

The use of computer graphics for solving problems in singularity theory

R.J. Morris

November 17, 1997

Abstract

We explore two investigation in singularity theory in which mathematical visualisation played an important part in the proof. We also describe a computer package which has been used to aid the experimental investigation of singularity theory and outline some of the computational problems involved in rendering singular surfaces.

1 Introduction

Many problems in differential geometry and singularity theory naturally lend themselves to graphical solutions. For example there is considerable interest in calculating ridges and sub-parabolic lines various surfaces which can be readily visualised. In other more theoretical problems, such as classifying the singularities of maps, graphical solutions can play an important part in the solution. Some of the applications in which visualisation have already helped include: symmetry sets and rotational symmetry sets of plane curves [5, 10]; ridges and sub-parabolic lines of surfaces [1, 10, 16]; maps from \mathbf{R}^2 to \mathbf{R}^2 [7]; binary differential equations [3]; duals of surfaces [2, 18]; robotic grasp [6]; cusp tracking [4]; and quartic curves [17].

In section 2 we will look at a set of programs we have developed to tackle the basic graphical problems which occur in singularity theory. We will discuss some of the user interface issues as well as the algorithms used to create accurate representations of singular surfaces.

In section 3 we will look at the role experimental results have to play in the development of mathematical proof. We will examine two cases studies which are good examples of this experimental method and illustrate some of the techniques needed for successful experimental analysis. In both cases there was a conflict between mathematical conjectures and the results generated by computer

graphics. This conflict caused both the mathematics and the experimental results to be reexamined leading to a resolution of the problems. In the first case a the experimental results were shown to be correct and in the second both methods were partially correct and the final results had a much richer structure.

2 A toolkit for exploring singularity theory

Singularity theory provides a wide range of problems for visualisation. However, a little analysis shows that most of these can be solved by a relatively small number of programs. We have developed a set of modules called the *Liverpool Surface Modelling Package* which tackle most of basic problems encountered. Each module has been written as a separate program, with an easy to use graphics interface. Equations can be entered in an editor window, and are evaluated using a reverse polish calculator. The modules automatically display any parameters which the equations depend on and these can be changed using the mouse. The whole package uses *Geomview* [15] as an underlying platform for visualising the results and communicating between the modules.

2.1 The modules

The modules fall into a number of different types:

acurve, acurve3, asurf These modules calculate algebraic curves in 2 and 3 dimensions and algebraic surfaces in 3 dimensions. The algorithms behind these modules are discussed in section 2.5.

pcurve, psurf Calculates parameterised curves and surfaces in three and four dimensions. Both programs can quickly recalculate a curve or surface each time a parameter is changed.

mappings Given a geometric object S and a map $f : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ the *mapping* program calculates $f(S)$. The mapping can be applied to any object in geomview, which allows quite complicated situations such as discriminants (see section 3.1) to be studied.

intersect Calculates the intersection of geometric object read in from geomview with either $f^{-1}(0)$, $\{x : f(x) \geq 0\}$ or $\{x : f(x) \leq 0\}$ where $f : \mathbf{R}^3 \rightarrow \mathbf{R}$. This uses a simpler algorithm than the algebraic

curves program, and it may not produce correct representations of singularities, however it can find the zeros of non-polynomial equations. Surfaces can be trimmed by finding the intersections of the surface with a half-space, which can improve the visual effect.

dualise The set of planes tangent to a surface S is called the *dual* of the surface. We can represent the dual as a surface in $\mathbf{R}P^3$, if (a, b, c) is a point on the surface with normal (l, m, n) then the dual of the tangent plane at that point is $(l, m, n, al + bm + cn)$. Taking the union of all such duals points forms the surface. The *dualise* program calculates duals in $\mathbf{R}P^3$ and performs various projections from $\mathbf{R}P^3$ into \mathbf{R}^3 which enable the dual to be visualised. Full details of the algorithm can be found in [13]. Duals capture considerable geometric information about the original surface and the results of this program have been used in a number of investigations [2] [18].

icurve Given a set of differential equations of the form

$$\frac{\partial x}{\partial t} = f(x, y, z), \frac{\partial y}{\partial t} = g(x, y, z), \frac{\partial z}{\partial t} = h(x, y, z),$$

the *icurve* program calculates the integral curves, vector field and wavefronts of these equations. Starting points for the integral curves can be read in from objects created by the other modules in the package. An option allows un-oriented vector field like principal directions and asymptotic directions to be handled. These are defined by binary differential equations [3] of the form

$$a(x, y) \left(\frac{\partial x}{\partial t} \right)^2 + b(x, y) \left(\frac{\partial x}{\partial t} \right) \left(\frac{\partial y}{\partial t} \right) + c(x, y) \left(\frac{\partial y}{\partial t} \right)^2 = 0.$$

We can solve the quadratic to give to give formula for $\frac{\partial x}{\partial t}$ and $\frac{\partial y}{\partial t}$. However this equation just defines an un-oriented line a line rather than an un-oriented direction and we need special algorithms to cope with these un-oriented vector fields, these are discussed in [14].

2.2 Graphical User Interface

A common style of graphical user interface has been developed for all the modules. The interface for the parameterised surface program is shown in figure 1. The main window (top left) allows the range of the two variables to be changed as well as the number of steps making up the grid and several other options. An equation can be entered in the editor window at the bottom of the figure. If the equation depends

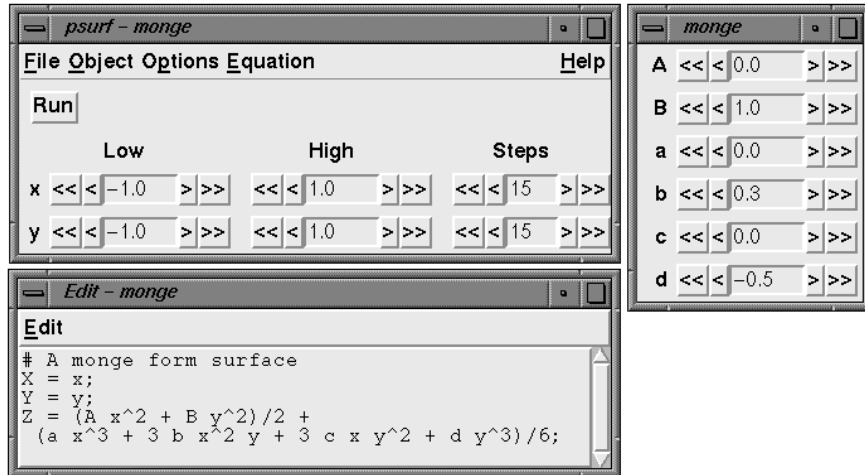


Figure 1: The user interface for the parameterised surface program.

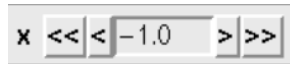


Figure 2: A widget for controlling values of parameters.

on any parameters a small window (right) allows the parameters to be interactively changed. The parameterised curves and surface programs are fast enough to recalculate the surface when ever a parameter is changed, the other programs require the **Run** button to be pressed each time the user wants to recalculate the surface.

The widget used for controlling the values of parameters is a development of the standard counter device and is shown in figure 2. Pressing the <<, <, > or >> buttons by the side of the number changes the value by -10, -1, 1 or 10 units in the last decimal place shown. This allows more precise control of the value of the number than sliders or dials which is important for investigations in singularity theory. Often it is necessary to change the value by a very small amount, for instance when trying to find the exact value at which a transition occurs. This can be accomplished by changing the value of the **precision** menu option which alters the number of decimal places shown and hence the amount the numbers change with each button press.

2.3 Intermodule Communication

Much of the power of the package comes from the way in which results data from one module can be used as input in another. We have used a simple scheme where each module reads and writes data to a common pool of objects maintained by *Geomview*. For example to perform a mapping on an object S , the object is first selected in *Geomview*, then the **Run** button in the *mapping* program is pressed. The object is read into the *mapping* program and the image is calculated. The result is then written out, and a new object is created in *Geomview*. This allows the same mapping to be applied to several objects and for any combinations of operations to be applied as the user wishes; one or two mouse clicks is normally enough to perform each operation. This method is both simple to use and very flexible. Most problems in singularity theory only require a few operations and we felt this did not justify the added complexity of using more sophisticated schemes such as the data flow methods used by packages such as *Explorer* and *Orange*.

2.4 Handling Equations

Most of the modules allow T_EX-style equations to be entered in a special editor window. These equations are first converted to a binary tree structure where each node is either a variable name a number or an operator such as +. This tree is either converted to an array of polynomial coefficients which are used by the algebraic curves and surface programs, or to a string of instructions for use by a reverse polish calculator. The calculator interpretes each instruction in turn, and performs operations on a stack, such as pushing a number onto the stack or adding the two numbers at the top of the stack. For example the formula $x + y$ would be represented by the string **xy+**. First the values of x and y are pushed onto the stack, then the top two numbers in the stack are added producing the required result. This calculator is heavily used in some of the modules, is very time critical. For example calculating a parameterised surface with a 20 by 20 grid of points, will require 3600 evaluations, three calculations to find the position of each point and six to find each normal. Considerable work has been put into optimising this calculator; which enables such parameterised surface to be calculated in near real time.

We are likely to encounter some quite complex formula such as the condition for a parabolic line on a parameterised surface $s(x, y)$. This is $lm - n^2 = 0$ where $l = \frac{\partial^2 s}{\partial x^2} \cdot \vec{N}$, $m = \frac{\partial^2 s}{\partial x \partial y} \cdot \vec{N}$, $n = \frac{\partial^2 s}{\partial y^2} \cdot \vec{N}$, and \vec{N} is a unit normal to the surface calculated by taking the vector

product $\frac{\partial s}{\partial x} \wedge \frac{\partial s}{\partial y}$ and dividing by its length. To enable such formula to be easily specified, we have made a number of extensions to the standard mathematical syntax:

1. The equation can be split up into several lines, where the first line gives the top level definition. All the subsequent lines are automatically substituted into the first line to make up the equation.
2. Multiplication signs are not needed between variables, a space is all that is needed to avoid confusing $\mathbf{x} \mathbf{y}$ with the single variable \mathbf{xy} . This is more akin to what might be seen on the printed page.
3. Equations can be written in terms of vectors. A vector is written as $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ and we use $.$ for the dot product, \wedge for the vector product and $*$, $/$ for multiplication and division by scalars. We see that the \wedge sign now has two meanings, this does not cause any problems as its meaning can be inferred from the type of its arguments.
4. The symbolic differential operator $\mathbf{diff}(f, \mathbf{x})$ can be used to represent the derivative of a function f with respect to the variable x .
5. Conditional statements like $\mathbf{if}(\mathbf{a}, \mathbf{b}, \mathbf{c})$ can also be used. If $a \geq 0$ then this has the value b otherwise it gives the value c .

As well as improving the clarity of the definition, the chance of errors is also reduced, especially as the user does not need to calculate a derivative in his head. Using these features enables the parabolic condition to be written as

```

1 n=m^2;
l = diff(S_x,x).N; m = diff(S_x,y).N; n = diff(S_y,y).N;
# Produces a unit length vector
N=N1/sqrt(N1.N1);
# Use tangent vectors and vector product to find the normal
N1 = S_x ^ S_y;
S_x = diff(S,x); S_y = diff(S,y);
#definition of the surface
S = (x,y,x y + x^2);

```

2.5 Algebraic Curves and Surfaces

One of the hardest tasks of all the modules in the package is finding $f^{-1}(0)$ where $f : \mathbf{R}^m \rightarrow \mathbf{R}^n$ and each of the component functions of f is a polynomial. We are particularly interested in the cases $m = 2$, $n = 1$ (algebraic curves in the plane); $m = 3$, $n = 1$ (algebraic

surfaces in \mathbf{R}^3); $m = 3, n = 2$ (algebraic curves in \mathbf{R}^3). The case $m = 4, n = 3$ is also of interest as this enables us to calculate the intersection of two parameterised surfaces.

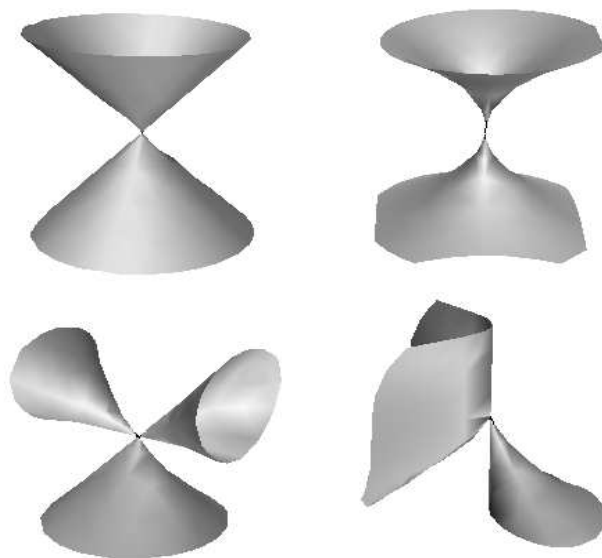


Figure 3: Some algebraic surfaces, top row $x^2 + y^2 - z^2 = 0$, $x^2 + y^2 - z^4 = 0$, bottom row $x^2z - z^3 - y^2 = 0$, $x^3 + xz^3 - y^2 = 0$.

Examples from singularity theory pose particular problems as the structure around singular points can be quite complicated, as can be seen in figure 3. These pictures have been obtained using the algebraic surface program which is briefly discussed below, we also discuss extensions to the other cases. This algorithm uses a recursive technique based on Bernstein polynomials, which was first used by Geisow [8] to find algebraic curves in the plane. Full details of the algebraic surface algorithm can be found in [11].

2.5.1 Bernstein polynomials

All the routines rely heavily on Bernstein polynomials which offer an easy test for finding possible zeros of a polynomial. To simplify the discussion of these polynomial we will look at the one-dimensional case. All the results below can be easily adapted to work for higher dimensions. Efficient algorithms for manipulating Bernstein Polynomials can be found in [9].

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

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

The b_i 's are the *Bernstein coefficients*. We are only interested in Bernstein polynomials which are defined over the range $[0, 1]$. 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 $[0, 1]$. In two dimensions the Bernstein basis elements for polynomial of degrees l , m , in x and y , are of the form

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

where $0 \leq i \leq l$ and $0 \leq j \leq m$. The three and four dimensional cases follow similarly.

2.5.2 Algebraic Surfaces

The first step in calculating an algebraic surface is to re-scale the domain of interest until it fits into the unit cube and then construct Bernstein polynomial for f and the three partial derivatives. Then we split the unit cube into a eight of smaller cubes, constructing new Bernstein polynomials for each cube and testing the signs of the Bernstein coefficients to eliminate those cubes which the surface does not pass through. We repeat this step a number of times until a user specified resolution is reached and we end up with a number of smaller cubes which contain the surface. For each of the smaller cubes we find three types of points:

1. The intersections of the surface with the edges of the cube.
2. Points on the surface of the cube where $f = 0$ and a least one of the partial derivatives $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ or $\frac{\partial f}{\partial z}$ vanish.
3. Points in the interior of the cube where $f = 0$ and at least two of the partial derivatives vanish. If all three derivatives vanish we will have a singularity of the surface.

All the points are found using recursive sub-division algorithms and the points found are linked together to form the facets approximating the surface. Finding points where one of the partial derivatives vanish helps to reduce problems with the surface being rounded off.

2.5.3 Algebraic Curves

A similar scheme is used for algebraic curves in the plane. Here we divide the unit square into smaller squares and find solutions on the edges. We also find turning points in the interior of the squares where either $\frac{\partial f}{\partial x}$ or $\frac{\partial f}{\partial y}$ vanish and singularities where both vanish.

Algebraic curves in \mathbf{R}^3 can be thought of as the intersection of two algebraic surfaces, $f^{-1}(0)$ and $g^{-1}(0)$ say. The tangent to the curve is

$$(l, m, n) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \wedge \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}, \frac{\partial g}{\partial z} \right)$$

and this gives three conditions $l = 0$, $m = 0$, $n = 0$ for turning points. These conditions also pick out the singularities which can only happen whenever $f^{-1}(0)$ and $g^{-1}(0)$ are tangent or one is singular. This implies that all of l , m and n are zero.

For algebraic curves in four dimensions we use a technique similar to above. We use the 4D analogue of the cross product to find the tangent to the curve, and hence its turning points and singularities. We calculate

$$\begin{pmatrix} k \\ l \\ m \\ n \end{pmatrix} = \begin{vmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} & \frac{\partial f}{\partial w} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} & \frac{\partial g}{\partial z} & \frac{\partial g}{\partial w} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial z} & \frac{\partial h}{\partial w} \\ \vec{i} & \vec{j} & \vec{k} & \vec{l} \end{vmatrix},$$

to give the tangent, where \vec{i} , \vec{j} , \vec{k} and \vec{l} are the Euclidean basis vectors. The conditions for turning points are $k = 0$, $l = 0$, $m = 0$ and $n = 0$.

3 Two problems in Singularity Theory

In this section we will study two examples in which experimental results proved important solving particular mathematical problems. We shall also discuss some of the problems which are frequently encountered in experimental investigations in singularity theory.

3.1 Co-rank 2 maps

Our first example has application in studying the configuration spaces of mechanisms, a full account is given in [7]. The general question is to classify maps from the plane to the plane up to diffeomorphism in the source and target. The simplest models are shown in table below. Most maps will be equivalent to one of these models.

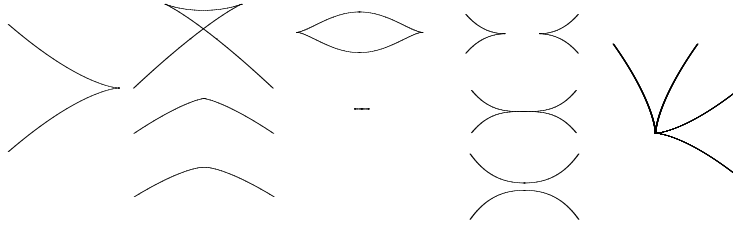


Figure 4: Discriminates for the cusp, swallowtail, lips, beaks and sharksfin maps.

Type	Normal Form
Immersion	$f : (x, y) \rightarrow (x, y)$
Fold	$f : (x, y) \rightarrow (x, y^2)$
Cusp	$f : (x, y) \rightarrow (x, xy + y^3)$
Swallowtail	$f : (x, y) \rightarrow (x, xy + y^4)$
Lips	$f : (x, y) \rightarrow (x, y^3 + x^2y)$
Beaks	$f : (x, y) \rightarrow (x, y^3 - x^2y)$
Butterfly	$f : (x, y) \rightarrow (x, xy + y^5 \pm y^7)$
Goose	$f : (x, y) \rightarrow (x, y^3 + x^3y)$
Gulls	$f : (x, y) \rightarrow (x, xy^2 + y^4 + y^5)$
Sharksfin	$f : (x, y) \rightarrow (x^2 + y^3, y^2 + x^3)$
Deltoid	$f : (x, y) \rightarrow (x^2 - y^2 + x^3, xy)$

We can visualise the structure of these maps by considering the critical sets and discriminants. The map $f : \mathbf{R}^2 \rightarrow \mathbf{R}^2$, f is said to be *singular* at a point (x, y) if the Jacobian has rank less than 2. The set of points (x, y) in the source where f is singular is called the *critical set* and its image under f is called the *discriminant*. The discriminants of some of the normal forms are shown in figure 4. The swallowtail, beaks and lips typically occur in one parameter families, and typical transitions are shown in the figure.

The particular question we will look at is what happens when we slightly deform the sharksfin map. All the possible deformations are exhibited in the two parameter family

$$f_{a,b} : (x, y) \rightarrow (x^2 + y^3 + ay, x^3 + y^2 + bx).$$

We wish to find the examples of discriminants for various values of a and b close to $(0, 0)$ and also find the transitions which can occur in this family.

The first graphical problem we encountered is one of size and scale. A particular feature of a curve may well be very small and

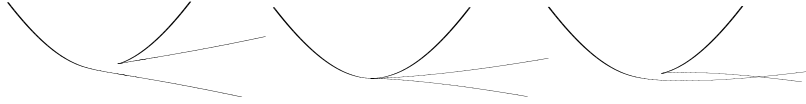


Figure 5: The first experimental results obtained.



Figure 6: Several thousand fold magnification reveals the correct transitions.

easily obscured by the other components. Related to this is sensitivity of parameters, a small change can lead to a drastic change in the topology of the curves. Both these problems occurred in our example. The first experimental pictures gave misleading results and are shown in figure 5. Here we have taken $a = 0.1$ and b ranges from -0.01 to 0.01 .

As this transition did not fit into our standard list we were suspicious of the results. We used an iterative process to deduce what was really happening. At each stage in the process we first magnified the image and then reduced the both the range of the parameter b and size of the domain, so that we homed in on the point of interest. After a number of iterations we produce the transition shown in figure 6. We can see the transition is actually composed of a beak transition and a swallowtail transition. Here the scaling is much larger with a magnification by a thousand in the x direction and fifty thousand in the y direction. At this scale we only need to use a very small range and a small variation of the parameter b which ranges from -0.0002 to 0.0002 .

These results were particularly interesting as it was thought that no swallowtail transition would occur for sufficiently small values of a and b . The pictures suggested that such a transition does occur. There are still some questions which these pictures do not answer. Does a swallowtail transition actually occur, or is there still a very tiny swallowtail shape which we can not see even at this resolution?

Was the value of a small enough, do we still have the same transitions for all sufficiently small a greater than zero.

The first question can be answered by finding the points on the critical set which correspond to cusps in the discriminant. We can find such points by finding the intersection of the critical set, $g(x, y) = 0$ with the curves

$$\begin{aligned} h_1 &= \frac{\partial f_1}{\partial x} \frac{\partial g}{\partial y} - \frac{\partial f_1}{\partial y} \frac{\partial g}{\partial x} = 0 \\ h_2 &= \frac{\partial f_2}{\partial x} \frac{\partial g}{\partial y} - \frac{\partial f_2}{\partial y} \frac{\partial g}{\partial x} = 0, \end{aligned}$$

where $f(x, y) = (f_1(x, y), f_2(x, y))$. When all three curves intersect we will have a cusp or worse on discriminant. A swallowtail transition would be represented by two points of intersection coming together and then vanishing. Computer generated examples do indeed show that such a transition does occur.

We can get some way to answering the second question by converting the problem to one of studying a one parameter family of surfaces rather than a two parameter family of curves. Let $a = r \cos(\theta)$, $b = r \sin(\theta)$ and define a family of maps

$$F_\theta(x, y, r) \rightarrow (x^2 + y^3 + r \cos(\theta)y, x^3 + y^2 + r \sin(\theta)x, r).$$

The discriminant of each map will give a singular surface S in \mathbf{R}^3 . Taking slices through S for a constant value of r will give us one of the original discriminant curves. We can think of this surface as a whole set of discriminant curves stacked one on top of each other. In effect this enables us to study all values of a at the same time. The results did suggest that we had a swallowtail transition for all values of a which was later confirmed mathematically by N. P. Kirk.

3.2 Sub-parabolic lines and problems with genericity

Let C_α be a family of curves or surfaces which depend on a finite or infinite number of real parameter. A property P of the family of objects is *Generic* if for any object C_α for which P holds then for all β sufficiently close to α the property P also holds for C_β . We also say that C_α is generic with respect to P . Typically we omit a precise definition of P and just say that C_α is generic with respect to all the important properties. For example transverse crossing of two curves is a generic property, slightly deforming the curves will still produce a crossing. Tangency of two curves is not generic, deforming the curves will typically produce either two transverse crossings or no crossing at all. However, tangency is generic in a one parameter family of curves.

Genericity poses one of the main problems for an experimental investigation. Whilst we can easily provide an example we must establish whether the example is generic or is special in some way. To illustrate some of these problems we will examine our second example, sub-parabolic lines on a surface. Let

$$s : (x, y) \rightarrow (x, y, (Ax^2 + By^2)/2 + (ax^3 + 3bx^2y + 3cxy^2 + dy^3)/6 + O(4))$$

be a Monge form parameterisation of a surface. At all points on the surface we can define a normal N which is orthogonal to the tangent plane at the point. At a generic point on the surface we can find two *principal directions* P, Q , and two corresponding *principal curvatures* κ_p, κ_q such that the normal curvature of a curve through x with tangent to P or Q is κ_p or κ_q respectively. Furthermore, κ_p and κ_q are the extremal values of the normal curvatures for all smooth curves through x . (The normal curvature of a curve on S with tangent V is the curvature of the intersection of the surface and the plane containing the normal and V .) At some isolated points called *umbilics* every tangent direction is principal with the same principal curvature. At such points the surface closely approximates a sphere. We define a *ridge* to be the set of points such that $d\kappa_p < P > = 0$. And a *sub-parabolic line* to be points such that $d\kappa_q < P > = 0$. Here $d\kappa_q < P >$ denotes the derivative of $d\kappa_q$ in the direction P . If $P = (u, v)$ then

$$d\kappa_q < P > = u * \left(\frac{\partial \kappa_q}{\partial x} \right) + v * \left(\frac{\partial \kappa_q}{\partial y} \right)$$

where $v = (v_1, v_2)$. Both the ridges and sub-parabolic lines generally form curves on the surfaces which hold important geometric information about the surface [1, 12].

Here we are interested in finding the transformations which occur on the sub-parabolic lines in the *birth of umbilics* transition. In this transition two umbilics come together at a single point and then annihilate. The first examples produced [10] pp 154-155, showed a transition where five sub-parabolic lines pass through the degenerate umbilic at the moment of transition, however two of the lines were tangent. At the time of the very little was known about the transition being studied so we did not know whether the tangency always occurred in the transition or if it was some artifact of the particular family of surfaces being used.

In fact the pictures turned out to be non-generic and another family (Fig. 7) was found which did not contain the tangency, on examining the equation we note that one of the fourth order terms was zero. We still did not know whether these new pictures were generic,

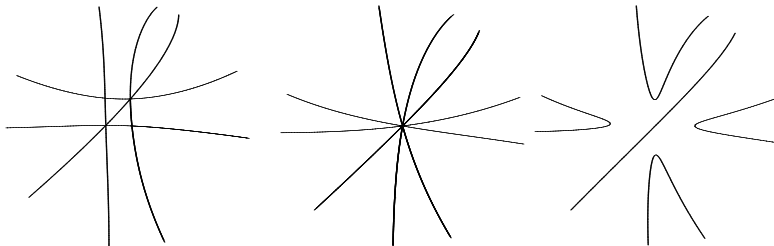


Figure 7: One form of the birth of umbilics transition.

there still could be some non-generic property which depended on the particular family chosen, for instance all the fifth order terms were zero, did this have an effect on the pattern of curves? No matter how many examples we generate there is still a chance that they all share some relation amongst the parameter to create non generic results.

To help resolve the genericity problems a more theoretical investigation [1] was undertaken. The first results from this investigation indicated that there should be three lines through the transitional umbilic, which conflicted with the experimental results. If the experimental results were non-generic it should be possible to find a slight perturbation of the results which did result in a generic situation, but we could not find any such perturbation, indeed we could not think of a way in which the five line transition could actually be deformed to form a three line transition.

Further mathematical and experimental analysis provided the solution to the problem. Rather than just being one generic transition there were actually five different generic transitions. One with 1 line through the transitional umbilic, one with 3 lines and three different transitions with 5 lines. Which particular transition we obtained depended on inequalities amongst the fifth order coefficients. This illustrates another point which needs to be considered when trying an experimental investigation: have we found all the cases? This can be an especially difficult problem as the range of parameter which give a particular case can be very small and hard to find by just picking random values for the parameters.

References

- [1] J. W. BRUCE, P. J. GIBLIN, F. TARI, *Families of surfaces in Euclidean 3-Space, Focal Sets, Ridges and Umbilics*. Preprint, University of Liverpool 1994.
- [2] J. W. BRUCE, P. J. GIBLIN, F. TARI, *Families of Surfaces: Height Functions, Gauss Maps and Duals*, in “Real and Complex Singularities”, Ed. W. L. Marar, Pitman Research Notes in Math. 333, 1995.
- [3] J. W. BRUCE, F. TARI, *On Binary Differential Equations*, Nonlinearity 8 (1995), 255-271.
- [4] R. CIPOLLA, G. J. FLETCHER, P. J. GIBLIN, *Following Cusps*, To appear Int. J. of Computer Vision.
- [5] P. J. GIBLIN, F. TARI, *Local Reflectional and Rotational Symmetry in the Plane*, Singularity Theory and its Applications, Springer Lecture Notes in Math. 1462, 154-171.
- [6] C. DESMOND, *Grasp in 2 and 3 dimensions: theory and simulated experiments*, M.Sc Thesis, University of Liverpool, 1995.
- [7] C. G. GIBSON, W. HAWES AND C. A. HOBBS, *Local Pictures for General Two-Parameter Motions of the Plane*. In “Advances in Robot Kinematics and Computational Geometry”, Kluwer Academic Publishers (1994) 49-58.
- [8] A. GEISOW, *Surface Interrogation*, Ph.D. thesis, 1982, University of East Anglia.
- [9] P. MILNE, *On the Algorithms and Implementation of a Geometric Algebra System*, Computer Science Technical Report 90-40, University of Bath 1990.
- [10] R. J. MORRIS, *Symmetry of Curves and the Geometry of Surfaces: Two Explorations with the aid of Computer Graphics*, Ph.D. Thesis, University of Liverpool, 1990.
- [11] R. J. MORRIS, *A New Method for Drawing Algebraic Surfaces*, in “Design and Applications of Curves and Surfaces”, Ed R.B. Fisher, Clarendon Press, Oxford, 1994.

- [12] R. J. MORRIS, *The Sub-Parabolic Lines of a Surface*, to appear Proc. Sixth I.M.A. conference on the Mathematics of Surfaces, Brunel University, Sept 1994.
- [13] R. J. MORRIS, *Visualising Duals of Surfaces*, preprint, University of Liverpool, 1995.
- [14] R. J. MORRIS, *Visualising Binary Differential Equations*, preprint, University of Liverpool, 1996.
- [15] M. PHILIPS, *Geomview reference manual*, Geometry Center research reports GCG 62, Geometry Center, Minneapolis.
- [16] M. J. PUDDEPHAT, *Robust Features of Bumpy Spheres*, M.Sc Thesis, University of Liverpool, 1994.
- [17] C. T. C. WALL, *Real rational quartic curves*, in "Real and Complex Singularities", Ed. W. L. Marar, Pitman Research Notes in Math. 333, 1995.
- [18] J. M. WEST, *The Differential Geometry of the Cross-cap* Ph.D. Thesis, University of Liverpool, 1995.