MM Research Preprints, 8–24
No. 15, April 1997. Beijing

Solving Geometric Constraint Systems II. A Symbolic Approach and Decision of Rc-constructibility[1)]

Xiao-Shan Gao[2)] and Shang-Ching Chou[3)]

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. We give complete methods of deciding whether the constraints are independent and whether a constraint system is over-constraint. Also, over-constrainted systems can be handled naturally. 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).

1. Introduction

In Part I of this paper [10], we use a construction approach to solve geometric constraint systems. More precisely, the algorithm tries to find a drawing procedure for a constrainted diagram with ruler and compass. 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 becomes easy. Unfortunately, construction approaches are generally not complete, that is, the methods can not find drawing procedures for some of the diagrams that can be drawn with ruler and compass. In this paper, we study two complete approaches of solving geometric constraint systems: the symbolic approach and a decision method for ruler and compass constructibility based on symbolic computation.

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 Ritt-Wu's decomposition algorithm [23, 25] or the Groebner 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). The advantage of the symbolic approach is that once an equation set of triangular form is obtained, the solution of these equations is much easier than that of the equation set in general form. In some of the numerical approaches [16, 19], the equations are also simplified, while triangulation could be considered the ultimate simplification.

[1)] This work was supported in part by the NSF Grant CCR-9420857 and the Chinese National Science Foundation.

[2)] Institute of Systems Sciences, Academia Sinica, Beijing 100080. email:xgao&mmrc.iss.ac.cn

[3)] Department of Computer Science, The Wichita State University Wichita KS 67260-0083, USA e-mail: choucs.twsu.edu

In this paper, we use Ritt-Wu's decomposition algorithm [25, 23] to solve geometric constraint systems. We will discuss the problem of using special coordinates to simplify the triangular sets and how to transform the solutions of a triangular set with special coordinates to the general position.

Distinctive feature of this approach includes. (1) We can compute robust numerical solutions from an equation set of triangular form. (2) We give complete methods of deciding whether the constraints are independent and whether a constraint system is over-constraint. (3) Different from the construction approach [11, 15, 20, 10], both the numerical and the symbolic approaches can be used to solve constraint systems in $2D$ and $3D$ uniformly, since both approaches are capable of handling equations with any number of variables.

Our program based on Ritt-Wu'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.

Related to the symbolic approach, Hoffmann used the Grobner 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 [4], Ritt-Wu's method and the Grobner 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 functions of parameters as coefficients can be solved in terms of square roots. Our decision procedure is quite efficient for the special case of quartic equation. Most geometric problems give equations of degree either two or four. For instance, 90% of the 134 problem reported in [10] give equations of degree one or two and all the remaining problems give equations of degree four.

In [20], Owen proved that his algorithm of solving constraint systems is complete if the constraint system uses only one predicate – distance between two points – and the distances used are algebraiclly independent. Of course, Owen's approach is not based on symbolic computation. It is still very desirable to develop a complete rc-constructibility method without using symbolic computations. A challenge problem is to find a set of construction rules which can be used to draw regular polygons with five or seventeen sides.

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

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

2.1. Basic Results of Ritt-Wu's Decomposition Algorithm. We only introduce the results needed in this paper. For more discussions about the method, see [23, 25, 3]. 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 $\mathbf{Q}$, like the complex number field, 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 $\mathbf{Q}[X]$, we define

$$Zero(PS/D) = Zero(PS) - Zero(D).$$

For a polynomial set PS in $\mathbf{Q}[X]$, *Ritt–Wu'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 [25, 3]

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

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 [23, 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 reason we use triangular form is that 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. According to (2.1), we need to find the zero set $Zero(TS_i/I_i)$. Since $I_i \neq 0$, we can discard any solutions that vanishes I_i. Thus for each set of values for the u_i, either $Zero(TS_i/I_i)$ is empty or we can find solutions for all the y_i according to the above procedure.

2.2. A Non-trivial Example. Let us first give an example. We will use this example in the following sections whenever it is needed.

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

We first assign the following coordinates to the five points: $A = (0,0), B = (r,0), C = (x_2,y_2), D = (x_3,y_3), E = (x_4,y_4)$. 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 to determine the six variables $x_2, y_2, x_3, y_3, x_4, y_4$. The following are these constraints and their corresponding algebraic equations.

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

Using Rit–Wu'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 real zeros, 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 1 and 2).

2.3. Coordinate Transformation. In this section, we will discuss the problem of how to draw the constrainted diagram on a window system after a triangular decomposition like (2.1) has been given.

To improve the efficiency of Ritt-Wu's decomposition 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 (i.e., no redundancies) constraints to determine the shape of the corresponding 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 Ritt-Wu's decomposition algorithm, we always assume that the coordinates of P_0 are $(0,0)$ and the coordinates of P_1 are $(r,0)$ where r is a variable representing the length of P_0P_1. 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. Before we can do that, let us consider three basic transformations for a well-defined diagram. For a well-defined constraint system, let P_0 be the point with free position and P_0P_1 the line with free direction. Then we usually can perform the following transformations.

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.

The coordinate transformations for the three transformations are well known. We will show that in our case what modifications can be made. For case (1), let the original and the new coordinates of point P_0 be (x_0, y_0) and (x_0', y_0'). The new coordinates for other points can be obtained with the following transformation

$$x' = x + (x_0' - x_0), y' = y + (x_0' - x_0). \tag{2.5}$$

For case (2), let the old coordinates of P_0 and P_1 be (x_0, y_0) and (x_1, y_1). Suppose, the new direction of P_0P_1 is determined by P_0Q where Q is a new point $Q = (x_2, y_2)$. The new coordinates for all points can be obtained with the following transformation

$$x' = x_0 + cos\theta * (x - x_0) + sin\theta * (y - y_0), y' = y_0 - sin\theta * (x - x_0) + cos\theta * (y - y_0) \tag{2.6}$$

where $\theta = arctan(\frac{(y_2-y_0)(x_1-x_0)-(y_1-y_0)(x_2-x_0)}{(x_1-x_0)(x_2-x_0)+(y_2-y_0)(y_1-y_0)})$ is between $-\pi/2$ and $\pi/2$. To avoid the computation of trigonometric functions, the new position of point P_1 can be obtained as the intersection of the line P_0Q and the circle with center P_0 and radius P_0P_1. Of the two intersections, we choose the one which is near P_1. For case (3), since the parameters are in the equations explicitly, the solution is straightforward.

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 transformations (2.5) and (2.6) 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 involving coordinates or the number of points in the diagram. In the first step, we need to solve a fixed number of univariate polynomial equations. We can use Newton's iteration method to do that [7]. 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. More discussions about numerical methods of solving constraint systems can be found in [16, 19].

Comparing to the previous approaches, our approach has clear advantages. First, 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 so-called "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 define a constraint system to be *well-defined* if the shape of the corresponding diagram has only a finite number of cases. We will show later that a well-defined constraint system of n points may have more than $2*n-3$ constraints.

Using algebraic languages, we can give a criteria for well-defineness. 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 the constraint system with n points is well-defined iff in a decomposition like (2.1) each triangular set TS contains $2n-4$ 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 [25, 3].

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

By Theorem 2.2, the pentagon problem in Section 2.2 is well-defined. But the seven independent constraints do not properly define a pentagon. From the above table, we see that only TS_1 defines a pentagon (Figure 1a). The solutions for TS_2 and TS_3 are not pentagons (Figures 1b and 1c). To define a proper pentagon, we need to add at least another constraint

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

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. (1) 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. Now we need only to consider the two solutions given by the quadratic equation about x_2. The two solutions are two regular pentagons, one with r as an edge another with r as a diagonal (Figure 2).

In our case, this problem can be solved due to the way of solving equations. As we mentioned in the preceding section, for each set of segment length and angle values, we only solve a fixed number of non-linear univariate equations once and the coordinates for other new positions are given by linear transformations.

Another method is to reduce the branches of solutions by introducing more constraints. For the pentagon example, we can assume that $ABCDE$ is a concave pentagon by introducing a condition $\theta_1 = (|AD|^2 > |AB|^2)$ which means that the diagonal AD is longer than the edge AB. When a set of solutions are found, we check if θ_1 is true. If it is not true, we discard it. 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 values. If they are valid, we have found a set of 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 and to select one satisfying θ_1 and θ_2.

(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, 24]. 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[4] if in any subsystem of n points, the number of constraints $e \leq 2n - 3$. This and other results reported in [21], though important, have the following 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 or regular polygons let along constraint systems using the implicit constraints like "two segment are equal" and "two angles are equal". 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 is provided by the quantifier elimination algorithm [6].

Theorem 2.3 ([25]) *A set of polynomial equations PS does not have solutions over the field of complex numbers iff $k = 0$ in decomposition[5] (2.1).*

If $k = 0$ in decomposition (2.1), the constraint system is also conflict over the field of real numbers.

[4] 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.

[5] Here we assume that the triangular sets are irreducible. For the definition of irreducible triangular sets, see [23, 25].

In the pentagon example, 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 [4], both the Groebner basis and Ritt-Wu'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 Ritt-Wu's decomposition algorithm, we have

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

where TS_5 and TS_6 are polynomial sets in triangular form and I_5 and I_6 are the product 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

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

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 [25, 3]. We may use these efficient algorithms to prove whether (2.8) is true.

Let us use the pentagon problem to show how Ritt-Wu's method is used to check the independency. Applying Theorem 2.4 to (2.4), we know that the seven constrains 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. Let us check whether a new constraint $|BD| = |BE|$ (with equation $h_8 = y_3^2 + x_3^2 - 2rx_3 - y_2^2 - x_2^2 + 2rx_2$) is independent to the eight constraints in (2.7). Equivalently, we want to know whether the new constraint is valid on $Zero(\{h_1, ..., h_7\}) = Zero(TS_1/I_1) \cup Zero(TS_4/I_4)$. This can be done by computing the remainder of the pseudo division of polynomial h_8 with respect to TS_1 [23, 25]. This remainder is zero iff $h_8 = 0$ is valid on $Zero(TS_1/I_1)$. Following this procedure, we can show that $h_8 = 0$ is valid on $Zero(TS_1/I_1)$ but not on $Zero(TS_4/I_4)$. Since TS_4 does not has solutions in the field of real zeros, $h_8 = 0$ is not independent to $h_1 = 0, ..., h_7 = 0$ over the field of real number.

3. A Decision Procedure for Rc-constructibility

3..1. Basic Results on Rc-constructions. Let $\mathbf{Q}$ be the field of rational numbers. Using the symbolic approach mentioned before, solving a geometric constraint system can be reduced to solving a set of algebraic equations. We thus allow us to abuse terms by saying that the numerical solutions of the equations 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 $x_1, ..., x_{i-1}$. We assume that x_{n-1} occurs in Q_n. Otherwise, η_{n-1} is not needed to define η. Since Q_n is linear in x_{n-1}, from $Q_n(\eta_1, \eta_2, ..., \eta_n) = 0$, we have

$$\eta_{n-1} = \frac{\eta_n^2 + e_{n-1}(\eta_1, ..., \eta_{n-2})\eta_n + f_{n-1}(\eta_1, ..., \eta_{n-2})}{m_{n-1}(\eta_1, ..., \eta_{n-2})\eta_n + n_{n-1}(\eta_1, ..., \eta_{n-2})}.$$

Substituting η_{n-1} into Q_{n-1} and taking the numerator, we have

$$\eta_n^4 + g_3\eta_n^3 + g_2\eta_n^2 + g_1\eta_1 + g_0 = 0$$

where g_i are polynomials of $\eta_1, ..., \eta_{n-2}$. Since Q_i is of degree two in η_i, we can assume that g_i are linear in $\eta_1, ..., \eta_{n-2}$. Then

$$\eta_{n-2} = \frac{\eta_n^4 + P_{n-2}(\eta_1, ..., \eta_{n-3}, \eta_n)}{Q_{n-2}(\eta_1, ..., \eta_{n-3}, \eta_n)}$$

where P_{n-2} and Q_{n-2} are polynomials with degree three in η_n. Repeat this process, 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* $k > 0$. Therefore, if the roots of an irreducible polynomial equation $P(x) = 0$ $(P(x)) \in \mathbf{Q}[x])$ can be constructed by ruler and compass, its degree must be a power of two. 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 us first consider a special case: we will give an efficient method of deciding whether the roots of a quartic equation are rc-constructible. 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 $\mathbf{Q}(u_1, ..., u_m)$. If $K = \mathbf{Q}(u_1, ..., u_m)$, we say that the roots of the equation (3.3) are *rc-constructible* if for all possible rational numbers $r_1, ..., r_m$ the roots of the quartic equation obtained by substituting r_i for u_i are rc-constructible.

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

$$x_1^2 + bx_1 + c = 0$$
$$y^2 + (-mx_1 + f)y - nx_1 + g = 0$$

The first equation can be simplified 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 quartic 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

$$8g^3 - 4h_2g^2 + (2h_1h_3 - 8h_0)g - h_0h_3^2 + 4h_0h_2 - h_1^2 = 0 \tag{3.4}$$

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

$$y^2 + f_1y + g_1 \pm (m_1y + n_1) = 0.$$

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.

Since $2mn = 2fg - h_1$ is a constant for fixed values of f and g, for each nonzero value of m there is a unique value for n. Furthermore, if m changes signs, n also changes its sign. From this observation, one of the two quadratic equations, say $(ax^2+2bx+c+2ao)-(2mx+n)=0$ is enough to obtain all the roots of equation (3.3) if we take the positive and negative square roots for m or n. Here is a summary of what we have proved.

Theorem 3.1 *(1) Use the notations given above. The roots of equation (3.3) are rc-constructible iff the cubic polynomial in (3.4) has a linear factor over the field K.*

(2) Let $g - g_0$ be a linear factor of the polynomial in (3.4). Then the four roots of equation (3.3) are given by the following triangular set of linear and quadratic equations

$$\begin{aligned}
&y^2 + fy + g - (my + n) = 0\\
&2mn - 2fg + h_1 = 0\\
&m^2 - 2g - f^2 + h_2 = 0 \ \textit{(or } n^2 - g^2 + h_0 = 0 \textit{ if } m = 0\textit{)}\\
&2f - h_3 = 0\\
&g - g_0 = 0
\end{aligned}$$

Example 3.2 (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 $O = (0,0), A = (w,0), P = (u,v), B = (w,s), X = (x_1,y_1), Y = (x_2,0)$. The algebraic equations of the constraints are

$$\begin{aligned}
&f_1 = wy_1 - sx_1 = 0 && X, O, B \text{ are collinear}\\
&f_2 = (y_1 - v)x_2 - uy_1 + vx_1 = 0 && X, P, Y \text{ are collinear}\\
&f_3 = x_2^2 - 2x_1x_2 + y_1^2 + x_1^2 - d_1^2 = 0 && |XY| = d_1
\end{aligned}$$

If considering w, u, v, and s as non-zero constants, with Ritt-Wu's characteristic method, the above equations can be reduced to the following triangular form.

$$\begin{aligned}
&f_2 = (y_1 - v)x_2 - uy_1 + vx_1 = 0\\
&f_1 = wy_1 - sx_1 = 0\\
&f_4 = s^2(s^2+w^2)x_1^4 - 2ws^2(vs+wu)x_1^3 + (v^2+u^2-d_1^2)w^2s^2x_1^2 + 2d_1^2w^3vsx_1 - d_1^2w^4v^2 = 0
\end{aligned}$$

With Theorem 3.1, we can calculate a cubic equation from the quartic equation $f_4 = 0$. Using factorization algorithms from [22], we can show that this cubic polynomial is irreducible. This means that the diagram cannot be drawn with ruler and compass.

Example 3.3 (Pappus' Problem II) *All requirements are the same as in Example 3.2 with an additional condition that point P is on the bisector of angle AOB.*

We need only to add a new constraint $\angle AOP = \angle POB$ (with equation $d_3(v^2 - u^2) + 2d_2uv = 0$) to the constraint set of Example 3.2. Using Ritt-Wu'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

$$-12(u^2+v^2)^2g + 2(u^2-v^2)(u^4-v^4) + (u^2-v^2)^2d_1^2 = 0.$$

Then the quartic equation can be reduced to two quadratic equations and two linear equations as shown in Theorem 3.1 and the line can be drawn with ruler and compass.

Finally, let us comment that the above approach for quartic equations is similar to the well-known techniques of solving quartic equations. What we demonstrate here is that how this method can be obtained naturally in the context of solving rc-constructibility problems. Also, in the next section, we will use the same idea to solve the rc-constructibility problem in the general case.

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

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

in $K[x]$ are rc-constructible. Let us first assume that the polynomial in (3.5) is irreducible which can be checked with the algorithm in [22].

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. However the algorithm has exponential time complexity.

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

$$x_1 = \frac{y^{2^{n-1}} + P(y)}{Q(y)} \text{ and } x_1^2 - c = 0$$

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 always 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 about p_i and q_i. It is clear that the solutions of PS are in a one to one corresponding with the following factorization of H

$$H(x) = (y^{2^{n-1}} + P(y) - R(y)) * (y^{2^{n-1}} + P(y) + R(y)) \tag{3.7}$$

Lemma 3.4 *Use the notations introduced above. The equation set PS has a finite number of solutions over any extension filed of K.*

Proof. Suppose that $H(x)$ has a factorization like

$$(3.8) \qquad H(x) = (y^{2^{n-1}} + s_{2^{n-1}-1}x^{2^{n-1}-1} + ... + s_0)(y^{2^{n-1}} + t_{2^{n-1}-1}x^{2^{n-1}-1} + ... + t_0).$$

Then by comparing the coefficients of equations (3.7) and (3.8), we have $p_i - q_i = t_i, p_i + q_i = s_i$ or $p_i - q_i = s_i, p_i + q_i = t_i$. In either case, we can determine p_i and q_i uniquely: $p_i = (s_i + t_i)/2, q_i = (s_i - t_i)/2$ or $p_i = (s_i + t_i)/2, q_i = (t_i - s_i)/2$. Since polynomial H can have only a finite number of factorizations like (3.8), there exist also a finite decompositions like (3.7). Since each set of solution of PS will provide a decomposition (3.7), PS has only a finite number of solutions over any extension field of K. ▮

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 in triangular form 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. We have reached the following result.

Lemma 3.5 *Use the notations introduced above. If $q_{k-1} \neq 0$, the solutions of the equation set PS can be obtained as follows. First we need to solve a polynomial equation PS_1 for the variables $p_0, ..., p_{k-1}$. Second $q_0, ..., q_{k-1}$ can be obtained with the triangular equations $R_1 = 0, ..., R_k = 0$. PS_1 also has a finite number of solutions.*

We need only to show that PS_1 has a finite number of solutions. To prove this, we need only to show that each set of solutions of PS_1 will lead to a set of solutions for PS. For a set of solutions of PS_1, if $q_{k-1} \neq 0$ then we can obtain a set of of solutions for the q_i via $R_i = 0$. If $q_{k-1} = 0$, let l be the largest index such that $q_{l-1} \neq 0$. Then we can obtain a similar set of equations $R_1, ..., R_l$ from which the rest of the q_i can be obtained.

Lemma 3.6 *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 set of solutions 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

$$\begin{aligned}
&y^k + e_{k-1}y^{k-1} + ... + e_0 - (q_{k-1}y^{k-1} + ... + q_0) = 0\\
&R_1 = q_{k-1}^2 + f_1(p_0, ..., p_{k-1}) = 0 \ (q_{k-1} \neq 0)\\
&R_2 = 2q_{k-1}q_{k-2} + f_2(e_0, ..., e_{k-1}) = 0\\
&R_3 = 2q_{k-1}q_{k-3} + f_2(e_0, ..., e_{k-1}, q_{k-2}) = 0\\
&\cdots\\
&R_k = 2q_{k-1}q_0 + f_k(e_0, ..., e_{k-1}, q_{k-2}, ..., q_1) = 0.
\end{aligned}$$

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. The proof is similar to that of Theorem 3.1. For the first part, we already proved that each set of solutions of PS_1 can be extended to a set of solutions for PS. This means that each solutions of PS_1 will lead to a decomposition (3.6). Noting that in decomposition (3.6) $P(y)$ and $Q(y)$ are polynomials in $K[y]$. Then decomposition (3.6) exists iff PS_1 has a set of solutions in K.

For the second part, similar to the proof of Lemma 3.5, if $q_{k-1} \neq 0$ we can eliminate the variables $q_{i-2}, q_{i-3}, ..., q_1$ from R_i for $i \geq 2$.

$$\begin{aligned}
&R'_2 = 2q_{k-1}q_{k-2} + g_2(p_0, ..., p_{k-1}) = 0\\
&R'_3 = 2q_{k-1}q_{k-3} + g_2(p_0, ..., p_{k-1}) = 0\\
&\cdots\\
&R'_k = 2q_{k-1}q_0 + g_k(p_0, ..., p_{k-1}) = 0.
\end{aligned}$$

Therefore, for any values of $p_0, ..., p_{k-1}$, $q_{k-1}q_i$ is a constant. When q_{k-1} changes signs, q_i $(i = 0, ..., k-2)$ will also change signs. Since $R_1 = 0$ will give raise to a positive and a negative values for q_{k-1}, choosing one of the two equations $y^{2^{n-1}} + P(y) - R(y) = 0$ or $y^{2^{n-1}} + P(y) + R(y) = 0$ is sufficient to find all the solutions of (3.5). ∎

Before giving the algorithm, let us 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.

Algorithm 3.7 *The algorithm will decide whether a geometric constraint system is rc-constructible, and if it is, represents the coordinates of the points in the geometric problems as solutions of linear and quadratic equations.*

Step 1. Choose a coordinate system. Let PS be the algebraic equations of the geometric constraints.
Step 2. Using Ritt-Wu's decomposition algorithm, we can represent the solutions of PS as solutions of polynomial sets in triangular form.

$$Zero(PS) = \bigcup_i Zero(TS_i/I_i)$$

where I_i is the product of the initials of the polynomials in TS_i.
Step 3. Let $TS_i = \{p_1(U, x_1), p_2(U, x_1, x_2), ..., p_k(U, x_1, ..., x_k)\}$ where $U = u_1, ..., u_m$ are the segment length and angle value parameters. If all the p_i are linear and quadratic, then the diagram defined by TS_i is rc-constructible. Otherwise, by (2.3), $Zero(TS_i/I_i)$ can be represented in the following form[6)]

$$R_i(U, w) = 0, x_j = M_j(U, w)/N_j(U, w), (j = 1, ..., k)$$

where w is a new variable and R, M_i, N_i, are polynomials in $K[U, w]$.

[6)] A more efficient way is to allow quadratic equations of x_i and we need only to change those p_i with degree higher than two. Now, $Zero(TS_i/I_i)$ can be represented as $R_i(U, w) = 0, Q_j(U, w, x_j) = 0, (j = 1, ..., k)$ where Q_j is either a linear or a quadratic polynomial in x_j.

Step 4. Now it is clear that the constraint system is rc-constructible iff the solution of each $R_i(U, w)$ is rc-constructible. We assume that $R_i(U, w)$ is irreducible, since otherwise we can factor it with the efficient algorithms in [22]. If the degree of R_i in w is not a power of two, the roots of R_i are not rc-constructible. If it is, do Step 5.

Step 5. Let R_i be of degree 2^m. Using Lemma 3.6, 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. This is because, in the case of quartic equations, the rc-constructibility is reduced to the reducibility of a cubic equation. 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 of the degree of the polynomial equation. Nevertheless, this seems to be first decision procedure for the rc-constructibility problem. Also, in practice equations with degree greater than four are rarely encountered.

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, *Mechanical Geometry Theorem Proving*, D.Reidel Publishing Company, Dordrecht, Netherlands, 1988.

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

[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] S.D. Conte & C. de Boor, *Elementary Numerical Analysis*, Prentice-Hall, 1976.

[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] P. S. Wang, An Improved Multivariate Polynomial Factorization Algorithm, *Math. Computation*, 32, p.1215-1231.

[23] J. F. Ritt, *Differential algebra*, Amer. Math. Sco. Colloquium, (1950).

[24] W. Whiteley, The union of Matroids and the Rigidity of Framework, *Siam J. Disc. Math.*, vol 1, 237, 1988.

[25] 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.