A new method for drawing Algebraic Surfaces

Richard Morris

University of Liverpool

1 Introduction

In this paper we describe a program for drawing algebraic surfaces in $\mathbf{R}^3$. Strictly speaking it draws the zero sets $f^{-1}(0)$ of implicit polynomials $f : \mathbf{R}^3 \rightarrow \mathbf{R}$. The program produces a model of the surface which can be viewed using a general rendering package. The program has been specially designed to draw singular surfaces accurately and find all the components of the surface.

There are many programs already existing which do this to a limited extent. Programs to draw surfaces of the type $z = f(x, y)$ have been around for a number of years. CAD systems can draw specific types of algebraic surfaces but not normally singular ones. There are a few existing programs which draw general algebraic surfaces. Ray tracers [5] can be used to produce beautiful pictures of algebraic surface but they are slow and only gives the view from one direction. Other programs such the contour tracer written by the Geometry Supercomputer Project [2], produce good results but do not pay mush attention to singularities and may give topologically incorrect results.

A hybrid method has been used in the algorithm using recursive sub-division, Bernstein polynomials and a little differential geometry. The division of the surface into cells also plays an important part in the algorithm.

Thanks must go to R. Martin who introduced me to A. Geisow's method [3] for drawing algebraic curves in the plane which is based on Bernstein polynomials, D. Marsh for useful conversations and an overview of possible techniques, C. Gibson for doubting whether it was possible and hence providing the motivation and finally all those in the Liverpool University Pure Mathematics Department who produced many practical examples to test the routine on.

The pictures produced in this paper have been produced using an implementation of the algorithm running on a Silicon Graphics Iris 4D-GT graphics workstation provided by the S.E.R.C. under a computer science initiative grant. The models were displayed using the geomview package written at the Geometry Center, University of Minnesota.

2 A parade of surfaces

Before we start discussing the algorithm for drawing algebraic surfaces it is worth looking at some examples of what we might expect. An equation $f(x, y, z) = 0$ will typically define a surface, however it can also define a curve ($x^2 + y^2 = 0$), or an isolated point ($x^2 + y^2 + z^2 = 0$).

If all three partial derivatives $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$, $\frac{\partial f}{\partial z}$ are non-zero at a point on a surface then the surface is smooth at this point. If $\frac{\partial f}{\partial x} = 0$, $\frac{\partial f}{\partial y} \neq 0$, $\frac{\partial f}{\partial z} \neq 0$ at a point of a surface then the surface is smooth at that point and the x-axis is tangent to the surface at that point. Such points generally lie along curves on the surface. If $\frac{\partial f}{\partial x} = 0$, $\frac{\partial f}{\partial y} = 0$, $\frac{\partial f}{\partial z} \neq 0$ then the surface is smooth and the x-y plane is tangent to the surface. Generically such points are isolated, lying at the intersections of the curves $\frac{\partial f}{\partial x} = 0$ and $\frac{\partial f}{\partial y} = 0$. For the surface $x^2 + y^2 - z = 0$ shown in Fig. 1 all the points on the curve $x = 0$, $y^2 = z$, satisfy $\frac{\partial f}{\partial x} = 0$, all the points on the curve $y = 0$, $x^2 = z$, satisfy $\frac{\partial f}{\partial y} = 0$ and at the origin both partial derivatives vanish. It is possible for two partial derivatives to vanish along a curve on the surface. An example of this is the surface $(x - y)^2 + z = 0$ where $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$ along the curve $x = y$, $z = 0$. Very exceptionally we can expect one of the derivatives to vanish at all points of the surface. For example the plane $y + z = 0$ has $\frac{\partial f}{\partial x} = 0$ everywhere and the plane $z = 0$ will have $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$ at all points of the plane.

FIGURE 1. Two smooth surfaces showing curves along which the partial derivatives vanish

If all three partial derivatives vanish then the surface is singular. Generically the singular points are isolated. In fact all the germs of maps $\mathbf{R}^3 \to \mathbf{R}$ which have finite positive co-dimension have isolated singularities. For example the germs A_k, (normal form $x^{k+1} \pm y^2 \pm z^2$), D_k, (normal form $x^2 y \pm y^{k-1} \pm z^2$) and E_6, E_7, E_8 (normal forms $x^3 \pm y^4 \pm z^2$, $x^3 + xy^3 \pm z^2$, and $x^3 + y^5 \pm z^2$) all have isolated singularities at the origin. Some of these surfaces are shown in Fig. 2. See [1] pp158–166 for a classification of germs of maps from $\mathbf{R}^3$ to $\mathbf{R}$ and [4] for a gentle introduction to Singularity theory.

FIGURE 2. Zero contours of some of the standard germs

Not all surfaces have isolated singularities. A result of Whitney shows that the image of a stable map from a 2-manifold into 3-space typically consist of smooth points, transverse intersections ($xy = 0$), triple points ($xyz = 0$) and cross-caps ($x^2 z + y^2 = 0$), which are shown in Fig. 3. Note that when given algebraically a cross-cap also contains a line, which is the analytic continuation of the self-intersection curve.

FIGURE 3. A cross-cap and a triple point

The intersection of two surfaces will generically give a curve which may itself be singular. For example the intersections of $x^2 + z = 0$ and $y^3 + z = 0$ is the curve $x^2 = y^3 = -z$ which has a cusp at the origin. The intersection of two surfaces, $f = 0$, $g = 0$ can be expressed algebraically as $f^2 + g^2 = 0$. The union of two surfaces, $fg = 0$, will have self-intersection along the curve of intersections.

Another interesting type of surface is a *bifurcation set.* For a three parameter family of functions, $f_{abc} : \mathbf{R}^n, 0 \to \mathbf{R}^p, 0$, the bifurcation set is the set of parameter values in parameter space for which the function is not stable. For example the quartic equation, $x^4 + ax^2 + bx + c = 0$, will have a repeated root if the discriminant, $-4c^3b^2 - 27b^4 + 16ac^4 - 128a^2c^2 + 144ab^2c + 256a^3$, is zero. The set of parameter values $(a, b, c) \in \mathbf{R}^3$ for which this holds gives a swallowtail surface Fig. 4. The smooth part of this surface correspond to functions with one repeated real root, the cuspidal edges to functions with one real root repeated three time, the self-intersection curve to functions with two repeated real roots and the origin to the function x^4. This algebraic surface also contains a tail, $b = 0$, $4a = c^2$, $c \leq 0$, which corresponds to a repeated complex root, this is not normally thought of as part of the discriminant surface. Note this surface shows the limits of the current algorithm, and there are some mistakes in the model produced, see section 5, for a discussion of why these errors occurred.

FIGURE 4. A swallowtail surface

One particularly nasty example of a singular function is the square of a function $f^2 = 0$ which is singular at all points of the surface.

3 Bernstein polynomials

The computations involved in the program are made much simpler by the use of Bernstein polynomials. To simplify the discussion we will look at the one-dimensional case. All the results below can be easily adapted to work for higher dimensions. All the results in this section are well known and the algorithms have been taken from a method for drawing algebraic curves in 2D, described by A. Geisow [3].

A **Bernstein polynomial** $B(x)$ of degree n is written as

$$B(x) = \sum_{i=0}^{n} b_i \binom{n}{i} (1-x)^i x^{n-i}. \tag{3.1}$$

The b_i's are the **Bernstein coefficients**. We are only interested in Bernstein polynomials which are defined over the range $[0, 1]$.

3.1 A test for zeros

The most useful property of Bernstein polynomials is a simple test for zeros. If all the Bernstein coefficients have the same sign, (all strictly positive or all strictly negative), then the polynomial has no zeros between 0 and 1. This is easily proved by noting that $(1-x)^i x^{n-i}$ is non-negative for $x \in [0, 1]$ and $0 \leq i \leq n$. Note the converse does not always hold and it is possible to construct a Bernstein polynomial which has coefficients of different signs but no zeros on $[0, 1]$. We have to check the signs of n coefficients so this algorithm is of order n.

3.2 Derivatives of Bernstein Polynomials

The second useful property is any easy way of finding the derivatives. For the Bernstein polynomial, $B(x)$, above the derivative is

$$B'(x) = \sum_{i=0}^{n-1} n(b_{i+1} - b_i)\binom{n-1}{i}(1-x)^i x^{n-i-1}.$$

This is a Bernstein polynomial of degree $n-1$ with coefficients $n(b_{i+1}-b_i)$. This algorithm is of order n.

3.3 Constructing a Bernstein Polynomial

We need a method for constructing a Bernstein polynomial, $B(y)$, defined over $[0,1]$, from a polynomial $p(x) = \sum a_i x^i$ define over $[x_l, x_h]$. We require $B(y) = p(x)$ where $y = (x - x_l)/(x_h - x_l)$. First we define a sequence $p_i(x)$ of polynomials of degree i which satisfy the properties,

$$\begin{aligned} p_0(x) &= a_n, \\ p_i(x) &= a_{n-i} + xp_{i-1}(x). \end{aligned}$$

Note that the final polynomial in this sequence, $p_n(x)$, is our polynomial $p(x)$. We now convert each of the $p_i(x)$ into polynomials

$$B_i(y) = \sum_{j=0}^{i} b_{i,j}(1-y)^j y^{i-j}.$$

This is not quite a Bernstein polynomial as the binomial coefficients are missing. Now

$$\begin{aligned} B_i(y) &= p_i(x) = a_{n-1} + xp_{i-1}(x) \\ &= a_{n-i} + xB_{i-1}(y) \\ &= a_{n-i}((1-y)+y)^n + (x_l(1-y)+x_h y)\left(\sum_{j=0}^{i-1} b_{i-1,j}(1-y)^j y^{i-1-j}\right). \end{aligned}$$

So the coefficients of B_i are:

$$\begin{aligned} b_{0,0} &= a_n, \\ b_{i,0} &= a_{n-i} + x_l b_{i-1,0}, \\ b_{i,j} &= \binom{n-i}{j} a_{n-i} + x_l b_{i-1,j} + x_h b_{i-1,j-1}, \\ b_{i,i} &= a_{n-i} + x_h b_{i-1,i-1}, \end{aligned}$$

where $1 \leq j < i \leq n$. We can now proceed inductively by degree to get an expression for $B_n(y)$. The final step is to divide by the binomial coefficients to get the required Bernstein polynomial.

3.4 Evaluating a Bernstein Polynomial

We will frequently need to evaluate a Bernstein polynomial (3.1) at a point $x = y$. Let

$$B_{i,j}(x) = \sum_{k=0}^{i} b_{j+k}\binom{i}{k}(1-x)^{i-k}x^k.$$

We observe that

$$\begin{aligned} B_{0,j}(x) &= b_j, \\ B_{i,j}(x) &= (1-x)B_{i-1,j}(x) + xB_{i-1,j+1}(x). \end{aligned}$$

Now, let $b_{i,j} = B_{i,j}(y)$. We have

$$\begin{aligned} b_{0,j} &= b_j, \\ b_{i,j} &= (1-y)b_{i-1,j} + yb_{i-1,j+1}, \end{aligned}$$

for $1 \le i \le n$, $0 \le j \le n-i$. This gives us an inductive process which ends when we have calculated $b_{n,0} = B_n(y) = B(y)$. This routine uses $(n-1)(n-2)$ multiplications to evaluate a Bernstein polynomial of degree n, so the algorithm is of order n^2. A short cut can be taken if we wish to calculate $B(0)$ or $B(1)$ which are b_0 and b_n.

3.5 Splitting a Bernstein Polynomial

The final algorithm we need is to split the polynomial, $B_n(x)$ into two polynomials, $B_n^+(y) = B_n(y/2)$ and $B_n^-(y) = B_n((y-1)/2)$. This will be useful in the recursive sub-division process where to find the solutions in $[0,1]$ we look for the solutions in $[0,\frac{1}{2}]$ and $[\frac{1}{2},1]$. Again we work inductively by degree defining polynomials, $B_{i,j}^+(y) = B_{i,j}(\frac{y}{2})$, $B_{i,j}^-(y) = B_{i,j}(\frac{(y-1)}{2})$. Now

$$\begin{aligned} B_{0,j}^+(y) &= b_j, \\ B_{i+1,j}^+(y) &= B_{i+1,j}(y/2) \\ &= (1-y/2)B_{i,j}(y/2) + \frac{1}{2}yB_{i,j+1}(y/2) \\ &= (1-y)B_{i,j}^+(y) + \frac{1}{2}y\left(B_{i,j}^+(y) + B_{i,j+1}^+(y)\right). \end{aligned}$$

which is a polynomial in y. Similarly

$$\begin{aligned} B_{0,j}^-(y) &= b_j, \\ B_{i+1,j}^-(y) &= \frac{1}{2}(1-y)\left(B_{i,j}^-(y) + B_{i,j+1}^-(y)\right) + yB_{i,j+1}^-(y). \end{aligned}$$

So if we know the coefficients of $B_{i,j}^\pm$ for $0 \le j \le n-i$ we can calculate the coefficients of $B_{i+1,j}^\pm$. This routine is also of order n^2.

3.6 Bernstein Polynomials in three dimensions

In three dimensions the Bernstein basis elements for polynomial of degrees l, m, n in x, y, z, are of the form

$$\binom{l}{i}\binom{m}{j}\binom{n}{k}(1-x)^i x^{l-i}(1-y)^j y^{m-j}(1-z)^k z^{n-k},$$

where $0 \le i \le l$, $0 \le j \le m$ and $0 \le k \le n$. The routines to evaluate a Bernstein polynomial and to split a polynomial into eight are both of order $\max(l^2mn, lm^2n, lmn^2)$. The routines to test for zero and to find the partial derivatives are of order lmn.

4 Drawing the surface

Let $f(x, y, z)$ be a polynomial of finite degree. We wish to draw the set given by $f^{-1}(0)$ inside a bounding box, $[x_l, x_h] \times [y_l, y_h] \times [z_l, z_h]$. We will assume that none of the derivatives vanish at all points of the surface and also that the surface is in general position.

The first step is to rescale the coordinates to fit the unit box $[0, 1] \times [0, 1] \times [0, 1]$, and construct a Bernstein polynomial b using the three dimensional version of the algorithm given in section 3.3.

To get a basic level of approximation we divide the bounding box into a number of smaller sub-boxes. An approximation of the surface will be calculated in each of these sub-boxes.

The three dimensional version of the splitting algorithm given in section 3.5 gives eight Bernstein polynomials each each of which is defined on a box whose sides are half the length of the original box. The splitting process is recursively repeated a predefined number of times. Three levels of recursion were found to give a reasonable approximation which can be calculated in a fairly short time. Furthermore, the resulting model is small enough for the rendering program to handle well and allow near instantaneous rotation of the surface.

4.1 Cellular division of the surface

Within each box we divide the surface into 0-cells, 1-cells and 2-cells. The 2-*cells* consist of points where none of the partial derivatives vanish. Points on the surface where a least one of the partial derivatives vanish make up the 1-cells and 0-cells. If these points form a curve then the curve will be called a *1-cell* otherwise the point will be a *0-cell.* Typically the 1-cells consist of the curves $\frac{\partial f}{\partial x} = 0$, $\frac{\partial f}{\partial y} = 0$, $\frac{\partial f}{\partial z} = 0$, transverse self-intersections of the surface, curves where the surface degenerates to a curve and curves along which two or more of the derivatives vanish. Examples of 0-*cells* are the points where the curves $\frac{\partial f}{\partial x} = 0$, and $\frac{\partial f}{\partial y} = 0$ intersect, isolated points and other isolated singularities. The cross-cap, $x^2z + y^2 = 0$, can be split into four 2-cells, four 1-cells and a 0-cell as shown in Fig. 5.

FIGURE 5. Dividing a cross-cap into cells

A few assumptions about the surface are necessary to avoid degenerate cases. We assume that none of the 0-cells lie on the faces of the box, none of the 1-cells intersect the edges and that no 2-cells intersect the corners of the box. A further assumption is that 1-cells intersect the faces in isolated points which we shall call *nodes.* All these assumptions are satisfied for a surface in general position. Any surface can be put in general position by applying a small translation.

4.2 Approximating the cells within a box

The intersection of each 2-cell with the edges of the box are represented by a set of points. An algorithm for finding these points is given in the next section, (4.3).

The intersection of a 2-cells with a face of the box is approximated by a straight lines whose end points are either solutions on the edge or nodes in the interior. The algorithm to find the nodes is given in section 4.4.

Within each box the 1-cells are approximated by straight lines whose end points are either nodes on the faces of the box or 0-cells lying in the interior. Each 2-cells is approximated by a *facet* whose boundary is made up of a set of straight lines. These lines are either the approximations of the 1-cells or the lines approximating the intersection of the 2-cell with the faces of the box. The algorithm for finding 0-cells is given in section 4.5.

For the cross-cap shown in Fig. 5, the surface will be approximated as shown in Fig. 6. Note that by including points where one or two of the partial derivatives vanish we obtain an approximation which does not truncate the surface.

FIGURE 6. approximating the cross-cap of Fig. 5 by a set of facets and lines

4.3 Finding solutions on edges

The first step in constructing an approximation of the surface is to find the solutions along the edges of the box. These can easily be found using a simple 1-dimensional sub-division algorithm.

We construct the 1-dimensional Bernstein polynomials for the restriction of the function to the edge. If all the coefficients of the Bernstein polynomial are the same sign then we know that there is no solution on the edge (Fig. 7 A). Otherwise we check the partial derivatives. If the Bernstein coefficients are all of one sign for each of the partial derivatives then there is exactly one solution. (Fig. 7 B). This solution can be found by repeatedly sub-dividing the edge and looking for a change of sign. The sub-division is carried out until sub-pixel level is reached. If the coefficients of any of the partial derivative are not all the same sign then we split the edge into two, calculating the new Bernstein polynomials for each of the sub-edges, and repeat the process. (Fig. 7 C).

FIGURE 7. Finding solutions on an edge

4.4 Finding nodes on faces

Another recursive procedure is needed to find the intersection of the 1-cells with the faces of the box. This is a two dimensional algorithm which is computationally slower than the 1-dimensional case. However, not all the faces will need to be tested so the effect on execution time will not be too great. This routine is also used to tell which of the solutions on the edges lie on the same 2-cells to ensure that we get the correct topology.

First we take a face, F, and construct the 2-dimensional Bernstein polynomial, b. We also construct the Bernstein polynomial of the three partial derivative functions. If the coefficients of b are all one sign then the surface does not intersect the face and we can ignore this face. If the coefficients of b are not all of one sign and the coefficients of each of the partial derivative are all of one sign, then there are exactly two solutions on the edges of the face. These solutions will lie on the same 2-cell. If any of the derivatives fail to be all of one sign we divide the face into four sub faces testing each in turn. This process is carried out recursively until a pre-defined depth, typically pixcel level, is reached.

When the bottom level of recursion is reached we might expect a 1-cell to intersect the sub-face in which case we create a node in the center of the face. However, this is not always the case and it will cause problems later if points are incorrectly marked as such. We need to look at the geometry a little more closely to tell if a face contains a node or not.

If only one of the derivatives, say $\frac{\partial f}{\partial x}$, fails to be all of one sign then we test for turning points, e.g. $x^2+y+z=0$, or higher inflections, e.g. $x^n+y+z=0$. Two points, a, b on the edges of the face will belong to the same 2-cell if $\frac{\partial f}{\partial x}(a)$ and $\frac{\partial f}{\partial x}(b)$ have the same sign and $\frac{\partial^2 f}{\partial x^2}$ is all of one sign. If $\frac{\partial f}{\partial x}(a)$ and $\frac{\partial f}{\partial x}(b)$ have different signs then the points lie on different 2-cells. Both 2-cells will be adjacent to a 1-cell passing through the sub-face so we create a node in the middle of the sub-face. If $\frac{\partial f}{\partial x}$ has the same signs for the two points and $\frac{\partial^2 f}{\partial x^2}$ is not all of one sign then we cannot always distinguish whether the two points lie on the same 2-cell. We err on the side of caution and assume they do not, creating a node in the middle of the sub-face. It is possible to have four solutions lying on the edges and here we check the solutions pairwise to see which lie on the same 2-cells. In the example shown in Fig. 8 the partial derivative $\frac{\partial f}{\partial x}$ will have different signs at the points a, b so a node lies within face A. On face B the points a and c have the same sign of $\frac{\partial f}{\partial x}$, points b and d also have the same signs. Hence two 2-cells intersect the face and there are no nodes on the face. On face C points d and e have the same signs of $\frac{\partial f}{\partial x}$ but $\frac{\partial^2 f}{\partial x^2}$ vanishes so we assume that a node lies on the face.

FIGURE 8. Faces where one of the partial derivatives vanishes

In the case when two of the derivatives, say $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ fail to be all of one sign we first test that all the second derivatives, $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial xy}$ and $\frac{\partial^2 f}{\partial y^2}$ are all of one sign and that the curves $\frac{\partial f}{\partial x}=0$ and $\frac{\partial f}{\partial y}=0$ do not intersect. If the last condition fails then the intersection with the x-y-plane may form an isolated point, e.g. $x^2+y^2=0$, or two intersecting curves, e.g. $x^2-y^2=0$. We can test this condition by finding the two solutions to $\frac{\partial f}{\partial x}=0$ round the edges and seeing if $\frac{\partial f}{\partial y}$ has the same sign at the two points. If any of the above fail then it is assumed that there is a node in the interior of the face. If the above conditions are all satisfied we check the solutions pair-wise to see if their derivatives have the same sign, if so they must lie on the same 2-cell. In the examples shown in Fig. 9 the points a, b, c all have the same sign for each of the partial derivatives. The points d, e, f also have the same sign. The curves $\frac{\partial f}{\partial x}=0$ and $\frac{\partial f}{\partial y}=0$ intersect on face B but not in A hence a node lies on face B but not on face A.

FIGURE 9. Faces where two partial derivatives vanish

The case where all of the partial derivatives fail to be all of one sign follows similarly to above. This is actually more likely as it is the condition for a self-intersection of the surface to pass through the face.

4.5 Finding 0-cells in boxes

Once we have calculated an approximation for each of the faces of a box, B, we work out how the nodes are joined within the box and find any 0-cells in the box. Again we use a recursive procedure; this time it works in three dimensions so can be expensive. However, we only expect a small number of 0-cells, so the routine will not be used very often. To make it a little faster we stop the recursion a little above pixcel level, but not so it is apparent to the naked eye.

Again for each box we test whether the coefficients of the Bernstein polynomial are all of one sign in which case there is no component of the surface in the box. If the coefficients of the Bernstein polynomials for each of the partial derivatives are all of one sign then there are no 0-cells or 1-cells in the box so we can ignore the box. If one or more of the partial derivatives fail to be all of one sign we count the number of nodes on the faces of the box. If there are two nodes for which the signs of the partial derivatives match, either both zero, both positive or both negative, then we assume that they both lie on the same 1-cell. If there are four nodes, we check the signs of the partial derivatives pairwise so see if they lie on the same 1-cell. Isolated points only occur if all three derivatives are zero so we can also disregard boxes where there are no nodes on the faces and one of the derivatives is all of one sign. If none of the above happen we cannot easily tell what happens in the box so we sub-divide it into eight sub-boxes, calculating the positions of the solutions and nodes on the edges and the faces of the sub-boxes. When the bottom level is reached then the sub-box is assumed to be a 0-cell, any of the 1-cells which pass through the faces of the box are assumed to have the 0-cell as an end point.

5 Conclusion

This program has already shown its usefulness in producing accurate models of surfaces with isolated singularities in a matter of minutes. For smooth surfaces the routine is even quicker running in under a minute. The pictures produce are not quite the same quality as those produced by a ray tracer, but the advantages of producing three dimensional models within seconds means the program is a very useful as a mathematical tool.

Problems start to arise when we consider cuspidal edges or more exotic surfaces. Along a cuspidal edge several of the higher order partial derivatives vanish, causing some of the tests to give incorrect results. Even here the program will produce a good model of the surface away from the singularities and a user can quickly tell what type of singularity you have. A "dust" of isolated points may surround the cuspidal edges as shown in Fig. 4. Running times can slow down dramatically as the routine to find isolated points is computationally expensive. The routine also has difficulties if the surface contains planes parallel to the sides of the boxes.

If we look at the mathematics behind the algorithm we find that the tests for nodes and 0-cells are too weak and the test for whether two nodes belong to the same 1-cell is too strong. These errors, may give rise to spurious points or facets which have the wrong vertices. Some improvements could be made by examining the higher derivatives but this would also slow the program down considerably. This will not solve all the problems as all the partial derivatives of the functions $f(x,y,z)+c=0$ are identical for all values of c, but the resulting surfaces may have different topologies. Checking the signs of the Bernstein coefficients does not always help as the coefficients may have different signs even when the function has no solutions. Some other test, for example the generalization of Sterm sequences to two and three dimensions discussed in [6], could be used to give a foolproof algorithm.

We have not mentioned many of the subtleties of the implementation, which improve performance. For example the solutions on edges and faces are only found once. These solutions being used each time the edge or face is examined.

The author was supported by a grant from the S.E.R.C.

References

[1] V. I. Arnold, 1981: Singularity Theory. *London Mathematical Society Lecture Notes* **53**.

[2] D. P. Dobkin, S. V. F. Levy, W. P. Thurston, A. R. Wilks, 1988: Contour Tracing by Piecewise Linear Approximations I & II. *Research report, Geometry Supercomputer Project, University of Minnesota.*

[3] A. Geisow, 1982, Surface Interrogation, Ph.D. thesis, University of East Anglia.

[4] C. G. Gibson, 1979: Singular points of smooth mappings. *Research Notes in Mathematics*, **23**.

[5] D. Kalra, A. H. Barr, 1989: Guaranteed Ray Intersection with Implicit Surfaces. *Computer Graphics*, **23(3)** 297–304.

[6] P. Milne, 1991, Zero set of Multivarient Polynomial Equations, *Mathematics of Surfaces IV.*