
|AB| = |DE|: & h_4 = y_4^2 - 2y_3y_4 + x_4^2 - 2x_3x_4 + y_3^2 + x_3^2 - r^2 = 0 \\
|AD| = |DB|: & h_5 = 2rx_3 - r^2 = 0 \\
|AC| = |CE|: & h_6 = y_4^2 - 2y_2y_4 + x_4^2 - 2x_2x_4 = 0.
\end{array}
$$

Using Wu–Ritt's decomposition algorithm, we have

$$Zero(\{h_1, ..., h_6\}) = \cup_{i=1}^{4} Zero(TS_i/I_i) \tag{2.4}$$

where TS_i are given below.

$TS_1 =$	$TS_2 =$	$TS_3 =$	$TS_4 =$
$y_4 - y_2$	$ry_4 + (2x_2 - 2r)y_2$	$y_4 - y_2$	y_4
$x_4 + x_2 - r$	$2x_4 - 2x_2 + r$	$x_4 + x_2 - r$	$x_4 + r$
$ry_3 + (-2x_2 + r)y_2$	$ry_3 + (2x_2 - 3r)y_2$	$ry_3 + (2x_2 - 3r)y_2$	$y_3 - y_2$
$2x_3 - r$	$2x_3 - r$	$2x_3 - r$	$2x_3 - r$
$4y_2^2 - 2rx_2 - r^2$	$4y_2^2 - 2rx_2 - r^2$	$4y_2^2 - 2rx_2 - r^2$	$4y_2^2 + 5r^2$
$4x_2^2 - 6rx_2 + r^2.$	$4x_2^2 - 6rx_2 + r^2.$	$4x_2^2 - 6rx_2 + r^2.$	$2x_2 + r.$

TS_4 does not have non-zero solutions, since there is a polynomial $4y_2^2 + 5r^2$. Each of TS_1, TS_2, and TS_3 contains six polynomials and has total degree four, i.e., for a non-zero value of r, each of TS_1, TS_2, and TS_3 gives four solutions to the problem (see Figures 2 and 3).

2.3. Coordinate Transformation. To improve the efficiency of Wu-Ritt's algorithm, we always use special coordinates for some points. In the pentagon example, points $A = (0,0), B = (r,0)$ have special coordinates. This example actually gives some insight into the general case. If a plane geometric constraint system involves n points, we generally need $2n-3$ independent constraints to determine the shape of the diagram. Such a constraint system or its corresponding diagram is called *well-defined*. There are three free parameters in a well-defined diagram: the coordinates of a point P_0 and the direction of a line P_0P_1. When using Wu-Ritt's decomposition algorithm, we always assume that the coordinates of P_0 are $(0,0)$ and the coordinates of P_1 are $(r,0)$. This will make the decomposition much faster.

When drawing the constrainted diagram on a window, coordinates of point A are usually not $(0,0)$ and line P_0P_1 is usually not horizontal. We thus need to do certain coordinate transforms before we can use the triangular equations to find the coordinates of a diagram in the general position. We need three basic transformations for a well-defined diagram: (1) Translate point P_0. (2) Rotate line P_0P_1 around point P_0 with the length of P_0P_1 fixed. (3) If the length of a segment or the value of an angle is given as a parameter, we can change the segment length or the angle value. Now we can give the procedure of computing the coordinates of the points in a diagram.

1. For a given set of numerical values for the segment lengths and angles, we can use the original triangular set TS to obtain a set of solutions for the coordinates of the points.
2. For each given new position of P_0 and a new direction of line P_0P_1, we can use the above transformations to find the new coordinates for all other points.

For the complexity of the above procedure, the second step is clearly linear in the number of points in the diagram. We also need to solve univariate equations which is also fast [6].

In this approach, instead of solving equations in general form, we need only solve univariate equations due to the fact that the equations are in triangular form. As a consequence, we can find all the solutions if needed. This makes the selection of the robust solutions possible. Second, for each set of segment lengths and angle values, we need only to solve a fixed number of univariate equations for *one time*. After a set of solutions is obtained, for each new position of P_0 and each new direction of P_0P_1 we need only to do a linear transformation to obtain their positions. Of course, in our approach, we need first to find a triangular decomposition, which might be time consuming. Fortunately, we need only to do that once for each constraint system. So we may adopt the following strategy: if in a fixed period of time, say five minutes, the program still cannot find the triangular decomposition then we will abandon the process and use the general numerical approach.

2.4. Over-Constrainted Systems. In many papers including Part I of this paper [11, 20, 10], a constraint system is called *well-defined* if it has n characteristic points and lines and $2*n-3$ independent constraints. As we will show below, though good enough for the construction and AI approach to constraint solving, this definition does not give the whole picture. Generally, let us call a constraint system *well-constrainted* if the shape of the corresponding diagram has only a finite number of cases; *under-constrainted* if the shape of the corresponding diagram has infinite solutions; *over-constrainted* if the corresponding diagram has no solution. From the discussion in the last paragraph in Section 2.1, the following result is easy to prove.

Theorem 2.2 *Let P_0 and P_1 be any two points in the constraint system such that length of P_0P_1 is r. Let $P_0=(0,0), P_1=(r,0)$. Then in a decomposition like (2.1), (1) the system with n points is well-constrainted iff each triangular set TS contains $2n-4$ polynomials; (2)*

the system is over-constrainted iff there is no triangular set if all length and angle parameters are considered as constants; (3) the system is under-constrainted iff at least one triangular set contains less than $2n-4$ *polynomials.*

We use the pentagon example to show that sometimes we need more than $2n-3$ constraints to define a constraint system with n points. The following table gives the result of checking three more geometric conditions on the three diagrams defined by TS_1, TS_2, and TS_3 using the automated theorem proving methods given in [22].

	TS_1	TS_2	TS_3
$\|AD\| = \|AC\|$	true	false	false
$\|BD\| = \|BE\|$	true	true	false
$\|EB\| = \|EC\|$	true	false	true

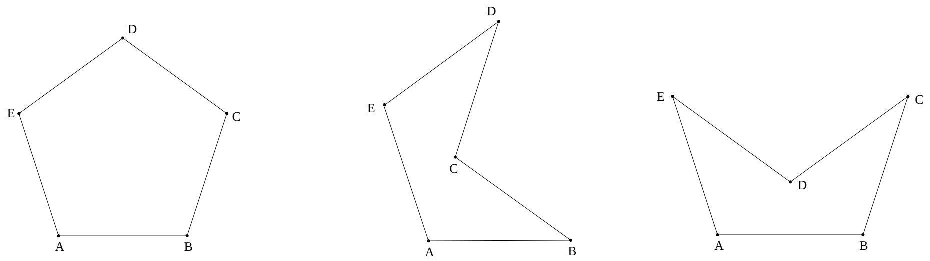

Figure 2a Figure 2b Figure 2c

By Theorem 2.2, the pentagon problem in Section 2.2 is well-defined. But the seven independent constraints do not properly define a regular pentagon. To define a regular pentagon, we need to add at least another constraint

$$|AD| = |AC|: \quad h_7 = y_3^2 + x_3^2 - y_2^2 - x_2^2.$$

Now there is only one useful triangular set TS_1 in the decomposition:

$$Zero(\{h_1, ..., h_7\}) = Zero(TS_1/I_1) \cup Zero(TS_4/I_4). \tag{2.7}$$

But this constraint system is *over-constraint* in the sense that there are five points but eight constraints. Note that the last constrain $h_7 = 0$ does not reduce the dimension of freedom of the problem; it only eliminates eight of the twelve solutions of the original problem.

We have observed that for some constraint systems, we must use more than $2n-3$ consistent constraints to properly determine the shape of the corresponding diagram. It is difficult for iteration numerical methods to deal with this kind of problems [16]. In the symbolic approach, this kind of "over-constraint" can be solved naturally. Furthermore, the triangular set of equations generally becomes simpler when more consistent constraints are added.

2.5. Robustness of Numerical Solving. It is well known (see e.g., [16]) that the Newton iteration method is unreliable, especially when the number of constraints are larger than the number of variables. From the symbolic viewpoint, this unreliability can be divided into two cases. The first case occurs when a triangular set TS has more than one solutions. When changing the shape of the diagram continuously, the solver may "jump" from one branch of solutions to another branch. For instance, TS_1 for the pentagon has four branches of solutions. The two solutions given from the quadratic equation about y_2 are easy to understand: if P is a solution pentagon then the reflection of P with the x-axis, obtained by changing y_i to $-y_i$, is also a solution. The two solutions given by the quadratic equation about x_2 are two regular pentagons, one with r as an edge another with r as a diagonal (Figure 3).

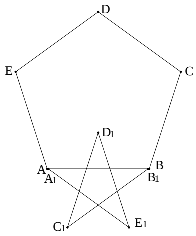

Figure 3

A trick of reducing the branches of solutions is to introduce more constraints. For the pentagon example, we can assume that $ABCDE$ is a concave pentagon by introducing a condition $\theta_1 = (|AD|^2 > |AB|^2)$. Similarly, we can assume that $A-B-C$ is in counterclockwise direction by introducing anther condition $\theta_2 = (y_2 > 0)$. Now the pentagon problem has a unique solution and the solution will always be robust. In summary, the numerical method works as follows. Let us assume that the current value for a variable x is x_0. When we drag a diagram, the next solution for x is near x_0. Thus, we can use x_0 as an initial value to find the new value for x. We then check whether the constraints θ_1 and θ_2 are valid for these new numerical values. If they are valid, we have found a solution. Otherwise, most probably we are near a "critical point" where more than one branches of solutions meet. Now we need to find all the solutions of the triangular set and to select one satisfying θ_1 and θ_2.

The second case occurs if the problem is *reducible*, i.e., its solutions can be written as the union of the solutions of more than one triangular equation sets. If $h_7 = 0$ is not added, the pentagon problem has three group of solutions, and hence is reducible. In a reducible problem, when changing the shape of the diagram continuously, the solver may "jump" from the solutions of one triangular set to that of another triangular set. This problem can be solved trivially if a triangular decomposition is already given: different solutions are given by different triangular sets and will not be mixed.

2.6. Independent and Conflict Constraints. Recall that a set of constraints are said to be *independent* if no one of them can be removed and the new constraint system still defines the same diagram. A set of constraints are said to be *conflict* if no diagram in the Euclidean plane satisfying this constraint system.

Many results about the independence and conflict of constraints are reported in [21]. One of the interesting result is as follows. If the only type of constraints is distance between two points and all the distances do not have algebraic relations, a constraint system is independent and non-conflict[3] if in any subsystem of n points, the number of constraints $e \le 2n-3$. This and other results reported in [21], though important, have the following strong limitation. The distances and angles have to be algebraic independent which means that we can not deal with problems involving two points on the same circle. On the other hand, the symbolic approach can provide complete solution to the problem of independency and conflict in the general case.

Let us first comment that in the general case there is no connection between the concept of independency and conflict and the critical number $2n-3$. Actually, there are conflict and/or non-independent systems which have less than $2n-3$ constraints, as well as non-conflict and/or independent systems with more than $2n-3$ constraints. For instance, the regular pentagon is a non-conflict and independent system with more than $2n-3$ constraints. Also, almost every geometry theorem provides an example for non-independent systems with less than $2n-3$ constraints.

2.6.1. Decision of Conflict Constraints. For a constraint system CS, let PS be the corresponding algebraic equations of the constraints. The conflict of CS means that PS does not have solutions. In the case of complex number field, this can be decided with the following result. For the case of real numbers, a complete algorithm can be provided by the quantifier elimination algorithm [6].

Theorem 2.3 ([22]) *A set of polynomial equations PS does not have solutions over the field of complex numbers iff $k=0$ in decomposition[4] (2.1). If $k=0$ in decomposition (2.1), the constraint system is also conflict over the field of real numbers.*

If we want to draw a regular pentagon with edge 1 and diagonal 2, we will get a conflict constraint system. In other words, $Zero(PS)=\emptyset$ where PS consists of equations for $|AB| = |BC| = |CD| = |DE| = |EA| = 1$ and $|AD| = |DB| = |BE| = |EC| = 2$.

If two dimensional constrains are conflict, we can find the relation between the two dimensions. In [14], this is done with the Groebner basis. In [3], both the Groebner basis and Wu-Ritt's method are used to find such relations. In the regular pentagon problem, we just showed that the edge r and the diagonal d are two conflict dimensions. To find a relation between them, let PS be the set of equations for the constraints $|AB| = |BC| = |CD| = |DE| = |EA| = r$ and $|AD| = |DB| = |BE| = |EC| = d$. Using Wu-Ritt's decomposition algorithm, we have

$$Zero(PS) = Zero(TS_5/I_5) \cup Zero(TS_6/I_6)$$

[3] Here the concept of conflict is over the complex numbers. That is, a constraint system is non-conflict if there is a diagram over the field of complex numbers satisfying all the constraints.

[4] Here we assume that the triangular sets are irreducible. For the definition of irreducible triangular sets, see [22].

where TS_5 and TS_6 are polynomial sets in triangular form and I_5 and I_6 are the products of the corresponding initials. The first polynomials in TS_5 and TS_6 are

$$d^2 + rd - r^2 = 0 \text{ and } d^2 - rd - r^2 = 0.$$

Since d and r are positive, the two reasonable solutions are $d = \frac{1+\sqrt{5}}{2}r$ and $d = \frac{-1+\sqrt{5}}{2}r$. The first one represents the case that r is the edge and d is the diagonal. The second one represents the case that r is the diagonal and d is the edge.

2.6.2. Decision of Independency. Let PS be the equation set of a constraint system. This system is not independent iff there is a polynomial P in PS such that

$$Zero(PS - \{P\}) \subset Zero(P) \text{ or } Zero(PS - \{P\}) = Zero(PS). \tag{2.8}$$

Theorem 2.4 *If $|PS| \leq 2n-3$, the constraints are independent iff in decomposition (2.1) there is at least one irredundant triangular set TS_i containing $|PS|$ polynomials.*

To prove this theorem, we need a theorem from algebraic geometry which says that the triangular set representing an irredundant component of $Zero(PS)$ cannot contain more than $|PS|$ polynomials [8]. If the constraints are not independent, $Zero(PS) = Zero(PS - \{P\})$. According to above theorem, an irredundant triangular set contains less than $|PS|$ polynomials. This is contradict to the hypothesis of the theorem.

In the general case, we need to check whether there is a polynomial in PS such that (2.8) is valid. Note that to check (2.8) is actually to prove a geometry theorem. Let C_P be the geometric constraint of $P=0$ and $C_1, ..., C_n$ the geometric constraints for other equations in PS. Then (2.8) is equivalent to the following geometry theorem

$$\forall \text{ points}[(C_1 \wedge C_2 \cdots \wedge C_n) \Rightarrow C_P].$$

The problem of *automated geometry theorem proving* has been studies extensively [22]. We may use these efficient algorithms to prove whether (2.8) is true.

In the regular pentagon example, applying Theorem 2.4 to (2.4), we know that the seven constraints given in Section 2.2 are independent. Since adding constraint $|AD| = |AC|$ eliminates two components from (2.5) (see (2.7)), this new constraint is also independent to the seven constraints. A new constraint $|BD| = |BE|$ is dependent to the eight constraints in (2.7), because it is true on $Zero(TS_1/I_1)$.

3 A Decision Procedure for Rc-constructibility

3.1. Basic Results of Rc-constructions. Let $\mathbf{Q}$ be the field of rational numbers. We allow us to abuse terms by saying that the numerical solutions of a diagram can be constructed with ruler and compass if the corresponding diagram can be constructed with ruler and compass. It is well-known [13] that a number η can be constructed with ruler and compass iff there exist n real numbers $\eta_1, \eta_2, ..., \eta_n = \eta$ and n quadratic equations

$$\begin{aligned}
&Q_1(x_1) = x_1^2 + b_1 x_1 + c_1 = 0\\
&Q_2(x_1 x_2) = x_2^2 + b_2(x_1)x_2 + c_2(x_1) = 0\\
&...\\
&Q_n(x_1, x_2, ..., x_n) = x_n^2 + b_n(x_1, ..., x_{n-1})x_n + c_n(x_1, ..., x_{n-1}) = 0
\end{aligned}$$

such that $Q_i(\eta_1, \eta_2, ..., \eta_i) = 0, (i = 1, ..., n)$. Since Q_i is of degree two, we can further assume that the b_i and c_i are linear in the variables $x_1, ..., x_{i-1}$. From the above equations, we have

$$\eta_1 = \frac{\eta_n^{2^{n-1}} + P_1(\eta_n)}{Q_1(\eta_n)} \tag{3.1}$$

where P_1 and Q_1 are polynomials of degree $2^{n-1} - 1$. Substituting (3.1) into $Q_1(\eta_1) = \eta_1^2 + b_1\eta_1 + c_1 = 0$ and taking the numerator, we have

$$\eta_n^{2^n} + h_{2^n-1}\eta_n^{2^n-1} + ... + h_0 = 0. \tag{3.2}$$

As a consequence, we have the well known result: *an rc-constructible number is the root of an equation of degree* 2^k *for some* $k > 0$. Thus *the root of a cubic polynomial equation is rc-constructible iff the polynomial is reducible.*

Since we can efficiently factor polynomials with computer algebraic systems such as Maple [2], the only remaining problem is to decide whether the roots of polynomial equations of degree 2^k are rc-constructible.

3.2. Quartic Equations. Let a quartic equation in $K[x]$ be

$$x^4 + h_3x^3 + h_2x^2 + h_1x + h_0 = 0 \tag{3.3}$$

where K is field. In our case, K could be the field of rational numbers $\mathbf{Q}$ or the field of rational functions $\mathbf{Q}(u_1, ..., u_m)$.

Let us assume that the quartic polynomial in (3.3) is irreducible. If a root y of (3.3) is rc-constructible, y can be written as the solution of the following two equations

$$\begin{aligned}
&x_1^2 + bx_1 + c = 0\\
&y^2 + (-mx_1 + f)y - nx_1 + g = 0
\end{aligned}$$

The first equation can be reduced to $x_1^2 + d = 0$. Since the roots are real numbers, we can further assume that the first equation is $x_1^2 - 1 = 0$. From the second equation we have $x_1 = \frac{y^2+fy+g}{my+n}$. Substitute this into $x_1^2 - 1 = 0$, the numerator is $(y^2+fy+g)^2 - (my+n)^2 = 0$. Comparing the coefficients of this equation with that of (3.3), we obtain a set of equations

$$2f - h_3 = 0$$

$$m^2 - 2g - f^2 + h_2 = 0$$
$$2mn - 2fg + h_1 = 0$$
$$n^2 - g^2 + h_0 = 0$$

Eliminating f, m, and n, we have

$$8g^3 - 4h_2 g^2 + (2h_1 h_3 - 8h_0)g - h_0 h_3^2 + 4h_0 h_2 - h_1^2 = 0 \tag{3.4}$$

Let g_1 be a root of equation (3.4), f_1, m_1 and n_1 solutions for f, m and n with g substituted by g_1. Then all the roots of equation (3.3) can be obtained from the quadratic equations

$$y^2 + f_1 y + g_1 \pm (m_1 y + n_1) = 0.$$

Therefore, the roots of equation (3.3) can be obtained by successively solving several quadratic and a cubic equations. By the discussion in the end of Section 3.1, roots of a cubic equation are rc-constructible iff the cubic polynomial is reducible, i.e., it has a linear factor. Then we have a method of deciding whether the roots of a quartic equation can be constructed with ruler and compass.

Example 3.1 (Pappus' Problem I) *Given a fixed point P, a fixed angle AOB, and a fixed length d, draw a line passing through point P and cutting lines OA and OB with a segment of length d.*

Let the segment be XY. Use the following coordinates $O = (0,0), A = (w,0), P = (u,v), B = (w,s), X = (x_1,y_1), Y = (x_2,0)$. If considering w,u,v, and s as non-zero constants, with Wu-Ritt's method, we can find a triangular set which contains the following quartic equation $f = s^2(s^2+w^2)x_1^4 - 2ws^2(vs+wu)x_1^3 + (v^2+u^2-d^2)w^2s^2x_1^2 + 2d^2w^3vsx_1 - d^2w^4v^2 = 0$. We can calculate a cubic equation from the quartic equation $f = 0$. Using factorization algorithms from [2], we show that this cubic polynomial is irreducible. This means that the diagram generally cannot be drawn with ruler and compass.

Example 3.2 (Pappus' Problem II) *All requirements are the same as in Example 3.1 with an additional condition that point P is on the bisector of angle AOB.*

We need only to add a new constraint $\angle AOP = \angle POB$ to the constraint set of Example 3.1. Using Wu-Ritt's characteristic set method, we still get a quartic equation of x_1. This time the corresponding cubic equation is reducible and has the following linear factor of variable g:$-12(u^2+v^2)^2 g + 2(u^2-v^2)(u^4-v^4) + (u^2-v^2)^2 d^2 = 0$. Then the quartic equation can be reduced to linear and quadratic equations and the line can be drawn with ruler and compass.

3.3. The General Case. We want to know whether the roots of the following equation

$$H(x) = x^{2^n} + h_{2^n-1}x^{2^n-1} + \ldots + h_0 = 0 \tag{3.5}$$

in $K[x]$ are rc-constructible. Let us first assume that the polynomial in (3.5) is irreducible which can be checked with maple [2].

If $K = \mathbf{Q}$, there exists a polynomial time algorithm to decide whether $H = 0$ is solvable using radicals [17]. But in the general case, $K = \mathbf{Q}(u_1, ..., u_q)$. The method presented in [17] does not work in this case. One possibility is to give a probabilistic algorithm as follows. For a given polynomial H involving parameters $u_1, ..., u_q$, we take q random numbers $e_1, ..., e_q$ from $\mathbf{Q}$ and substitute u_i for e_i in H to obtain a polynomial $H' \in \mathbf{Q}[x]$. Then we can use the algorithm in [17] to decide whether the roots of $H' = 0$ are rc-constructible or equivalently whether they can be represented by square roots. If the roots of $H' = 0$ are not rc-constructible then the original problem is not rc-constructible. In the following, we will give a decision algorithm for the general case.

Similar to the quartic case, if a root y of equation (3.5) is rc-constructible, by (3.1) there is a number x_1 such that $x_1 = (y^{2^{n-1}} + P(y))/Q(y)$ and $x_1^2 - c = 0$ where P and Q are polynomials of degree $2^{n-1} - 1$. Substituting x_1 into $x_1^2 - c = 0$, we have $(y^{2^{n-1}} + P(y))^2 - cQ(y)^2 = 0$. Since an rc-constructible number is a real number, we can assume $c > 0$. Then the above equation becomes

$$(y^{2^{n-1}} + P(y))^2 - (\sqrt{c}Q(y))^2 = (y^{2^{n-1}} + P(y))^2 - R(y)^2 = 0. \tag{3.6}$$

Let $P = p_{2^n-1}x^{2^n-1} + ... + p_0$ and $R(y) = \sqrt{c}Q = q_{2^n-1}x^{2^n-1} + ... + q_0$. Substituting them into (3.6) and comparing coefficients with (3.5), we obtain a set of equations PS.

Lemma 3.3 *Use the notations introduced above. The equation set PS has a finite number of solutions over any extension filed of K.*

Proof. See [4]. ∎

Now let us consider equation (3.6) more carefully. Let $k = 2^{n-1}$. Considering the coefficients of $y^{2(k-1)}, y^{2k-3}, ..., y^{k-1}$ in $R(y)^2$, the corresponding equations in PS have the form

$$\begin{aligned}
R_1 &= q_{k-1}^2 + f_1(p_0, ..., p_{k-1}) = 0\\
R_2 &= 2q_{k-1}q_{k-2} + f_2(p_0, ..., p_{k-1}) = 0\\
R_3 &= 2q_{k-1}q_{k-3} + f_2(p_0, ..., p_{k-1}, q_{k-2}) = 0\\
&\cdots\\
R_k &= 2q_{k-1}q_0 + f_k(p_0, ..., p_{k-1}, q_{k-2}, ..., q_1) = 0.
\end{aligned}$$

If $q_{k-1} \neq 0$, $RS = \{R1, ..., R_k\}$ is a triangular set with leading variables $q_{k-1}, q_{k-2}, ..., q_0$. Let P be a polynomial in PS but not in RS. If P contains variables q_i, from equation (3.6) we see that $P = P_1(p_0, ..., p_{k-1}) + P_2(q_0, ..., q_{k-2})$ and $P_2(q_0, ..., q_{k-2})$ is a homogeneous quadratic polynomial. Since $q_{k-1} \neq 0$, $P = 0$ is the same as $q_{k-1}^2 P = 0$. Using $R_1, ..., R_k$, we can eliminate all the q_i from P. Let PS_1 be the set of polynomials obtained as above.

Lemma 3.4 *Use the notations introduced above. (1) Equation (3.5) has a decomposition like (3.6) iff PS_1 has a set of solutions in K.*

(2) Let a solution of PS_1 be $(e_0, ..., e_{k-1}), e_i \in K$. Then all roots y of (3.5) can be obtained as follows

$$y^k + e_{k-1}y^{k-1} + ... + e_0 - (q_{k-1}y^{k-1} + ... + q_0) = 0$$
$$R_1 = 0,\ R_2 = 0,\ \cdots,\ R_k = 0,\ (q_{k-1} \neq 0).$$

If $q_{k-1} = 0$, we can use the largest l such that $q_l \neq 0$ as the leading coefficient for R to find a similar set of equations.

Proof. See [4]. ∎

Let us first note that if a polynomial equation system PS_1 has a finite number of solutions over any extension field of K. Then the problem of deciding whether PS_1 has a set of solutions over K is easy. This is equivalent to the fact that in a decomposition like (2.2), there exists a triangular set consisting of linear polynomials only.

Now we can briefly describe the algorithm as follows. First by using Wu-Ritt's decompositions algorithm and formula (2.3), we need only consider whether the roots of an irreducible univariate polynomial can be constructed with ruler and compass. If the degree of R_i in w is not a power of two, the roots of R_i are not rc-constructible. Let R_i be of degree 2^m. Using Lemma 3.4, we can decide whether the roots of $R_i = 0$ can be written as the roots of a set of linear and quadratic equations and the roots of an equation of degree 2^{m-1}. Using this process repeatedly, we can decide whether the roots of R_i can be written as the solutions of a set of linear and quadratic equations. If the answer is positive, the geometric problem is rc-constructible; otherwise, the geometric problem is not rc-constructible.

Although this decision procedure is an extension of the efficient algorithm for quartic equations, it is not efficient. For higher degree polynomials, the rc-constructibility is reduced to the solution of a set of quadratic equations PS_1. It is known that solving zero dimensional equations is single exponential in the number of variables [18]. Then our algorithm should also be of single exponential in the degree of the polynomial equation.

References

[1] B. Buchberger, Gröbner Bases: an Algorithmic Method in Polynomial Ideal Theory, *Recent Trends in Multidimensional Systems theory* (ed. N.K. Bose), D.Reidel Publ. Comp., 1985.

[2] B. W. Char, et al, *Maple V*, Springer-Verlag, 1991.

[3] S. C. Chou & X. S. Gao, Mechanical Formula Derivation in Elementary Geometries, *Proc. ISSAC-90*, ACM, New York, 1990, p. 265–270.

[4] X. S. Gao & S. C. Chou Solving Geometric Constraint Problems, II. A Symbolic Approach and Decision of Rc-constructibility, WSUCS-96-3, CS Dept, Wichita State University, 1996.

[5] G.E. Collins, Infallible Calculation of Polynomial Zeros to Specified Precision, *Mathematical Software*, p. 35–68, Academic Press, New York, 1977.

[6] G. E. Collins, Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition, Lect. Notes Comp. Sci. 33, 134–183, Springer-Verlag, 1975.

[7] C. Gallo and B. Mishra, Efficient Algorithms and Bounds for Wu-Ritt Characteristic Sets, TR, Courant Institute of Math. Sciences, 1989.

[8] X.S. Gao & S.C. Chou, On the Dimension for Arbitrary Ascending Chains, *Chinese Bull. of Scis.*, vol. 38, pp.396–399, 1993.

[9] X.S. Gao & S.C. Chou On the Theory of Resolvents and Its Applications, MM-Research Preprints, No.6, Inst. of Systems Science, Academia Sinica, (submitted to J. of Symbolic Computation).

[10] X.S. Gao & S.C. Chou, Solving Geometric Constraint Systems, Part I. A Global Propagation Approach, WSUCS-96-1, CS Dept, Wichita State University, 1996 (submitted).

[11] C. Hoffmann, Geometric Constraint Solving in R^2 and R^3, in "Computing in Euclidean Geometry", eds D. Z. Du and F. Huang, p. 266–298, World Scientific, 1995.

[12] C. Hoffmann and I. Fudos, Constraint-based Parametric Conics for CAD, *Geometric Aided Design*, vol 28, No. 2, p.91–100, 1996.

[13] N. Jacobson, *Basic Algebra*, Vol. 1, San Francisco, Freeman, 1985.

[14] K. Kondo, Algebraic method for manipulation of dimensional relationships in geometric models, *Geometric Aided Design*, 24(3), p. 141-147, 1992.

[15] G. Kramer, *Solving Geometric Constraint Systems*, MIT Press, 1992.

[16] A. Heydon & G. Nelson, The Juno-2 Constraint-Based Drawing Editor, *SRC Research Report 131a*, 1994.

[17] S. Landau & L. Miller, Solvability by Radicals Is in Polynomial Time, *J. of Computer and System Sciences*, 30, p.179-208, 1985.

[18] D. Lazard, Solving Zero-Dimensional Algebraic Systems, *J. of Symbolic Computation*, 13/2. 117-131.

[19] V.C. Lin, D.C. Gossard, & R.A. Light, Variational Geometry In Computer-Aided Design, *Computer Graphics*, vol. 15, no. 3, 171–177, 1981.

[20] J. Owen, Algebraic Solution for Geometry from Dimensional Constraints, in *ACM symp. Found. of Solid Modeling*, p. 397-407, Austin TX, 1991.

[21] J. Owen, Constraints of Simple Geometry in Two and Three Dimensions, To appear in *Inter. J. of Comp. Geometry and Its Applications.*

[22] Wu Wen-tsün, *Basic Principles of Mechanical Theorem Proving in Geometries*, Volume I: Part of Elementary Geometries, Science Press, Beijing (in Chinese), 1984. English version, Springer-Verlag, 1993.