The Fourth International Workshop on

Automated Deduction in Geometry

Abstracts Booklet

September 04-06, 2002

Research Institute for Symbolic Computation
Castle of Hagenberg, Linz, Austria

Program for the Workshop

Wednesday, September 04

08:00	Bus leaves from IBIS Hotel
09:00 - 09:15	**Opening of the Workshop**
09:15 - 10:15	D. Scott Introducing more abstract algebraic proofs in projective geometry
10:15 - 10:45	**Coffee Break**
10:45 - 11:15	L. Yang Distance coordinates used in geometric constraint solving
11:15 - 11:45	G. Bodnár Testing the normal crossing property of hypersurfaces
11:45 - 12:15	J. C. Owen, S. C. Power The nonsolvability by radicals of generic 3-connected planar graphs
12:15 - 14:00	**Lunch Break**
14:00 - 14:30	Z. Zeng, H. Fu Subresultants and discriminant sequences
14:30 - 15:00	I. J. Tchoupaeva Analysis of geometrical theorems in coordinate-free form by using anticommutative Gröbner bases method
15:00 - 15:30	H. Li Algebraic representation, expansion and simplification in automated geometric theorem proving
15:30 - 16:00	**Coffee Break**
16:00 - 16:30	X. S. Gao, Q. Lin MMP/Geometer - A software package for Automated Geometric Reasoning - A progress report
16:30 - 17:00	H. G. Gräbe The SymbolicData Proof Scheme Collection as a multiplatform project
17:00 - 17:30	J. Robu Geometry theorem proving in the frame of Theorema Project
18:00	Bus leaves for IBIS Hotel

Thursday, September 05

08:30	Bus leaves from IBIS Hotel
09:15 - 10:15	H. Pottmann Understanding and reconstructing three-dimensional shapes from point clouds
10:15 - 10:45	**Coffee Break**
10:45 - 11:15	Q. N. Tran Implicitization of geometric objects under affine transformations using Gröbner walks
11:15 - 11:45	S. Covez, F. Rouillier Using computer algebra tools to classify serial manipulators
11:45 - 12:15	X. Chen, D. Wang The projection of quasi varieties and it's application on geometry theorem proving
12:15 - 14:00	**Lunch Break**
14:00 - 15:00	J. Schicho Parametric Varieties
15:00 - 15:30	Y. Wu, H. Shi A special central configuration
15:30 - 16:00	**Coffee Break**
16:00 - 16:30	C. Jermann, B. Neveu, G. Trombettoni A new structural rigidity for geometric constraint systems
16:30 - 17:00	A. Sosnov, P. Macé Rapid algebraic resolution of 3D geometric contraints and control of their consistency
17:00 - 17:30	P. Conti, C. Traverso Deciding topological properties: Compactness of basic real semialgebraic sets
17:45 - 18:45	**Wine and Cheese Reception**
19:00 - 21:30	**Workshop Banquet**
21:30	Bus leaves for IBIS Hotel

Friday, September 06

08:30	`Bus leaves from IBIS Hotel`
09:15 - 10:15	B. Buchberger Automated proofs of automated geometry provers
10:15 - 10:45	**Coffee Break**
10:45 - 11:15	B. Jüttler, J. Schicho, M. Shalaby C^1 spline implicitization of planar curves
11:15 - 11:45	M. Peternell Rational parametrizations of the Minkowski sum of two quadrics in 3-space
11:45 - 12:15	A. Pasko, V. Adzhiev Function-based shape modeling using a specialized language
12:15 - 14:00	**Lunch Break**
14:00 - 14:30	M. C. Ko, Y. C. Choy Feature-preserving simplification of polygonal surface based on half-edge contraction manner
14:30 - 15:00	B. Jüttler The shape of spherical rational quartics
15:00 - 15:30	**Coffee Break**
15:30 - 16:00	G. Landsmann Implicitization of Algebraic Varieties
16:00 - 16:30	D. Wang GEOTHER 1.1: Handling and proving geometric theorems automatically
16:30 - 17:00	H. Crapo, W. Schmitt Straightening in the Whitney algebra of a matroid
17:15	`Bus leaves for IBIS Hotel`

Testing the normal crossing property of hypersurfaces

Gábor Bodnár
RISC, Johannes Kepler University, A-4040 Linz, Austria.
Email: `Gabor.Bodnar@risc.uni-linz.ac.at`

Given finitely many hypersurfaces, in some nonsingular ambient variety W. They are normal crossing if each of their irreducible components are nonsingular, and when r such irreducible components meet at a point $a \in W$, their local defining equations form a part of a regular system of parameters at a. We give a straightforward algorithm to test this property. The test relies on partial differentiation over W and Gröbner basis computation.

Introduction

The normal crossing property has a prominent role in the problem of resolution of singularities (see e.g. [2] for the foundations of the classical theory, or [1] for a more recent survey on the field). The definition of embedded resolution of singularities requires that a desingularization morphism $\pi : W' \to W$ of an embedded variety $X \subset W$ (where W, W' are nonsingular) has the property that the irreducible components of $\pi^{-1}(S)$ (also called as the set of exceptional divisors) are normal crossing, where $S \subset X$ is the set of singularities of X and π is an isomorphism outside S.

The normal crossing test itself is a straightforward application of the available computer algebra machinery on the problem. I am aware of the fact that the difficulty of the solution falls into the category of the textbook exercises; still, it might be worthwhile to be documented, since the normal crossing property is ubiquitous in resolution of singularities and in singularity theory, and the solution provides a first constructive approach to the problem.

The normal crossing test

Let k be an algebraically closed field of characteristic zero, let W be a nonsingular affine variety of dimension n with coordinate ring $k[W] = R/J$, where $R = k[x_1, \dots, x_m]$, $0 \neq J \subset R$. Let $\partial_{p_1}, \dots \partial_{p_n} \in \mathrm{Der}(W)$ be derivations on W, coming from a basis $dp_1, \dots, dp_n$ of the module differential forms $\Omega(W)$.

Let $X_1, \dots, X_s \subset W$ be defined by $f_1, \dots, f_s \in k[W]$ respectively. The set of singularities of the hypersurface $X_i \subset W$ is

$$\mathrm{Sing}(X_i) = \{a \in X_i \mid \partial_{p_j} f_i(a) = 0,\ j = 1, \dots, n\}.$$

Definition 1. The hypersurfaces $X_1, \dots, X_s \subset W$ are *normal crossing* if for all i: $\mathrm{Sing}(X_i) = \emptyset$ and for all $\{i_1, \dots, i_l\} \subseteq \{1, \dots, s\}$: if $a \in X_{i_1} \cap \cdots \cap X_{i_l}$, the elements of $\{f_{i_1}\mathcal{O}_{W,a}, \dots, f_{i_l}\mathcal{O}_{W,a}\}$ are linearly independent $\bmod \mathfrak{m}_a^2$.

The *Jacobian ideal* of $f_{i_1}, \dots, f_{i_l}$ is generated by all the minors of maximal size of the matrix

$$\mathrm{Jacobian}_W(f_{i_1}, \dots, f_{i_l}) = \begin{pmatrix} \partial_{p_1} f_{i_1} & \dots & \partial_{p_n} f_{i_1} \\ \vdots & & \vdots \\ \partial_{p_1} f_{i_l} & \dots & \partial_{p_n} f_{i_l} \end{pmatrix}.$$

Under the assumption $l \leq n$ and considering that all the $k[W]$ elements and ideals are given by representatives from R, if

$$\texttt{gbasis}(\langle f_{i_1}, \dots, f_{i_l} \rangle + \mathrm{Jacobian\_ideal}_W(f_{i_1}, \dots, f_{i_l}) + J,\ \texttt{tdeg}(x_1, \dots, x_m))$$

contains a nonzero constant, $X_{i_1}, \dots, X_{i_l}$ are normal crossing *within their intersection*, otherwise they are not (and the result is a Gröbner basis, defining the points where they fail to have the property). When $n < l$, it is enough to check whether the intersection of the hypersurfaces is empty, since the rank of $\mathrm{Jacobian}_W(f_{i_1}, \dots, f_{i_l})$ cannot reach l at any point of W.

Because the method checks the normal crossing property for a selected subset of hypersurfaces along their intersection, the complete algorithm has to go through all the nonempty subsets of $X_1, \dots, X_s$. A reasonable strategy is do the easier tests first, i.e. to check subsets by increasing cardinality.

```
Input: $W$ a nonsingular affine variety, $(f_1, \dots, f_s)$ a list of defining equations
    of hypersurfaces in $W$.
Output: a boolean indicating the normal crossing property of the input
    hypersurfaces in $W$.
isNormalCrossing($W$,$(f_1, \dots, f_s)$)
$n := \dim(W)$; $R/J := k[W]$; $O :=$ tdeg(Generators($R$));
for $l = 1$ to $s$ do
    for each $\{i_1, \dots, i_l\} \subseteq \{1, \dots, s\}$ do
        if $l \leq n$ then
            $G :=$ gbasis($\langle f_{i_1}, \dots, f_{i_l} \rangle + \mathrm{Jacobian\_ideal}_W(f_{i_1}, \dots, f_{i_l}) + J, O$);
        else   $G :=$ gbasis($\langle f_{i_1}, \dots, f_{i_l} \rangle + J, O$);
        if NormalForm($1, G, O$) $\neq 0$ then   return false;
return true;
```

References

[1] Hauser H., *Seventeen obstacles for resolution of singularities,* In Singularities. The Brieskorn Anniversary Volume, Arnold V. I., Greuel G.-M., and Steenbrink J., Eds. Birkhäuser, Boston, 1998.

[2] Hironaka, H., *Resolution of singularities of an algebraic variety over a field of characteristic zero I-II,* Ann. Math. 79 (1964), 109–326.

Automated Proofs of Automated Geometry Provers

Bruno Buchberger
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, A-4040 Linz, Austria.
Email: `buchberg@risc.uni-linz.ac.at`

In this talk I address three interrelated topics:

(A) the use of algebraic algorithms as the kernel of automated provers for certain classes of geometric theorems,

(B) the use of automated theorem provers for proving these algebraic algorithms,

(C) the transition from the logic level on which algebraic algorithms are proved to the level in which these algorithms are used as complex inference rules.

(A) is well known by the work of W.T. Wu and others, see for example [Wu 1978] for the characteristic sets algorithm, and by the work of Kutzler and others, see for example [Kutzler, Stifter 1986], for the Gröbner bases algorithm.

I posed (B) as a research problem at the Calculemus Meeting in Rome 1996. In fact, my current research interest in general automated theorem proving and the design and implementation of the Theorema system was heavily motivated by the desire to automate the type of proving necessary for expanding, generalizing, modifying and applying Gröbner bases theory, see [Buchberger 1996] and [Buchberger et al. 2000]. Meanwhile, a couple of research groups started to study the possibility of establishing Gröbner bases theory by automated theorem proving systems.

So far, the implementation of a formalization of Gröbner bases theory by L. Thery, see [Thery 2001], seems to be the most advanced study into this direction. The formalization is carried out in the Coq system and all proofs are checked by the system. Also, the algorithms that are proved correct are executable by producing an Ocaml version automatically. The formalization and proofs within Coq suffer, however, from the fact that Coq is basically only a proof checker, i.e. the proofs have to be composed by the users and are only checked to be correct by the system. Since all proofs had been given completely formally in my original papers on Gröbner bases, see for example the concise summary of the formal proofs in [Buchberger 1998], in the working style, there is very little difference between the situation where you use a "professor" or a "student" as a proof checker and the situation where you use a system like Coq.

Another work into the direction of (B) is [Ruiz-Reina et al. 2001], in which they give an automated proof of the Knuth-Bendix algorithm using the automated prover ACL 2, which is based on the well-known Boyer-Moore prover. Also, [Coquand, Person 1999] contribute to (B) by giving a constructive proof of Dickson's lemma using open induction. The work by Ruiz-Reina seems to provide much more automation in the composition of proofs than the approach by Thery. In fact, they provide a statistics on the proofs that demonstrates that only very few "hints" had to given for finding the proofs. The main human interaction with the system goes into the set-up of the definitions and the layers of lemmata and the sequence in which they are given to the prover. In my view, this is natural and desirable, because the user wants to have control on the systematic build-up of mathematical theories.

In the talk I will analyze the advances made in the area (B). In particular, I will also discuss in which way I envisage Theorema to contribute to (B). Although the current implementation of Theorema is still too weak to establish a complete proof of Gröbner bases theory I will be able to demonstrate a few features of Theorema that promise to make a computer-supported proof of Gröbner bases theory quite attractive:

- definitions, theorems, and algorithms are given in one logic frame (a version of higher-order predicate logic) and in a very user-friendly notation
- algorithms are in fact theorems; their execution is nothing else than the application of a fragment of the underlying logic inference system
- proofs are generated automatically; human interaction is mainly necessary for building up knowledge bases in a reasonable order and for designing appropriate lemmata
- as a consequence, changes in definitions and the set-up of the theory do not make it necessary for the user to re-formulate all proofs; thus, the emphasis of research in Gröbner bases theory (and related theories) can be on the design of domains, definitions, and elementary and advanced propositions
- by the functor construct available in Theorema both for building up domains and knowledge bases, theories and algorithm libraries can be built up in relatively small blocks that can be combined flexibly and allow a high degree of abstractness and generality.

Finally, in the talk, I would like to discuss (C). This is a very subtle issue, which I think is the key point why the ordinary style of building up mathematical theories is efficient and feasible. The transition from proving algebraic algorithms to using them as complex inference rules, i.e. the transition from object level knowledge to meta-level inferencing, is by no means a specific trick in geometry theorem proving but, rather, a paradigm that seems to be ubiquitous in building up mathematical theories in an efficient and manageable way. We will discuss in detail how this transition should be supported by appropriate tools in future computer-supported mathematical software systems in order to make them true mathematical knowledge management systems.

References

[Buchberger 1996] Buchberger B., *Symbolic Computation: Computer Algebra and Logic,* In: Proceedings of FROCOS 1996 (1st International Workshop on Frontiers of Combining Systems), March 26-28, 1996, Munich, (F. Bader, K.U. Schulz eds.), Applied Logic Series, Vol.3, Kluwer Academic Publisher, Dordrecht - Boston - London, 1996, pp. 193-220.

[Buchberger 1998] Buchberger B., *Introduction to Gröbner Bases,* In: Groebner Bases and Applications (B. Buchberger, F. Winkler, eds.), Cambridge University Press, 1998, pp.3-31.

[Buchberger et al. 2000] Buchberger B., Dupre C., Jebelean T., Kriftner F., Nakagawa K., Vasaru D., Windsteiger W., *The Theorema Project: A Progress Report,* In: Symbolic

Computation and Automated Reasoning (Proceedings of CALCULEMUS 2000, Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, August 6-7, 2000, St. Andrews, Scotland, M. Kerber and M. Kohlhase eds.), A.K. Peters, Natick, Massachusetts, pp. 98-113.

[Coquand, Person 1999] Coquand T., Persson H., *Gröbner Bases and Type Theory,* In: T. Altenkirch, W. Naraschweski, B. Reus (eds.), Types for Proofs and Programs, Lecture Notes in Computer Science, Vol 1657, Springer-Verlag, 1999.

[Kutzler, Stifter 1986] Kutzler B., Stifter S., *On the Application of Buchberger's Algorithm to Automated Geometry Theorem Proving,* Journal of Symbolic Computation, Vol. 2/4, pp. 289-298.

[Ruiz-Reina et al. 2001] Ruiz-Reina J. L., Alonso J. A., Hidalgo M. J., Martin F. J., *A Mechanical Proof of Knuth-Bendix Critical Pair Algorithm (Using ACL 2),* Technical Report, Dept. of Computer Science and Artificial Intelligence, Faculty of Informatics and Statistics, University of Sevilla, Avenida Reina Mercedes, 11012 Sevilla, Spain, 2001.

[Thery 2001] Thery L., *A Machine-Checked Implementation of Buchberger's Algorithm,* Journal of Automated Reasoning, Vol. 26, pp. 107-137, 2001.

[Wu 1978] Wu W. T., *On the Decision Problem and the Mechanization of Theorem Proving in Elementary Geometry,* Sciencitia Sinica, Vol. 21, pp. 150-172.

The Projection of Quasi Variety and Its Application on Geometric Theorem Proving

Xuefeng Chen
Institute of Systems Science
Academia Sinica
Beijing 100080,P.R. China
Dingkang Wang
Institute of Systems Science
Academia Sinica
Beijing 100080,P.R. China
Email: `dwang@mmrc.iss.ac.cn`

Let PS be a finite set of polynomials in $K[x_1,\cdots,x_n]$, K is a field of characteristic 0. Let E be and algebraic closed extension field of K. Let DS be also a finite set of polynomials in $K[x_1,\cdots,x_n]$

$$Zero(PS/DS)=\{x|x\in E^n, P(x)=0, \forall P\in PS, \exists d\in DS, d(x)\neq 0\}$$

We define $\bigcup_i Zero(PS_i/DS_i)$ or $\bigcup_i Zero(PS_i/D_i)$ as a quasi variety.

After setting up a coordinate system, geometric theorem can be translated into algebraic form. The hypotheses can be represented by $h_1(x)=0,\cdots,h_p(x)=0$,$PS=\{h_1,\cdots,h_p\}$, $PS=0$ and the conclusion can be represented by $C(x)=0$, $C=0$.

Definition $g\neq 0$ is called the non-degenerate condition if

$$\begin{aligned}(2.A)(\forall x\in E^n)(\quad & PS(x)=0\wedge g(x)\neq 0\\ & \Rightarrow C(x)=0)\\ (2.B)(\exists x\in E^n)(\quad & PS(x)=0\wedge g(x)\neq 0)\end{aligned}$$

Under the non-degenerate condition $g\neq 0$, if $PS(x)=0$, then $C(x)=0$. If $PS=0\Rightarrow C=0$ then the theorem $T=(PS,C)$is universally true.

Wu's non-degenerate condition

Suppose $CS=C_1,\cdots,C_m$ is the characteristic set of PS, then $J=\prod_i I_i$ where I_i is the initial of C_i, if $Prem(C,CS)=0$, i.e. $I_1^{s_1}\cdots I_m^{s_m}C=Q_1C_1+\cdots+Q_mC_m+0$, then $J\neq 0$ is the non-degenerate condition for the theorem to be true. In this case, $\forall x, PS(x)=0, J(x)\neq 0,\Rightarrow C(x)=0$.

Kapur's non-degenerate condition

Let G_1, G_2 be the Groebner basis of ideal (PS) and $(PS\cup\{C*z-1\})$.

if $G_1\neq\{1\}$ and $G_2\neq\{1\}$, then the theorem is true when $g_i\neq 0$. If g_i satisfy

(a)$g_i\in G_2\cap K[x]\wedge g_i\notin (PS)$

(b)$1\notin$ GroebnerBasis$(PS\cup\{g_i*z-1\})$

Winkler's simplest non-degenerate condition.

In $K[x]$, all the polynomials which satisfy (2.A) consists of an ideal N, $N=[radical(PS):(C)]$.

Let G be the Groebner basis of N. If there is a polynomial in N which satisfy condition (2.B) then there is a polynomial in G which also satisfy (2.B).

Let G' be a set of polynomials which satisfy (2.B). Let g be the least polynomial in G' for a given monomial order. then we call $g \neq 0$ the simplest non-degenerate condition.

The non-degenerate conditions given above are too "strong". Sometimes, the non-degenerate conditions are not necessary for a theorem to be true.

Projection of Quasi-variety

The projection map is given as following:
$Proj_{x_{m+1},\cdots,x_n} : E^n \to E^{(n-m)}$

$$Proj_{x_{m+1},\cdots,x_n} : (x_1, \cdots, x_n) \to (x_1, \cdots, x_m)$$

For polynomial set PS and polynomial D in $K[x]$, we define the projection with $x_{m+1}, \cdots, x_n$ as follows

$Proj_{x_{m+1},\cdots,x_n} Zero(PS/D) = \{e \in E^m \mid \exists a \in E^{(n-m)} s.t.(e, a) \in Zero(PS/D)\}$

How to compute the projection of quasi-variety?

We will give a algorithm to compute the projection of a quasi variety in the following two steps.

In the first step, we will give the zero decomposition as

step 1. Zero Decompostion

$$Zero(PS/D) = \bigcup_i Zero(AS_i/D_i)$$

in which AS_i is an ascending set.

step 2. Compute the projection of $Zero(AS/D)$.

We combine step 1 and step 2 together, then we can get the projection of $Zero(PS/D)$.

Theorem For polynomial sets PS, DS and polynomial C as shown above,let $X = (x_1, \cdots, x_n), Z = (x_1, \cdots, x_m), Y = (x_{m+1}, \cdots, x_n)$. Variables order is $x_1 < \cdots < x_n$.
then

(1) if $Proj_Y Zero(PS/C) = \emptyset$
and $Proj_Y Zero(PS/C) \neq \emptyset$, then the theorem T is universally true.

(2) if $Proj_Y Zero(PS/C) \neq \emptyset$, then it gives sufficient and necessary condition for the theorem $T = (PS, C)$ to be false.
$\forall z_0 \in Proj_Y Zero(PS/C)$, then there is a x_0, which makes $PS(x_0) = 0$ but $C(x_0) \neq 0$.
if x_0 which makes $PS(x_0) = 0$ and $C(x_0) \neq 0$, then there is a $z_0 \in Proj_Y Zero(PS/C)$

According to the above theorem, we can give the sufficient and necessary condition for a geometric theorem to be true.

Deciding Topological Properties: Compactness of Basic Real Semialgebaic Sets

Pasqualina Conti
Dipartimento di Matematica
Via Buonarroti 2
I-56127 PISA, ITALY
Email: `conti@dm.unipi.it`
Carlo Traverso
Dipartimento di Matematica
Via Buonarroti 2
I-56127 PISA, ITALY
Email: `traverso@dm.unipi.it`

We discuss the problem of deciding if a basic real semialgebaic set, i.e. a subset of $\mathbb{R}^n$ defined by a finite set of equations and inequalities, is compact.

Compactness is not a first-order property, hence one cannot apply quantifier elimination, but on the real field it is equivalent to closedness and boundedness, that are first order properties, hence the problem is decidable; of course we are looking for better algorithms than elimination of quantifiers in general.

We will use a few standard notations in real algebraic geometry, for which the standard reference is [1]. The basic proof tools rely on quantifier elimination, but it is never used in algorithms, that just use standard commutative algebra tools, real root counting for zero-dimensional ideals being the only specific real algebraic geometry tool.

References

[1] Bochnak J., Coste M., Roy M.-F., *Géomé - trie Algébrique Réelle*, Erg. der Mathematik, Vol. 12, Springer (1987)

Using computer algebra tools to classify serial manipulators

Solen Corvez
IRMAR, Université de Rennes I, France,
Email: `corvez@univ-rennes1.fr`
Fabrice Rouillier
LORIA, INRIA-Lorraine, Nancy, France
Email: `Fabrice.Rouillier@loria.fr`

Industrial robotic 3-DOF manipulators are currently designed with very simple geometric rules. In order to enlarge the the possibilities of such manipulators, it is interesting to relax some constraints.

The behavior of the manipulators when changing posture depends strongly on the design parameters and it can be very different from the one of manipulators commonly used in Industry. P. Wenger and J. El Omri [6], [10] have shown that for some choices of the parameters, 3-DOF manipulators may be able to change posture without meeting a singularity in the joint space. This kind of manipulators is called **cuspidal**.

We restrict the study to 3-DOF manipulators as described in figure 1 and manage to caraterize the values of the parameters for which such a manipulator cuspidal.

In the first part, we show that caraterizing the values of the parameters for which such a manipulator cuspidal is equivalent to find the values of $d_4, d_3, r2$ for which a polynomial $P(t) \in \mathbb{Q}[R, Z, d_4, d_3, r_2][t]$ (where $R = x^2 + y^2 + z^2$ and $Z = z^2$) of degree 4 has a triple real root or in other words, to solve the following system :

$$\left\{ \begin{array}{lcl} P & = & 0 \\ \frac{\partial P}{\partial t} & = & 0 \\ \frac{\partial^2 P}{\partial t^2} & = & 0 \end{array} \right. \tag{1}$$

The goal is then to compute a partition of the parameter's space such that the number of real solutions of system 1 in each cell is constant. Due to practical constrains, we are only interested computing one sample point or a bowl in the cells of highest dimension : the other possible cells will be embedded inside strict algebraic subsets of the parameter's space.

Our first step consists in eliminating two of the variables t, R and $Z = z^2$ in the system 1 : the generic solutions can be viewed as *regular* (with respect to the terminology of [1]) roots of a triangular system with the following shape :

$$\left\{ \begin{array}{l} surf(R, d_4, d_3, r_2) = 0 \\ lc_Z(d_4, d_3, r_2)Z + tr_Z(R, d_4, d_3, r_2) \\ lc_t(d_4, d_3, r_2)t + tr_t(R, Z, d_4, d_3, r_2) \end{array} \right. \tag{2}$$

System 2 describes all the solutions of the problem for values of the parameters taken outside the two algebraic varieties $lc_Z(d_4, d_3, r_2) = 0$ and $lc_t(d_4, d_3, r_2) = 0$, which are closed

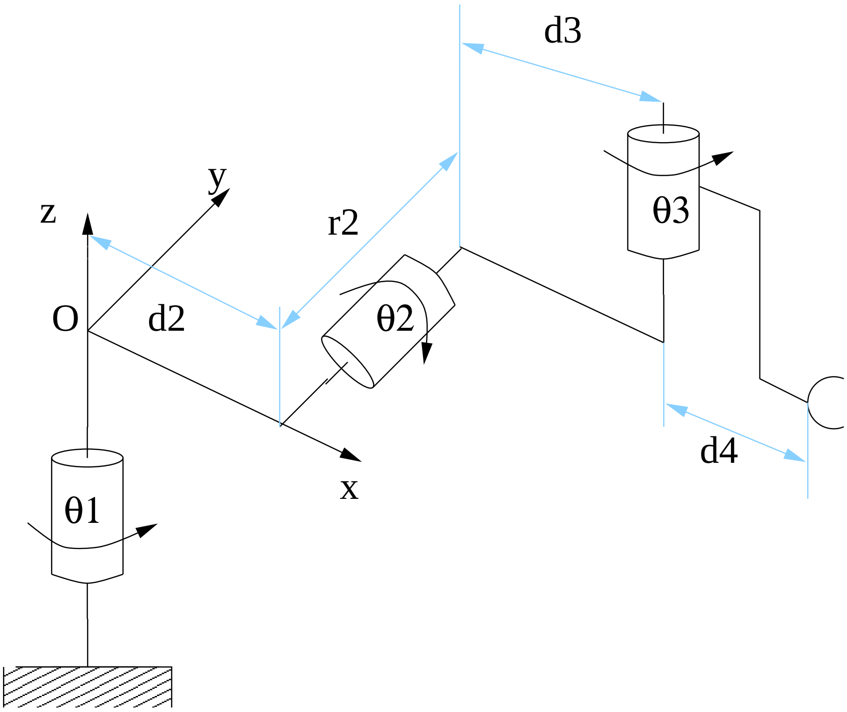


Figure 1: The manipulator under our hypothesis

subsets of strict smaller dimension of the parameter's space and so can be excluded for practical issues.

The number of real roots of $surf$ varies if and only if its discriminant $(dis_R(d_4, d_3, r_2))$ or its leading coefficient with respect to R $(lc_R(d_4, d_3, r_2))$ vanishes.

So, the last set of equations to be computed for defining our partition of the parameters' space in cells where the number of real solutions to system 2 is constant is defined by these two conditions.

The real roots of system 1 must verify $Z = z^2 > 0$ and $R - Z = x^2 + y^2 > 0$ to be admissible. Adding the condition $Z = 0$ (resp. $R - Z = 0$) to the initial system, give us (after an elimination process) two polynomials in the parameters d_3, d_4 and r_2 :$Hyp_{Z=0}(d_4, d_3, r_2)$ and $Hyp_{R-Z=0}(d_4, d_3, r_2)$.

So, in each connected subset of the parameter's space where none of the following polynomials vanish $dis_R(d_4, d_3, r_2)$, $lc_R(d_4, d_3, r_2)$, $Hyp_{Z=0}(d_4, d_3, r_2)$, $Hyp_{R-Z=0}(d_4, d_3, r_2)$, $lc_Z(d_4, d_3, r_2)$, $lc_t(d_4, d_3, r_2)$, the system 1 has a constant number of real solutions.

The best way for representing our partition is now to compute a partial CAD (Cylindrical Algebraic Decomposition - see [3]) of $\mathbb{R}^3$ adapted to this set of polynomials. For practical reasons, we are only interested in finding one point or a bowl in the cells of higher dimension, embedding the other cells inside algebraic subsets of the parameters' space. This make much more easier the projection (much less resultant computations) and lifting phases (no computations with real algebraic numbers) of the CAD.

We mix several technics for this purpose : [5] for reducing the number of polynomials to be computed, [2] for testing if some algebraic sets have real roots or not, etc.

The final result of the full computation is a partial cellular decomposition of the parameter's space so that for each point taken in the interior of any cell, the number of solutions to the system 1 is constant.

Precisely, we have computed :

- at least on point in each cell, as far as possible from the boundaries of the cell;
- the equations of the algebraic sets that bounds these cells;

In practice, we provided 6 polynomials and 105 sample points which represents a reasonable output since it allows roboticians to analyse the results.

References

[1] Aubry P., Lazard D., Moreno Maza M., *On the theories of triangular sets,* Journal of Symbilic Computation, Vol. 28, pp. 105–124, 1999.

[2] Aubry P., Rouillier F., Safey M., *Real Solving for Positive Dimensional Systems,* Research report number RR-3992, INRIA, submitted to Journal of Symbolic Computation, 2000

[3] Collins G. E., *Quantifier elimination for real closed fields by cylindrical algebraic decomposition,* Springer Lecture Notes in Computer Science 33, Vol. 33, pp. 515-532, 1975

[4] Faugère J.-C., *A New Efficient Algorithm for Computing Gröbner bases (F4),* Journal of Pure and Applied Algebra, Vol. 139, N. 1-3, pp. 61-88, 1999

[5] McCallum S., *An improved projection operator for cylindrical algebraic decomposition of three dimensional space,* J. Symbolic Computation, 5:141-161, 1988.

[6] El Omri J., *Analyse géomtrique et cinématique des mécanismes de type manipulateur,* Thèse de Doctorat de l'Université de Nantes. 16 février 1996.

[7] Rouillier F., *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux,* PHD Thesis, Université de Rennes I, 1996.

[8] Rouillier F., *Solving zero-dimensional systems through the rational univariate representation,* Journal of Applicable Algebra in Engineering, Communication and Computing, Vol. 9, N. 5, pp. 433–461, 1999

[9] Wenger P., *Classification of* 3-*R positioning manipulators,* Journal of mechanical design. June 1998 Vol 120, pp :327-332.

[10] Wenger P., El Omri J., *Changing posture for cuspidal robot manipulators,* IEEE 1996 pp:3173-3178.

Straightening in The Whitney Algebra of a Matroid

Henry Crapo
CAMS, EHESS, 54 bd Raspail, 75230 Paris Cedex, France
Email: `crapo@ehess.fr`
William Schmitt
George Washington University, Washington, DC, USA
Email: `wschmitt@gwu.edu`

The concept of a *matroid*, as introduced by Hassler Whitney, captures the abstract combinatorial properties of linear (or even algebraic) dependence of finite configurations of points in projective space. The information carried by the matroid structure is often 'all you need to know' about a configuration of projective points, that is, all its *intrinsic* properties: which sets of points are collinear, coplanar, cospatial, etc. The matroid (that is, the matroid of linear dependence of points) does not, however, carry information about how the configuration has been embedded in projective space. The matroid does not determine whether a given set of points lies together on a non-linear algebraic surface, nor does it predict whether three pairs of points generate lines that are concurrent, unless a point of concurrence happens to be itself a point of the configuration. Such properties could be termed *extrinsic* to the configuration, requiring information about the environment of the figure.

The deepest problems in matroid theory involve the search for combinatorial obstacles to the representability of matroids in projective spaces over specified fields. This is another sign that the gap between intrinsic and extrinsic properties of matroids is highly significant.

The best-known earlier attempts to bridge this gap are the Orlik-Solomon algebra, extensively used to study hyperplane configurations, and Neil White's bracket ring. Both of these constructions start by taking anticommutative products of elements of the matroid.

We propose a further step in this direction, associating with each matroid an appropriately weakened form of Hopf algebra, called a *lax Hopf algebra*, constructed from the free exterior algebra generated by the elements of the matroid, forming tensor products, then taking the quotient by the ideal generated by *homogeneous components of coproducts of dependent words*. In this manner we effectively impose those algebraic relations that necessarily would hold in any representation of the matroid, and over any field. We call this quotient the *Whitney algebra* of the matroid.

That this is a natural construction follows from a simple observation. Consider the Hopf algebra structure of the exterior algebra $\Lambda = \bigoplus \Lambda^k$ generated by a finite set of points in a projective space. Recall that the coproduct $\delta : \Lambda \to \Lambda \otimes \Lambda$ is the multiplicative map determined by $\delta(a) = a \otimes 1 + 1 \otimes a$, for all vectors $a \in \Lambda^1$; for example,

$$\begin{aligned}\delta(abc) &= \delta(a)\ \delta(b)\ \delta(c)\\ &= abc \otimes 1 \ + \ ab \otimes c \ - \ ac \otimes b \ + \ bc \otimes a\\ &\qquad + \ c \otimes ab \ - \ b \otimes ac \ + \ a \otimes bc \ + \ 1 \otimes abc,\end{aligned}$$

for vectors a, b, c (where the signs are determined by anticommutativity). Now if the set $\{a, b, c\}$ is dependent, then the wedge product abc is equal to zero in Λ, and hence the coproduct $\delta(abc)$ is also zero. Since Λ is graded by the nonnegative integers $\mathbf{N}$, the tensor product $\Lambda \otimes \Lambda$ is thus graded by $\mathbf{N} \times \mathbf{N}$, and an element of $\Lambda \otimes \Lambda$ is equal to zero if and only if all its $(\mathbf{N} \times \mathbf{N})$-homogeneous components are zero. Hence, in particular, if $\{a, b, c\}$ is linearly dependent, then the homogeneous component $a \otimes bc - b \otimes ac + c \otimes ab$ of *shape* $(1, 2)$ in the coproduct $\delta(abc)$ is equal to zero. We obtain similar relations in each component $T^k(\Lambda) = \Lambda \otimes \cdots \otimes \Lambda$ of the tensor algebra $T(\Lambda) = \bigoplus T^k(\Lambda)$ from the fact that the iterated coproduct $\delta^k(a_1 \cdots a_r)$ is zero for any dependent set of vectors $\{a_1, \ldots, a_r\}$.

In this talk, we exhibit our main technical result, the *Zipper lemma*, a cancellation theorem that points to the basic *exchange properties* of the Whitney algebra of a matroid, and proves the (skew) commutativity of a geometric product. This geometric product is essentially the product defined by Hermann Grassmann in his 1844 *Ausdehnungslehre*, roughly speaking, the tensor product of join and meet operators.

In the spirit of the present assembly, we demonstrate the straightening algorithm for the Whitney algebra of uniform matroids, based on the Rutherford method of interpolants, here applied to tableaux with *holes*. Such Whitney algebras are torsion-free, so this method yields a basis for each homogeneous component of the algebra. An analogous method for Whitney algebras (not necessarily torsion free) of arbitrary matroids is not yet known, but we will indicate a possible approach. On the other hand, a simple, but highly exponential, algorithm yields a normal form for the integer matrix of coefficients of the defining relations of each homogeneous component of the algebra. The torsion properties of this algebra reveal coordinatizability properties of the underlying matroid.

MMP/Geometer - A Software Package for Automated Geometry Reasoning

Xiao-Shan Gao
Institute of System Science, AMSS, Academia Sinica, China
Email: `xgao@mmrc.iss.ac.cn`
Qiang Lin
Institute of System Science, AMSS, Academia Sinica, China
Email: `qlin@mmrc.iss.ac.cn`

MMP (Mathematics-Mechanization Platform) is a stand-alone software platform which implements various versions of Wu-Ritt characteristic set (CS) method and its applications [4].

MMP/Geometer is a package of MMP for automated geometry reasoning. The aim of MMP/Geometer is try to automate some of the the basic geometric activities, mainly geometric theorem proving, geometric theorem discovering, and geometric diagram generation. The current version is mainly for plane Euclidean geometries and the differential geometry of space curves.

The goal of MMP/Geometer is to provide a convenient and powerful tool to learn and use geometry by combining the methods of geometric theorem proving and geometric diagram generation. The introduction of computer into geometry may give new life into the learning and study of the classical field. Geometric problems are abundant in the field of robotics, CAD, and computer vision. We expect that MMP/Geometer may have applications in these fields.

Automated Geometric Theorem Proving and Discovering

Study of automated geometric theorem proving (AGTP) may be traced back to the landmark work by Gelernter in the late fifties. The extensive study of AGTP in the past twenty years is due to the introduction of Wu's method in late seventies [4], which is surprisingly efficient for proving difficult geometric theorems. AGTP is now one of the successful fields of automated reasoning. There are few areas for which one can claim that machine proofs are superior to human proofs. Geometry theorem proving is such an area.

Our experiments show that MMP/Geometer is a quite efficient in proving geometry theorems. Within its domain, it invites comparison with the best of human geometry provers. Precisely speaking, we have implemented the following methods.

Wu's method might be the most powerful method in terms of proving difficult geometric theorems and applying to more geometries [4]. Wu's method is a coordinate-based method. It first transfers geometric conditions into polynomial or differential equations in the coordinates of the points involved, then deals with the equations with the characteristic set method.

The area method uses high-level geometric lemmas about geometry invariants such as the area and the Pythagorean difference as the basic tool of proving geometry theorems [1]. The method can be used to produce human-readable proofs for geometry theorems.

The deductive database method can be used to generate the fixpoint for a given geometric configuration[2]. With this method, we can not only find a large portion of the well-known facts about a given configuration, but also to produce proofs in traditional style.

Comparing with other provers, MMP/Geometer has the following distinct features. First, it implements some of the representative methods for AGTP, while most previous provers are for one method. MMP/Geometer is stand-alone, while most of the previous provers are implemented in Lisp or Maple. Third, MMP/Geometer is capable of producing human-readable proofs and proofs in traditional style. Finally, MMP/Geometer has a powerful graphic interface, which will be introduced below.

Automated Geometric Diagram Generation (AGDG)

MMP/Geometer implements AGDG methods for the following reasons. First, a geometry theorem prover needs a nice graphic user interface (GUI). Second, AGDG methods may enhance the proving scope for AGTP methods with constructive statements as input by finding the construction sequence. Third, with AGDG methods, MMP/Geometer may be used in application areas such as robotics, linkage design and computer vision.

Dynamic geometry software systems, noticeably, Gabri, Geometer's Sketchpad, Cinderella, and Geometry Expert [3] may generate diagrams interactively based on ruler and compass construction. These systems are mainly used to education and simulation of linkages. It is well known that the drawing scope of ruler and compass construction has limitations. To draw more complicated diagrams, we need the method of automated geometry diagram generation (AGDG).

In MMP/Geometer, by combining the idea of dynamic geometry and AGDG we obtain what we called the *intelligent dynamic geometry*, which can be used to input and manipulate diagrams more easily. It can be used to manipulate geometric diagrams interactively as dynamic geometry software and does not have the limitation of ruler and compass construction.

References

[1] Chou S. C., Gao X. S., Zhang J. Z., *Machine Proofs in Geometry,* World Scientific, Singapore, 1994.

[2] Chou S. C., Gao X. S., Zhang J. Z., *A Deductive Database Approach To Automated Geometry Theorem Proving and Discovering,* J. Automated Reasoning, 25, 219-246, 2000.

[3] Gao X. S., Zhang J. Z., Chou S. C., *Geometry Expert (in Chinese),* Nine Chapter Pub., Taipai, Taiwan, 1998.

[4] Wu W. T., *Mathematics Mechanization,* Science Press/Kluwer, Beijing, 2000.

The *SymbolicData* GEO Records - A Public Repository of Geometry Theorem Proof Schemes

Hans-Gert Gräbe
Univ. Leipzig, Germany
Email: `graebe@informatik.uni-leipzig.de`
`http://www.symbolicdata.org`

The collection of geometry theorem proof schemes that I will present in my talk is a subproject of the larger *SymbolicData* project to set up and run a publicly available repository of digital test and benchmark data from different areas of symbolic computation. This work is part of the benchmark activities of the German “Fachgruppe Computeralgebra” who also sponsored the web site [3] as a host for presentation and download of the tools and data developed and collected so far. We kindly acknowledge support also from UMS MEDICIS of CNR/École Polytechnique (France) who provides us with the needed hard- and software for establishing and running this web site.

About the *SymbolicData* project

For easy reuse of the *SymbolicData* data both in the repository and at a local site we concentrated on free software tools and concepts. The data is stored in a XML like ASCII format (using tag/value pairs) in records that can be edited with your favorite text editor. Records with similar attributes (tags) are grouped into tables. Table descriptions are stored in the same XML-like format and can be manipulated, changed, extended etc. as easily as records. The tools are completely written in Perl using Perl 5 modular technology.

The project is organized as a free software project. The CVS repository is equally open to people joining the *SymbolicData* project group. Tools and data are freely available also as tar-files (via HTML download from our Web site) under the terms of the GNU Public License.

The *SymbolicData* project and geometry theorem proving

Due to the research interests of the people involved so far with the *SymbolicData* project we mainly collected various data related to polynomial system solving. Geometry theorem proving via the Descartes-Wu coordinate method is one of the sources of challenging benchmark examples for that area. Such systems can be derived automatically with appropriate computer algebra software (CAS) from proof schemes that formalize more informal geometric statements. [1] is probably the most prominent source of such examples, where S.-C. Chou collected and worked out 512 examples this way. The *SymbolicData* proof scheme collection contains more than 200 of these examples, but also various examples from other sources.

Proof schemes and geometry theorem proving

Even if this conference refers in its title to “automated deduction in geometry” many authors speak about “mechanized geometry theorem proving” since the whole process of proving includes some stages that require human intervention and intuition. The first and dominant such step is the translation of the informal geometric statement into a formal proof scheme that takes into consideration the special proof approach (and software). It is that part of Chou's work that required most diligence and erudition. Hence it is desirable to collect such geometry proof schemes as the starting point of further transformations.

Given such a proof scheme and reliable software the geometric problem can be translated automatically into an algebraic one. Even the solution of the algebraic problem can be automated if it requires only standard techniques. This applies, e.g., to constructive proof schemes since the corresponding algebraic problem reduces to simplification of rational expressions. For other algebraic approaches (e.g., Gröbner bases) the solution may be less straightforward and require more human interaction. This is reflected by a `solution` tag in such *SymbolicData* GEO records that stores some information about a possible treatment of the corresponding algebraic problem.

Towards a proof scheme repository

Since a public repository of proof schemes should serve different geometry theorem provers it is not desirable to store proof schemes in a special language of one of them. Instead one should invent and use a generic proof scheme language that is sufficiently general to easily map to the special proof scheme languages of the different provers. There are several concepts for generic data exchange on the way (XML, OpenMath, MathML).

For our first experiments within the *SymbolicData* project we invented our own generic proof scheme language – the *GeoCode.* It arose from a prototypical test implementation of a geometry theorem prover based on the coordinate method – the *GeoProver* [2]. *GeoProver* versions exist for the four major Computer Algebra Systems Maple, Mathematica, MuPAD, and Reduce. The translation of the generic *GeoCode* proof schemes to each of the target systems is realized with appropriate Perl tools within the *SymbolicData* tools framework.

With *GeoProver*, version 1.2, the *GeoCode* description was separated from the *GeoProver* and stored in a special table `GeoCode` within the *SymbolicData* project. Hence the *GeoCode* description may be extended, modified and adopted in the same way as other *SymbolicData* records. It requires further discussion with interested parties to work out the necessary changes.

A first comparison with the syntax of other geometry provers ([4, 5]) given in the literature encourages that with some more Perl programming efforts the collected proof schemes are valuable not only for other provers based on the coordinate method but – at least for constructive proof schemes – also to provers using Cayley algebra computations. It seems also possible to fix these definitions in a more common OpenMath-compliant format. We started first experiments to translate constructive proof schemes in a format to be displayed by the dynamical geometry software GeoNext.

References

[1] Chou S.-C., *Mechanical Geometry Theorem Proving,* Reidel, Dortrecht, 1988.

[2] Gräbe H.-G., *GeoProver - a small package for mechanized plane geometry,* 1998–2002. With versions for Reduce, Maple, MuPAD, and Mathematica. Some prototypes were compiled in cooperation with M. Witte. See `http://www.informatik.uni-leipzig.de/~compalg/software`.

[3] *The SymbolicData Project,* 2000–2002. See `http://www.SymbolicData.org` or the mirror at `http://symbolicdata.uni-leipzig.de`.

[4] Wang D., GEOTHER: *A geometry theorem prover,* In M.A. McRobbie and J.K. Slaney, editors, Automated deduction – CADE-13, volume 1104 of LNCS, pages 166 – 170, 1996.

[5] Wang D., *Clifford algebraic calculus for geometric reasoning with applications to computer vision,* In D. Wang, editor, Automated Deduction in Geometry, Toulouse 1996, volume 1360 of LNAI, pages 115 – 140. Springer, Berlin, 1997.

On The Structural Rigidity For GCSPs

Christophe Jermann*
COPRIN Team, INRIA-I3S-CERMICS, 2004 route des lucioles,
BP 93, 06902 Sophia Antipolis, France
Email: `Christophe.Jermann@@sophia.inria.fr`
Bertrand Neveu
COPRIN Team, INRIA-I3S-CERMICS, 2004 route des lucioles,
BP 93, 06902 Sophia Antipolis, France
Email: `Bertrand.Neveu@@sophia.inria.fr`
Gilles Trombettoni
COPRIN Team, INRIA-I3S-CERMICS, 2004 route des lucioles,
BP 93, 06902 Sophia Antipolis, France
Email: `Gilles.Trombettoni@@sophia.inria.fr`

The rigidity concept is in the heart of many *geometric constraint satisfaction problems* (GCSP) applications. In particular, constructive solving methods use this property to decompose GCSPs into solvable subsystems. Rigidity detection procedures can be classified in 2 categories: pattern-based approaches depend on a repertoire of known rigid subsystems which cannot cover all practical instances; flow-based approaches use flow machinery to identify subsystems verifying a structural property: the *structural rigidity.* The latter approaches are more general although structural rigidity is only an approximation of rigidity.

A New Structural rigidity

A **GCSP** $S = (O, C)$ is composed of a set O of *geometric objects* (points, lines, planes, ...) represented by their generalized coordinates; and a set C of *geometric constraints* (distances, angles, incidences, parallelisms, ...) represented by systems of equations and inequalities on the generalized coordinates. Each object has a number of *degrees of freedom* (DOF) equal to the number of its independant coordinates; each constraint removes a number of DOF equal to the number of its independant equations. The number of DOF of a GCSP is equal to the sum of its objects' DOF minus the sum of its constraints' DOF. Figure 1(a) presents an example of a GCSP in 3D.

Intuitively, a GCSP is **rigid** if it is indeformable and can be displaced anywhere in the considered geometric space. It is **over-rigid** if it cannot be displaced anywhere, and **under-rigid** otherwise. In the example of figure 1(a), the subsystem CDF is rigid, AF is under-rigid and $ACDEF$ is over-rigid.

---

* Supported by CNRS and Région Provence Alpes Côte d'Azur

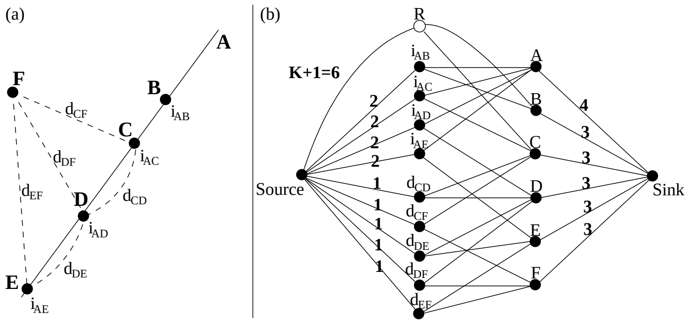


Figure 1: (a)A GCSP in 3D composed of 1 line (A, 4 DOF) and 5 points (B, C, D, E and F, 3 DOF each) constrained by 4 point-line incidences (i_{AB}, i_{AC}, i_{AD}, i_{AE}, 2 DOF each) and 5 point-point distances (d_{CD}, d_{CF}, d_{DE}, d_{DF}, d_{EF}, 1 DOF each) represented in dots. (b) The objects-constraints network corresponding to this GCSP. Capacities are represented in bold on arcs.

The structural rigidity is a generalization of Laman [1] characterization of rigidity for bar frameworks in 2D. A GCSP S is **structurally rigid** (s_rig1) in dimension d if it verifies $DOF(S) = \frac{d(d+1)}{2}$ and $\forall S' \subset S$, $DOF(S') \geq \frac{d(d+1)}{2}$. Indeed, $\frac{d(d+1)}{2}$ is the number of independant displacement (rotation+translation) in a geometric d-space, and a rigid GCSP must have all these displacements and admit no other one. It is well known that redundant or non-generic GCSPs mislead structural rigidity. We highlight here another limit which applies even in the non-redundant and generic case: there exists rigid GCSP having less than $\frac{d(d+1)}{2}$ DOF in dimension d. Such subsystems appear in figure 1(a): CD and $ACDE$ are rigid but have only 5 DOF, which makes them over-s_rig1 (since $\frac{d(d+1)}{2} = 6$ in 3D); also, $ABCD$ is under-rigid and has 6 DOF but contains CD and is found over-s_rig1 (or exact-s_rig1 if subsystems having less than d objects are not considered).

We introduce the **degree of rigidity** (DOR) of a set of objects O in the context of a GCSP S: $DOR(O, S)$ is the number of DOF that the subsystem of S induced by O has if it is rigid. Determining $DOR(O, S)$ is equivalent to geometric theorem proving in the general case since it depends on the geometric properties S induces on O (parallelisms, incidences, alignment, ...). We then reformulate the structural rigidity definition as follows: A GCSP $S = (O, C)$ is **structurally rigid** (s_rig2) in dimension d if it verifies $DOF(S) = DOR(O, S)$ and $\forall S' = (O', C') \subset S$, $dof(S') \geq DOR(O', S)$. This definition remains a heuristic for rigidity but is strictly better than the initial one. For instance, CD and $ACDE$ have $DOR = 5$ and are then detected s_rig2. Also, ABCD has DOR=5 and is detected under-s_rig2.

Algorithms

We propose algorithms, corresponding to our new definition, for the main problems of rigidity: deciding if a GCSP is rigid and detecting rigid or over-rigid subsystems. They use flow machinery on a network $G(S)$ introduced in [2] to represent a GCSP S. Constraints and objects are nodes in $G(S)$, and the capacities are the DOFs. Figure 1(b) presents the network corresponding to the GCSP in figure 1(a).

A flow in $G(S)$ is a distribution of the constraints' DOF among the objects' DOF. If a maximum flow in $G(S)$ does not saturate all arcs outgoing the source, this means that some constraints' DOF cannot be ***absorbed by*** the objects' DOF, i.e. $\exists S' \subseteq S$ s.t. $DOF(S') < 0$. We propose to add a node R, with capacity $K+1$ from the source (see figure 1(b)). Hence, maximum flow allows to identify $S'' \subseteq S$ s.t. $DOF(S'') \leq K$: It was proved [2] that S'' is induced by the set O'' of objects traversed during the last search for an augmenting path. Taking $K = DOR(O', S)$, we obtain $DOF(S'') \leq DOR(O'', S)$ which implies S'' is over or exact s_rig2. Linking R successively to all *DOR-minimal* subsets of objects in the GCSP allows to identify an over or exact s_rig2 subsystem if there exists one. A **DOR-minimal** set O of objects in a GCSP S verifies $\forall O' \subsetneq O$, $DOR(O, S) > DOR(O', S)$. DOR-minimal subsets contains at most 3 objects for GCSPs built on points, lines and planes in 3D.

A GCSP $S = (O, C)$ is s_rig2 if it contains no over-s_rig2 subsystem and verifies $DOF(S) = DOR(O, S)$. To find an over-s_rig2 subsystem, we use the same algorithm with capacity $K = DOR(O', S) - \mathbf{1}$ instead of $DOR(O', S)$.

These algorithms are polynomial in the size of DOR-minimal sets if the DOR can be computed in polynomial time. Computing the DOR is polynomial for mechanisms and GCSPs limited to metric constraints. It is difficult for GCSPs with incidence and parallelism constraints since they introduce explicit degeneracies. Other easy classes have to be found.

References

[1] Laman G., *On graphs and rigidity of plane skeletal structures,* J. Eng. Math., Vol. 4, 331–340, 1970.

[2] Hoffmann C., Lomonossov A., Sitharam M., *Finding solvable subsets of Constraint Graphs,* Principles and Practice of Constraint Programming CP'97, 463–477, 1997.

The shape of spherical rational quartics

Bert Jüttler
Johannes Kepler University, Linz, Austria
Email: `bert.juettler@jku.at`

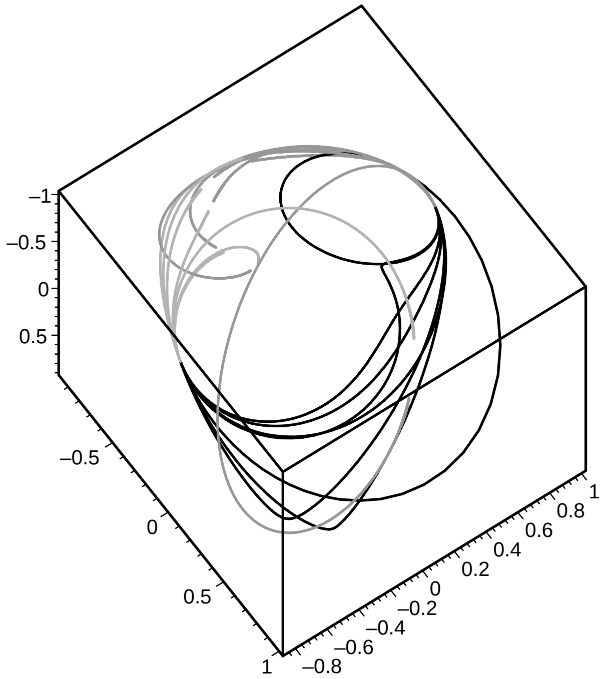
Spherical curves have various applications in geometric modelling and in kinematics and animation. For instance, curves on the 4D unit sphere can be identified with spherical (i.e., rotational) motions, and they can therefore be used for computer animation and robot motion planning.

Rational curves on quadric surfaces can be seen as solutions to certain Diophantine equations in the ring of polynomials. In the case of the sphere $\mathbb{S}^2$, which is a representative of the class of oval quadrics, this equation takes the form $w^2 = x^2 + y^2 + z^2$. All irreducible solutions can be generated with the help of a classical representation formula from number theory, which was first noted by V.A. Lebesgue in 1868 [1]. More recently, this formula has been used to define a mapping from real projective 3–space onto the unit sphere, $\delta : P^3(\mathbb{R}) \to \mathbb{S}^2$, which has been called the *generalized stereographic projection* [2]. Due to its algebraic origin, this mapping can be used to generate any rational curve of degree $2n$ on the sphere as the image of a curve of degree n.

Besides, this mapping has been be analyzed from a geometrical point of view. It can be shown to identify the points of the unit sphere with a very special two–parameter system of lines, which is called a elliptic linear congruence. (See [3] for more information on line geometry). As a major advantage, the generalized stereographic projection avoids the dependency on the choice of the center of projection, which is always introduced by using the standard stereographic projection.

We use the generalized stereographic projection to generate and to analyze the solutions to the C^1 Hermite interpolation problem with spherical rational curves on the sphere $\mathbb{S}^2$. Given two points with associated first derivatives on a sphere, we interpolate these data with a rational curve segment of degree 4. (In the 4D case, the data span a three–dimensional space, and is natural to ask for solutions which are contained in it. Note that also non–3D solutions exist; they have no singularities.)

The given data can be interpolated with a two–parameter family of curves. Some examples are shown in the following figure.



The black curve segments match the same C^1 Hermite boundary data.

Using the generalized stereographic projection, each solution can be identified with a point in a certain parameter plane. We discuss the shape of the solutions, which is characterized by the presence of cusps or double points. This results in a so–called *characterization diagram:* the parameter plane is subdivided in different regions which correspond to solutions exhibiting the same shape.

This talk is based on joint work with Wenping Wang (The University of Hong Kong, China).

References

[1] Dickson L. E., *History of the Theory of Numbers,* Vol. II, Chelsea, New York, 1952.

[2] Dietz R., Hoschek J., Jüttler B., *An algebraic approach to curves and surfaces on the sphere and on other quadrics,* Computer Aided Geometric Design 10 (1993), 211–229.

[3] Pottmann H., Wallner J., *Computational Line Geometry,* Springer, 2001.

C^1 Spline Implicitization of Planar Curves

B. Jüttler
Institute of Analysis, Dept. of Applied Geometry
Johannes Kepler University, Linz, Austria
Email: `bert.juettler@jku.at`
J. Schicho
Research Institute for Symbolic Computation
Johannes Kepler University, Linz, Austria
Email: `jschicho@risc.uni-linz.ac.at`
M. Shalaby
Research Institute for Symbolic Computation
Johannes Kepler University, Linz, Austria
Email: `shalaby@risc.uni-linz.ac.at`

In CAGD, there are two possibilities to define planar curves: the implicit form $f(x, y) = 0$, or by parametric equations $x = x(t)/w(t)$ and $y = y(t)/w(t)$ where $x(t), y(t)$, and $w(t)$ are polynomials. Both the parametric and implicit representation have its own advantages. The availability of both often results in simpler computation. For example, if both implicit and parametric representation are available, the intersection of two curves is obtained easier than if only implicit or parametric representation is available.

From classical algebraic geometry, it is known that each rational parametric curve has an implicit representation, while the converse is not true. The process of converting the parametric equation into implicit form is called *implicitization.* A number of established methods for *exact* implicitization exists: Resultants [4], Gröbner bases [1], and Moving curve and surface [9].

However, *exact* implicitization has not found widespread use in CAGD. This is in part due to the following facts:

- Exact implicitization often produces large data volumes.
- Exact implicitization process is relatively complicated, especially, in the case of high polynomial degree.
- A single exact implicitized parametric curve may have unwanted components or self intersections.

For these reasons, *approximate implicitization* has been proposed. A number of methods are available for approximate implicitization: Montaudouin and Tiller [8] employed a power series method to obtain local explicit approximation (about a regular point) to polynomial parametric curves and surfaces. Chuang and Hoffmann [3] extend this method using what they called “implicit approximation”. Dokken [5] proposed a new way to approximate the parametric curve and surface, globally, in the sense that the approximation is valid within the whole domain of the curve segment or surface patch. Sederberg et al. [10] employed monoid curves and surfaces to find an approximate implicit equation and approximate inversion map of a planar rational parametric curve or a rational parametric surface .

This paper discusses the problem of constructing what we call a spline implicitization for planar curves: a partition of the plane into polygonal pieces, and an implicit polynomial for each piece. On the boundaries, these polynomial pieces are joined together with C^m continuity for suitable choice of m.

Recall that the parametric and implicit representations of a planar curve have the same polynomial degree n. However, the number of the coefficients in the parametric case is $2(n+1)$ while it is $(n+1)(n+2)/2$ in the implicit case, i.e., in the implicit case, high polynomial degree will lead to expensive computation.

Therefore, the main goal of this paper is to find a C^1 *low* degree spline implicit representation of a given parametric planar curve. We extend our technique proposed in [6]. First, to ensure the low degree condition, quadratic B-splines are used to approximate the given curve via orthogonal projection in Sobolev spaces. Adaptive knot removal, which is based on spline wavelets [2], is used to reduce the number of segments. The resulting quadratic B-spline segments are implicitized. These implicitized quadratic segments are joined together with C^0 continuity along suitable transversal lines. Using results from classical differential geometry [7], the asymptotic behavior of these transversal lines for small step size is analyzed. We showed that these lines are always well behaved, except at inflections of the original curve.

Finally, by multiplying with suitable polynomial factors, the segments are joined with C^1. The main problem was to "localize" the construction, as otherwise the degree of the implicit representation would depend on the number of segments. In order to localize the construction, we introduce some additional partition lines and multiply each segment by a piecewise quadratic multiplier, defined as a quadratic polynomial in each sub-patch. In this case, we get a global C^1 spline curve of degree 4. As an advantage, we have a smaller data volume compared with the case of *exact* implicitization.

References

[1] Buchberger B., *Application of Gröbner bases in non-linear computational geometry,* Mathematical Aspects of Scientific Software(J. Rice, Ed.), Springer-Verlag, New York/Berlin, 1988.

[2] Chui C. K., Quak E., *Wavelets on a bounded interval,* Numerical Methods in Approximation Theory, D. Braess, L. L. Schumaker (eds.), Birkhauser Verlag, Basel, 1992, volume 9; pp. 53–75.

[3] Chuang J. H., Hoffmann C. M., *On local implicit approximation and its application,* ACM Trans. Graphics 8, 4, 1989, pp 298-324.

[4] Cox D., Little J., O'Shea D., *Ideals, varieties and Algorithms,* Springer-Verlag, New York, 1997.

[5] Dokken T., *Approximate implicitization,* Mathematical Methods in CAGD, Lyche T., Schumaker L. L. (eds.), Vanderbilt University Press, Nashville & London, 2001.

[6] Jüttler B., Schicho J., Shalaby M., C^0 *spline Implicitization of Planar Curves,* submitted to Curves and Surfaces 2002.

[7] Kreyszig E., *Differential geometry,* Dover, Toronto, 1991.

[8] de Montaudouin Y., Tiller W., Vold H., *Application of power series in computational geometry,* CAD 18, 10, 1986, pp 93-108.

[9] Sederberg T. W., Chen F., *Implicitization using moving curves and surfaces,* Computer Graphics (SIGGRAPH 95 Conference Proceedings)(R. Cook, Ed.), Vol. 29, pp. 301-308. Addison-Wesley, Reading, MA, 1995.

[10] Sederberg T. W., Zheng J., Klimaszewski K., Dokken T., *Approximate Implicitization Using Monoid Curves and Surfaces,* Graphical Models and Images processing 61, 1999,pp 177-198.

Feature-preserving Simplification of Polygonal Surface based on Half-edge Contraction Manner

Myeong-Cheol Ko
Multimedia/Graphics Lab. Dept. of Computer Science, Yonsei Univ.
134 Shinchon-Dong, Seodaemun-Gu, Seoul 120-749, Korea
Email: `zoo@rainbow.yonsei.ac.kr`
Yoon-Chul Choy
Multimedia/Graphics Lab. Dept. of Computer Science, Yonsei Univ.
134 Shinchon-Dong, Seodaemun-Gu, Seoul 120-749, Korea
Email: `ycchoy@rainbow.yonsei.ac.kr`

In recent years, with the introduction of reverse engineering in the fields of 3D modeling, there has been increased need for complex and high detailed models, which are generally created by 3D laser scanner, in many computer graphics applications. Current high-end graphics systems are capable of rendering tens of millions of polygons per second. However, the complexity of large geometric datasets appears to be growing at a faster rate as compared to the rendering capabilities of the graphics systems. The number of polygons much affects the rendering speed of the system as well as memory usage. Although the highly complex and detailed models can provide a convincing level of realism, it must be noted that the complexity of the model does not mean the degree of recognition of the model. Because the human cannot distinguish the degree of details from models with various resolution more than some level of complexity. This means that over highly complex model can cause rather problems. Fortunately, the full complexity of the models is not always required and desirable in applications, such as simulation and virtual reality systems, focusing on the real-time interactivity. Since, it is acceptable to decrease the fidelity of the models for increasing the runtime efficiency. Moreover, in such systems, an object might be projected on the screen at various scales, according to the distance from the viewer. We, however, cannot recognize the degree of visual details, as the size gets smaller. Therefore, it is useful to have various simpler versions of original complex models according to its uses in various applications. The surface simplification is one of the methodologies to solve the problems. In polygonal surface simplification, the goal is to take a complex polygonal model as input and automatically generate a simplified model as output without a loss of geometric properties of the original model if possible.

The goals of this works are to generate a various simpler versions of original complex model with retaining the characteristic features of the original model even after drastic simplification process. The primary contributions of this work are as following:

1. Error metric guaranteeing feature preservation: We have developed an error metric that describes and reflects the geometric features of surface and the geometric changes before and after simplification. To define the error metric, we introduced the orientation component of local surface as an additional property.

2. Surface simplification algorithm based on half-edge contraction manner: We have developed a surface simplification algorithm based on half-edge contraction manner utilizing

the error metric is implemented. The half-edge based scheme is the method that repositions a new vertex after the edge contraction by merging one of the two vertices to the other in previous edge having directionality. It is more efficient in memory usage and useful in real-time applications, which require progressive transmission of large amount of surface data and instant rendering compared to existing methods adopting optimal positioning scheme.

In addition, the proposed simplification algorithm has some additional features or advantages. The algorithm assumes a high simplification rates. In many cases, according to our observations, when the original model is simplified less than some degree of simplification ratio, there is almost no visual difference. Moreover, applications for real-time rendering might demand higher rate of simplification to decrease the complexity of a scene. So, the proposed algorithm is concentrated on the preservation of geometric features of the original model even after performing drastic level of simplification. Another feature of the algorithm is that the reduction process is completely automated. A number of existing algorithms require a user to specify an error term, i.e. distance. However, requiring a user intervention to specify the desired fidelity of the approximation is not practical and thus not a good approach. Since, to run the algorithm, a user should set the different error terms according to the various models to be simplified, although the user may not have any knowledge about the models. The user intervention in the proceeding of the algorithm often means that the absence of the error metric. A surprising number of algorithms use no metric at all. In the proposed algorithm, the only constraint to halt the algorithm is the target triangle or vertex count, which is specified by a user. Whether or not to support the triangle or vertex-budget approach is important for time-critical rendering on which we focus.

Implicitization of Algebraic Varieties[1]

Günter Landsmann
Research Institute for Symbolic Computation,
Johannes Kepler University, A-4040 Linz, Austria
Email: `gland@risc.uni-linz.ac.at`

An affine algebraic variety is the set of solutions of a system of polynomial equations. Given a set of polynomials $f_1, \ldots, f_m$, the geometric aspects of finding and describing their common solutions $\mathbf{V}(f_1, \ldots, f_m)$ may be summarized by the notion 'parametrization of varieties'. While there is an obvious need for a concise description of the solution set of a polynomial system, in applications the problem often arises just the other way round. For instance, in Computer Aided Geometric Design one often asks for a finite set of polynomial equations, whose set of solutions contains all image points of a given parametrized surface in 3-space. Of course one is interested in such a system which contains these image points in a minimal way. Finding such a system from a given parametrization is called 'implicitization'. Thus, if ρ is a parametrizing map, implicitization is the problem of finding the smallest algebraic variety X that contains the parametrized set $\mathrm{im}(\rho)$.

We consider several methods of computing the implicit form. Among others, they make use of Gröbner Bases, resultants and interpolation techniques. All these methods are related to Elimination Theory which in its modern developments lies at the heart of implicitization.

Interpolation

Finding the implicit representation of a paramtrized variety means finding the coefficients of a finite set of polynomials. So, if we know bounds for the degrees of the desired polynomials, we may evaluate the given parametrizing functions in some finite set of interpolation nodes, obtaining a linear system L. A nontrivial solution of L yields an answer to the implicitization problem. Although such degree bounds are easy to compute, the direct approach often fails due to the huge size of the linear system. Recently there has been progress in the development of methods taking advantage of the special type of such a linear system.

Gröbner Bases

The fundamental theorems on implicitization describe the variety $X = \overline{\mathrm{im}(\rho)}$ with the aid of elimination ideals which therefore are of central interest.

Theorem 1 (Elimination). *Let* $\mathfrak{a} \subseteq \mathbf{k}[t_1, \ldots, t_m, x_1, \ldots, x_n]$ *be an ideal and* G *a Gröbner basis of* $\mathfrak{a}$ *with respect to an elimination order* $t_i >> x_j$*. Then* $G \cap \mathbf{k}[x_1, \ldots, x_n]$ *is a Gröbner basis of* $\mathfrak{a} \cap \mathbf{k}[x_1, \ldots, x_n]$.

Taking into account the existence of algorithms for computing the reduced Gröbner basis out of any ideal basis, the problem of finding the implicit representation of varieties given by

---

[1] Partially supported by the Austrian Science Fund (FWF) under the special research area SFB F013.

rational parametrizations is completely solved. However, computing Gröbner bases for the mentioned ideals is a costly task. This is the reason to consider also different methods of implicitization.

Resultants

Consider positive degrees $d_0, \dots, d_n$ representing the spaces of homogeneous polynomials in $\mathbf{k}[x_0, \dots, x_n]$. For each pair of indices i, α where $0 \le i \le n$ and $|\alpha| = d_i$ we introduce a variable $y_{i,\alpha}$ constructing the polynomial ring $\mathbb{Z}[y_{i,\alpha}]$. If $\Omega \in \mathbb{Z}[y_{i,\alpha}]$ and $F_i = \sum_{|\alpha|=d_i} c_{i,\alpha} x^\alpha$ are homogeneous polynomials in $\mathbf{k}[x_0, \dots, x_n]$ of degree d_i $(0 \le i \le n)$ then $\Omega(F_0, \dots, F_n)$ is the result of replacing in Ω the variable $y_{i,\alpha}$ by the scalar $c_{i,\alpha}$.

Theorem 2. *Let* $\mathbf{K}$ *be algebraically closed and fix positive integers* $d_0, \dots, d_n$. *Then there is a unique polynomial* $\mathrm{Res} \in \mathbb{Z}[y_{i,\alpha}]$ *with the following properties:*

1. *If* $F_0, \dots, F_n \in \mathbf{K}[x_0, \dots, x_n]$ *are homogeneous of degrees* $d_0, \dots, d_n$ *then* $\mathbf{V}(F_0, \dots, F_n) \ne \emptyset$ *if and only if* $\mathrm{Res}(F_0, \dots, F_n) = 0$.
2. $\mathrm{Res}(x_0^{d_0}, \dots, x_n^{d_n}) = 1$.
3. Res *is irreducible in* $\mathbf{K}[y_{i,\alpha}]$.

Roughly speaking the implicit representation of a parametrized variety is given by the resultant of a family of homogeneous polynomials. The problem is that the resultant vanishes identically in case the parametrization has base points. One solution to this problem is given by the following concept.

Moving Varieties

Let X be given in parametrized form by rational functions $\frac{f_1}{f_0}, \dots, \frac{f_r}{f_0}$ where $f_\mu \in \mathbf{k}[s_1, \dots, s_m]$. A moving variety of type d and multi-degree $(\sigma_1, \dots, \sigma_m)$ is a polynomial

$$\sum_{i_1=0}^{\sigma_1} \cdots \sum_{i_m=0}^{\sigma_m} \sum_{|\alpha|=d} A^{\alpha}_{i_1 \dots i_m} x^\alpha s_1^{i_1} \cdots s_m^{i_m} \tag{1}$$

For each fixed value of $s_1, \dots, s_m$ (1) is the implicit representation of a variety in $I\!P^r(K)$. The moving variety is said to follow X if

$$\sum_{i_1=0}^{\sigma_1} \cdots \sum_{i_m=0}^{\sigma_m} \sum_{|\alpha|=d} A^{\alpha}_{i_1 \dots i_m} f^\alpha s_1^{i_1} \cdots s_m^{i_m} = 0. \tag{2}$$

In its simplest instances the method of moving varieties presents itself as a generalization of the classical resultant method.

Algebraic Representation, Expansion and Simplification in Automated Geometric Theorem Proving

Hongbo Li
Institute of Systems Science
Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing 100080, China
Email: `hli@mmrc.iss.ac.cn`

In this talk we consider the problem of multiple algebraic representations and multiple elimination results, and the solution by combining expansion and simplification techniques, in automated geometric theorem proving with Grassmann-Cayley algebra, Clifford algebra and their related bracket algebras.

To better understand the situation of the problem, we divide the content into three parts.

Projective geometry with Grassmann-Cayley algebra and bracket algebra

Multiple algebraic representations usually occur in problems related to conics. The basic constraint that six points in the projective plane are on the same conic has 15 different representations by bracket equalities. In proving a theorem involving such a constraint in either its hypothesis or conclusion, the proving procedure can be made drastically different (either much simplified or much more difficult) by choosing different representations. In general all possible representations of a geometric constraint must be considered, and techniques on optimal selection should be developed.

Multiple elimination results come from different expansions of the same Cayley expression into bracket polynomials. For example, let $\mathbf{a}, \mathbf{b}, \mathbf{c}$ be three points in the projective plane, let $\mathbf{a}$ be the intersection of lines $\mathbf{12}$ and $\mathbf{34}$, let $\mathbf{b}$ be the intersection of lines $\mathbf{1'2'}$ and $\mathbf{3'4'}$, and let $\mathbf{c}$ be the intersection of lines $\mathbf{1''2''}$ and $\mathbf{3''4''}$. In Grassmann-Cayley algebra the constructions of $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are represented by the following Cayley equalities:

$$\mathbf{a} = \mathbf{12} \wedge \mathbf{34}, \quad \mathbf{b} = \mathbf{1'2'} \wedge \mathbf{3'4'}, \quad \mathbf{c} = \mathbf{1''2''} \wedge \mathbf{3''4''}.$$

Here the juxtaposition $\mathbf{12}$ denotes the join of vectors $\mathbf{1}, \mathbf{2}$, and the wedge symbol denotes the meet of two bivectors. If we want to eliminate points $\mathbf{a}, \mathbf{b}, \mathbf{c}$ from the bracket $[\mathbf{abc}]$, we only need to substitute the corresponding Cayley expressions of the three points into the bracket, and we get

$$[\mathbf{abc}] = [(\mathbf{12} \wedge \mathbf{34})(\mathbf{1'2'} \wedge \mathbf{3'4'})(\mathbf{1''2''} \wedge \mathbf{3''4''})].$$

The right side of the equality has 16847 different expansions into bracket polynomials. Choosing different expansions has considerable influence upon the computational complexity of later algebraic manipulations. Thus, techniques on optimal selection of expansions should be developed.

In the polynomial ring of coordinates, simplification techniques of a polynomial expression contain expansions, collections, factorizations, etc. In the bracket ring, similar simplification techniques must be developed in order to carry out efficient computation. The corresponding techniques are called Cayley expansions, contractions and Cayley factorizations respectively.

Affine geometry with affine Grassmann-Cayley algebra, affine bracket algebra and heterogeneous Grassmann-Cayley algebra

The most common model of an affine space is a hyperplane away from the origin in a vector space with one more dimension. The hyperspace composed of all displacement vectors fully characterizes the affine space in that a vector in the surrounding vector space represents an affine point if and only if it does not belong to the hyperspace. Vectors in the hyperspace represents points at infinity of the affine space. In this model we can simply adopt the Grassmann-Cayley algebra and bracket algebra for the projective geometry of all the affine points and the points at infinity, under the modification that a new operator should be developed to distinguish affine points from points at infinity. This operator, known as the boundary operator in elementary topology, is simply a bracket operator in bracket algebra. The modified version of Grassmann-Cayley algebra and bracket algebra are called affine Grassmann-Cayley algebra and affine bracket algebra respectively, and a parallel development of the expansion and simplification techniques is necessary.

By discarding the homogeneous feature of the affine Grassmann-Cayley algebra, we get another version of the algebra, which generalizes the classical Grassmann algebraic representation of affine geometry.

Metric geometry with different Clifford algebras and Clifford bracket algebras

For the same metric geometry, there are usually several models to realize them in inner product spaces. For Euclidean geometry, for example, one of the most useful models is to realize the Euclidean space in the null cone of a Minkowski space with two more dimensions. For each model there is a Clifford algebra generated by the corresponding inner product space. Each algebra presents a different algebraization of the same geometric problem.

The techniques of expansion and simplification can be further developed in Clifford algebra. In fact, they constitute the main contents of the so-called expansion theory of general Clifford algebras and Clifford bracket algebra. Powerful new computational tools for metric geometries can be developed in this way.

In this workshop, we intend to talk about our recent research work on the above three topics: algebraic representation, expansion and simplification in applying Grassmann-Cayley algebra and Clifford algebra in automated geometric theorem proving.

The Nonsolvability by Radicals of Generic 3-connected Planar Graphs

J.C. Owen
D-Cubed Ltd
Park House
Cambridge, CB3 0DU
United Kingdom
Email: `john.owen@d-cubed.co.uk`
S.C. Power
Department of Mathematics and Statistics
Lancaster University
Lancaster, LA1 4YF
United Kingdom
Email: `s.power@lancaster.ac.uk`

A fundamental problem in Computer Aided Design is the formulation of effective approximation schemes or algebraic algorithms which solve for the location of points on a plane, given a set of relative separations (dimensions) between them. For CAD applications, an important class of configurations are those for which the dimensions are just sufficient to ensure that the points are located rigidly with respect to one another. These configurations are known to correspond to maximally independent graphs. A number of algebraic and numerical methods have been proposed for solving these configurations (Owen [4], Bouma et al [1], Light and Gossard [3]) and these have been successfully implemented in CAD programs.

These algebraic methods assemble the solution for complete configurations from the solutions of rigid subcomponents. The assembly process involves only rigid body transformations and the solution of quadratic equations. The simplest subcomponent is a triangle of points and this is solvable by quadratic equations. All the other subcomponents in this process are represented by graphs which are (vertex) 3-connected and so the problem of solving any general configuration is reduced to the problem of solving just those configurations which are represented by 3-connected graphs.

We have previously suggested that with generic dimension values a subcomponent which is represented by a 3-connected graph cannot be solved by quadratic equations (Owen [4]). This would mean that for generic dimensions the existing algebraic methods solve all configurations which can be solved by quadratic equations. Such configurations are also known as "ruler and compass constructible". Gao and Chou [2] have given a procedure for determining in principle if any given configuration is ruler and compass constructible. However their analysis is based on the detail of derived elimination equations and they do not address the problem of solvability for general classes of graphs.

We propose to strengthen the suggestion above to the following conjecture:

Conjecture. A generic configuration of dimensioned points on the plane whose graph is 3-connected is not solvable by radicals.

We have proved the conjecture for the (infinite) class of graphs which have a planar embedding. For this class we establish a reduction scheme through which we can prove that the solution to any rigid 3-connected graph with a planar embedding leads to the solution of configurations with the maximally independent 3-connected graph shown in Figure 1, which we call the doublet.

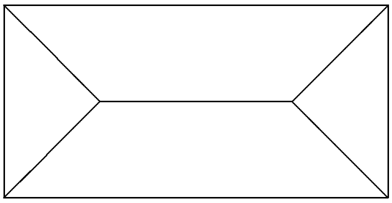


Figure 1.

The proof is both long and eclectic, drawing on new and known results from graph theory, algebraic geometry and Galois theory. Moreover, the final step uses the Maple Mathematical software package to show that there are certain dimension values for which the doublet has non-radical roots. This requires an examination of the generators of single variable elimination ideals which are polynomials of degree 28.

We will give an outline of the main ideas and state the relevant theorems, the proofs of which will appear elsewhere.

References

[1] Bouma W., Fudos I., Hoffmann C., Cai J., Paige R. *A geometric constraint solver,* Computer Aided Design 27 487-501 (1995)

[2] Gao, X.-S., Chou, S.-C., *Solving geometric constraint systems I & II.,* J. CAD

[3] Light R., Gossard D., *Modification of geometric models through variational constraints,* Computer Aided Design 14 209 (1982)

[4] Owen J. C., *Algebraic solution for geometry from dimensional constraints,* In ACM Symposium on Foundations in Solid Modeling, pages 397-407, Austen, Texas (1991)

Function-based Shape Modeling Using a Specialized Language

A. Pasko
Hosei University, Tokyo, Japan
Email: *pasko@k.hosei.ac.jp*
V. Adzhiev
Bournemouth University, Bournemouth, UK

In our talk, we describe the following different aspects of modeling multidimensional point sets (shapes) using real-valued functions of several variables:
- algebraic system as a formal framework
- representation of shapes, operations, and relations using real-valued functions
- internal representation and algorithms
- specialized language for function-based modeling
- extension to point sets with attributes (hypervolumes)
- some applications.

Algebraic system. The concepts of the function representation FRep [1] can be presented as an algebraic system (Objects, Operations, Relations). A complex object is defined as $F(X) \geq 0$, where F is a single continuous function of several variables (coordinates X of a point in a multidimensional space). Operations are unary, binary, and k-ary operations closed on the object representation. Relations as, for example, "interpenetration" are defined on the set of objects using predicates.

Representation. A complex object can be constructed by applying different operations to primitive objects. A primitive is considered a "black box" with the defining function given by a known function evaluation procedure. There is a rich system of operations closed on the representation, i.e., resulting in a continuous real function: set-theoretic operations and Cartesian product defined using R-functions (exact C^k continuous definitions for set operations on functionally defined arguments), blending and bounded blending, offsetting, sweeping, projection, deformation, metamorphosis, and extended space mapping, which combines spatial and functional transformations (mappings). FRep can be considered a combination and generalization of Constructive Solid Geometry (CSG), implicit surfaces, sweeping, and other known shape models.

Internal representation. The real-valued function F defining the point set is associated with a tree structure that serves as its underlying representation. The function F is evaluated at the given point by a procedure traversing a tree structure with primitives in the leaves and operations in the nodes of the tree. In general, a k-ary tree should be supported. Specific details of processing different operations are discussed.

Specialized language. HyperFun is a specialized high-level modeling language suitable for specifying FRep models [2]. While being minimalist, it supports all main notions of FRep.

HyperFun is also intended to serve as a lightweight exchange protocol for FRep models to support platform independence and Internet-based collaborative modeling. A minimal API for interrogating HyperFun models includes the parsing and the function evaluation procedures. The applications developed for HyperFun include a polygonizer (surface mesh generator), plug-in to a ray-tracer, and a set of Web-based modeling tools such as translator to Java, polygonizer in Java, and interactive modeler based on empirical modeling principles and provided as an applet.

Constructive hypervolume model. Multidimensional point sets with multiple attributes (hypervolumes) can be used to model heterogeneous objects with internal distribution of material, density, temperature, and other scalar attributes. FRep was recently applied to define a constructive hypervolume model [3]. The underlying representation can be defined in a similar way by introducing a set of tree structures. Along with the tree corresponding to a function F defining the point set, there are constructive trees associated with functions S_j defining attributes and reflecting the construction logic of the attribute definition. We describe an extension of HyperFun providing for using it to model hypervolumes.

Applications. Main current application areas of FRep and HyperFun include education (geometry and geometric modeling, computer graphics, programming languages), animation and multimedia, and computer art. The constructive hypervolume models can be applied in multiple material rapid prototyping, geological and biological modeling, physics based simulations, and volume graphics. We are also planning to develop an advanced computer-aided design system based on several geometric representations including FRep and the constructive hypervolume model.

References

[1] Pasko A., Adzhiev V., Sourin A., Savchenko V., *Function representation in geometric modeling: concepts, implementation and applications,* The Visual Computer, vol. 11, No. 8, 1995, pp. 429–446.
URL `http://wwwcis.k.hosei.ac.jp/~F-rep/`

[2] V. Adzhiev et al., *HyperFun project: a framework for collaborative multidimensional F-rep modeling,* Implicit Surfaces '99 Workshop (Bordeaux, France), J. Hughes and C. Schlick (Eds.), 1999, pp. 59-69.
URL `http://www.hyperfun.org`

[3] Pasko A., Adzhiev V., Schmitt B., Schlick C., *Constructive hypervolume modeling,* Graphical Models, special issue on Volume Modeling, 64(2), 2002.
URL `http://wwwcis.k.hosei.ac.jp/~F-rep/Fhypervol.html`

Rational Parametrizations of the Minkowski Sum of two Quadrics in 3-space

Martin Peternell
Institute of Geometry, University of Technology Vienna
Wiedner Hauptstraße 8-10, A–1040 Wien, Austria
Email: *martin@geometrie.tuwien.ac.at*

We study the Minkowski sum S of two regular quadrics Q_1, Q_2 in real affine 3-space A^3 and will answer the question, under which conditions the Minkowski sum S admits real rational parametrizations and how they can be computed. A regular quadric Q is given by a quadratic equation

$$X^t M X = 0,$$

where M is a real symmetric 4x4 matrix and $X = (1, x, y, z)^t$ is a vector containing the unknowns x, y, z. It is assumed that M has full rank and that Q possesses real points which implies that Q is either an ellipsoid, a paraboloid or a one or two sheet hyperboloid. It is known that any quadric Q is birationally equivalent to the projective plane P^2. By stereographic projection $\sigma : P^2 \to Q \subset P^3$ it is possible to construct rational parametrizations for Q. Quadrics in affine space A^3 possess same properties and $\sigma : A^2 \to Q \subset A^3$ parametrizes $Q - \{e, f\}$, where e, f are a pair of real or conjugate complex intersecting lines.

The Minkowski sum S of two sets A, B is defined by

$$S = a + b, a \in A \text{ and } b \in B, a, b \in A^3.$$

It is known that if A, B are convex then S is convex too and the boundary of S is given by boundary points $a \in \partial A$ and $b \in \partial B$ whose oriented tangent planes T_a and T_b at a and b respectively, are parallel. If A and B are not convex, the boundary ∂S of S is at least a subset of $\partial A + \partial B$. In the following we assume A and B to be smooth surfaces, in particular quadrics and concentrate on the computation of the Minkowski sum S of these surfaces.

A pointwise construction of S can be obtained as follows: We look for points $a \in A$ and $b \in B$ whose oriented tangent planes T_a, T_b are parallel. Then, at least in the convex case, $s = a + b$ is a point of the boundary surface of the Minkowski sum S and the tangent plane T_s at s is parallel to T_a, T_b.

Our problem can be reformulated as follows: Given two quadrics Q_1, Q_2, is it possible to rationally parametrize both of them in a way that the oriented tangent planes T_{q_1} and T_{q_2} are parallel. More precisely we ask for rational parametrizations $q_1(u, v)$ and $q_2(u, v)$, where q_i parametrizes Q_i, such that the oriented tangent planes to corresponding parameter values (u, v) are parallel. The parameter domain is assumed to be a subset of the affine plane A^2.

At first we notice that if there exists a similarity mapping α (the corresponding linear mapping is a scalar multiple of the identity I) which maps Q_1 onto Q_2, the problem is rather simple since corresponding points $q_1, q_2 = \alpha(q_1)$ with respect to α possess parallel tangent planes T_{q_1}, T_{q_2}. In particular this holds for spheres Q_1, Q_2.

In some kind similar is the case where Q_1, Q_2 are both paraboloids. These are quadrics which are tangent to the plane at infinity in the projective extension P^3 of A^3. It can be shown that the computation of rational parametrizations q_1, q_2 for which the parallelity of T_{q_1} and T_{q_2} holds, is a linear problem.

At second we observe that the problem is invariant with respect to affine transformations. Suppose that Q_2 is an ellipsoid, we assume it to be given by the equation

$$Q_2 : x^2 + y^2 + z^2 - 1 = 0.$$

From a Euclidean point of view Q^2 equals the unit sphere S^2 and the Minkowski sum S is nothing else but the outer offset surface (outer parallel surface) of Q_1. It has been proved earlier that the offset surfaces of quadrics admit rational parametrizations, see [4]. Further, this indicates a kinematic generation of the Minkowski sum S. It has to be noticed that the outer offset S is not a rational surface in the common sense (birationally equivalent to P^2), since its algebraic closure contains in general also the inner offset as a second component. Nevertheless, there exist real rational parametrizations of the outer offset S which shall be called improper, since the inverse mapping is in general no longer rational.

Theorem 1. *The Minkowski sum S of a regular quadric Q_1 and an ellipsoid Q_2 is affinely equivalent to an outer offset of Q_1 and it admits rational parametrizations.*

If Q_1 and Q_2 are both regular ruled quadrics (which contain one dimensional subspaces) it is already proved in [3] that their Minkowski sum $S = Q_1 + Q_2$ admits rational parametrizations. Moreover, this property holds for general rational non developable ruled surfaces Q_1, Q_2.

Finally it shall be proved that the more general statement holds:

Theorem 2. *The Minkowski sum S of two regular quadrics Q_1, Q_2 in affine space A^3 admits rational parametrizations.*

References

[1] Landsmann G., Schicho J., Winkler F., Hillgarter E., *Symbolic Parametrization of Pipe and Canal Surfaces,* Proc. ISSAC–2000, ACM Press, 194–200.

[2] Landsmann G., Schicho J., Winkler F., *The Parametrization of Canal Surfaces and the Decomposition of Polynomials into a Sum of Two Squares,* J. of Symbolic Computation, 32(1-2):119-132, 2001.

[3] Mühlthaler H., Pottmann H., *Computing the Minkowski sum of ruled surfaces,* Technical Report No. xx, Vienna Univ. of Technology, 2002.

[4] Peternell M., Pottmann H., *A Laguerre geometric approach to rational offsets,* Computer Aided Geometric Design 15 (1998), 223–249.

[5] Peternell M., *Rational Parametrizations for Envelopes of Quadric Families,* PhD Thesis, Vienna Univ. Technology, 1997.

[6] Sendra J. R., Sendra J., *Rationality Analysis and Direct Parametrization of Generalized Offsets to Quadrics, Applicable Algebra in Engineering,* Communication and Computing 11(2): 111-139 (2000).

Understanding and Reconstructing Three-dimensional Shapes from Point Clouds

Helmut Pottmann
Institute of Geometry
Vienna University of Technology
Email: *pottmann@geometrie.tuwien.ac.at*

Modern 3D measurement devices (laser range scanners, structured light based measurement,...) produce a large amount of 3D data of geometric objects. These data are more or less structured point clouds. We have a variety of methods for processing these clouds of points: triangulation, mesh decimation, automatic CAD model generation (reverse engineering) through surface fitting [8]. Together with rapid prototyping and 3D printing, there is a complete chain for the emerging area of *3*D technology. For the essential steps such as data acquisition, CAD model building, model modification and printing there are already good solutions on the market.

Whereas the basic concepts and algorithms for 3D Vision and Reverse Engineering of geometric objects are available, the degree of automation and intelligence in the systems still has be increased. A reverse engineering system should not just fit any surface to the data as long as it is within tolerance. For several reasons including functionality and the choice of the right manufacturing tools, it is important to understand the shape, i.e. recognize special shapes, symmetries and other geometric constraints, and build an according CAD model [8].

In the talk, the speaker will survey recent advances on shape understanding and reconstruction and present problems which might be efficiently solvable using techniques developed for automatic deduction in geometry.

A basic entity for shape understanding is the detection of symmetries [3]. The computation of exact symmetries of polyhedra has been treated by several authors. The methods used are those of Computational Geometry; at the heart of most algorithms is a linear time graph isomorphism algorithm [1]. All these results are not efficiently applicable to the decision whether rather dense point sets such as those coming from modern 3D scanners, represent symmetric objects. This is so because of errors in the data, the high number of data points, and the fact that the measurement data do generally not possess the same symmetries as the underlying objects. In a recent contribution to approximate symmetry detection for reverse engineering, Mills at al. [4] treat models with up to about 200 vertices. There seems to be no algorithm which would be capable of detecting symmetries and approximate symmetries of scanned objects at hand of the data point clouds arising from the measurement system.

So far, we have considered discrete symmetries. Objects which are invariant under a continuous group of motions, are general cylinders, surfaces of revolution and helical surfaces. If objects exhibit such surfaces and are just given by a rather dense cloud of measurement points, a solution to the shape detection and reconstruction problem has been found via line geometry. One first estimates surface normals of the data points and then fits a linear line complex to those normals [7]. This approximation problem in line space requires the solution of a general eigenvalue problem. In case that a good fit is possible, the characteristics of the linear complex allow us to compute the kinematic generation of the underlying shape (i.e., the rotational axis for a surface of revolution, helical axis plus pitch for a helical surface, the translational direction in case of a cylinder surface). It is then rather simple to compute a generating profile curve and finally the approximating surface.

We will also discuss approximation in the space of planes and its application to the recognition and reconstruction of planar faces and developable surfaces in point clouds [6].

After one has solved the segmentation problem, i.e. the partition of the point cloud into simpler regions, and one has detected geometric constraints, one still has to find a final approximant which satisfies the detected constraints and fits the data points [2, 5]. This is a combination of a constraint solving problem and an approximation problem.

There seems to be a wide unexplored area of applications for automatic deduction in geometry to the present shape understanding and reconstruction problems. This problem area also illustrates the necessity of combining symbolic and numerical computations.

References

[1] Alt H., Mehlhorn K., Wagener H., Welzl E., *Congruence, similarity, and symmetries of geometric objects,* Discrete & Computational Geometry **3** (1988), 237–256.

[2] Benkö B., Varady T., *Direct segmentation of smooth multiple point regions,* Geometric Modeling and Processing 2002, IEEE Press, pp. 169–178.

[3] Davis L. S., *Understanding shape: II. Symmetry,* IEEE Transactions on Systems, Man and Cybernetics **7** (1977), 204–212.

[4] Mills B. I., Langbein F. C., Marshall A. D., Martin R. R., *Approximate symmetry detection for reverse engineering,* Proceedings Sixth ACM Symposium on Solid Modeling and Applications, Ann Arbor, MI, 2001.

[5] Langbein F. C., Marshall A. D., Martin R. R., *Numerical methods for beautification of reverse engineered models,* Geometric Modeling and Processing 2002, IEEE Press, pp. 159–168.

[6] Peternell M., Pottmann H., *Approximation in the space of planes – Applications to geometric modeling and reverse engineering,* Preprint, Vienna Univ. of Technology, 2001.

[7] Pottmann H., Wallner J., *Computational Line Geometry,* Springer-Verlag, 2001.

[8] Várady T., Benkö P., Kós G., *Reverse engineering regular objects: simple segmentation and surface fitting procedures,* Int. J. Shape Modeling **4** (1998), 127–141.

Geometry Theorem Proving in the Frame of *Theorema* Project

Judit Robu
Babes-Bolyai University Cluj, Romania
Email: `robu@cs.ubbcluj.ro`
RISC-Linz, Austria
Email: `jrobu@risc.uni-linz.ac.at`

During the last few years many algorithms for automated geometry theorem proving appeared and there exist some very good implementations of these algorithms. We present an implementation of some of these algorithms in the frame of the *Theorema* system. *Theorema* isa system that provides a uniform frame for theorem proving in all areas of mathematics, which is programmed in *Mathematica*. *Theorema* is being developed at the RISC Institute by the *Theorema* Group under the direction of Bruno Buchberger. Several examples of geometry proof generated by our implementation are provided. Besides the implementation of known proving methods we also present two new approaches: systematic exploration of geometric configurations and a new method for proving nontrivial geometry theorems involving order relations.

Rational Parametrizations of Curves and Surfaces over Various Fields

Josef Schicho
Research Institute for Symbolic Computation (RISC),
Johannes Kepler University, A-4040 Linz, Austria.
Email: `jschicho@risc.uni-linz.ac.at`

An algebraic curve/surface V can be given by a polynomial equation in two/three variables ("implicit representation"). The parametrization problem asks for a parametric representation, in terms of rational functions in two/three parameters. In other words, we look for a *general solution* of the given algebraic equation, in terms of rational functions.

Such a general solution may not exist; so, a first step in an algorithmic solution is to decide the existence. In the affirmative case, we say that V is *unirational.*

A parametrization which is birational (as a rational map from the line/plane to the curve/surface V) is called *proper.* If p is a proper parametrization of V, then any parametrization of V can be obtained by specializing from p. Thus, a proper parametrization may be also seen as a *most general solution.* The variety V is called *rational* iff it admits a most general solution.

Assume that we have given a parametrization; we ask the question whether we can find a simpler parametrization for the same variety. Obviously, the answer depends on your concept of simplicity.

"Simple" could mean that the degree of the polynomials in the numerator or denominator of the rational functions are small. In the curve case, the degree is minimal precisely for the proper parametrizations. More precisely, this degree is equal to the degree of the curve iff the given parametrization is proper, and it is larger otherwise. So, simplification in this sense is equivalent to properly parametrizing an improperly parametrized curve. This problem is also called the Lüroth problem. Algorithmic solutions have been proposed in [10, 5].

In the surface case, there are proper parametrizations of arbitrarily large degree, for a fixed surface. So, simplification is quite different from the Lüroth problem. If, however, the input parametrization is already proper, we can always find the simplest parametrization by specialization, because we have a most general parametrization.

We give here a new algorithm that produces a reparametrization with degree at most r times the degree of the smallest possible reparametrization, where $r = 5$. The computational cost is a polynomial number of field operations and solutions of univariate equations of polynomial degree, where the measure for the input is the degree of the given parametrization. For the complex case, the algorithm will also be presented in [9]. In this case, the constant r above may be reduced to $r = 2$. Here, we also discuss the necessary adaptions to the case where the ground field is not algebraically closed.

Note that we do not attempt to simplify the coefficient field of the parametrization, as the authors in [1, 12, 2] do for the curve case.

Our problem is similar to the reduction of linear systems of plane curves by Cremona transformations. This problem has been considered by the many authors, see [3, 4, 6]. The main difference to our problem is that there, one attempts to do a reduction by quadratic Cremona transformation, and this gives in turn a proof of the classical result that the Cremona transformations are generated by the quadratic ones (see [11]). Unfortunately, the classical methods work only for linear systems with genus less than or equal to 4 (this corresponds to the simplification of parametrizations of surfaces of sectional genus less than or equal to 4.) Also, the reduction algorithms are quite complicated, and we think that the computational costs would be large.

For arbitrary genus, the first result bounding the degree of the reduced form was given in [8], formulated in the terminology of parametrizations. More precisely, theorem 4 in that paper gives an upper bound for the smallest possible parametrization in terms of the sectional genus. The same paper contains also techniques for finding nontrivial lower bounds for the degree of a parametrization (nontrivial means not the bound that follows immediately by Bezout's theorem), which will be essential for proving the main statement in this paper. However, the proofs in [8] are not constructive.

The main idea for the simplification algorithm in this paper is to simulate the parametrization algorithm [7]. Since we already have a parametrization available, we do not need to resolve the singularity of the surface, and this is the most expensive subtask in [7]. Essentially, the resolution of the singularities can be replaced by the analysis of the base points of the parametrization.

The author has been supported by the Austrian Science Fund (FWF) in the frame of SFB-project F13.

References

[1] Andradas C., Recio T., Sendra R., *Relatively optimal rational space curve reparametrization through canonical divisors,* In Proc. ISSAC 1997 (1997), ACM Press, pp. 349–355.

[2] Andradas C., Recio T., Sendra R., *Base field restriction techniques for parametric curves,* In Proc. ISSAC 1997 (1999), ACM Press.

[3] Castelnuovo G., *Ricerche generali sopra sui sistemi lineari di curve piane,* In Memorie scelte, Zanichelli, 1891, pp. 137–187.

[4] Conforto F., *Le superfici razionali,* Zanichelli, 1939.

[5] Gutierrez J., Rubio R., Sevilla D., *Unirational fields of transcendence degree one and functional decomposition,* In Proc. ISSAC 2001 (2001), ACM Press.

[6] Nagata M., *Rational surfaces I + II,* Mem. Coll. Sci. Kyoto 32 and 33 (1960), 351–370+271–293.

[7] Schicho J., *Rational parametrization of surfaces,* J. Symb. Comp. 26, 1 (1998), 1–30.

[8] Schicho J., *A degree bound for the parameterization of a rational surface,* J. Pure Appl. Alg. 145 (1999), 91–105.

[9] Schicho J., *Simplification of surface parametrizations,* In Proc. ISSAC 2002 (2002), ACM Press. to appear.

[10] Sederberg T. W., *Improperly parametrized rational curves,* Comp. Aided Geom. Design 3 (1986), 67–75.

[11] Shafarevich I. R., Ed., *Algebraic surfaces,* Proc. Steklov Inst. Math., 1965. transl. by AMS 1967.

[12] van Hoeij M., *Rational parametrizations of algebraic curves using canonical divisors,*J. Symb. Comp. 23 (1997), 209–227.

Introducing More Abstract Algebraic Proofs in Projective Geometry

Dana Scott
Carnegie Mellon University Pittsburgh, Pennsylvania, USA
Email: `Dana_Scott@gs2.sp.cs.cmu.edu`

Introduction

On four years (1989, 1993, 1997, 1998), the author offered a one-semester course on the topic of algebraic curves in the classical complex projective plane. The purpose of the present lecture is to review this experience and suggest what might be some desirable future developments.

Algebraic geometry in projective form standardly employs homogeneous polynomials. In the plane, the zeros of a homogeneous **xyz**-polynomial represent an algebraic curve. The dual, **uvw**-polynomials, in a dual set of variables, represent the tangential form of curves, that is to say envelopes. The first question to be answered is how the full impact of duality—familiar from elementary axiomatic projective geometry as the duality between points and lines—can be extended to curves of higher degree and be put in a suitable algebraic for appropriate to the use of symbolic computation.

In order to derive the necessary formulae it is convenient to introduce the ring of ***differential operators***. If we consider the space of all **xyz**-polynomials as a space of (continuous) infinitely differentiable multivariate functions, then we know that on such functions, **f**, the operators of partial differentiation, $\partial/\partial\mathbf{x}$, $\partial/\partial\mathbf{y}$, and $\partial/\partial\mathbf{z}$, which can be written as **D[f, x]**, **D[f, y]**, and **D[f, z]** in *Mathematica* notation, are associative and commutative (under composition of operators). These compositions are also linear operators, and any linear combination of linear operators is again a linear operator. Moreover, composition of linear operators distributes over all linear combinations of operators; hence, as is well known, the differential operators generate a***commutative ring of operators***.

It is thus possible to define a ring homomorphism from the (free) ring of **uvw**-polynomials to these operators; and this homomorphism is actually an isomorphism. In particular, if **f** is an **xyz**-polynomial and **g** is a **uvw**-polynomial of ***lower*** degree, then we can notate by **PD[f, g]** the result of letting **g** operate on **f** as a differential operator by calling up this homomorphism. However, differentiation between **xyz**-polynomials and **uvw**-polynomials can be made ***symmetric*** or ***dual*** in specific sense. Indeed, if **f** is a homogeneous **xyz**-polynomial, and **g** is a homogeneous **uvw**-polynomial, then **PD[f,g]**, an operation the author calls ***polydifferentiation***, may be defined as the result of allowing the polynomial of lower degree to operate on the polynomial of higher degree as a differential operator. In other words, the symbol **u** can be regarded as the operator $\partial/\partial\mathbf{x}$, or the symbol **x** can be regarded as the operator $\partial/\partial\mathbf{u}$, and similarly for other variables. In the case of polynomials of ***equal degree***, by reference to monomials, it is easy to see that in evaluating **PD[f, g]** it makes ***no difference*** which is taken to be the operator and which is the operand.

The ***linear homogeneous*** **xyz**-polynomials, **f**, can be regarded as ***lines***; while the linear **uvw**-polynomials, **g**, can be taken as points (up to a constant, non-zero factor, as with homogeneous coordinates). Then the equation **PD[f, g] == 0** means that the point **g** ***lies***

on the line **f**. (In the linear case, partial differentiation works just like a dot product between vectors.) The equation also means the line **f** ***passes through*** the point **g**, and this remark is the basis for the explanation of the duality between points and lines. What about higher degrees?

We need is a special case of a theorem of Euler which tell us that, if **f** is a homogeneous **xyz**-polynomial of degree **n**, and if **g** is the ***linear form*** **a u + b v + c w**, then we have the equation

PD[f, gn] == n! f[a, b, c],

where **f[a, b, c]** is shorthand for the evaluation of the polynomial as a function of **xyz** at **abc**. Among other consequences, this theorem tells us that the equation **PD[f, gn] == 0** means that the point **g** ***lies on the curve*** **f**. But there are many algebraic consequences.

For example, by a simple use of the Binomial Theorem, we can argue that if **g** represents an **m**-tuple point of the curve **f**, then the polynomial **P[f, gn-m]** of degree **m** factors into the **m** linear factors representing the ***tangents*** to the curve **f** at the point **g**. The trick here is that pairing **PD[f, g]** is a bilinear paring between the vector spaces of **n**-degree homogeneous **xyz**-polynomials and **n**-degree homogeneous**uvw**-polynomials that makes then ***dual vector spaces***. But the polynomials are not just elements of vector spaces, since they are parts of a pair of commutative rings, and **PD** has many simple properties with respect to the ring structure. A suitable axiomatization of these properties makes the ring of **xyz**-polynomials fully dual to the ring of **uvw**-polynomials, and the formal operations—as indicated—can have geometric interpretations.

Mathematica well implements commutative algebra (and many algebraic algorithms) and of course it implements partial differentiation. It does not directly implement a algebra of differential operators, however. The author—with the help of graduate-student assistants—was able to do that by defining the function **DegP** for computing the degree of homogeneous polynomials, and by transcribing the axiomatic characterization of **PD** into rewrite rules between the two kinds of polynomials using *Mathematica* rule sets. Additionally some ideas of exterior algebra had to be represented (on linear forms) and some polynomial invariants using determinants had to be used. Otherwise, polynomial simplification, factorization, and root finding (all built into *Mathematica*) were the only other constructs needed. One big advantage discovered by using symbolic computation in this way was that ***formulae*** for the solution of geometric problems could be developed and then used in numerical computation (for example, in creating graphical illustrations). The improvement in *Mathematica's* ImplicitPlot routine made good pictures of (real parts of) curves possible.

Looking to the future

The author found this approach to classical, plane projective geometry most satisfactory, as the symbolic polynomial algebra really gives ***a middle way*** between synthetic and analytic geometry. Several difficult problems remain to be investigated, nevertheless:

(1) How to extend this approach to fields of ***characteristic*** p? **(2)** How to make good use of symbolic computation with ***algebraic numbers***? **(3)** How to refine the methods to apply to the ***real projective plane***? **(4)** How to move to***higher dimensions*** beyond plane geometry? **(5)** How to make use of modern methods of symbolic computation in ***ideal theory***? **(6)** How to connect with other methods of ***automated theorem proving***?

As regards that last point, *Mathematica*—by doing algebraic transformations and symbolic rewriting—***does*** actually prove theorems; however, these proofs are not fully automatic, as the user of the system has to issue the commands to be carried out. The user also has to supply the interpretation of the results, often resulting in a non-automated search for significant output. Nor does the system formulate conclusions by itself.

Still, the experience in this course development was encouraging in that the algebra generated could not be carried out with any pleasure ***by hand***, and the use of the computer definitely helped the author to ***think***. But, more developments are needed before courseware can be written that fully uses theorem proving.

Rapid Algebraic Resolution of 3D Geometric Constraints and Control of their Consistency

Alex Sosnov
Département Informatique, Ecole des Mines de Nantes, France
Email: `sosnov@emn.fr`
Pierre Macé
Tornado Technologies Europe, Nice, France
Email: `mace@emn.fr`

CAD systems still remain modeling tools instead of being design ones. Thus, conceptual design is mainly done on the paper; once spatial structure of an object is conceived, the designer should again model it with a CAD system and hence loses his time. Therefore, the problem of design really aided by computer is receiving increased attention. The major trend is to improve human-computer interface of CAD systems by automating certain geometric constructions: The designer defines an object by describing its properties and the system infers an appropriate 3D form. The common way to describe properties of an object is to use *3D geometric constraints*, which represent elements composing the object and their relations: incidence, parallelism, orthogonality, distances and so on.

Most geometric constraint solvers use expensive numerical, symbolic or problem decomposition approaches. These solvers work usually in a 3D Euclidean space and hence should treat many complex particular cases of parallelism and orthogonality. However, there exist tools that allow to improve significantly "intelligence" and efficiency of geometric constraint solvers — that are, *geometric algebras*. We present a new method to resolve formally systems of 3D projective, affine and orthogonality constraints as well as detect rapidly logical and numerical contradictions. The method is based on uniform representation of constraints, control of their structural and numerical consistency and their formal resolution with the Grassmann-Cayley geometric algebra.

We represent any 3D object with only low-level elements (*points, lines* and *planes*) and constraints (*collinearity, coplanarity, parallelism* and *orthogonality*). On the other hand, elementary objects can be grouped in primitives of any complexity. Due to usage of the Grassmann-Cayley algebra and projective geometry we can represent elementary objects and constraints in a very simple and homogeneous form. Indeed, points, lines and planes are projective subspaces of 3D projective space. The Grassmann-Cayley algebra provides operators that allow to construct unions, intersections and dual subspaces of such subspaces. We enhance this algebra with the new *orthinf* operator, which constructs for any projective subspace a subspace of its orthogonal directions and hence allows to perform certain *metric* constructions. Thus, we can decompose projective, affine and orthogonality constraints into sets of projectively invariant *incidence* relations and hence represent any scene as a constraint graph, which vertices represent 3D elementary objects while edges represent inferred incidences.

Simplicity of the representation allows to ensure efficiently *structural consistency* of constraints. Indeed, geometry of incidences implies only five logical contradictions, such as a pair of distinct lines incident to the same pair of points. The idea is to prevent creation of contradictory configurations. To achieve it, we use *geometry laws* about incidences and *orthinfs* that allow efficient realization as interrelated *graph update procedures*, which infer automatically new constraints that are consequences of the created ones and applied laws. If such a procedure infers a contradiction, the created constraint is not consistent with already established ones. Thus, we create only *saturated* constraint graphs that contain all the consequences of all the imposed constraints. It allows to improve significantly efficiency of resolution. Furthermore, graph update procedures may be used as a rapid automated geometry theorem prover.

We use a *generic* geometric constraint solver: It first constructs a *formal coordinate-free* solution, which consists of expressions of the Grassmann-Cayley algebra that determine each of the low-level elements composing a 3D object so that constraints imposed on the element would be satisfied. Then, it computes a *numerical solution* from the obtained formal one and particular constraint values — for instance, given spatial coordinates of certain points of the 3D object. Such an approach improves efficiency because it allows to solve entire classes of constraint problems and to recompute rapidly numerical solutions once numerical constraint values are changed. Due to *constructive* nature of the Grassmann-Cayley algebra, we build a formal solution of a constraint problem by very efficient *propagation of known data*. Because we use saturated constraint graphs, multiple local solutions are possible. To choose the best one, we establish the *operation priorities* that ensure minimal computation cost and maximal accuracy of the whole solution. Remaining *alternative local solutions* are used to ensure *numerical consistency* and to recover from precision errors. Furthermore, we can reject many degenerate solutions without any computations.

We determine automatically whether a constraint problem is under- or over-constrained. In the first case, the user is demanded to specify new constraints, or the resolution is stopped. In the second case, consistency of the obtained solution is checked during numerical evaluation.

We compute a numerical solution from the formal one by expressing operations of the Grassmann-Cayley algebra as exterior and matrix products. We perform all the computations in *doubles*, and only if a problem occurs, we use exact arithmetics as backup. It allows to compute efficiently and reliably. We ensure *numerical consistency* of the solution by computing, if necessary, alternative local solutions obtained in the formal resolution phase. Separation of *structural* and *numerical* consistency simplifies detection of logical contradictions in constraints as well as numerical contradictions in their parameters.

We tested successfully our method for constrained geometric constructions in 3D space as well as for rapid reconstruction of 3D scenes from their perspective views and constraints that describe their spatial structure.

Analysis of geometrical theorems in coordinate-free form by using anticommutative Gröbner bases method

Irina J. Tchoupaeva
Moscow, MSU, mech-math
Email: `ladyirina@shade.msu.ru`

In this paper we consider the Gröbner bases of Grassman algebra and its application to the algebraic geometry. Geometrical statements of constructive type should be given in the coordinate-free form.

Implicitization of Geometric Objects under Affine Transformations using Gröbner Walks

Quoc-Nam Tran
Department of Computer Science
Lamar University, TX 77710, U.S.A.
Email: `tranqn@hal.lamar.edu`

In computer aided geometric design, a geometric object is often a combination of several patches. These patches may actually come from a smaller subset of source patches using affine transformations such as rotation, scale and shear. As the result, some algebraic or even transcendental constants such as $\sin(\pi/3)$ and $\cos(\pi/5)$ may appear on the equations, which makes the implicitization much harder.

Usually, parametric forms such as Bezier patches or NURBS are used for designing the objects. But implicit forms are also needed in several occasion such as for finding the intersection of the patches or for finding whether or not a point is on a patch. There are several methods for implicitization such as resultants, Gröbner bases and moving curves/surfaces.

The method of Gröbner bases has the important advantage that they can solve the implicitization problem in full generality. However, Gröbner bases are known to be very slow in implicitizing bicubic patches.

This paper investigate the use of new theoretical results in the method of Gröbner bases over the last five years to improve the efficiency of algorithms for implicitization.

The main result of this paper is that we can dramatically improve the efficiency of implicitization algorithms using the deterministic Gröbner walk conversion. The main ideas are that: first, since the parametric equations of a patch is already or very close to a Gröbner basis with respect to any elimination order for the parameters, one can convert a Gröbner basis by partitioning the computation of the basis into several smaller computations following a path in the Gröbner fan of the ideal generated by the system of equations. The method works with ideals of zero-dimension as well as positive dimension. Typically, the target point of the walking path lies on the intersection of very many cones, which ends up with initial forms of a considerable number of terms and therefore huge intermediate polynomial systems. The deterministic method in [Tra00] to vary the target point can ensure the generality of the position, i.e. we always have just a few terms in the initial forms. Second, if a patch is an image of a known patch under an affine transformation, then one can use the structure of the Gröbner bases to convert from the basis of the known patch rather than computing the Gröbner bases from scratch. This approach, initiated in [Hon97] will speed up the computation, especially when the image patch has algebraic or transcendental constants.

Our experiments show the superlative performance of our improved Gröbner walk method in comparison with the traditional ones. The average performance is 5×10^2 to 10^3 times faster than the direct computation of the reduced Gröbner basis with respect to pure lexicographic term order (using the Buchberger algorithm and the sugar cube strategy).

References

[Hon97] Hong H., *Groebner basis under composition I,* Journal of Symbolic Computation, 11, 1997.

[Tra00] Quoc-Nam Tran, *A fast algorithm for Gröbner basis conversion and its applications,* Journal of Symbolic Computation, 30:451–468, 2000.

GEOTHER 1.1: Handling and Proving Geometric Theorems Automatically

Dongming Wang
Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie – CNRS
4 place Jussieu, F-75252 Paris Cedex 05, France
Email: `Dongming.Wang@lip6.fr`

We refer to the proceedings of the first three International Workshops on Automated Deduction in Geometry (published as LNAI 1360, 1669, and 2061 by Springer-Verlag in 1997, 1999, and 2001 respectively) and the Bibliography on Geometric Reasoning (`http://calfor.lip6.fr/\~{}wang/GRBib`) for the current state-of-the-art on automated theorem proving in geometry. The construction of theorem provers has been a common practice along with the development of effective algorithms on the subject. The GEO - THER environment described in this paper is the outcome of the author's practice for more than a decade. An early version of it was ready for demonstration in 1991, and in 1996 was published a short description of the enhanced version GEO - THER 1.0. The current version GEO - THER 1.1 provides an environment for handling and proving theorems in elementary (and differential) geometry automatically.

In this environment, geometric theorems are represented by means of predicate specifications. The following is a typical example of predicate specification:

```
Simson := Theorem(
         [arbitrary(A,B,C), oncircle(A,B,C,D), perpfoot(D,P,A,B,P),
          perpfoot(D,Q,A,C,Q), perpfoot(D,R,B,C,R)],
          collinear(P,Q,R), [x5, x6, x7, x8, x9] );
```

which will be used throughout the paper for illustration. In general, a geometric theorem specified in GEO - THER has the following form

```
T := Theorem(H,C,X)
```

where `Theorem` is a predicate specially reserved, `T` is the name, `H` the hypothesis and `C` the conclusion of the theorem, and the optional `X` is a list of *dependent* variables.

In order to perform translation, drawing and proving, coordinates have to be assigned to points, so that geometric problems may be solved by using algebraic techniques. In GEO - THER the assignment of coordinates can be done either manually or automatically. The user may also transform the geometric relations (for the hypothesis and conclusion) into algebraic equations manually and provide the set of hypothesis-polynomials to `H` and the conclusion-polynomial or the set of conclusion-polynomials to `C`.

We list some of GEO - THER's capabilities as follows.

- Automatic translation of the predicate specification `T` of a geometric theorem into an English or Chinese statement, into an algebraic specification, or into a logic formula.

- Proving `T` automatically using one of the five provers implemented on the basis of some sophisticated elimination algorithms using characteristic sets, triangular zero decompositions, and Gröbner bases.

- Drawing geometric diagrams automatically from `T`; the drawn diagrams may be modified and animated by mouse click and dragging.

  Automated generation of an HTML file (with Java applet) or a PostScript file, documenting the last manipulations and machine proof of `T`.

- Automated interpretation of the geometric meanings of the produced algebraic nondegeneracy conditions (with respect to `T`), in most cases.

- Searching for and loading the specification of a theorem from the built-in library to the GEO - THER session, automated GEO - THER demonstration, mouse-driving GEO - THER interface, and online help.

The majority of GEO - THER code has been written as Maple programs, and one can use GEO - THER as a standard Maple package. Some of the GEO - THER functions need external programs written in Java (and previously in C) and interact with the operating system. This concerns in particular the functions for automatic generation of diagrams and documents and the graphic interface, which might not work properly under certain operating systems and Java installations. GEO - THER has been included as an application module in the author's Epsilon library which will be made available publicly in later 2002. The reader will find more information about GEO - THER from `http://calfor.lip6.fr/~wang/GEOTHER`.

There are several similar geometric theorem provers implemented on the basis of algebraic methods. As a distinct feature of it compared to other provers, GEO - THER is designed not only for proving geometric theorems but also for handling such theorems automatically. Our design and full implementation of new algorithms for (irreducible) triangular decomposition make GEO - THER's proof engine also more efficient and complete. We shall discuss some implementation strategies and report experimental data on the performance of GEO - THER's algebraic provers in the paper.

Distance Coordinates Used in Geometric Constraint Solving

Lu Yang
Guangzhou University, Guangzhou 510405, China
Chengdu Institute of Computer Applications, Chinese Academy of Sciences
Email: `yanglu36@hotmail.com`

In this paper, an invariant method based on distance geometry is proposed to construct the constraint equations for spatial constraint solving.

A Special Central Configuration

Wu Yuchun
Institute of Systems Science, CAS, Beijing
100080,China
Shi He
Institute of Systems Science, CAS, Beijing
100080,China
Email: `hshi@mmrc.iss.ac.cn`

In this paper, we consider the flat central configurations of bodies using the characteristic set method. We solve a symmetry four bodies problem.

Subresultants and Discriminant Sequences

Zhenbing Zeng
Laboratory for Automated Reasoning and Programming
Chengdu Institute of Computer Applications
Chinese Academy of Sciences
610041 Chengdu, P. R. China
Email: `zeng@mail.sc.cninfo.ne`
Hongguang Fu
Laboratory for Automated Reasoning and Programming
Chengdu Institute of Computer Applications
Chinese Academy of Sciences
610041 Chengdu, P. R. China

Many geometric optimal problems and geometric inequalities can be reduced to the existence of the real roots of the following polynomial

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

with indeterminate coefficients $a_n, a_{n-1}, \cdots, a_1, a_0 \in R[u_1, u_2, \cdots, u_s]$. Some time we need to classify the real roots of this polynomial, that is, give an explicit condition P_k on coefficients for determining if the polynomial has k distinct real roots. When the coefficients are constant numbers, this problem can be easily done using the well-known Sturm's Theorem and Euclidean successive polynomial division. For indeterminant coefficients, the Euclidean successive polynomial division cannot be performed in the coefficient domain. In this case, there are classical works on computing polynomial remainder sequences of two polynomials through subresultant chain (see [1-4], [6] and [7]). This makes it possible to construct a Sturm sequence via the subresultants of a polynomial and its derivative in a recursive way, and then compute the number of real roots for the polynomial by counting the variation in signs of the leading coefficients of the obtained Sturm sequence(see [5] for an example). This process becomes much complicated if the subresultant chain is defective. It would be more convenient if one could simply take the principal subresultants coefficients for counting the variation in signs. This idea is actually feasible. Yang Lu and his cooperators have proved (see [11-12]) that Sturm's theorem can be translated to the principal minor determinants (called "discriminant sequence") of a slightly modified Sylvester resultant matrix by way of the variation in sign of a so-called "revised sign list". This quite surprising result has been found useful in many applications. But the original proof does not show the connection of this property with other already-existed real lgebra results. The goal of this paper is to provide a constructive description to this work in the accepted context of subresultants and polynomial remainder sequences.

Using the standard notation of matrix associated with a sequence of polynomials and the determinant polynomial associated to a matrix, we define the modified subresults (called negativesubresults) for polynomials

$$\begin{aligned} f_1 &= a_{n+1} x^{n+1} + a_n x^n + \cdots + a_0, \\ f_2 &= f_1' = (n+1) a_{n+1} x^n + n a_n x^{n-1} + \cdots + a_1 \end{aligned}$$

and any integer $k, 0 \leq k < n$ with following way:

$$\begin{aligned}
&M_k^* = \max(x^{n-k}f_1, x^{n-k}f_2, x^{n-k-1}f_1, x^{n-k-1}f_2, \cdots, f_1, f_2),\\
&S_k^* = a_{n+1}\mathrm{detpol}(\mathrm{M_k^*}),\\
&S_{n+1}^* = f_1, S_n^* = f_2.
\end{aligned}$$

Then we prove that for any Tarski's remainder sequence $f_1, f_2, \cdots, f_{r-1}, f_r$ of f_1, f_2 with

$$c_i = \deg(f_i, x), \quad c_i = \mathrm{coeff(f_i, x, n_i)}, \quad \mathrm{(i = 1, 2, \cdots, r)}$$

the following two relations between a generalized sturm sequence and negative subresultant chain:

$$\begin{aligned}
S_{n_j}^* S_{n_j-1}^* \quad &\approx f_j f_{j+1},\\
S_{n_j}^* S_{n_{j+1}}^* \quad &\approx (-1)^{\frac{1}{2}(n_j - n_{j+1})(n_j - n_{j+1} - 1)}(c_j c_{j+1})^{n_j - n_{j+1} - 1} f_j f_{j+1},
\end{aligned}$$

hold for $1 \leq j < r$ (if $f_2 = f_1'$) starting from the Fundamental Theorem of p.r.s. This leads to the following result of L. Yang, X. Hou and Z. Zeng:

Theorem 1. *Let f_1 be a polynomial of degree $n+1$ with determined or indetermined real coefficients. Let $S_{n+1}^*, S_n^*, S_{n-1}^*, \cdots, S_0^*$ be the negative subresultant chain generated by $f_1, f_2 = f_1'$, $R_{n+1}^*, R_n^*, R_{n-1}^*, \cdots, R_0^*$ the formal leading coefficients of negative subresultants, $V_{+\infty}$ the modified number of variation in signs of the sequence. Let $N(f_1)$ be the number of the distinctive real roots of f_1, $n_1 = n+1$ and n_r the degree of the last regular negative subresultant. Then*

$$N(f_1) = n_1 - n_r - 2V_{+\infty}.$$

Where the modified number of variation in signs is defined for any real number sequence $a_1, a_2, a_3, \cdots, a_m$ is defined by the following procedure:

```
Let $N = 0$.
For $i$ from 1 to $m - 1$ do
    if $a_i \cdot a_{i+1} < 0$, then $N \leftarrow N + 1$;
    if $a_{i+1} = 0$, and $a_{i+j}$ is the first non-zero number in $a_{i+1}, \cdots, a_m$,
    then $N \leftarrow N + \nu(a_i, a_j; j - i - 1)$ in which
        $\nu(a, b; l) = \frac{1}{2}(l+1) - \frac{1}{2}\sum_{k=1}^{l}(-1)^k - \frac{1}{2}(-1)^{\frac{1}{2}l(l+1)}\mathrm{sign(a \cdot b)}$
        and sign$(\cdot)$ is the sign function.

Return N.
```

In the end of the paper we give an easy-to-use method to calculate the modified number of variation in sign and a Maple program for computing generalized Sturm sequences via negative subresultants.

References

[1] Aubry P., Lazard D., Moreno Maza M., *On the theories of triangular sets,* Journal of Symbilic Computation, Vol. 28, pp. 105–124, 1999.

[2] Brown W. S., Traub J. E., *On Euclid's Algorithm and the Theory of Subresultants,* JACM 18, 505-514(1971).

[3] Brown W. S., *The Subresultant PRS Algorithm,* ACM TOMS 4. 237-249(1978).

[4] Collins G. E., *Polynomial Remainder Sequences and Determinants,* Am. Math. Mon., 73, 708-712(1966).

[5] Collins G. E., *Subresultants and Reduced Polynomial Remainder Sequences,* JACM 14, 128-142(1967).

[6] Fu H., Yang L., Zeng Z., *A recursive algorithm for constructing generalized,* Sturm sequences, Science in China, Ser. E 43:1, 32-41(2000).

[7] Gonzalez-Vega L., Lombardi H., Recio T., Roy M. F., *Sturm-Habicht sequence,* in Proceedings of ISSAC'89, pp. 136-146, New York, ACM Press.

[8] Habicht, W., *Eine Verallgemeinerung des Sturmschen Wurzaehlverfahrens,* Comm. Math. Helvetici 21, 99-116(1948).

[9] Loos R., *Generalized Polynomial Remainder Sequences,* Computing, Supl. 4, 115-137(1982).

[10] Rosset M., *Normalized symmetric functions, Newton's Inequalities, and a New Set of Stronger Inequalities,* Am. Math. Mon., 96, 815-819(1989).

[11] Tarski A., *A decision method for Elementary Algebra and Geometry,* 2nd ed. Berkeley: Uni. Of California Press 1951.

[12] Yang L., Hou X., Zeng Z., *A complete discrimination system for polynomials,* Science in China, Ser. E 39:6, 628-646(1996).

[13] Yang L., *Recent advances on determining the number of real roots of parametric polynomials,* J. Symbolic Computation, 28, 225-242, 1999.

Appendix

A Maple Program for Computing Generalized Sturm Sequences via Negative Subresultants

```
with(linalg):
### matrix associated to a sequence of polynomials
mat := proc(plist, x)
local l, i, p, cf, M;
    l := max(op(map(degree, plist, x)));
    M := [];
    for p in plist do
        cf := [subs(x = 0, p)];
        for i to l do cf := [coeff(p, x, i), op(cf)] od;
        M := [op(M), cf]
    od;
    M
end

### determinant polynomial generated by a matrix
detpol := proc(M, x)
local i, j, k, l, k1, cf, mj;
    k := nops(M);
    l := nops(op(1, M));
    cf := [];
    k1 := seq(i, i = 1 .. k - 1);
    for j from k to l do
        mj := submatrix(M, 1 .. k, [k1, j]);
        cf := [op(cf), det(mj)]
    od;
    mj := 0;
    for i to l - k + 1 do mj := mj + op(i, cf)*x^(l - k + 1 - i)
    od;
    mj
end

### subresultant chain of two polynomials
sres := proc(f, g, x)
local m, n, k, i, p, sk;
    m := degree(f, x);
    n := degree(g, x);
    sk := [];
    for k from 0 to min(m, n) - 1 do
        p := [];
        p := [seq(x^i*g, i = 0 .. m - k - 1),
            seq(x^i*f, i = 0 .. n - k - 1)];
        p := map(collect, p, x);
        sk := [detpol(mat(p, x), x), op(sk)]
    od;
    sk
end

### negative subresultant chain of polynomials f and g with degree(f,x)=degree(g,x)+1
s_res := proc(f, g, x)
local m, n, k, i, p, sk;
    m := degree(f, x);
    n := degree(g, x);
    sk := [];
    for k from 0 to min(m, n) - 1 do
        p := [];
        for i from 0 to max(m, n) - k - 1 do
            p := [x^i*f, x^i*g, op(p)]
        od;
        p := map(collect, p, x);
        sk := [coeff(f, x, m)*detpol(mat(p, x), x), op(sk)]
    od;
    map(primpart, [f, g, op(sk)])
end

### formal leading coefficients of f and diff(f,x)
psc_ := proc(f, x)
local i, n1, sr, r;
    n1 := degree(f, x);
    sr := s_res(f, diff(f, x), x);
    print(sr);
    r := [];
    for i to n1 + 1 do
        r := [op(r), coeff(op(i, sr), x, n1 - i + 1)]
    od;
    r
end

### construct generalized sturm sequence for polynomials with indeterminates coefficients
res_sturm := proc(f, x)
local n1, sn, i, j, j1, r, nj, fj, stm;
    n1 := degree(f, x);
    sn := s_res(f, diff(f, x), x);
    stm := [f, diff(f, x)];
    nj := [];
    for i to n1 + 1 do
        if degree(op(i, sn), x) = n1 - i + 1 then
            nj := [op(nj), n1 - i + 1]
        fi
    od;
    r := nops(nj);
    for j from 3 to r do
        j1 := n1 + 2 - op(j - 1, nj);
        fj := op(j1, sn)*lcoeff(op(j1 - 1, sn), x)*
            lcoeff(op(j - 1, stm), x);
        stm := [op(stm), fj]
    od;
    map(primpart, stm)
end

### an example
> res_sturm(x^5+a*x+b,x);
> [x^5+a*x+b, 5*x^4+a, -4*a*x-5*b, (-256*a^5-3125*b^4)*a^4]
```