CSG SET-THEORETIC SOLID MODELLING AND NC MACHINING OF BLEND SURFACES

by

Dayong Zhang and Adrian Bowyer

The School of Engineering
University of Bath
Bath
England BA2 7AY

0. Summary

This paper presents a new solid modelling system capable of representing three dimensional objects consisting of both simple and complicated surfaces and, equally importantly, blend surfaces between them. Implicit polynomial inequalities and set theoretic techniques are used to specify the models. The modelling system can generate shaded pictures and NC machining instructions for cutting complicated objects automatically.

The modelling system uses a technique for generating polynomials for blends and fillets that allows the user complete control over their extent without requiring him or her explicitly to devise the polynomial inequalities needed to represent them.

1. Introduction

Early solid modelling systems were mainly used to represent simple-shaped engineering components which could be made from cuboids, quadrics, and the torus. However, there are a great number of components that are not just made of those simple primitives, but also have blends between them.

The representation and manipulation of blended and complicated surfaces has been one of the major outstanding difficulties in developing solid modelling systems. There have been some attempts to solve these problems [7], but few of them have achieved success so far [1,5].

This paper presents an efficient, consistent, solid modelling system capable of representing blend surfaces and generating NC machining instructions for cutting them.

2. Problems in incorporating complicated surfaces in solid models

Historically, there have been two main methods used to represent the shape of an object in a computer [12]. *Boundary representation* modellers use surface modelling techniques and topological checking to ensure the solidity of the objects that they represent. Some incorporate parametric patches as well as simpler surfaces. Such systems are mainly used in the aircraft, shipbuilding and vehicle manufacturing industries. *Constructive Solid Geometry* (CSG) or *Set Theoretic* modellers use primitive solids, such as cuboids, cylinders, cones and spheres which are put together using set-theoretic operators. This technique is often used for representing machined parts in mechanical engineering, and the modeller described in this paper is of this type.

There are two types of relationship between the shape elements used in constructing a component. One is a direct joint or cut, such as that obtained by applying Boolean operators (UNION, INTERSECTION, or DIFFERENCE) to solid primitives. In this case, any face of the modelled object is part of the surface of some primitive. The other relationship is a smooth transition or blend between primitives. In this case, a more complicated surface is required to link two or more primitives together with continuity in both slope and position (that is C1 continuity).

Despite their widespread use, it is not easy to generate complicated surfaces using parametric patches. It is inconvenient to generate low order surfaces and difficult to define the required blend patch surface between two parametric patches, let alone to define a blend between three or more patches.

It is also difficult to combine parametric patches with solid primitives in one system [8], because parametric patches are defined by meshes in parametric space, whereas solid primitives are defined by implicit inequalities or *half-spaces* and operated on by set-theoretic operators. The set-theoretic operations use the fact that an implicit function has an inside and an outside (solid and air), which parametric patches do not have. It is also not easy to generate a blend between a parametric patch and an implicit half-space's surface, because the relationship of the blend to these different types of surface is difficult to establish consistently.

Low order half-spaces with exponents not higher than two or three can be simply specified using implicit equations. Their surfaces are easy to understand and manipulate. However, it is not so easy to define a high order surface by a single equation in the traditional way, that is to say by choosing coefficients of the polynomial explicitly or by requiring the polynomial to pass through certain points in space. To meet the requirement to generate blends, there needs to be some other way of defining high order surfaces.

3. Using implicit polynomials to represent complicated and blended surfaces

In aircraft fuselage cross-section design in two dimensions, Liming [4] used straight line equations as elements in constructing implicit quadratic curves. For example, a quadratic curve can be defined in the following form:

$$(1-\lambda)PQ + \lambda RS = 0 \qquad (1)$$

Where P,Q,R and S are the linear functions

$$\begin{aligned} P &= A_p x + B_p y + C_p \\ Q &= A_q x + B_q y + C_q \\ R &= A_r x + B_r y + C_r \\ S &= A_s x + B_s y + C_s \end{aligned} \qquad (2)$$

and λ lies in the range [0,1].

Equation (1) is equivalent to a general quadratic and has the property that it passes through all the points where P or Q cut R or S, as shown in Fig. 3.1.

If line S is moved towards line R, eventually the lines will merge (Fig. 3.2). When S and R become identical, equation (1) becomes:

$$(1-\lambda)PQ + \lambda R^2 = 0 \qquad (3)$$

It is clear from Fig. 3.2 and equation (3) that:

a. In (1), four lines are used to define a quadratic, but in (3) only three are needed.

b. The four lines in (1) have equal importance in the definition of a quadratic, while the three lines in (3) have different contributions: lines P and Q become two tangent lines of the quadratic curve.

This idea of constructing a quadratic from straight lines can be extended to complicated surfaces and blended surfaces. This

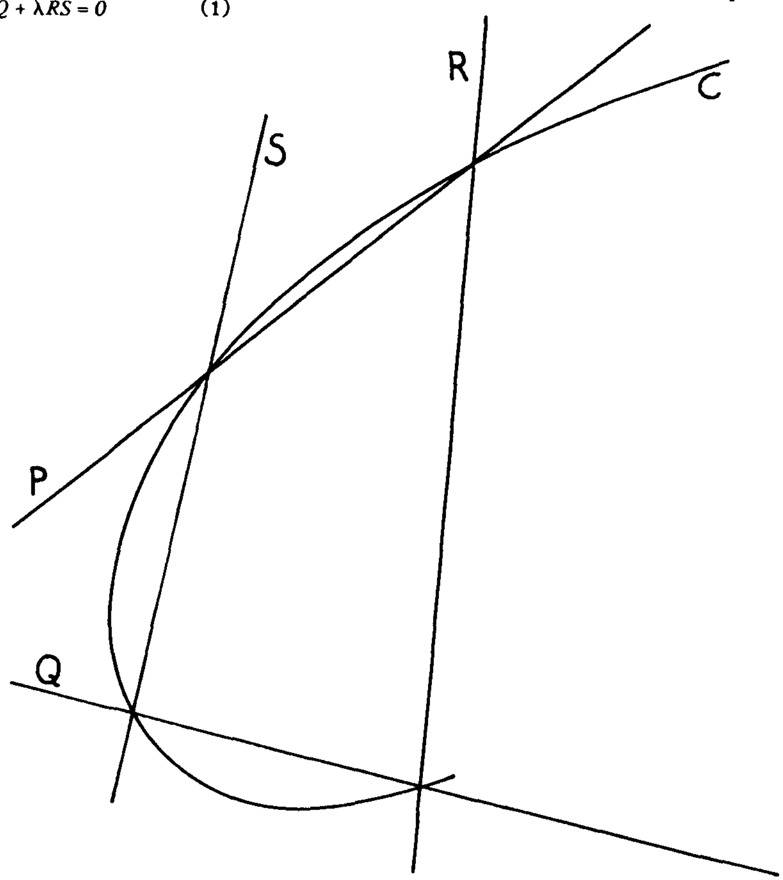

Figure 3.1 Quadratic, C, through the intersection of four straight lines: $C = (1-\lambda)PQ + \lambda RS$

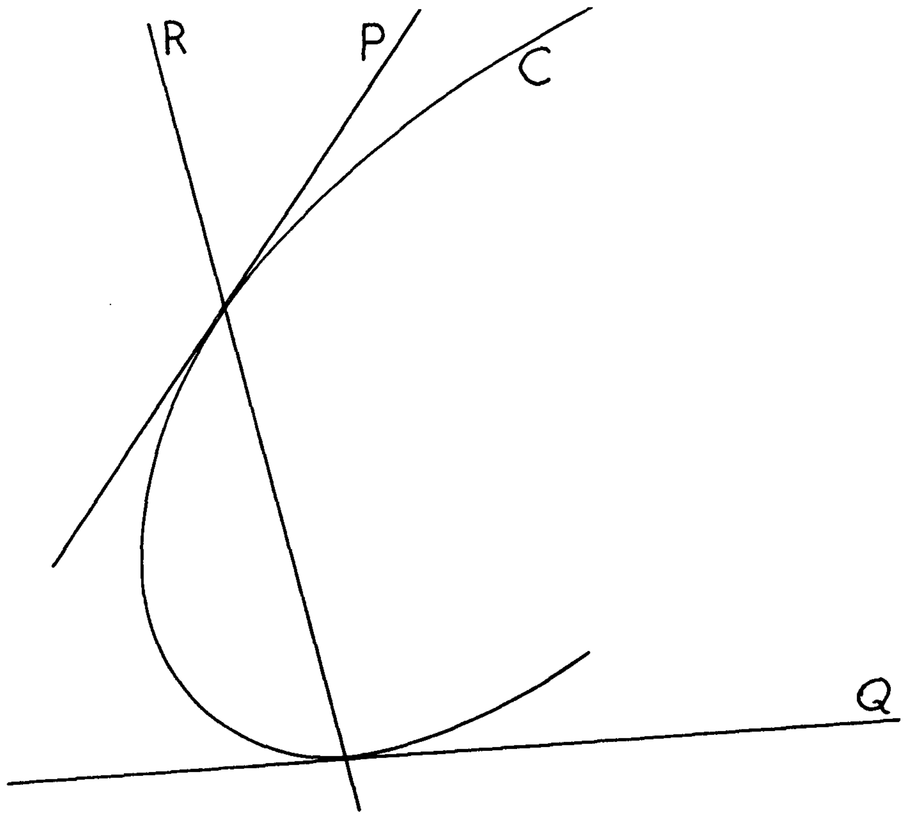

Figure 3.2 Quadratic, C, tangential to two lines where they intersect a third: $C = (1 - \lambda)PQ + \lambda R^2$

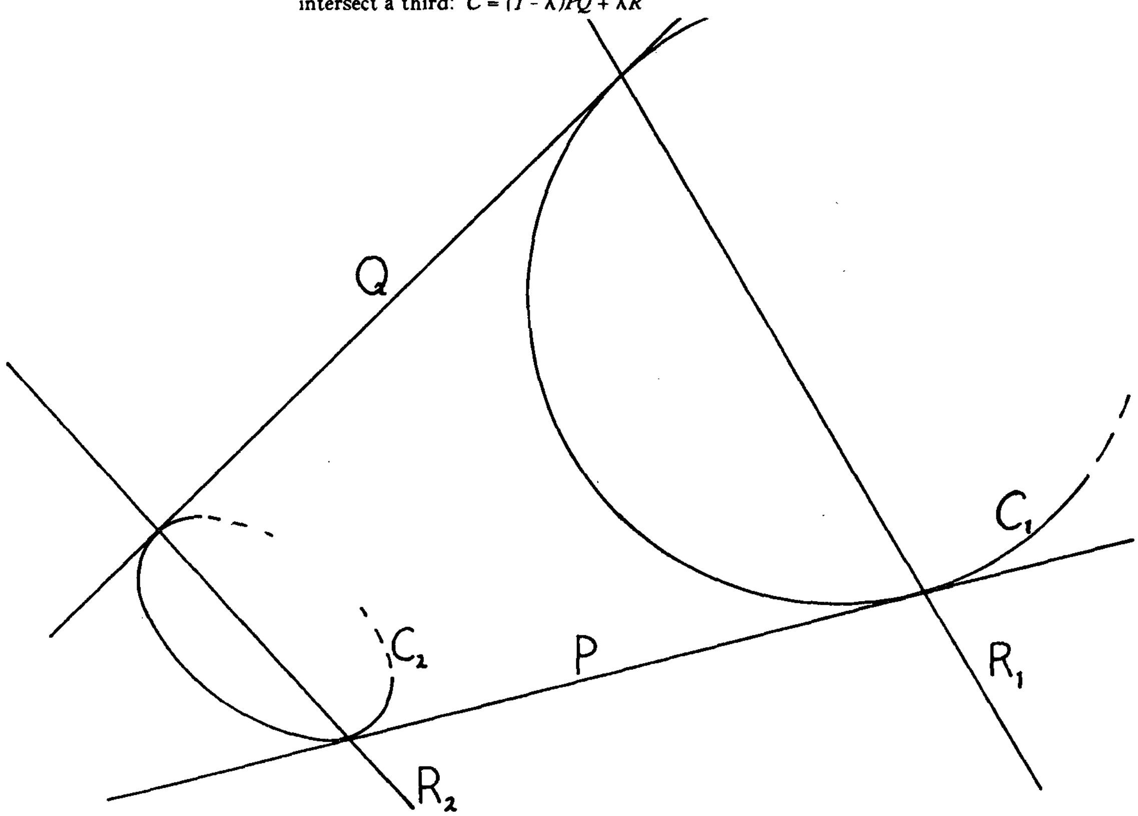

Figure 3.3 Control of the range of a blend.

was first thought of and implemented by a colleague of the authors, John Woodwark [13], though the method described in [5] was devised more or less coincidentally, and is similar.

Firstly, by replacing straight lines in two dimensions with three dimensional planes, we can easily generate quadratic surfaces. Secondly, by changing the number of planes and their orientations and relative positions, we can generate all sorts of conic surfaces which are difficult to specify in traditional ways. Thirdly, by replacing some of the planes with higher order surfaces, or just with constant values, we can get very complicated surfaces of different orders. Finally, if we use the defined surfaces from one equation to define surfaces in another, we can build up required surfaces stage by stage. One very important feature of this is that all these extensions keep the property of this method that the defining surfaces and the defined surfaces are smoothly joined together with a common tangent. Therefore, it is an effective way of defining blend surfaces between primitives, and also blends between blends.

The reasons for choosing this method of defining blends in a solid modelling system are as follows:

Simple Specification:

Because blend surfaces are usually complicated, it is important to keep the specification simple in designing a model. The blend can be specified by the user from only the surfaces being blended and another to define the tangency. It is possible to define the tangency using more than one surface to limit it if that is convenient.

Continuity:

Blends defined in this way automatically meet requirements of continuity in slope as well as of position, which are difficult to achieve in other ways.

Controllability:

The range of blends is very easily controlled by the position of R in relation of P and Q (Fig. 3.3) and the extent of the blend is controlled by the value of λ: If λ is 0, no blend is generated and P and Q join together sharply; if λ is 1, P and Q are linked by R; if λ is between 0 and 1, a blend is generated that extends within the boundary: as λ increases, the blend becomes fatter and fatter. (Fig. 3.4). Using set-theoretic operations, we can chop the blend off just where we stop being interested in it (Fig. 3.5) simply by intersecting the blend with the half-space used to specify the tangent locations.

4. Model evaluation and picture generation

One of the problems in evaluating a model defined by implicit polynomials is the number of calculations required to find the position of each surface in space. This problem is dealt with by using recursive sub-division and pruning techniques.

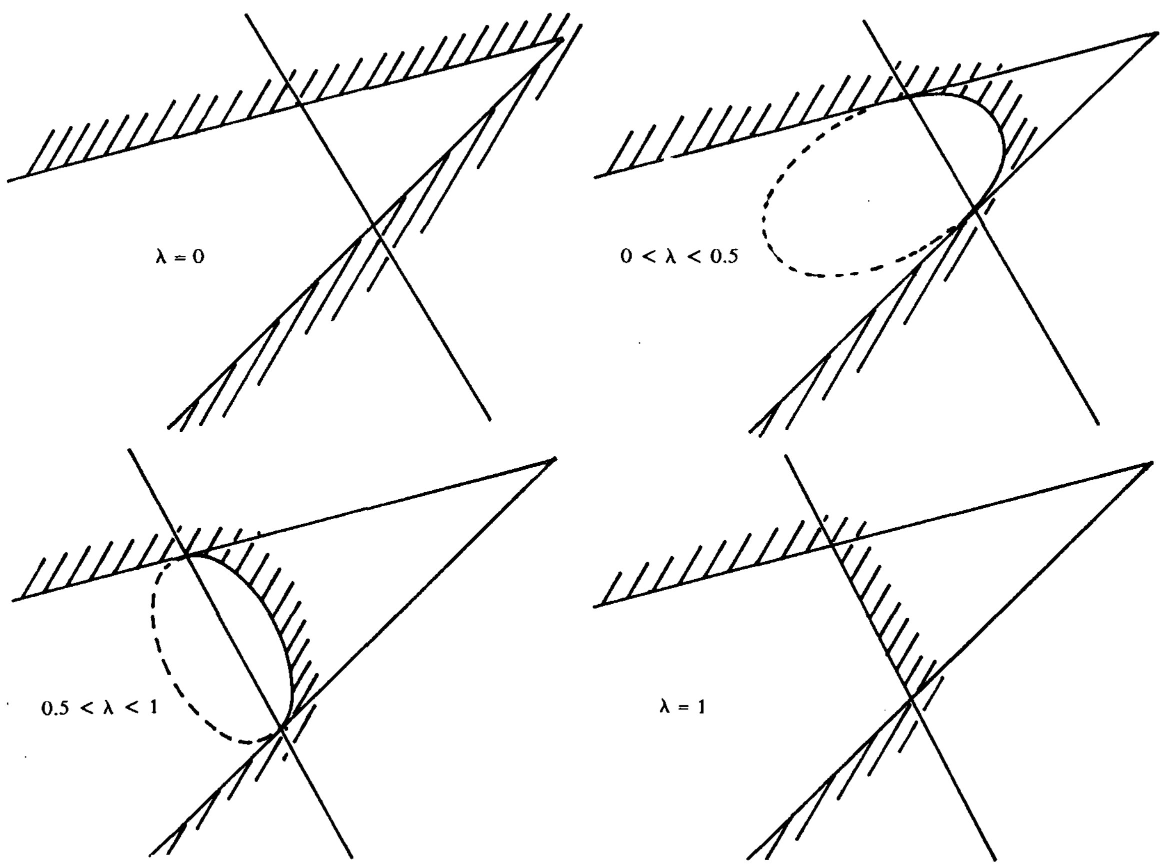

Figure 3.4 The effect of changing λ in the blend equation

The model has associated with it an object volume: a cuboid in space aligned with the coordinate axes that contains the model. In order to reduce the impact of the complexity of the model upon the time it takes to make a picture of the object modelled or the time taken to generate a cutter path for an NC mill to cut it, this volume is recursively divided in a manner similar to a three dimensional equivalent of Warnock's algorithm for hidden detail removal. The volume is divided into smaller cuboids by planes parallel to the coordinate directions. The decision on where to put these planes and which coordinate axis to align them with at each stage is taken by examining the model and optimising the complexity of the smaller cuboids against the complexity of their contents. For example, if the division is being performed to generate pictures, the appropriate things to include in the complexity calculation are the likely contents of the two smaller cuboids created by a division and the probability of a ray from the picture-generating ray-caster hitting the cuboids produced. Similar criteria are used for tool path generation.

As the model is divided it is also pruned [9,10,11]. The whole model starts out as a large (perhaps 2000 term) set-theoretic expression involving polynomial half-spaces and UNION, INTERSECTION and, (redundantly, but usefully) DIFFERENCE operators. Parts of the set-theoretic expression not contributing to the cuboids that result from a division are pruned out, so that, at the end of the division process, the model consists of a large number of little cuboids as leaves of the division tree. Each cuboid has a simple set-theoretic expression associated with it that represents the whole model locally in that cuboid.

This division process can be carried out quickly, and makes the performance of all subsequent processes such as picture generation and tool path generation faster than linear against object complexity; complexity being considered as proportional to the number of half-spaces comprising the model.

Because the divided and pruned data structure represents a model exactly, the shaded pictures of a model generated by ray casting appear exactly as it was defined and no approximation is introduced. The same is true of the use of the structure for tool-path generation. One of the advantages that set-theoretic modellers have over boundary modellers is that they are more numerically robust. This robustness is not compromised at all by the division algorithm.

Figure 4.1 shows a blend between three cylinders meeting. The largest and middle-sized cylinder were blended, and then the smallest cylinder was blended onto that blend. The ray-casting rendering program for the modeller was written by a colleague of the authors, Andrew Wallis.

5. Generating CNC machining instructions for cutting the models

A three axis CNC milling machine was available for use by the modelling project described in this paper. Cutter path generation for this will be described, though the divided model could, just as easily, be used to control a machine with more degrees of freedom.

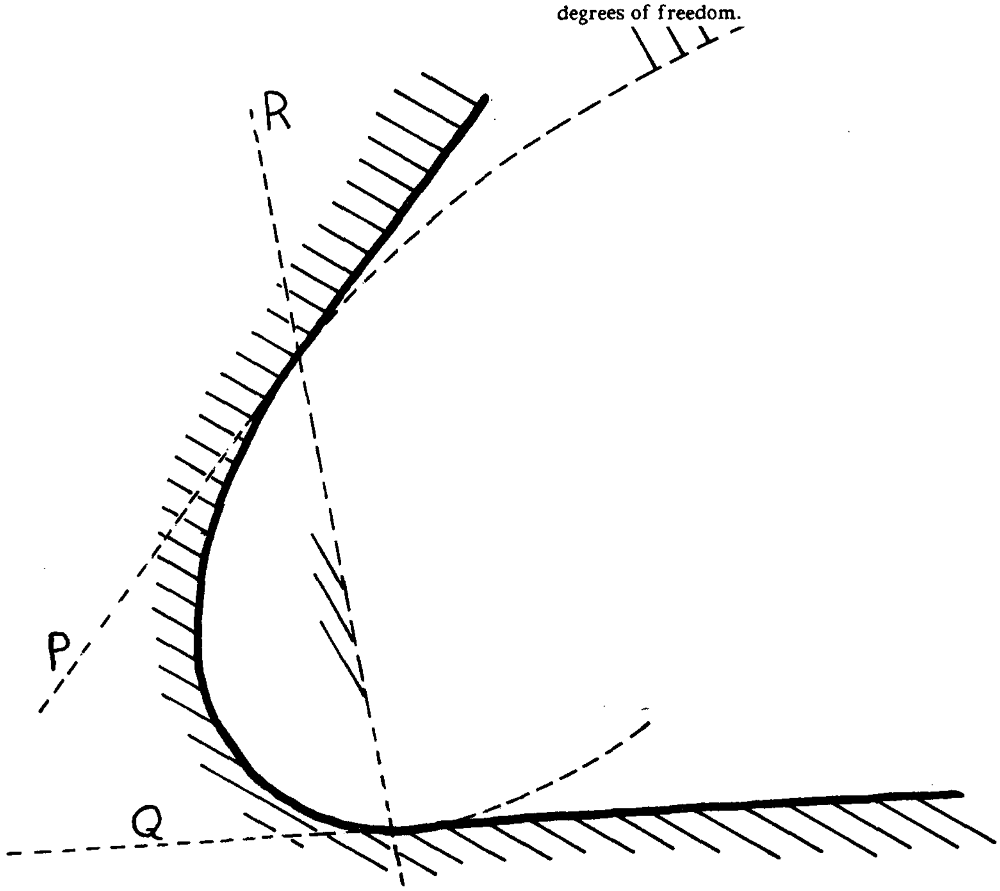

Figure 3.5 Set theoretic operators allow a blend to be truncated using its forming surface economically. The object is

$$P \cup Q \cup \{[(1-\lambda)PQ + \lambda R^2] \cap R\}$$

The shaded side of the half spaces is their solid side.

The modeller has been used to cut components that can be described by single valued functions. This is not a restriction of the modeller, but of the machine tool that the authors have available. This family of components include items such as dies, moulds and casting patterns. The cutter path generation simply ignores under cuts, so it is a simple matter to cut components of a more complicated shape by first cutting down in one direction, then turning the component over and cutting in another. This, of course, presumes that adequate fixturing for holding the component for the second and subsequent cuts is available. Figure 5.1 shows a picture of a handwheel with a number of blended surfaces modelled by the system. Figure 5.2 shows the same model with a check surface added for cutting, the purpose of which is to prevent the cutting tool plunging too deeply into the gaps between the spokes. Figure 5.3 is a photograph of the model as cut. This was cut in wood to save time, but, of course, cutting metal can be achieved just as straightforwardly.

The cutting data generated from the divided model are an x-y grid with different z values at each node. This is generated by ray-casting vertically into the model, the rays being parallel to the axis of the cutting tool. Because the information in the divided model is exact, the points on the grid are exactly on the surface. This means that by this stage no approximation has been introduced. Theoretically, the precision of a surface is only affected by the distance between grid points, which can be as small as the precision required. For the handwheel a 1mm grid was used.

The system described in this paper transforms the surface-point position grid to a tool-centre position grid. It assumes that each surface point has a corresponding tool centre above it. From this assumption, the tool centre will have the same x and y values as the surface point has and all tool positions are evenly located in x and y. Only the z values have to be calculated.

To work out the z value of the tool centre at a node on the grid, only those surface points within the circular area covered by the tool cross section need to be considered. In order to decide the tool height for a given x-y point all that the system needs to do is to inspect the neighbouring points in the grid that lie inside the circle representing the cross section of the tool (Fig. 5.4). For each of these the height of the tool can be calculated (knowing its shape) and the maximum of these values taken as the actual one for cutting. This automatically prevents unwanted interference between the tool and parts of the work not being cut and ensures the best (for a given tool size) approximation to the shape of the required component that it is possible to achieve. Several cutting passes with different sized and shaped cutters are performed to obtain the exact final shape required. Because the divided and pruned model is such an efficient structure for the ray-caster to interrogate this whole process can be carried out quickly.

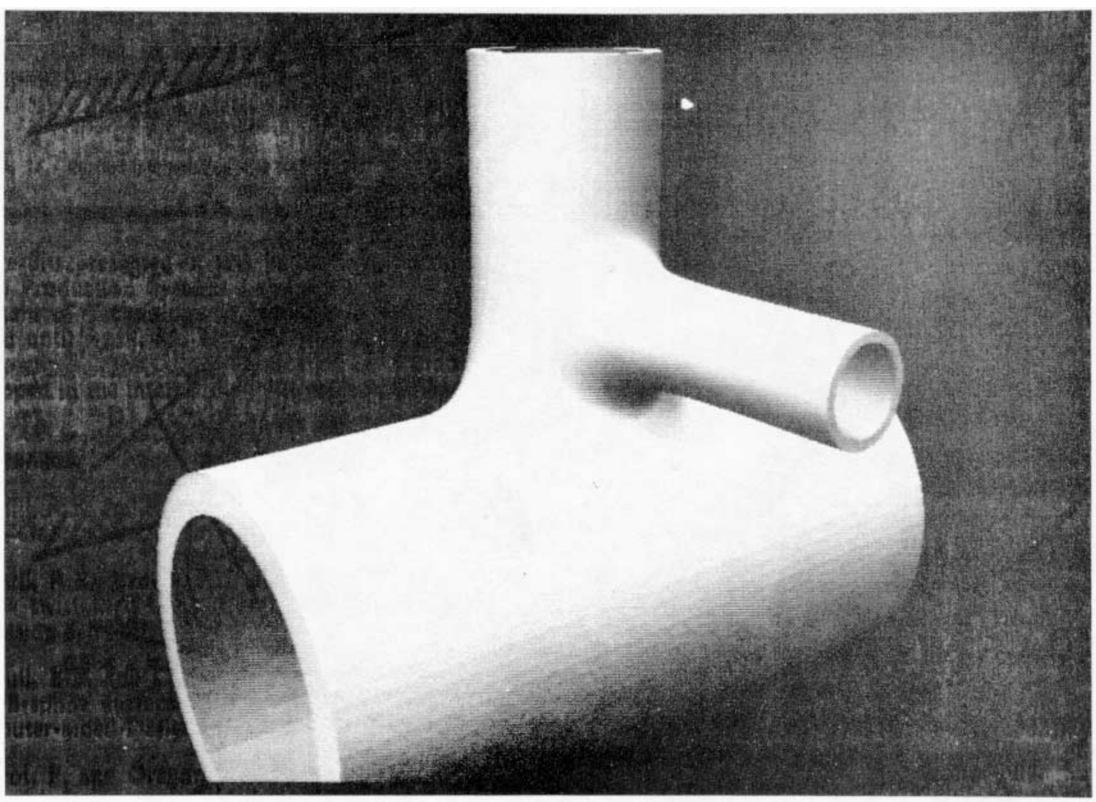

Figure 4.1 A blend between three cylinders of different diameters

Using this method, there is no need to calculate normal vectors in deciding the tool centre and interferences are automatically avoided during the course of transforming the data into a tool-centre position grid. The method can also be used for flat ended and cone ended cutters. So a better, more even product is achieved using a simpler cutting strategy. Flat areas and vertical surfaces are cut with appropriate cutters, and a ball-nosed cutter is only used when it is really needed. Also the cut can be performed in any order, so it is possible to group areas that are alike together and cut them all with one tool in a way that gives the best finish and most efficient cutter path.

6. Conclusions

The blend solids produced by this system are continuous in both position and slope, so that they link basic solid primitives together smoothly. The control of the boundary and extension of a blend surface can be very easily specified by the user of the system.

The CNC machining instructions generated by this system are free of interference and their precision can potentially meet any requirement, and in practice is only limited by the precision of the computer and the NC machine.

The solid modelling system described in this paper is one of very few that has *all* implicit polynomial inequalities as its primitive domain, and is the only such, as far as the authors are aware, capable of automatically manufacturing the components that it is used to model.

7. Acknowledgements

The authors acknowledge with gratitude the provision of funds by the United Kingdom's Science and Engineering Research Council and by the Chinese Government. They would also like to acknowledge the instigator of the project, John Woodwark, and their colleague Andrew Wallis for all their work. Dr Woodwark and Mr Wallis will be reporting this elsewhere.

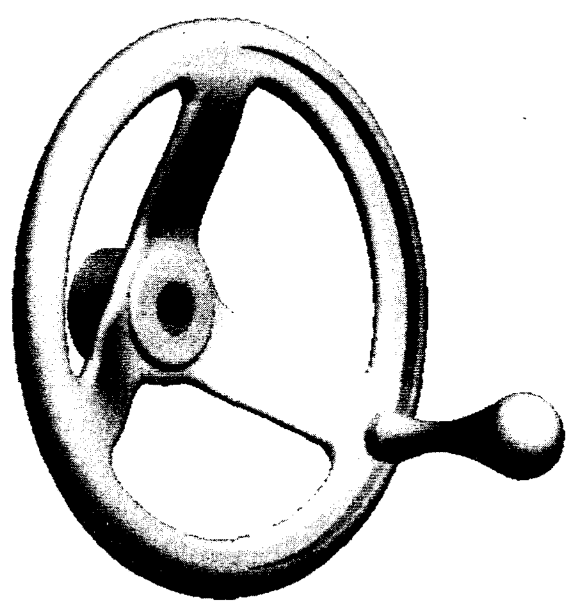

Figure 5.1 Model of a handwheel with several complicated surfaces and blends

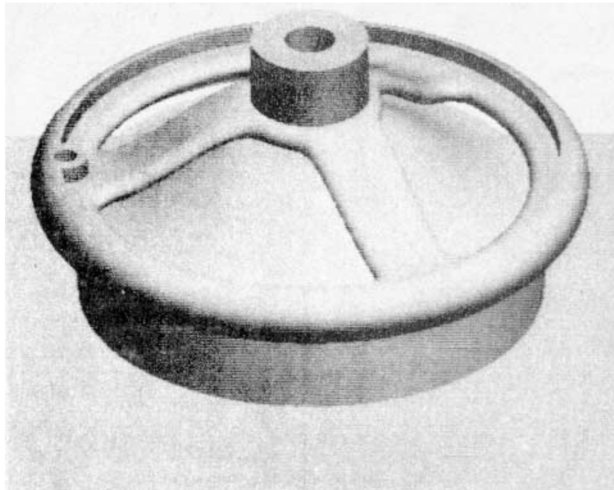

Figure 5.2 The handwheel with check surfaces added

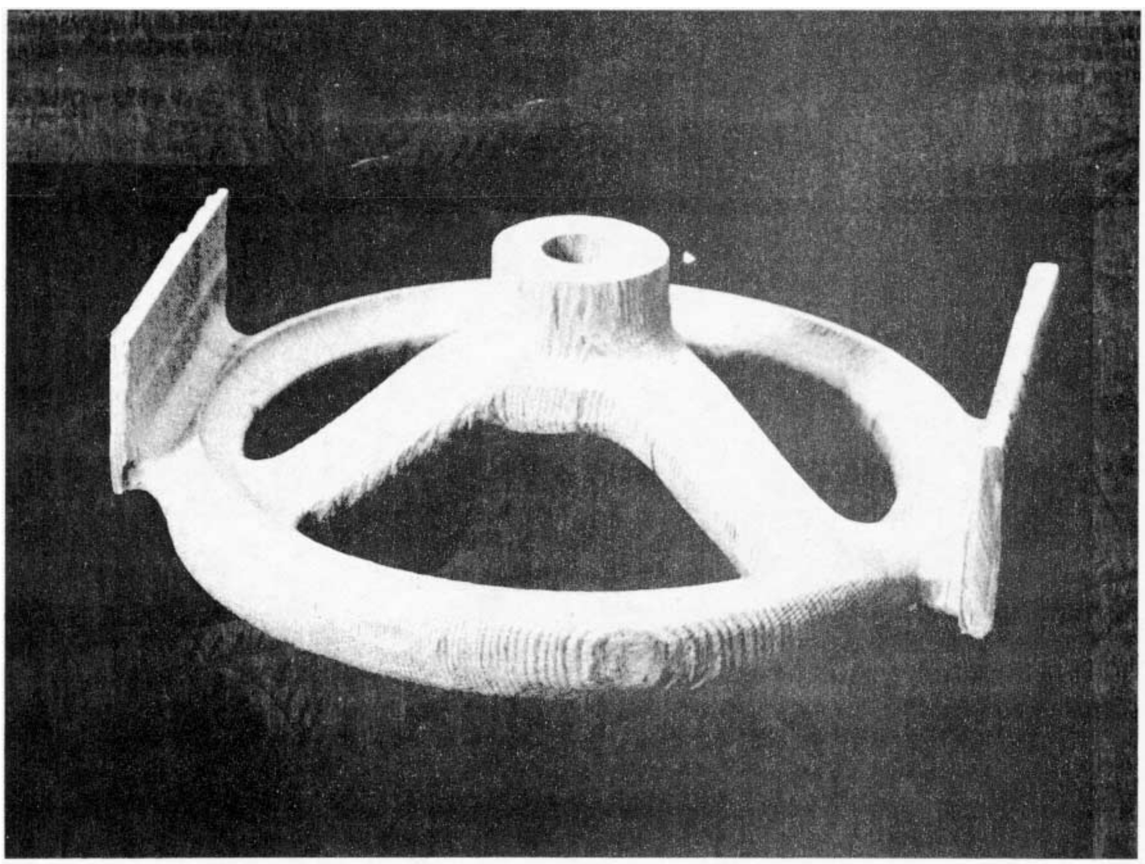

Figure 5.3 The handwheel after cutting. The rectangular projections were added to facilitate fixing, and would not be needed if the wheel were to be cut mounted by its central hole. The top surface has been finished, and the under surface is as left by the roughing cut.

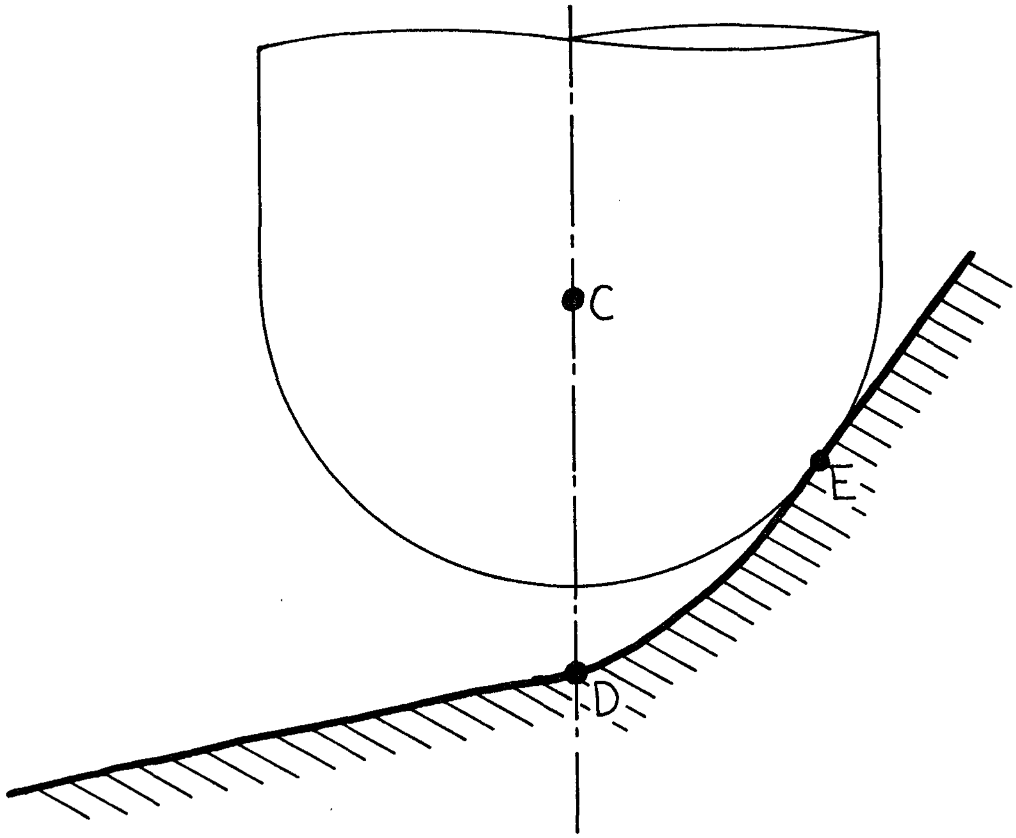

Figure 5.4 The offset centre (C) of the ball-nosed cutting tool above point D is actually decided by the surface at point E.

8. References

[1] Hoffmann, C. & Hopcroft, J. *Automatic Surface Generation in Computer Aided Design* Cornell University Computer Science Dept. Technical Report 85-661 (1985)

[2] Hoffmann, C. & Hopcroft, J. *Quadratic Blending Surfaces* Cornell University Computer Science Dept. Technical Report 85-674 (1985)

[3] Hoffmann, C. & Hopcroft, J. *The Potential Method for Blending Surfaces and Corners* Cornell University Computer Science Dept. Technical Report 85-699 (1985)

[4] Liming, R.A. *Practical Analytical Geometry with Applications to Aircraft* Macmillan, New York (1944)

[5] Middleditch, A.E. & Sears, K.H. *Blend Surfaces for Set Theoretic Volume Modelling Systems* Proc. ACM SIGGRAPH '85, San Fransisco, U.S.A. **19(1985)161**

[6] Phillips, M.B., & Garrett, M.O. *An Algorithm for Locating and Displaying the Intersection of Two Arbitrary Surfaces* IEEE CG&A (1984)48

[7] Ricci, A. *A Constructive Geometry for Computer Graphics* Computer Journal **16**(1972)157

[8] Sederberg, T.W. & Anderson, D.C. *Implicit Representation of Parametric Curves and Surfaces* Computer Vision, Graphics, and Image Processing **28**(1984)72

[9] Woodwark, J.R. & Bowyer, A. *Better and Faster Pictures from Solid Models* IEE Computer Aided Engineering Journal **3**(1986)17

[10] Woodwark, J.R. & Quinlan, K.M. *Reducing the Effect of Complexity on Volume Model Evaluation* Computer Aided Design **14**(1982)89

[11] Woodwark, J.R. & Quinlan, K.M. *The Derivation of Graphics from Volume Models by Recursive Subdivision of the Object Space* Proc. CG 80. Brighton, England (1980)530

[12] Woodwark, J.R. *Computing Shape* Butterworths ISBN 0-408-01402-4 (1986)

[13] Woodwark, J.R. *Pers. Com.* IBM U.K. Scientific Centre, Winchester, England.