MATHEMATICAL TECHNIQUES IN SOLID MODELING

Chandrajit Bajaj

CSD TR-764
April 1988
(Supercedes CSD TR-754)

MATHEMATICAL TECHNIQUES IN SOLID MODELING†

Chanderjit L. Bajaj

Computer Sciences Department
Purdue University
CSD-TR-764
April, 1988

Abstract

Solid Modeling has reached a plateau that cannot be elevated unless a number of basic computational problems in mathematics are solved efficiently and robustly. This paper focuses on these problems and shows how the merging of results from algebra, geometry and approximation theory, into effective tools, shall lead to a higher level of performance in solid modeling.

† Research supported in part by NSF grant MIP 85-21356 and ARO contract DAAG29-85-C-0018 under Cornell/MSI.

MATHEMATICAL TECHNIQUES IN SOLID MODELING†

Chanderjit L. Bajaj
Department of Computer Science
Purdue University

Abstract: *Solid Modeling has reached a plateau that cannot be elevated unless a number of basic computational problems in mathematics are solved efficiently and robustly. This paper focuses on these problems and shows how the merging of results from algebra, geometry and approximation theory, into effective tools, shall lead to a higher level of performance in solid modeling.*

1. INTRODUCTION

Current research in solid modeling has diversified into four interrelated activities‡.

(I) Constructing robust operations for existing solid modelers, which are primarily polyhedral or restricted to a special class of surfaces, [14, 30, 32, 33, 39, 47, 58, 60].

(II) Increasing the geometric coverage to include modeling operations on arbitrary curved surfaces. Extensions to non-rigid surfaces are also considered, [5-9, 11-22, 27-29, 31, 34, 37, 38, 40, 44, 49, 51, 52, 54-57, 63].

(III) Improving the user interface with graphical as well as textual languages for describing, editing and displaying object parts, [13, 22, 37, 50].

(IV) Using solid models in engineering analysis (heat flow, stress analysis) and dynamic simulations (kinematics with interference checking) for design and process verification, [13, 16, 38, 40, 52].

In this paper, we shall only delve into areas (I) and (II) and describe efficient computational techniques that are being developed to eliminate traditional bottlenecks in modeling operations. Solid modeling operations involve problems in solid intersections [14, 21, 22, 44, 51], offset generation [15, 17, 18, 31], topological reconstruction [39, 52], meshing of surface patches [55, 61, 63], interrogation of models [16, 19, 21] and the computation of volumetric

† Research supported in part by NSF grant MIP 85-21356 and ARO contract DAAG29-85-C-0018 under Cornell/MSI.

‡ In citing references, I confess to have at times opted for a more recent paper, containing an appropriate survey and bibliography of previous results.

properties of solids [52, 64]. In designing efficient and robust solutions for these problems, pertaining to solids with curved surfaces, much use is made of algebraic geometry, differential geometry, analysis and approximation theory. In the next section we shall show how effective mathematical techniques such as (i) singularity analysis and resolution, (ii) parameterization and implicitization, (iii) residue computation and chinese remaindering, (iv) evaluation and interpolation, (v) power series computations and localization, and (vi) membership within Ideals (i.e. special sets of polynomials), have had and shall have in the future, a significant impact on solid modeling.

2. MATHEMATICAL TECHNIQUES

2.1 Singularity Analysis and Resolution

A fundamental technique of algebraic geometry has been the resolution of singularities for algebraic curves and surfaces (and in general algebraic varieties). In in its simplest version by resolution is meant the birational transformation (i.e. by an almost one-to-one algebraic transformation) of every irreducible singular algebraic variety V, defined over some ground field K, into nonsingular ones. In the concrete viewpoint of algebraic geometry, a variety is something given by a finite number of polynomials (or power series) in several variables. Further to make a birational transformation means to substitute new variables for the old and see what effect this has on the original polynomials. In other words, the interest is in transforming systems of polynomial equations via simple substitutions into related polynomial systems with no singularities.

The resolution problem has so far been settled affirmatively in the following cases: for curves (dimension one variety), the solution is classical; for surfaces (dimesnion two variety) and K = the field of complex numbers, after several geometric solutions by the Italian geometers such as Albanese, Levi, etc., [65, Chapter 1], the first rigorous solution was given by [62]. Walker's solution is function theoretic and makes use of the local solution (i.e., solution of the local uniformization problem, which is a localized version of the resolution problem) given by Jung in [41]. Then, Zariski introduced the tools of local algebra and valuation theory into algebraic geometry, and thereby obtained a solution to the resolution problem for all varieties upto dimesnion three and K = a field of characteristic zero; at that time he also obtained a local solution for arbitrary dimension varieties and K = a field of characteristic zero. Subsequently, Abhyankar gave a solution for surfaces and K = a perfect field of nonzero characteristic, see [1] for a survey and references. Finally, Hironaka settled the resolution problem for arbitrary dimension varieties and K = field of characteristic zero. Hironaka's solution is especially marked by a vigorous induction and by his ability to deal with several simultaneous equations [36].

In solid modeling operations such as intersection, topological reconstruction, the need for effective singularity analysis is all pervasive and of paramount importance. Self intersections of curved surfaces and space curves give rise to singularities. These singularities occur frequently in practice. For example, when sweeping an object along a space curve, singular surface points are easily generated, [16-18, 31]. When intersecting two objects with curved surfaces, the intersection curves may have singularities even though none of the intersecting surfaces have singular points. Occurring singularities must be determined explicitly, for in the vicinity of a singularity, most algorithms needed to implement modeling operations will fail, [14, 51].

Though much progress has been made towards singularity resolution of varieties in this century, algorithmic procedures for singularity analysis and resolution are scant. Max Noether in 1876 gave a procedure of resolving the singularities of a curve in the complex plane by quadratic transformations. Resolution via quadratic and monoidal transformations is achieved by a process of locally "blowing up" the curve such that the various branches of the curve at the singular point are separated. It is not required that the curve be irreducible, but the branch to be traced must not be multiply contained. Noether's procedure can be generalized to include fields of nonzero characteristic, and also to the "mixed characteristic case", see [3]. Algorithmic proofs have also been given for surfaces in characteristic zero [4], however a truly constructive version yet needs to be developed. An effective singularity analysis and resolution technique also needs to be developed that works directly for space curves rather than for its planar projections [9]. To achieve this one needs the Ideal theoretic methods of section 2.5.

Desingularization yields enough information so that all the singularities of a variety can be analyzed. For curves there can only exist a finite number of point singularities and these can be systematically located together with the number of branches of the curve at each singular point. This in turn coupled with numerical tracing procedures, gives a robust and complete tracing procedure for algebraic curves, see [14]. A constructive version of the curve desingularization theorem has also been effectively used in the topological reconstruction of the offset of nonconvex planar models [15, 19]. Determination and complete analysis of curve singularities also proves essential in computing the parametric equations of curves from their implicit representations [6-9]. More details are given in section 2.2. Surfaces may contain both isolated singular points as well as curves of singular points. A complete analysis of them is more difficult, [65], however an algorithmic analysis procedure similar to curves would allow easy topological reconstruction of solids with singular surface boundaries and remove the usual forced choice of smooth surfaces made by present day solid modelers.

2.2 Birational Mappings

Computing the parametric equations for implicitly defined algebraic curves and surfaces is very attractive in solid modeling, since the parametric form lends itself to curve tracing, generating curves on surfaces, greater ease for transformation and shape control and also for linearly ordering points along a regular segment of a curve on the surface. Recently, various efficient methods have been given for obtaining the parametric equations for special low degree rational algebraic curves and surfaces: degree two and three curves and surfaces [6, 7, 57], the rational space curves arising from the intersection of certain degree two surfaces [44], and the rational space curves arising from the intersection of two rational surfaces [49]. The above rational parametric equations, together with their inverse, form a birational mapping (near one to one) between points on the curve and a line, or alternatively between points of a surface and a plane. Such birational mappings shall increasingly prove useful in solid modeling since they allow the solution of problems on complex geometries to be obtained from simpler ones. The above parameterization algorithms have also been extended to algebraic plane curves of arbitrary degree [8], and as well as the irreducible intersection curve of two algebraic surfaces [9].

For surfaces of degree higher than three, no rational parametric forms exist in general, although parametrizable subclasses can be identified. These methods can be specialized to work over rational or real fields, (both of characteristic 0). It is also important to obtain explicit parametrizations over fields of nonzero characteristic. Consider remark in conjunction with techniques of section 2.3.

Though all algebraic curves have an implicit representation only irreducible algebraic curves with $genus = 0$ are rational, i.e., have a rational parametric representation. Genus, a birational invariant of the curve, measures the deficiency of singularities on the curve from its maximum allowable limit. It is also equal to the topological genus (i.e. the number of handles) when the algebraic curve is viewed as a closed manifold in four dimensional real space. By being able to compute the genus one is able to determine whether a given implicit algebriac curve permits a rational parametric form. A variety of (complicated) algorithms have been presented for computing the genus of algebraic curves: by counting the number of linearly independent differentials of the first kind (without poles) [27], the computation of the Hilbert function [48], and the computation of ramification indices [29]. The method of [8] uses affine quadratic transformations and is noteworthy for its simplicity. For algebraic surfaces there exists a necessary and sufficient condition for rationality, namely, Castelnuovo's criterion : "simultaneous vanishing of the arithmetic genus and the second plurigenus". However a complete algorithmic method for the computation of these genera has yet to be developed.

For rational plane curves, there is also a subclass of polynomially parameterizable curves. Polynomial Parametrization is related to whether the rational curve has one or more places at infinity. Abhyankar has obtained an algorithmic irreducibility criterion for determining when a rational curve has one place at infinity which thereby determines when a rational curve has also a permissible polynomial parametrization, see [5]. The algorithmic irreducible criterion and resultant problem both have a bearing on the Jacobian conjecture in algebraic geometry, see [2]. Thus the interplay between computational algorithms and the underlying mathematics is clearly going to prove mutually beneficial.

In contrast to the parametric form, the implicit form is preferred for testing whether a point is above, on, or below the surface, where above and below is determined relative to the direction of the surface normal. The reverse problem then of converting from parametric to implicit equations for algebraic curves and surfaces, called implicitization is achieved by elimination methods, i.e., the computation of polynomial resultants, see [11, 56]. Efficient computation of the resultant of two polynomials, also known as the Sylvester resultant, has been considered by various authors: for univariate polynomials, [53], for multivariate polynomials [20, 26]. For simultaneously eliminating two variables in three polynomials or in general for eliminating $n-1$ variables from n polynomials, the multivariant resultant is needed [45]. Computing the multivariate resultant by taking the resultant of two polynomials at a time leads to extraneous factors that cannot be avoided. In practice, this means that the resulting implicit form describes not only the parametric surface, but in addition, other surfaces. The efficient computation of the multivariate resultant has yet to be undertaken. The multivariate resultant proves useful in deriving the implicit equation of a parametric surface without extraneous factors, for computing the inverse formulas for 1–1 rational mappings, the convolution of algebraic curves and surfaces, the common intersection points of three surfaces, etc.

2.3 Modular Techniques

In interrogating or manipulating solid models with algebraic curve and surface boundaries one is essentially reduced to finding the solution of systems of polynomial equations. This can be achieved by computing polynomial resultants and polynomial GCD's. For such applications amongst others, as we shall discuss below, it is convenient to do efficient integer and polynomial arithmetic in "modular" form, i.e., over fields of prime characteristic $p \neq 0$. That is, instead of representing an integer by a fixed radix notation, we represent the integer by its residues modulo a set of primes. If $p_1, p_2, \ldots, p_r$ are prime integers and $p = p_1 p_2 \cdots p_r$, then we can represent any integer q, $0 \leq q < p$, uniquely by the set of residues $q_1, q_2, \ldots, q_r$ where $q_i = q$ modulo p_i, for $1 \leq i \leq r$. Resultants analogous to those for integers, hold for polynomials. Let $f_1, \ldots, f_r$ be (univariate) irreducible polynomials and let $f = f_1 f_2 \cdots f_r$ and $d = d_1 + d_2 + \cdots + d_r$ where $d_i = \deg f_i$ = degree of f_i. Then each polynomial g with $\deg g < d$, can be represented uniquely by the sequence $g_1, g_2, \ldots, g_r$ or remainders obtained by dividing g by each f_i. That is, g_i is the unique polynomial with $\deg g_i < d_i$ such that $g = f_i s_i + g_i$ for some polynomial s_i. We write $g_i = g$ modulo f_i, in complete analogy with integer modular arithmetic.

The advantage of modular representation is chiefly that substantial portions of the arithmetic can be done simultaneously, since calculations are done for each modulus independently of the others. On parallel computers, they allow many operations to take place simultaneously resulting in substantial speed increases. Modular arithmetic can be a significant advantage even for addition, subtraction and multiplication. The same kind of decrease in execution time could not be achieved by conventional techniques, since carry propagation must be considered. "Real time" calculations make the inherent parallelism of modular arithmetic even more significant. Further decreases in computation time result from the bound on the size of integers. In a field with prime characteristic p the only integers allowed are less than p. Finally, the modular technique (treating the given floating point coefficients as exact rational numbers) gives a method for obtaining 'true' answers in less time than conventional methods can produce reliable "approximate" answers.

The disadvantage of "modular" representations is that it is comparatively difficult to test whether or not one number is greater than another. It is also difficult to test whether or not overflow has occurred as the result of an addition, subtraction or multiplication. The use of modular representations is, however, justified when fast means of conversion into and out of modular representation are available.

Thus in order to use modular arithmetic, algorithms are needed to convert from radix notation to modular notation and back. To go from radix notation to modular is easy as this entails computing a number of residues (modulus) with respect to primes. For polynomials one computes the residue of each of the polynomial coefficients.

The problem of converting from modular notation to radix notation requires a process known as Chinese remaindering. Suppose we are given relatively prime moduli p_1, $p_2, \ldots, p_r$ and the residues $q_1, q_2, \ldots, q_r$ and we wish to find the integer q such that q corresponds to $(q_1, q_2, \ldots, q_r)$. We may do so by the integer analog of the Lagrangian interpolation formula for polynomials. Let y_i be the product of all the p_j's except p_i (that is $y_i = p/p_i$ where $p = p_1 p_2 \cdots p_r$). Let $z_i \equiv 1/y_i$ modulo p_i (that is $y_i z_j \equiv 1$ modulo p_i and $0 \le z_i < p_i$). Then $q \equiv y_1 z_1 q_1 + y_2 z_2 q_2 + \cdots + y_r z_r q_r$ modulo p. The above holds for polynomials modulo $f_1, f_2, \ldots, f_r$ as it does for integers. Suppose $f_1(x), f_2(x), \ldots, f_r(x)$ are pairwise coprime univariate polynomials. Let $d = d_1 + \cdots + d_r$ where $d_i = \deg f_i(x)$ = degree of $f_i(x)$. Then given any polynomials $g_1(x), g_2(x), \ldots, g_r(x)$ where $\deg g_i(x) < d_i$ for $1 \le i \le r$, there exists an algorithm to compute the unique polynomial $g(x)$ of degree less than d such that $g(x)$ corresponds to $(g_1(x), g_2(x), \ldots, g_r(x))$. For details see [10, 42].

An important special case occurs when all the moduli have degree 1. If $f_i = x - a_i$ for $1 \le i \le r$, then the residue (the g_i's) are constants. If $g(x) \equiv g_i$ modulo $(x - a_i)$ then $g(x) = s(x)\,(x - a_i) + g_i$. Hence $g(a_i) = g_i$. Thus the unique polynomial of degree $< d$ such that $g(x)$ corresponds to $(g_1, g_2, \ldots, g_r)$ is the unique polynomial of degree $< d$ such that $g(a_i) = g_i$ for each i, $1 \le i \le r$, i.e., the *interpolating* polynomial. We can thus do polynomial arithmetic such as addition, subtraction and multiplication by evaluating polynomials at n points, performing the arithmetic on the values at these points and then interpolating a polynomial through the resulting values. If the answer is a polynomial of degree $d - 1$ or less, this technique will yield the correct answer.

An efficient "modular" algorithm is possible for the exact calculations of the Sylvester resultant of two multivariates polynomials with integer coefficients, see [26]. The algorithm applies modular homomorphisms and the Chinese remainder theorem, evaluation homomorphisms and interpolation, in reducing the problem to resultant calculation for univariate polynomials over finite fields of prime characteristic p, viz., GF(p), whereupon a polynomial remainder sequence is used.

The modular homomorphisms allow computations to occur with polynomials over GF(p_i) with reduced coefficients restricted to be less than various prime moduli p_i. By the Chinese remainder theorem, if the computations are performed for sufficiently many prime moduli, the actual resultant can be computed. The number of moduli needed depends on a bound for the coefficients of the resultant which can be easily determined. Evaluation homomorphisms allow specializations of multivariate polynomials with the final resultant being computed by interpolation. Here an easily determined bounded m on the degree of the final resultant allows for unique interpolation from the computations on $m + 1$ different specializations. See also [20] where modular techniques together with efficient divide and conquer methods for resultants of univariate polynomials yields an asymptotically faster algorithm.

Similar techniques as above can also be used to calculate the GCD of polynomials: GCD $[u(x), v(x)] = w(x)$. If the bound on the coefficients is so large that single-precision primes p are insufficient, we can compute $w(x)$ modulo several primes p until it has been determined via the Chinese remainder algorithm, see [23]. Alternatively, as suggested by Moses and Yun (see [42]), we can use Hensel's method [35], to determine $w(x)$ module p^e for sufficiently large e. Hensel's construction appears computationally superior to the Chinese remainder approach, but is valid only when either GCD $[w(x), \frac{u(x)}{w(x)}] = 1$ or GCD $[w(x), \frac{v(x)}{w(x)}] = 1$.

2.4 Power Series and Localizations

Polynomials and rational functions describe the global geometry of curves and surfaces. On the other hand, power series capture the local phenomenon. The use of power series in solid modeling has seen only limited use, see [14, 28], for adaptive step size selection for curve tracing methods. Power series, on the other hand, have been extensively studied in computer algebra, where efficient algorithms have been defined the addition, multiplication, division and reversion of power series, see [43]. Their application in solid modeling is only a matter of time.

The use of power series in solid modeling arises whenever local properties need to be computed, such as the behaviour of a curve at a singular point or the behaviour of a surface along a singular curve or at a singular point. Power Series expansions are possible in singular neighborhoods and in fact give a complete description of the various branches meeting together at the singularity. This then together with techniques for computing Pade' rational function approximants, can be used to provide efficient techniques for computing approximate implicit and parametric representations of local neighborhood of curves around simple and singular points, see [12].

Along with power series comes the power of localizations or *local rings*, which allow two surfaces meeting along a space curve to be viewed locally as two curves meeting at a point. Localizations have seen scant use so far, however with their inherent power of simplification they should prove very useful in solid modeling. Especially for the difficult and cumbersome task of analysing singular surface intersections.

2.5 Ideal Theoretic Methods

While plane curves and surfaces are represented by a single implicit equation, space curves require two or more. In the parametric definition in fact, there are no exceptions, all curves and surfaces require more than a single equation. In dealing with implicitly defined space curves or parametric curves and surfaces and applying operations on them, the usual equational methods are at times inappropriate and cumbersome. Here ideal theoretic methods which work directly with the ideals of geometric entities prove useful, see [24, 59]. These methods provide constructive solutions to many problems dealing simultaneously with two or more equations by reducing them to determining the membership of a certain polynomial in the Ideal of other polynomials. The impact of these methods on solid modeling has been limited because of their extremely prohibitive worst case behaviour [46]. However the practical significance of these powerful techniques for simple cases in solid modeling, remains to be fully explored. More research is needed to investigate the efficiency of these algorithms in the specific contexts of problems such as: analysis and resolution of singularities, implicitization, inversion of birational mappings, etc.

2.6 Approximations

The efficiency of almost all computational methods for problems dealing with curves and surfaces depend primarily on the algebraic degree of the equation being manipulated. Using lower degree surface approximations for the higher degree surfaces generated, e.g., for complex blending surfaces, is therefore a very attractive possibility that must be explored. In such an approach to modeling, one chooses a family of low degree rational algebraic surfaces that give sufficient flexibility in controlling shape so as to enable close approximations of high degree surfaces. Choosing a good family of approximating algebraic surfaces requires extensive experimentation and good graphics tools. It also requires research to find efficient computational methods for obtaining close approximations by rational parametric patches. Rational parametric surfaces represent a wider class of algebraic surfaces than those represented by polynomial parametric patches. Most of the work in the past has focussed on the approximation of functions, or the approximation of curves and surfaces by polynomial parametric patches, see [21]. Work reported by [54, 55, 61, 63], amongst others, which apply approximation, meshing and interpolation techniques directly to curves and surfaces in their implicit form, needs to be further pursued.

3. CONCLUSIONS

We have considered a spectrum of mathematical techniques and tried to indicate their usefulness in the various operations in solid modeling. Much more remains to be researched. Two major omissions in this paper of relevant mathematical areas are those of differential geometry and probability theory. Differential geometry essential concerns itselfs with local invariants of analytic varieties and consequently their techniques prove useful in constructive algorithms for convex decompositions of curves and surfaces, computing principal and geodesic curvature lines on surfaces, etc. Probabilistic computations on the other hand have yielded fast algorithms modulo some insignificant yet controllable error probability, for various verification and interrogation problems see [25, 53]. Their application to speeding up solid modeling operations remains one of the more significant open areas.

ACKNOWLEDGEMENTS:

The author has benefitted much from vigorous discussions on related topics with his fellow researchers: S. Abhyankar, C. Hoffmann, J. Hopcroft, E. Houstis, T. Korb and J. Rice.

4. REFERENCES

[1] Abhyankar, S., (1968)
On the problem of resolution of singularities, *Proceedings of the International Congress of Mathematics*, Moscow.

[2] Abhyankar, S., (1976)
Expansion Techniques in Algebraic Geometry, Tata Institute, Lecture Notes, Bombay.

[3] Abhyankar, S., (1983)
Desingularization of plane curves, *American Mathematical Society Proceedings of the Symposia in Pure Mathematics*, Vol. 40, Part 1, pp. 1–45.

[4] Abhyankar, S., (1988)
Good points of a hypersurface, *Advances in Mathematics*, (in press).

[5] Abhyankar, S., (1988)
The difference between a parabola and a hyperbola, *Mathematical Intellingencer*, (in press).

[6] Abhyankar, S. and Bajaj, C., (1987a),
Automatic Parameterization of Rational Curves and Surfaces I: Conics and Conicoids, *Computer Aided Design*, 19, 1, 11–14.

[7] Abhyankar, S. and Bajaj, C., (1987b),
Automatic Parameterization of Rational Curves and Surfaces II: Cubics and Cubicoids, *Computer Aided Design*, 19, 9, 499–502.

[8] Abhyankar, S. and Bajaj, C., (1987c),
Automatic Parameterization of Rational Curves and Surfaces III: Algebraic Plane Curves, *Computer Aided Geometric Design* (in press).

[9] Abhyankar, S. and Bajaj, C., (1987d),
Automatic Parameterization of Rational Curves and Surfaces IV: Algebraic Space Curves, Computer Science Technical Report, CSD-TR-703, Purdue University.

[10] Aho, A., Hopcroft, J. and Ullman, J., (1974)
The Design and Analysis of Algorithms, Addison Wesley, Reading, MA.

[11] Bajaj, C., (1987),
Algorithmic Implicitization of Algebraic Curves and Surfaces, Computer Science Technical Report, CSD-TR-697, Purdue University.

[12] Bajaj, C., (1988),
Approximate Implicitization and Parameterization of Algebraic Curves Manuscript.

[13] Bajaj, C., Dyksen, W., Hoffmann, C., Houstis, E., Korb, T., and Rice, J., (1988),
Computing About Physical Objects, Computer Science Technical Report, CSD-TR-696, Purdue University.

[14] Bajaj, C., Hoffmann, C., Hopcroft, J., and Lynch, R., (1988),
Tracing Surface Intersections, *Computer Aided Geometric Design* (in press).

[15] Bajaj, C. and Kim, M., (1987a)
Generation of Configuration Space Obstacles : The Case of Moving Algebraic Curves, *Proc of 1987 IEEE Conference on Robotics and Automation*, Raleigh, North Carolina, 979–984. Updated version to appear in *Algorithmica*.

[16] Bajaj, C. and Kim, M., (1987b)
Compliant Motion Planning With Geometric Models, *Proc. of the Third ACM Symposium on Computational Geometry*, Waterloo, Canada, 171–180.

[17] Bajaj, C. and Kim, M., (1988a)
Generation of Configuration Space Obstacles : The Case of Moving Spheres, *IEEE Journal of Robotics and Automation*, 4, 1, 94-99.

[18] Bajaj, C. and Kim, M., (1988b)
Generation of Configuration Space Obstacles : The Case of Moving Algebraic Surfaces, *International Journal of Robotics Research*, (in press).

[19] Bajaj, C. and Kim, M., (1988c)
Algorithms for Planar Geometric Models, (Proc of the Fifteenth Intl. Colloquium on Automata, Languages and Programming, ICALP 88), *Lecture Notes in Computer Science*, Springer Verlag, to appear.

[20] Bajaj, C. and Royappa, A., (1987)
On an Efficient Implementation of Sylvester's Resultant for Multivariate Polynomials, Computer Science Technical Report, CSD-TR-718, Purdue University.

[21] Boehm, W., Farin, A. and Kahmann, J., (1984),
A Survey of Curve and Surface Methods in CAGD, *Computer Aided Geometric Design*, 1, 1–60.

[22] Brown, C.M., (1982),
PADL-2: A Technical Summary, *IEEE Computer Graphics and Applications*, 2, 69–84.

[23] Brown, W., (1971)
On Euclid's algorithm and the computation of polynomial GCD, *Journal of the ACM*, Vol. 18, pp. 478–504.

[24] Buchberger, B., (1984),
Grobner Bases: An Algorithmic Method in Polynomial Ideal Theory, In: *Recent Trends in Multidimensional System Theory*, N. Bose (eds.), Reidel.

[25] Clarkson, K., (1988)
Applications of Random Sampling in Computational Geometry II, *Proc. of the Fourth ACM Symposium on Computational Geometry*, Urbana, Illinois, to appear.

[26] Collins, G., (1971),
The Calculation of Multivariate Polynomial Resultants, *Journal of the ACM*, 18, 4, 515–532.

[27] Davenport, J., (1979),
The Computerization of Algebraic Geometry, *Proc. of Intl. Symposium on Symbolic and Algebraic Computation*, EUROSAM'79 Lecture Notes in Computer Science, Springer-Verlag, 72, 119–133.

[28] de Montaudouin, Y. and Tiller, W. and Vold, H., (1984),
Application of Power Series in Computational Geometry, *Computer Aided Design*, 18, 10, 514–524.

[29] Dicrescenzo, C. and Duval, D., (1984),
Computations on Curves, *Proc. of Intl. Symposium on Symbolic and Algebraic Computation*, EUROSAM'84 Lecture Notes in Computer Science, Springer-Verlag, 174, 100–107.

[30] Edelsbrunner, H. and Muecke, E., (1988),
Simulation of Simplicity : A technique to cope with degenerate cases in geometric algorithms, *Proc. of the Fourth ACM Symposium on Computational Geometry*, Urbana, Illinois, to appear.

[31] Farouki, R., (1985),
Exact offset procedures for simple solids, *Computer Aided Geometric Design* Vol. 2, pp. 257–279. 33, 209–236.

[32] Farouki, R. and Rajan, V., (1987),
On the Numerical Condition of Algebraic Curves and Surfaces, Manuscript.

[33] Greene, D., and Yao, F., (1986)
Finite Resolution Computational Geometry, *Proc. of the 27th Annual Conference on Foundations of Computer Science*, 143-152.

[34] Goldman, R., (1986),
The Role of Surfaces in Solid Modeling, *Geometric Modeling*, (G. Farin, ed.), SIAM, 69–90.

[35] Hensel, K., (1908)
Theorie der Algebraischen Zahlen, Teubner, Leipzig.

[36] Hironaka, H., (1964)
Resolution of singularities of an algebraic variety over a field of characteristic zero, *Annals of Mathematics*, Vol. 79, pp. 109–326.

[37] Hoffmann, C. and Hopcroft, J., (1985),
Automatic Surface Generation in Computer Aided Design, *The Visual Computer*, 1, 92–100.

[38] Hoffmann, C. and Hopcroft, (1987),
Simulation of Physical Systems from Geometric Models, *IEEE J. of Robotics and Automation,June, 194-206.*

[39] Hoffmann, C., Hopcroft, J., and Karasick, M., (1988)
Towards Implementing Robust Geometric Computations, *Proc. of the Fourth ACM Symposium on Computational Geometry*, Urbana, Illinois, to appear.

[40] Hopcroft, J., (1986),
The Impact of Robotics on Computer Science, *Communications of the ACM*, 29, 6, 486–498.

[41] Jung, H.W.E., (1908)
Darstellung der Functionenen eines algebraischen Koerpers zweier unabhaengingen Veraenderlichen *x,y der Umgebung einer Stelle*, *Journal fuer die Reine and Angewandte Mathematik*, Vol. 133, pp. 289–318.

[42] Knuth, D., (1981)
The Art of Computer Programming, II: Seminumerical Algorithms, Addison Wesley, Reading, MA.

[43] Kung, H. and Traub, J., (1978)
All Algebraic Functions can be Computed Fast, *Journal of the ACM*, 25, 245–260.

[44] Levin, J., (1979),
Mathematical Models for Determining the Intersections of Quadratic Surfaces, *Computer Graphics and Image Processing*, 11, 73–87.

[45] Macaulay, F., (1916)
The Algebraic Theory of Modular Systems, Cambridge University Press, London.

[46] Mayr, E. and Meyer, A., (1982)
The complexity of the word problem for commutative semi-groups and polynomial ideals, *Advances in Mathematics*, Vol. 46, pp. 305–329.

[47] Milenkovic, V., (1986),
Verifiable Implementation of Geometric Algorithms Using Finite Precision Arithmetic, *International Workshop on Geometric Reasoning*, Oxford, England.

[48] Mora, F. and Moller, H., (1983),
Computations of the Hilbert Function, *Proc. of European Computer Algebra Conference*, EUROCAL'83 Lecture Notes in Computer Science, Springer-Verlag, 162, 157–167.

[49] Ocken, Schwartz, J. and Sharir, M., (1986),
Precise Implementation of CAD Primitives Using Rational Parameterization of Stanford Surfaces, *Planning, Geometry and Complexity of Robot Motion*, (Schwartz, Sharir, Hopcroft, eds.), Chapter 10, 245–266.

[50] Poppelstone, R., Ambler, A. and Bellos, I., (1980),
An Interpreter for a Language Describing Assemblies, *Artificial Intelligence*, 14, 79–107.

[51] Pratt, M., and Geisow, A., (1986)
Surface/surface intersection problems, *The Mathematics of Surfaces*, edited by J. Gregory, pp. 117–142, Oxford University Press.

[52] Requicha, A. and Voelcker, H., (1983),
Solid Modeling: Current Status and Research Directions, *IEEE Computer Graphics and Applications*, 25–37.

[53] Schwartz, J., (1980),
Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of the ACM*, 27, 4, 701–717.

[54] Sederberg, T., (1984),
Piecewise Algebraic Curves, *Computer Aided Geometric Design*, 1, 241–255.

[55] Sederberg, T., (1985),
Piecewise Algebraic Surface Patches, *Computer Aided Geometric Design*, 2, 53–59.

[56] Sederberg, T., Anderson, D. and Goldman, R., (1984),
Implicit Representation of Parametric Curves and Surfaces, *Computer Vision, Graphics and Image Processing*, 28, 72–74

[57] Sederberg, T., and Snively, J., (1987),
Parametrization of Cubic Algebraic Surfaces, Manuscript.

[58] Segal, M. and Sequin, C., (1985),
Consistent Calculations for Solid Modeling, *Proc. of the First ACM Symposium on Computational Geometry*, Baltimore, Maryland, 29-38.

[59] Seidenberg, A., (1974)
Constructions in Algebra, *Trans. Amer. Math. Soc.*, 197, 273–313.

[60] Sugihara, K., (1987),
On Finite Precision Representations of Geometric Objects, Dept. of Mathematical Engg., and Instrumentation Physics, Research Memo RMI-87-06, Tokyo University.

[61] Wachpress, E., (1975),
A Rational Finite Element Basis, Academic Press.

[62] Walker, R.J., (1935)
Reduction of singularities of an algebraic surface, Annals of Mathematics, Vol. 36, pp. 336–365.

[63] Warren, J., (1975),
On Algebraic Surfaces Meeting with Geometric Continuity, Ph.D. Thesis, Cornell University

[64] Wu, M., Bajaj, C., and Liu, R., (1988),
A Face Area Evaluation Algorithm for Solids, *Computer Aided Design*, 20, 2, 75 - 82.

[65] Zariski, O., (1935)
Algebraic Surfaces, Ergebnisse der Mathematik und ihre Grenzgebiete, Vol. 4.