Chapter 20

GEOMETRIC THEOREM PROVERS AND ALGEBRAIC EQUATIONS SOLVERS

Shang-Ching Chou, Xiao-Shan Gao, Zhuo-Jun Liu, Ding-Kang Wang and Dongming Wang

The methods of mechanizing mathematics are realized by means of computer software for solving scientific and engineering problems via symbolic and hybrid computation. This chapter provides a collection of geometric theorem provers and algebraic equations solvers that are pieces of mathematical software based mostly on Wu's method and were developed mainly by members of the extended Wu group. The early theorem provers, though efficient, were written in basic programming languages and on primitive computers. Now there exist more powerful and mature geometric theorem provers of which some have already been published as commercial software. On the other hand, building effective equations solvers is still at the experimental stage and remains for further research and development.

20.1 Introduction

The mechanization of mathematics consists of three layers: theory and algorithms, software, and applications. The value and significance of algorithmic developments are rendered visible by means of their successful applications. Moreover, computational problems from practical applications often serve as key sources for motivating further development and improvement on theory and algorithms. The role of software development lies in bridging algorithms and applications. Therefore, the design and implementation of powerful software tools are always a central issue to the development of mathematics mechanization. Only with effective implementation on computer, good algorithms can be put to use for solving practical problems from science and engineering.

When Wu proposed his method of automated geometry theorem proving (AGTP) in the late seventies, there was no computer available to him. The computation for the first few examples in his paper was carried out with pencil and paper, involving polynomials of hundred terms (Wu 1979). In the early eighties, Wu was able to use an HP personal computer and wrote his first geometric theorem prover in Fortran 77. This prover is capable of proving very difficult geometric theorems (including Feuerbach's and Morley's theorems) and was also used to discover several new ones. During 1983–1984, Chou developed a geometric theorem prover at the University of Texas based Wu's method (with modifications), first using the PASCAL language and then using Macsyma on Lisp Machines (Chou 1984, 1988). With this prover Chou was able to prove later about 600 non-trivial geometric theorems, which exerted a widespread influence in the field of automated reasoning.

Stimulated by the work and provers of Wu (1979, 1984) and Chou (1984), several students of Wu in Beijing (Gao 1987; Wang and Hu 1987; Wang and Gao 1987) and other groups of researchers in USA and Europe (Ko 1986; Kusche et al. 1989) implemented efficient geometric theorem provers in the middle eighties. These early provers were developed mainly for proving and discovering theorems in elementary geometries. A number of other software systems have been developed since then not only for automated theorem proving, discovering and formula derivation in elementary as well as differential geometries but also for solving systems of polynomial equations with applications to various problems in computer vision, solid modeling, linkage design, robotics, chemical computation, mechanics, theoretical physics and CAD as we have seen partially in the previous chapters.

A side remark: although computer algebra systems had already been available in the West, Wu and his students did not have access to such systems in the early and middle eighties. Each of them wrote his own Fortran programs, starting with representations and operations of big integers and multivariate polynomials. In some way they undertook the painful tasks of re-inventing the basic techniques for designing and implementing polynomial algebra systems.

In this chapter, we focus on the development of effective geometric theorem provers and polynomial equations solvers by researchers from the extended Wu's group. A short description for each software package or system is provided. Related work by other researchers is only mentioned briefly; the reader should consult the indicated references and sources for details.

As some of the provers for geometric theorems and the solvers for polynomial equations are based on the same algebraic method of Wu, such provers may also be used with minor modifications as solvers, and vice versa. Therefore, the classification between provers and solvers is made mainly according to their primary purposes.

20.2 Geometric theorem provers

Most of the early provers implemented only a simple version of Wu's method, that is based on the well-ordering principle and successive pseudoreduction. A complete version of Wu's method involves zero and variety decompositions, which provides an effective method for AGTP and a symbolic approach for solving systems of polynomial equations and hence have much wider scope of applications. The complete version of Wu's method for AGTP has been implemented in some of the later provers. A few special yet more powerful modules have also been designed for certain restricted classes of theorems. The provers described in this section are listed chronologically, roughly according to the periods when they were developed.

20.2.1 The China-Prover: A historical theorem prover for geometry

This historical China-Prover was written by Wu during 1980–1983 in Fortran 77 on some micro-computers HP9835A and HP1000 (Wu 1984). It is based on the method proposed by Wu (1978) himself; this method is called the simple version of Wu's method in Chap. 5. The prover consists of three parts which perform the three steps of the proving:

- translating the geometric relations into polynomial equations;
- transforming the hypothesis polynomial set into a triangular form using the well-ordering principle; and
- determining the successive pseudo-remainders of the conclusion polynomial with respect to the hypothesis polynomials in triangular form.

If the final remainder is zero then the theorem is valid under some non-degeneracy conditions. Otherwise, it is generally false.

This version of Wu's method, though quite simple, is powerful enough to prove many difficult theorems and to discover some new results. It is the first geometric theorem prover based on Wu's algebraic method, that was written by Wu to show the effectiveness of the method. The detailed implementation and extensive experimental study of Wu's method of AGTP are later carried out by his students and other researchers.

20.2.2 GEO: A Macsyma theorem prover at UT

The geometry theorem prover GEO developed by Chou (1984) at the University of Texas (UT) is based on Wu's method (Wu 1978, 1984). It was implemented first in PASCAL and then in MACSYMA on Lisp Machines during 1983-1985. This prover has been used to prove a large number of difficult geometric theorems and widely known in the area of AGTP. There are several distinctive features in the prover at that time.

- The input is purely geometric, i.e., the user does not have to select coordinates manually – this job is done by the prover automatically and optimally.

- It has graphics displays: the geometric input also generates the graphics diagram of the theorem.
- It decomposes a triangle form into irreducible components in the case of quadratic successive algebraic extension fields and the case of any degree over the base field **Q** (for theorems that have been run into this prover so far, there is no need of decomposition over algebraic extensions with degree > 2).
- For a class of constructive geometric theorems (i.e., geometric statements about the configurations that can be drawn by rulers and compasses), the output is also geometric, i.e., non-degeneracy conditions are given in geometric form such as "Point A, B, C are not collinear."

The prover was later used to prove over 600 theorems, of which 512 are reported in the book by Chou (1988). Some "new" theorems were also discovered with this prover (Chou 1988).

20.2.3 CPS/PS/DPMS: Early Fortran theorem provers at AC

These provers were written by Gao and D. Wang in Fortran 77 on an HP-1000 computer during 1985–1987 at Academia Sinica (AC). They are based on Wu's method with modifications for various reasons and were used to prove more than one hundred geometric theorems, of which 95 are included in Wang and Gao (1987).

The prover CPS implemented by D. Wang is based on Hilbert's mechanical method for pure intersection point theorems described in Wu (1982). In CPS, Hilbert's method is extended to cover theorems involving metric relations such as circles and angles (Wang 1989b). D. Wang also wrote a general theorem prover PS on the basis of Wu's well-ordering principle. The provers CPS and PS were used to prove a number of quite difficult theorems; several "new" theorems were also discovered by them (Wang 1989a).

DPMS (Differential Polynomial Manipulation System) was implemented by Gao (see Wang and Gao 1987) and was used to prove geometric theorems of constructive type, to prove identities of trigonometric functions and hyperbolic functions (Gao 1987, 1990), and to find geometric loci automatically. In DPMS, some techniques are used to transform reducible geometric theorems into irreducible ones, so that the prover is complete for this kind of constructive geometric theorems. DPMS also includes an implementation of Wu-Ritt's well ordering principle in differential case and Buchberger's Gröbner bases method. Gao tried to use the Gröbner basis program to prove the Jacobi conjecture for certain concrete cases. However, this program was quickly abandoned by Gao, because of the availability of more convenient programming languages such as Lisp and C/Xwindows.

20.2.4 GEOTHER: A theorem proving environment

GEOTHER (GEOmetry THeorem provER) is an environment implemented in Maple with drawing routines and interface written in C for manipulating and proving geometric theorems (Wang 1996). In GEOTHER a theorem is specified by means of predicates of the form `Theorem(H,C,X)` asserting that `H` implies `C`, where `H` and `C` are lists or sets of predicates that correspond to the geometric hypotheses and the conclusion of the theorem, and the optional `X` is a list of variables for the internal computation. The information contained in the specification may be all what is needed in order to manipulate and prove the theorem. From the specification GEOTHER can *automatically*

- assign coordinates to each point in some optimal manner;
- translate the predicate representation of the theorem into an English statement, into a first-order logical formula, or into algebraic expressions;
- draw one or several diagrams for the theorem;
- prove the theorem using any of the six algebraic provers;
- translate the generated algebraic non-degeneracy conditions into geometric/predicate form,
- generate a LaTeX and/or a PostScript file documenting the theorem and its proof.

The assignment of coordinates to points can be done optionally by the user. The algebraic provers are based on Wu's method (Wu 1984), the method of Gröbner bases (Buchberger 1985) and a method proposed in Wang (1995a,b). GEOTHER can run with an ATINF graphic interface using menu-driver and contains a collection of theorems in both elementary and differential geometries with sample specifications that have been proved. These proved theorems include those named after Steiner, Morley, Thébault, MacLane and Bertrand. The package also has functions for handling the collection of theorems and online help for all functions with examples. A new module for proving geometric theorems using Clifford algebra and rewrite rules is being added.

20.2.5 Euclid: A program for producing readable proofs

Euclid[1] is written by Chou, Gao and Zhang in Lisp on a Next workstation based on the methods of producing readable proofs (Chou et al. 1994). It uses

[1]The prover is available via ftp at emcity.cs.twsu.edu: pub/geometry/software/euc.tar.Z.

a polynomial factorization package written by P. S. Wang from Kent State University, USA.

Euclid implements the following methods of AGTP: *the area method* for Euclidean geometry, *the full-angle method* for Euclidean geometry , *the volume method* for solid geometry, *the vector method* for metric geometries, and *the argument method* for non-Euclidean geometries. Detailed discussion of these methods could be found in Chap. 7 or (Chou et al 1994a).

Most of the work related to the area method is originally developed and experimenced with this software. It is used to prove more than 400 theorems with the area method (Chou et al 1994a), 110 theorems with the full-angle method (Chou et al 1994b), 90 theorems with the argument method (Chou et al 1994c), and 135 theorems from the American Mathematical Monthly (Chou et al 1994d).

20.3 Algebraic equations solvers

20.3.1 SOLVER: A package based on the well-ordering principle

SOLVER was implemented by Wu on a Dual System 83/20 using Fortran77 (Wu 1986a, 1986b). It implements the well ordering principle **??**. As mentioned by Wu, the zero decomposition algorithm **??** could be carried out by repeatly using the program with input provided by the user. Wu proposed several improvements of his original algorithm to avoid the occurrence of large polynomials: (1) The concept of weak characteristic set (CS) is introduced. A CS is called weak if its initials are redcued with respect to CS. (2) The well ordering priciple is modified to avoid the appearance of too many polynomials. (3) Since factorization is expensive, (including implementation in Fortran), Wu used primitive polynomials instead of irreducible ones.

SOLVER aims at polynomial equations-solving and its diverse applications. Specificly, Wu (1986b) discussed how to use SOLVER to solve problems raised from eignevalue and eigen vectore calculation, theory of automatical control, design of mechanisms, mechanical derivation of geometric formulas, and automated geometry theorem proving.

20.3.2 DEC: An implementation of Wu-Ritt's decomposition method

DEC (decomposition) was implemented by Chou and Gao (1990a) in Lisp on both SUN workstations and Lisp machines. With DEC a detailed experimental study of techniques in the effective implementation of Wu-Ritt's zero decomposition theorem was carried out. These techniques are summarized in Chou and Gao (1992). DEC has the following modules:

- Wu-Ritt's zero decomposition method for polynomial systems and ordinary differential polynomial systems. The decomposition may be affected by the following parameters.
  - DEC-DIM: decide whether to use the dimension theorem.
  - DEC-VER: decide the type of ascending chains in the decomposition — weak form (Chou and Gao 1990a), Ritt's sense, Wu's sense or triangular form.
  - DEC-TRI: decide the type of elimination procedures — bottom-up triangulation, top-down triangulation, Gröbner bases reduction or multiple-p.r.s. (Li 1987).
  - DEC-FAC: decide when to perform factorization over $\mathbf{Q}$ (DEC uses a polynomial factorization package written by P. S. Wang).
  - DEC-AFAC: decide whether to find irreducible ascending chains (the factorization program was written by the authors of DEC).
  - DEC-PAR: this is a set of variables which are treated like parameters (i.e., the decomposition is performed over $\mathbf{Q}(\text{DEC-PAR})[x_1, \ldots, x_n]$).

- Quantifier elimination over algebraic closed fields. This part has two functions: projection of a quasi-algebraic set and quantifier elimination over an algebraic closed field. These functions are based on an improved version (Gao and Chou 1992) of an algorithm proposed by Wu (1990).

- Gröbner bases computation. The program can find the Gröbner basis of a polynomial set in $\mathbf{Q}(\text{DEC-PAR})[x_1, \ldots, x_n]$, where DEC-PAR is a set of variables treated like parameters.

DEC has been used to prove more than 400 geometry theorems with given non-degeneracy conditions (Chou and Gao 1990a), to derive geometric formulas for 120 problems (Chou and Gao 1989), and to prove about 100 theorems from differential geometry and mechanics (Chou and Gao 1991).

20.3.3 CharSets: A characteristic sets package in Maple

CharSets is a Maple package based on Wu-Ritt's method of characteristic sets (Ritt 1950; Wu 1984, 1989) implemented by Wang (1995). The first version CharSets 1.0 was made publicly available with Maple's share library in early 1991. Two subsequent versions 1.1 and 1.2 were also distributed. The current version CharSets 2.0 is available on the World Wide Web as:

http://www-leibniz.imag.fr/ATINF/Dongming.Wang/charsets-2.0.tar.Z

They have been used by a number of individuals for different purposes. Some of the routines have been adapted into other computer algebra systems including Singular and Risa/Asir.

The early versions of CharSets were implemented for computing characteristic sets and various zero decompositions of polynomial systems and other related polynomial calculations. Their capabilities include

- computing (modified) characteristic sets of polynomial sets (in different senses),
- decomposing polynomial sets into ascending sets and into irreducible and quasi-irreducible ascending sets,
- decomposing algebraic varieties into irreducible components,
- decomposing polynomial ideals into primary components,
- solving systems of polynomial equations, and
- factorizing multivariate polynomials over algebraic extension fields.

A new module has been added to CharSets version 2.0 that contains 8 functions for computing *differential* characteristic sets and the corresponding zero decompositions for systems of *ordinary differential* polynomials. See Wang (1997) for more details.

Some of the routines from the CharSets package are used by the provers in GEOTHER reviewed in Sect. 20.2.4. Several selected functions from CharSets and algebraic provers from GEOTHER were parallelized via workstation networks (Wang 1991).

20.3.4 WSOLVE: An implementation of Wu-Ritt's decomposition algorithm for solving polynomial systems

WSOLVE was implemented by D.-K. Wang in 1993 in the Maple system on both SUN workstations and PCs. WSOLVE has two main functions: *wsolve* and *e_val*. The former is an implementation of Wu-Ritt's zero decomposition algorithm, and its main aim is for solving systems of polynomial equations. The function *e_val* is an implementation of Wu's finite kernel theorem, that computes the extremal value of a polynomial under some polynomial constraints (see Chap. 1). It can be used to prove theorems involving inequalities.

WSOLVE uses the following techniques for improving the practical efficiency.

- Factorize every polynomial which is given in the initial set or produced in the process of computation.

- Use a parameter to decide which kinds of triangular form — ascending set, weak ascending set or triangular set — to compute.
- Check the redundant branches which are produced from the initials during the computations of pseudo-remainders.

The program WSOLVE has been used to solve problems involving polynomial equations and to proving trigonometric inequalities.

20.3.5 SACCS: A C library for the characteristic set method

The SACCS library was implemented by Zhi (see Wang and Zhi 1995; Zhi 1997) in the C language, making use of routines from SACLIB, a publicly available library for symbolic and algebraic computation. SACCS has functions for most of the computations (except for primary decomposition) via characteristic sets that are listed in Sect. 20.3.3 for ordinary polynomials. For polynomial factorization over algebraic extension fields, an algorithm suggested in Wang (1992; see also Chap. 14) was implemented in SACCS; an implementation of an algorithm proposed by Zhi (1997; Chap. 14) using modular techniques was not finished.

SACCS was developed as a first step towards a parallel implementation of characteristic sets algorithms with PARSAC, a parallel version of SACLIB. The parallel implementation has been done later by Iyad A. Ajwa from Ashland University, USA on the basis of the sequential SACCS.

Zhi also made a simplified version of the CharSets package in Maple which has been named CSET.

20.3.6 Li's CS implementation

20.4 New systems

The solvers and provers reviewed in Sects. 20.2 and 20.3 were implemented either in the classical programming languages like Fortran and Lisp, or on the top of popular computer algebra systems like Maple. Some of these solvers and provers are also quite preliminary and have limited capabilities. An attempt of the extended Wu group has been to develop more comprehensive and advanced software systems for Wu-type algorithms that are independent of the existing symbolic computation software and that take advantages of the new generation of computer hardware, operating systems and programming languages. The design and implementation of such systems may take into account the special features of the algorithms in question and various technical considerations in order to achieve ideal performance. The two systems described in this section are some of the results from our effort and trials.

20.4.1 Geometry Expert

A prototype of this software was first written in 1992 by Gao and Lin with C/Xwindow/Xview on a Sun workstation. The program consists of two parts: a prover and a graphic editor. The prover is basically a C transformation of that of DPMS (see Sect. 20.2.3). The editor is one for *dynamic geometry* that allows free dragging for the theorem prover. From 1993 to 1996, Gao added many new functions based on new AGTP methods introduced by Chou et al. (1994) and developed an Xwindow version of Geometry Expert (GEX)[2] (Chou et al. 1994). Several new features for dynamic diagram construction were also added. In 1997 Gao re-wrote the program with C++ on Window 95 platform, and a Chinese version of GEX was then published as a commercial software product (Gao et al. 1998).

GEX is a software tool for *dynamic geometric drawing* and *automated geometric theorem proving and discovering*. The *dynamic geometric editor* mainly has the following functions: (1) geometric input of geometric diagrams and statements for the prover, (2) dynamic geometric transformations and measurements, (3) animation and loci generation. With the loci generation function, we may generate diagrams for "elementary functions". We may also generate loci with linkages. As proved by Kempe (1876), the class of loci includes all those defined by algebraic curves of the form $f(x, y) = 0$, where $f(x, y)$ is a polynomial of x and y.

The dynamic geometry part of GEX was be used to CAI (computer aided instruction; see Gao et al. 1998a), geometry research, mechanics design (Gao et al. 1998b), and computer vision (Gao and Cheng 1998).

GEX is also an efficient computer program for geometric reasoning. Within its domain, it invites comparison with the best of human geometry provers. It implements the six AGTP methods: Wu's method, the area method, the deductive database method, Gröbner bases method, the vector method, the method based on full-angles. This part is basically a C transformation of the AGTP algorithms in Euclid (Sect. 20.2.5) and DPMS (Sect. 20.2.3). The prover of GEX had been used by many researchers to deal with various kinds of geometry problems.

20.4.2 ELIMINO: A realization for Wu's method

ELIMINO is a Latin word meaning *eliminate*. It has been taken as the name of an ongoing software project (Lin 1998) whose objective is to implement a software system that has its own packages for long integer computation, polynomial manipulation, and Wu's method of polynomial equations solving

[2] The binary codes for GEX can be obtained via ftp at emcity.cs.twsu.edu: /pub/geometry/software/(ge_sparc.tar.Z, ge_linux.tar.Z).

and geometric theorem proving; so ELIMINO also stands for the system. As a project, ELIMINO may be considered as a continuation of the previous work described in Ju (1996, 1997) and Liu (1996, 1999).

As a symbolic computation system, ELIMINO has very general capabilities for polynomial operations and computation of characteristic sets in Wu's method. It also has an interactive programming environment which allows the user to add new facilities.

From the object-oriented point of view, a system can be considered as a collection of objects. In the system ELIMINO, every object has an associated datatype, which can be thought as a property of the object. The datatype of an object determines the operations that are applicable to the object and clearly separates the various kinds of objects in the system. The following are some of implemented features of ELIMINO.

- Numbers are the most fundamental objects to be manipulated in the implementation of ELIMINO. ELIMINO provides several different kinds of numbers such as integers, fractions and floating numbers. When appropriate, they can be combined in the same computation because the system knows how to convert among these numbers automatically.

  Integers can be as large as desired, under the only limitation of total storage available to the system. Fractions are quotients of integers. Cancelation between numerators and denominators occurs automatically.

  For approximations, floating point calculations can be performed with any desired number of digits. The function `precision` or `PREC` can be used to set the number to use.

- Polynomial is another fundamental object treated by ELIMINO. It plays a central role in the system. Some basic operators for polynomial computation were implemented include arithmetic computation, GCD and factorization for the purpose of implementing Wu's method.

- As a software tool for mathematical research, ELIMINO provides an embedded C-like programming language. The language allows the user to program his/her ideas and algorithms using procedures and then run the procedures to verify the ideas or solve the problems.

- Functions can be as important as the values on which they act. In ELIMINO, there are two kinds of functions: system internal functions and user defined functions. The former are implemented in the kernel part, and they provide the common and fundamental facilities to the user. The user defined functions are those defined by the user himself/herself using the programming language and internal functions. In fact, they are part of the User API.

- Implementation of Wu's method is one of the primary goals for this system. Although it can be realized as a group of user defined functions, we implemented it in the kernel part of the system.

20.5 Related packages and systems

The software packages and systems presented in the previous sections were developed mainly by members of Wu's group at large. There are a number of relevant systems and packages developed by other researchers based on Wu's method and other similar methods. We are unable to give a comprehensive and fair survey on all such systems and packages. The information and pointers in this section are provided informally and to help the reader locate the desired literature and sources.

20.5.1 Provers and systems for geometry

First of all the reader is advised to look at Hong et al. (1995) for another collection of early geometric theorem provers.

Different pieces of software have been developed by L. Yang, J.-Z. Zhang and their collaborators over the years. See Chaps. 10, 3 and 7 and other relevant publications by these authors.

There are several methods for quantifier elimination over real closed fields including the general method of cylindrical algebraic decomposition introduced by G. E. Collins and improved by H. Hong (from North Carolina State University, USA) and others, and some specialized methods proposed by V. Weispfenning (from Universität Passau, Germany). These methods have been used for geometric theorem proving. The reader may contact these authors or consult the relevant publications by them and their co-workers for more information.

A few systems that have been developed for doing geometry (research and education) in general and for theorem proving in some special cases include the well-known

- Cabri-Géomètre (http://www-cabri.imag.fr/) initiated by J.-M. Laborde (from Universit Joseph Fourier de Grenoble and CNRS, France),

- Geometer's Sketchpad (http://www.keypress.com/sketchpad/) developed by N. Jackiw (from Key Curriculum Press, USA), the new systems

- Cinderella (http://www.cinderella.de/) developed by J. Richter-Gebert and U. Kortenkamp (from ETH Zurich, Switzerland), Dr. Geo (http://members.xoom.com/FeYiLai/dr_geo/doctor_geo.html) and Geometry Inventor.

20.5.2 Other implementations of Wu-Ritt's decomposition method

An implementation of Wu-Ritt's zero decomposition method in Lisp was made by Ko (1988) for the purpose of proving geometric theorems. Kapur and Wan (1990) implemented a geometry theorem prover based on a refutation approach using a modified version of Wu-Ritt's zero decomposition method. Mcphee (1994) implemented a prover for geometry theorems involving inequalities based on a modified version of Wu-Ritt's zero decomposition method and Hong's implementation of Collins' CAD method.

Part of the CharSets package was re-implemented in Risa/Asir by T. Shimoyama (from Fujitsu Laboratories Ltd., Japan) for the purpose of primary decomposition and in Singular (http://www.mathematik.uni-kl.de/~zca/Singular/) by M. W. Messollen (from Universität des Saarlandes, Germany).

A recent implementation of Wu' decomposition method in AXIOM/ALDOR has been done by Aubry and Moreno Maza (1999).

20.5.3 Implementations of other triangular sets methods

Besides Wu's method there are three known methods based on triangular sets proposed by Lazard (1991), Kalkbrenber (1993) and Wang (1993). The two of 1993 have been implemented and experimented by their proposers (see also Wang 1996). Recently, a comparative implementation of the four methods including Wu's in AXIOM/ALDOR has been made by Aubry and Moreno Maza (1999) for solving polynomial systems.

Some methods presented in Yang et al. (1995) based on resultants and in Wang (1998, 1999) based subresultant regular subchains have also been implemented by the authors themselves.

20.5.4 Gröbner bases packages

The method of Gröbner bases (Buchberger 1985) is one of the most relevant to Wu's method. This method may address many of the problems, in particular geometric theorem proving and algebraic equations solving, to which Wu's method can be applied. The Gröbner bases method has been implemented in most of the popular computer algebra systems including Mathematica, Maple, Macsyma and Axiom. There are also special-purpose implementations, for which we may mention the one in CoCoA (http://cocoa.dima.unige.it/) and the GB package developed by J.-C. Faugère (from Université Paris VI, France). The two mentioned have been considered among some of the best implementations for Gröbner bases. The software GB has been combined with a RealSolving package by F. Rouillier (from INRIA Lorraine, France) for wider applications.

Aubry, P., Moreno Maza, M. (1999): Triangular sets for solving polynomial systems: A comparative implementation of four methods. J. Symb. Comput. (to appear).

Chou S. C. (1984): Proving elementary geometry theorems using Wu's algorithm, *Automated Theorem Proving: After 25 years*, American Mathematics Society, Contemporary Mathematics, 29(1984), 243–286.

Chou S. C. (1988): *Mechanical geometry theorem proving*, D.Reidel Publishing Company, Dordrecht, Netherlands, 1988.

Chou S. C. and Gao X. S. (1989): A collection of 120 computer solved geometry problems in mechanical formula derivation, TR-89-22, Computer Sciences Department, The University of Texas at Austin, August 1989.

Chou S. C. and Gao X. S. (1990a): Ritt-Wu's decomposition algorithm and geometry theorem proving, *CADE'10*, M.E. Stickel (Ed.) pp 207–220, Lect. Notes in Comp. Sci., No. 449, Springer-Verlag, 1990.

Chou S. C. and Gao X. S. (1990c): DEC: A common lisp algebraic system, preprint, The Computer Science Department, The University of Texas at Austin.

Chou S. C. and Gao X. S. (1991) Geometry theorems proved mechanically using Wu's method, part on differential geometry, MM-Research Preprints, No.6, 1991, Inst. of Systems Science, Academia Sinica.

Chou S. C. and Gao X. S. (1992): An algebraic system based on the characteristic method, *Proc. of International Workshop on Mathematics Mechanization*, Beijing, International Academic Publishers, p.1–17, 1992.

Chou S. C., Gao X. S., and Zhang J. Z. (1994a): *Machine proofs in geometry*, World Scientific, Singapore, 1994.

Chou S. C., Gao X. S., and Zhang J. Z. (1994b): A collection of 110 geometry theorems and their machine produced proofs using full-angles, TR-94-4, CS Dept., WSU, March, 1994.

Chou S. C., Gao X. S., and Zhang J. Z. (1994c): Automated solution of 135 geometry problems from American Mathematical Monthly, TR-94-8, CS Dept., WSU, August, 1994.

Chou S. C., Gao X. S., Zhang J. Z. and L. Yang, (1994d): A collection of 90 mechanically solved geometry problems from Non-Euclidean geometries, TR-94-10, CS Dept., WSU, September, 1994.

Chou S. C., Gao X. S., and Zhang J. Z. (1996a): An Introduction to Geometry Expert, *Proc. CADE-13*, p. 235-239, 1996, L.N.A.I. No. 1104, (eds. M. A. McRobbie and J. K. Slaney), Springer-Verlag.

Gao X. S. (1987): Trigonometric identities and mechanical theorem proving in elementary geometries, *J. Sys. Sci. and Math. Scis.*, No.3.

Gao X. S. (1990): Transcendental functions and mechanical theorem proving in elementary geometries, *Journal of Automated Reasoning*, **6**, 403-417.

Gao X. S. and Cheng, H. F. (1998) On the solution classification of the "P3P" problem, *Proceedings of the Third Asian Symposium on Computer Mathematics*, eds Z. B. Li, pp. 185-200, LanZhou University Press, 1998.

Gao X. S. , Zhu C. C., and Huang Y. (1998a): Building dynamic mathematical models with Geometry Expert, I. geometric transformations, functions and plane curves, *Proceedings of the Third Asian Technology Conference in Mathematics*, eds W.C. Yang, pp. 216-224, Springer, 1998.

Gao X. S. , C. C. Zhu, and Y. Huang (1998b): Building dynamic mathematical models with Geometry Expert, II. linkages, *Proceedings of the Third Asian Symposium on Computer Mathematics*, eds Z. B. Li, pp. 15-22, Springer, 1998.

Gao X. S. , Zhang J. Z., and Chou S. C. (1998): *Geometry Expert*, Nine Chapters, Taiwan, (in Chinese).

Hong, H., Wang, D., Winkler, F. (eds. 1995): Short description of existing provers. Ann. Math. Artif. Intell. 13: 195-202.

Ju Y., Liu J. and Liu Z.J. (1997): On the interactive & programming environment for STAR, MM-Research Preprints, No. 15, 1997, Inst. of Systems Science, Academia Sinica.

Lin, D.-D. Liu, J. and Liu, Z.-J. (1998) Mathematical research software: ELIMINO, *Proceedings of the 3rd ASCM*, 107–114, Aug. 6-8, 1998. Lanzhou, China.

Liu J. and Liu Z.J. (1999) Upgrading & using lex and yacc to develop message-driven GUI Syntax Analyzing Software, *J. of Computer Engineering and Design*, Vol. 20, No. 1, 1999, pp. 17–23.

Liu Z-J. and Wu W.-D. (1996) STAR: small tool for algebraic research Proceedings of the 2rd ASCM, 121-129, Aug. 20-22, 1996. Kobe, Japan.

Kapur, D. and Wan, H. K. (1990): Refutational proofs of geometry theorems via characteristic sets, *Proc. of ISSAC-90*, pp.277-284, Tokyo, Japan.

Kempe, A. B. (1876): On a general method of describing plane curves of the nth degree by linkwork, *Proc. of L.M.S.*, 213–216.

Ko, H.-P. (1986): ALGE-prover II: A new edition of ALGE-prover. Tech. Rep. 86CRD081, General Electric Co., Schenectady, USA.

Ko H. P. (1988): Geometry Theorem Proving by Decomposition of Quasi-Algebraic Sets: An Application of the Ritt-Wu Principle, *Artificial Intelligence*, 37 95–120.

Kusche, K., Kutzler, B., Stifter, S. (1989): Implementation of a geometry theorem proving package in Scratchpad II. In: Springer's LNCS 387, pp.

246-257.

Li Z. M. (1987): On the triangulatin for any finite polynomial set, MM-Research Preprints, No.2, pp. 48-54, Inst. of Systems Science, Academia Sinica.

Mcphee N., Chou S. C. and Gao X. S. (1994): A combination of Ritt-Wu's method and Collins' method, *Proc. of CADE-12*, pp.401–415, France.

Wang, D.-K. (1993): *Polynomial Equations Solving and Mechanical Geometric Theorem Proving*, PhD Thesis, Institute of Systems Science, Academia Sinica.

Wang, D.-K., Zhi, L. H. (1995) Software development in MMRC. Proceedings of ATCM95, Singapore, Dec. 18-21, 1995.

Wang D. (1989): A new theorem discovered by computer prover, *J. of Geometry* **36**, 173–182.

Wang, D. and Gao, X. S. (1987): Geometry theorems proved mechanically using Wu's method, part on elementary geometry, MM-Research Preprints, No.2, pp. 75-10,6 Inst. of Systems Science, Academia Sinica.

Wu W. T. (1978): On the decision problem and the mechanization of theorem-proving in elementary geometry, *Scientia Sinica*, **21**, 159–172; re-published in *Automated Theorem Proving: after 25 Years*, (Ed. W. W. Bledsoe and D. W. Loveland), 1984, 235–242.

Wu W. T. (1982); Toward mechanization of geometry — Some comments on Hilbert's "Grundlagen der Geometrie", *Acta Math.Scientia*, **2**, 125–138.

Wu W. T. (1986a); A general description of the SOLVER package system, (in Chinese), *Math. in Practice and Theory*, No.2, 32–39.

Wu W. T. (1986b); Examples of applications of the SOLVER package system, (in Chinese), Math. in Practice and Theory, No.3, 1–11.

Wu W. T. (1984): Some recent advance in mechanical theorem-proving of geometries, in *Automated Theorem Proving: after 25 Years*, (Ed. W. W. Bledsoe and D. W. Loveland), 235–242.

Wu W. T. (1990): On a projection theorem of quasi-varieties in elimination theory, *Chin. Ann. of Math.*, 11B, 220–226.

L. Yang, Gao X. S. , S. C. Chou, and Z. J. Zhang, (1998): Automated proving and discovering of theorems in non-Euclidean geometries, in: *Automated Deduction in Geometry* (D. Wang, ed.), LNAI **1360**, Springer-Verlag, Berlin Heidelberg, 1998, pp. 171–188.