Mathematics and Computer Aided Design: A Personal Retrospective

Pierre J. Malraison

Spatial, Inc.

Abstract

The types of mathematics used in Computer Aided Design cover a wide range. This paper will look at some that I have been involved with over the past twenty years.

- Algebraic geometry, in the context of surface intersection
- Geometric semantics
- Graph theory and analytic geometry in the context of geometric constraints
- Splines, numerical techniques, and analytic geometry in construction of n-sided surfaces

1 Introduction

> "All men of whatsoever quality they be, who have done anything of excellence, or which may properly resemble excellence, ought, if they are persons of truth and honesty, to describe their life with their own hand; but they ought not to attempt so fine an enterprise till they have passed the age of forty."
>
> - *Autobiography of Benvenuto Cellini*

I am not as ambitious (nor as conceited) as Benvenuto Cellini, but writing this paper at the end of the millennium, I have picked as my topic the various types of mathematics I have encountered in my own career in computer-aided design.

The objective of this paper is to present some of the diferent types of mathematics which have proven useful in computer-aided design. In many cases results from the nineteenth century and from disciplines which are no longer studied in the general mathematical curriculum have proven valuable for CAD.

2 Quadric Intersections

> "Mathematicians are a species of Frenchmen: if you say something to them they translate it into their own language and presto! It is something entirely different" - *Goethe*

One of the earliest applications of algebraic geometry to CAD was Levin's work [5] using the representation of quadric surfaces as 4×4 matrices to derive parameterizations for intersection curves. More details and more examples can be found in [7].

The work starts from a representation of a quadric surface by a matrix. Let $X = (x, y, z, 1)$ be the homogeneous coordinates for a point in R^3. Then we can write the equation of a quadric as $X\ Q\ X^T = 0$ where Q is a 4×4 symmetric matrix. For example, for a sphere of radius r centered at the origin,

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & r^2 \end{pmatrix}$$

So if X lies on the intersection of Q_1 and Q_2 then $X\ Q_1X^T = 0$ and $X\ Q_2X^T = 0$ and further for any λ $X\ Q_1 + \lambda Q_2X^T = 0$. So a judicious choice of λ will provide a simple surface or a least a ruled surface which provide a nice domain for parameterizing the intersection curve.

Continuing with the example from above, let

$$Q_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Q_0 is a cone $x^2 + y^2 = z^2$.

$$Q + \lambda Q_0 = \begin{pmatrix} 1+\lambda & 0 & 0 & 0 \\ 0 & 1+\lambda & 0 & 0 \\ 0 & 0 & 1-\lambda & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and when $\lambda = -1$ we obtain the two planes $2z^2 = r^2$, and the intersection curves: two disjoint circles.

When we move this technique into the "real world" two problems arise. First, consider the quadric:

$$Q_0 = \begin{pmatrix} 1+\epsilon_1 & 0 & 0 & 0 \\ 0 & 1+\epsilon_2 & 0 & 0 \\ 0 & 0 & 1++\epsilon_3 & 0 \\ 0 & 0 & 0 & r^2 \end{pmatrix}$$

This is an ellipsoid. But, is it *really* an ellipsoid, or just a sphere whose values have been perturbed a little? In [4] the answer suggested was that we cannot tell, but it doesn't matter. In production limiting surface types to the "natural" quadrics (plane, sphere,cylinder, cone) should suffice.

The second problem similarly involves numerical issues but in a more fundamental problem. Solid models link topology and geometry. In a computer, topology is logical: this edge is or is not adjacent to that edge; but geometry is numerical: this point is (0.0,0.0,0.0) within a given tolerance. Now look at what happens to our sphere-cone intersection as we move the center of the sphere off the apex of the cone:

$$Q_2 = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & 0 & 0 & r^2 \end{pmatrix}$$

$$Q_2 + \lambda Q_0 = \begin{pmatrix} 1+\lambda & 0 & 0 & a \\ 0 & 1+\lambda & 0 & 0 \\ 0 & 0 & -\lambda & 0 \\ a & 0 & 0 & 0 \end{pmatrix}$$

When $a = r$ we get a figure-8 sort of curve, when a is close to zero we get two disjoint curves. When a is close to r, we get two disjoint curves which might be within tolerance of looking like a figure-8. The solution here is to analyze the possible states, and force the cases which are within tolerance of topological boundary states (like the figure 8) to be treated as though they were on the boundary.

A simple example of a state diagram is for two spheres, say S_1 with radius R_1 and S_2 with radius R_2. There is only one parameter: the distance between the spheres. Assume S_1 is centered at the origin. Tangency occurs when the distance between the centers is $R_1 + R_2$ and the state diagram looks like:

Variable is distance between centers

———————— ○ ————————

2 *circles* *tangent* *no intersection*

3 Geometric Semantics

> "I want to suggest to you today, that unless we have a tolerant attitude toward mistakes - I might almost say 'a positive attitude toward them' - we shall be behaving irrationally, unscientifically, and unsuccessfully. Now, of course, if you now say to me, 'Look here, you weird Limey, are you seriously advocating relaunching the Edsel?' I will reply, " 'No.' There are mistakes - and mistakes. There are true, copper-bottom mistakes like spelling the word "rabbit" with three Ms; wearing a black bra under a white shirt; or, to take a more masculine example, starting a land war in Asia. These are the kind of mistakes described by Mr. David Letterman as Brushes With Stupidity, because they have no reasonable chance of success." *-John Cleese*

Although there are several ways to represent solid objects in a computer, two have been used most widely. The Constructive Solid Geometry (CSG) representation describes a solid as the result of a sequence of Boolean operations on a given set of primitive solids. The Boundary representation (B-rep) describes the solid as an object in 3-space bounded by faces. Even systems which store solids as B-reps may use a CSG description of how the solid is built. I looked at how to minimize the cost of building solids in [8]. This was a mistake on two counts. First, Moore's Law [2] it is now eight years later so computers are about 2^5 times faster. Second, the construction history may capture some of the geometric semantics, for example, modeling the machining process - and that information has value.

So what I would like to discuss here is two versions of geometric semantics and a foundational issue.

3.1 B-rep to CSG

Semantics needs a syntax to make sense. The syntax in this case is the construction history of an object: how was it made using Boolean operations? Geometric semantics is provided by an evaluation process which will take a CSG tree and generate a boundary representation.

The reverse process: B-rep to CSG is trivial in a formal sense: make a tree with one primitive which is the evaluated solid. This is, however, stretching the notion of *primitive*. For CSG representations, primitives are defined in terms of half-spaces. Shapiro and Vossler in [16] and [15] identify two main problems to be solved. A set of half-spaces H_b divides R^3 into cells. If we can construct CSG representation for all of the cells inside S, the b-rep in hand can be built as the union of the CSG representations of the closures of those cells. [16] then discusses how to optimize this result to get a more efficient CSG tree.

3.2 Feature recognition

An alternative syntax to Booleans is the notion of features. The incorporation of features into commercial modeling systems is arguably responsible for a wider acceptance of solid modeling. This is a richer syntax, providing a more useful semantics.

Shen *et. al.* in [17] use similar techniques to Shapiro and Vossler to recognize features as opposed to reconstructing a CSG tree. Shirar *et. al.* in [18] provide an algebra for features in the context of machining operations. Mantyla *et. al.* in [11] provide a broader overview of the field.

3.3 Foundations

In [12] Raghothama and Shapiro note:

> "From its very inception solid modeling [see Requicha [14]] has been synonymous with unambiguous (informationaly complete) representations of homogeneously $n-$dimensional subsets of the Euclidean space. On the other hand, the recent rise of solid modeling as a principal information medium, first in engineering and now in consumer aplicaitons,probably has to do more with the development and succesful marketing of new parametric (*feature-based* and *constraint-based*) user interfaces than with the mathematical soundness of solid modeling systems...the new solid modeling systems no longer guanatee that the parametric models are valid or unambiguous, and the results of modeling operations are not always predictable."

Requicha [14] uses the set topological notion of *regular* sets to insure that solids obtained by evaluating CSG trees are realizable (i.e. machinable) pieces of space. A set X is regular if $X = c(i(X))$, the closure of its interior.

The acceptance of parametric or feature-based systemns demonstrates the need for a richer syntax and a correspondingly more complex geometric semantics. At the same time, [12] Raghothama and Shapiro provide examples of how the semantics are in general ambigous, and provide one possible direction for a solution: using the structure of an B-rep as a cell complex to control and evaluate the effects of changes in parameter.

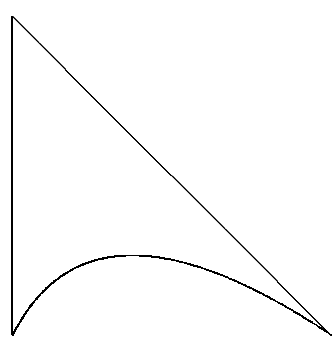
4 Geometric Constraints

> "On two occasions I have been asked [by members of Parliament], 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?' I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question." *Charles Babbage*

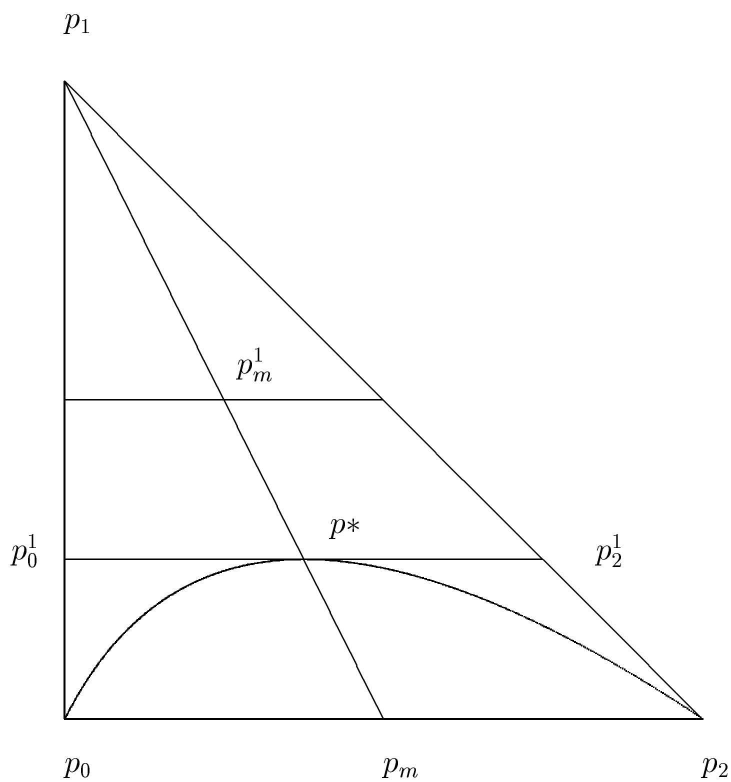
In early CAD systems, two lines drawn at right angles would not "know" that they were perpendicular. Geometric constraints add this knowledge.

One method is to encode all of the constraints as a series of equations and apply classical numerical methods. An alternative method, based on analytic solutions and analysis of a dependency graph, is given in [6]. The strength of this approach is reliable, reproducible results. A numerical solution may depend on (possibly random) initial guesses. A weakness of the analytic approach is that the types of geometry which can be handled are limited.

Here is an example of how to use a simpler geometry (the control net for a b-spline) to constrain a more complicated object (an elliptical arc). First, replace the elliptical arc by its rational quadratic Bezier curve.



Assuming the constraint solver cannot handle ellipses directly, build up a set of constraints starting with those two lines which force the quadratic Bezier to be an ellipse.



The rational Bezier form is

$\phi_0(t) = (1 \ - \ t)^2, \phi_1(t) = 2t(t \ - \ 1), \phi_2(t) = t^2$

$P(t) = \frac{\sum_{i=0}^{2} w_i p_i \phi_i(t)}{\sum_{i=0}^{2} w_i \phi_i(t)}$.

Let $w_0 = w_2 = 1.0$ and then

$w_1 = \frac{\|p+m-p*\|}{\|p_1-p*\|}$.

The geometric constraints needed to to get an ellipse are:

$\overline{p_0^1 \ p_2^1} \ \| \ \overline{p_0 \ p_2}$ (A)

and

$w - 1 < 1$ (B)

For (B) add:

- p_m is the midpoint of $\overline{p_0 \ p_2}$

- p_m^1 is the midpoint of $\overline{p_1\ p_m}$
- Now define a line L parallel to $\overline{p_0\ p_2}$ passing through p_m^1,
- a circle C_1 with center on $\overline{p_1\ p_m}$ tangent to L and $\overline{p_0\ p_2}$.
- Now let point p' lie on $C_1\ \cap\ \overline{p_0^1\ p_2^1}$.
- The final (magic) constraint is C_2 - a circle centered at p' tangent to L. The magic is that the constraint solver forbids zero-radius circles, so the radius of C_2 is never zero, forcing $w - 1 < 1$.

5 N-sided Surfaces

"I'm fixing a hole where the rain gets in
And stops my mind from wandering
Where it will go

I'm filling the cracks that ran through the door
And kept my mind from wandering
Where it will go"

-John Lennon

The problem being considered is:

Given $C_i, i = 1, ..., n$ n curves whose endpoints match, i..e (if we say $C_0 = C_n$) the end of C_{i-1} is the start of C_i, fill in the hole bounded by the C_i, possibly satisfying some additional boundary conditions. For example, in blending the C_i are the edges of faces and the filling surface or surfaces must be smooth across the edges.

This problem was discussed in [10]. Other surveys are: Dyn [1] for a review of John Gregory's contributions to the field, Gregory[3] is a survey on n-sided patches by Gregory and others, Varady[19] specific to vertex blends, and Vida[20] discusses blends in general with a section on n-sided issues.

There are two main approaches. Multiple patch approaches fill in the holes with three-sided or four-sided patches. The difficulty then is to insure cross-patch continuity. Single patch approaches find a single patch. This is easier to deal with in a solid modeling environment, but had difficulty because of the non-standard parameterizations. Recent work in [21] shows that several of the single patch approaches can be expressed in terms of toroidal varieties.

I close this section with two examples of extremely simple n-sided surfaces. Assume the C_i are all arcs. Define the *normal* for an arc to be the line defined by the center of the arc and the normal vector of the plane it lies in. Then,

Theorem 1 *The region bounded by the C_i can be filled with a plane if and only if N_i, the normals are all parallel.*

Proof: If all of the C_i are in the same plane, then the N_i must all be parallel to the normal for that plane. If the $N = i$ are all parallel, then the fact that C_i and C_{i+1} share an endpoint means that C_I and C_{i+1} are in the same plane, so all of the C_i lie in the plane defined by N_0 and the center of C_0.

Further, given

Lemma 1 *An arc C lies on a sphere S if and only if its normal N passes through the sphere's center.*

Proof: If the arc C is on the sphere, let c' be the sphere center, and c_0 be the arc center. Then for p on the arc, $(p - c_0)\dot(c_0 - c') = 0$ by looking at the great circle defined by p, c_0 and c'. So the normal vector for the plane of C is parallel to $(c_0 - c')$ and the normal (line) for C passes through c'. Conversely, if N passes the c_0, C lies on the intersection of its plane and S.

it follows that

Theorem 2 *The region bounded by the C_i can be filled with a sphere if and only if the N_i all intersect in a common point P which is the center of the filling sphere.*

Proof: Similar argument to the one for the plane. If all the arcs lie on a sphere, then by the lemma the normals all pass through the center of the sphere. Conversely, the common point of C_i and C_{i+1} forcea the sphere they each lie on to be the same sphere.

Some cases for which I do not have any nice answers: when can a collection of arcs bound a piece of a cylinder or a torus or a Dupin cyclide?

6 Conclusion

> "Mathematicians have long since regarded it as demeaning to work on problems related to elementary geometry in two or three dimensions, in spite of the fact that it is precisely this sort of mathematics which is of practical value" - *Branko Grünbaum and G. C. Shephard in the Handbook of Applicable Mathematics*

Any of the topics discussed here merit a complete and deeper discussion of their own. I hope this paper provides sufficient introduction to the literature and motivation for considering these (newer) examples of applied (older) mathematics.

7 Acknowlegdements

I would like to thank Ben Herbst for inviting me to present this talk, D-Cubed, Ltd and Spatial, Inc. for the software used in some of the referenced papers, Vadim Shapiro, Lee Cole, Jami Shah, and Christoph Hoffman for help with references, and Miki Magyar for helpful editing.

The opinions expressed in this paper are the author's and do not represent the views of Spatial, Inc.

References

[1] N. Dyn, *The Work of John Gregory: Rational Spline Interpolation, Subdivision Algorithms, and C^2 Polygonal Patches,* in Mathematics of Surfaces VI, Clarendon, 1996, 21–42

[2] Chris Gottbrath, Jeremy Bailin, Casey Meakin, Todd Thompson, J.J. Charfman, *The Effects of Moore's Law and Slacking on Large Computations* http://agave.as.arizona.edu/chrisg/mooreslaw/Paper.html Steward Observatory, University of Arizona

[3] J. A. Gregory, V. K. H. Lau and J. Zhou, *Smooth Parametric Surfaces and n-sided Patches,* in Computation of Curves and Surfaces, Kluwer, 1990, 457–498

[4] D.G. Hakala, R. C. Hillyard, P. Malraison, B. E. Nourse, *Natural quadrics in mechanical design,* Proc - AUTOFACT West, Vol. 1, CAD/CAM 8, Anaheim, Calif, USA, November 17-20, 1980 Publ by SME, Dearborn, Mich, USA, 1980, 363–378

[5] J. Levin, *A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces,* Communications of the ACM, Vol 19, No 10, 555–563

[6] J. Owen, *Constraints on Simple Geometry in Two and Three Dimensions,* International Journal of Computational Geometry and Its Applications, Vol. 6, No. 4, December 1996, 421–434

[7] P. Malraison, *Algebraic Geometry in CAD/CAM: The Role of 19th Century Geometry in Geometric Modeling,* The UMAP Journal, Vol. 11, No. 2, (1990), 161–169

[8] P. Malraison and D. Schlaepfer, *Designing Efficiently with Solids,* CADALYST, March 1992

[9] P. Malraison, *Constraining a Quadratic Rational Bezier to be an Elliptical Arc,* International Journal of Computational Geometry and Its Applications, Vol. 6, No. 4, December 1996, 435–441.

[10] P. Malraison, *N-sided surfaces: A Survey,* Curve and Surface Design: Saint-Malo 99, Pierre-Jean Laurent, Paul Sablonnière, and Larry L. Schumaker (eds), Vanderbilt University Press (2000).

[11] M. Mantyla, J. Shah and D. Nau, *Challenges in Feature–based Manufacturing Research,* Comm. of the ACM, Vol. 39, No 2, Feb 1996, 77–85.

[12] S. Raghothama and V. Shapiro, *Boundary Representation Deformation in Parametric Solid Modeling,* ACM Transactions on Computer Graphics, Vol. 17, No. 4, Oct 1998 259–286.

[13] S. Raghothama and V. Shapiro, *Consistent Updates in Dual Representation Systems,* Proceedings, 5th ACM conference on Solid Modeling, 65–75

[14] A. A. G. Requicha, *Representations for rigid solids: Theory, methods and systems,* Procedings, 5th ACM conference on Solid Modeling, 65–75 ACM Computing Surveys, Vol. 12, December 1980, 437–464

[15] V. Shaprio and D. Vossler, *Separation for Boundary to CSG conversion,* ACM Transactions on Computer Graphics, Vol. 12, No. 1, January 1993, 35–55

[16] V. Shaprio and D. Vossler, *Construction and optimization of CSG representations,* CAD, Vol. 23, No. 1 Jan/Feb 1991, 4–20

[17] Y. Shen and J. J. Shah, *Feature recognition by volume decomposition using half-space partitioning,* in ASME Design Technical Conference, 1994

[18] A. Shirur, D. Nau and J. Shah, *Challenges in Feature–based Manufacturing Research,* Journal of Manufacturing Systems Vol. 17, No. 3, 1998, 167–82.

[19] T. Varady and C. M. Hoffmann, *Vertex Blending: Problems and Solutions,* in Mathematical Methods for Curves and Surfaces II:Lillehammer, Vanderbilt University, 1998, 501–527

[20] J. Vida,R. R. Martin and T. Varady, *A survey of blending methods that use parametric surfaces,* Computer Aided Design, Vol. 26, no. 5, 1994, 341–365

[21] S. Zube, *N-Sided Toric Patches and A-resultants,* Curve and Surface Design: Saint-Malo 99, Pierre-Jean Laurent, Paul Sablonnière, and Larry L. Schumaker (eds), Vanderbilt University Press (2000).