The Bivariate Contouring Problem *

Thomas Grandine† Bogdan Craciun‡ Noel Heitmann§ Brian Ingalls¶
Quoc Thong Le Gia‖ Miao-jung Ou** Yen-hsi Richard Tsai††

July 28, 2000

Abstract

An algorithm is presented for determining a connected component of the zero level set of a function $f : \Omega \to \mathbb{R}^{n-2}$, where Ω is a bounded subset of $\mathbb{R}^n$. Two different numerical methods are employed and an error estimate procedure is indicated. Some examples that suggest possible applications are presented.

1 Problem Description

The problem of finding the zero level set of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ (with $m < n$) frequently occurs in practice. The case $m = n - 1$ has been extensively studied numerically (see [2] or [3]), while the study of other cases is still in an incipient phase.

The present report is concerned with the case $m = n - 2$. There are several motivations for studying this problem, such as the problem of finding the intersection of hypersurfaces in $\mathbb{R}^4$ (which could be trajectories in time of evolving three dimensional surfaces) or generating a parametric representation of an implicitly defined manifold. Another possible motivation comes from the need to determine the surface envelope of a swept volume, which is defined in the following way: assume that we are given a surface that is moving or deforming in time

$$S : D \times [0, T] \to \mathbb{R}^3 \text{ where } D \in \mathbb{R}^2 \text{ is bounded;}$$

the surface envelope is the boundary of the set

$$\{x \in \mathbb{R}^3 : \exists t \in [0, T] \text{ with } x \in S(D, t)\}.$$

*This work was done while the authors were participating in the Mathematical Modelling in Industry Workshop for Graduate Students at the Institute of Mathematics and its Applications at the University of Minnesota

†The Boeing Company (**thomas.a.grandine@boeing.com**)
‡California Institute of Technology (**santa@ama.caltech.edu**)
§University of Pittsburgh (**wallop@pitt.edu**)
¶Rutgers University (**ingalls@math.rutgers.edu**)
‖Texas A&M University (**qlegia@math.tamu.edu**)
University of Delaware (miao@udel.edu**)
††University of California, Los Angeles (**ytsai@math.ucla.edu**)

Before describing the problem, it should be mentioned that several simplifying assumptions have to be made to ensure an attainable goal due to the time contraint of this workshop. For this reason, we assume solutions to other problems that would be contained in a fully realized algorithm, as will be noted in the following. These problems are mathematically interesting and will surely be the subject of future research.

In mathematical terms, our problem can be set as follows. Given a smooth function $f : \Omega \to \mathbb{R}^{n-2}$ (where Ω is a bounded subset of $\mathbb{R}^n$), we wish to parametrically represent the set $\{x \in \mathbb{R}^n \ : \ f(x) = 0\}$ as the image of a function $x : [0,1]^2 \to \mathbb{R}^n$ (for simplicity, we have chosen our parametric domain to be $[0,1]^2$ here).

In addition to this, we need boundary conditions. Designing appropriate boundary conditions could be accomplished by an algorithm similar to the one described in [3] for the codimension equals one case. We will simply assume that we are given proper boundary conditions, i.e. the prescribed data lie on a connected subset of the zero level set of f and that a consistent solution exists.

Another difficult matter is uniqueness. Lack of uniqueness may result from two different sources. One is the possibility of several manifolds included in the zero level set of f sharing the same (given) boundary. Such cases are often met in practice, as suggested by simple examples such as

$$f : [-2,2]^3 \ \to \ \mathbb{R}, \ f(x,y,z) = (x^2+y^2+z^2-1)(2x^2+y^2+z^2-1).$$

We will assume that this is not the case (being avoidable by choice of appropriate boundary conditions). This and the assumption made in the previous paragraph ensure uniqueness of the manifold solution.

However, non-uniqueness of x will automatically follow from the fact that generically this manifold has an infinite number of parametrizations. To single out a unique one, we need to impose additional conditions. Among many possibilities, we considered the following two:

1. isoparms traversed with constant velocity

$$\|x\|_u = c_1(v), \quad \|x\|_v = c_2(u). \tag{1}$$

2. area element dA is constant

$$\|x_u \times x_v\| = c.$$

For the purpose of our numerical study of the problem, we chose to work with the first set of conditions. Although the second case is more natural geometrically, our choice is easier to implement and certainly mathematically sound.

A direct consequence of (1) is

$$\begin{aligned} x_u \cdot x_u &= c_1^2(v) \\ x_v \cdot x_v &= c_2^2(u). \end{aligned}$$

Differentiating the first equation with respect to u and the second equation with respect to v eliminates the unknown functions c_1 and c_2 and leaves us with

$$\begin{aligned} x_u \cdot x_{uu} &= 0 \\ x_v \cdot x_{vv} &= 0. \end{aligned}$$

The result is a complete nonlinear partial differential algebraic system of n equations in n unknowns:

$$\begin{aligned} f(x) &= 0 \\ x_u \cdot x_{uu} &= 0 \\ x_v \cdot x_{vv} &= 0, \end{aligned} \tag{2}$$

with the solution made unique by the prescribed boundary data

$$\begin{aligned} x(u,0) &= x^{bottom}(u) && \forall u \in [0,1] \\ x(u,1) &= x^{top}(u) && \forall u \in [0,1] \\ x(0,v) &= x^{left}(v) && \forall v \in [0,1] \\ x(1,v) &= x^{right}(v) && \forall v \in [0,1]. \end{aligned}$$

Two numerical approaches for obtaining the solution to this boundary value problem were followed. The first used finite differences to approximate the partial differential equations in (2). The second used the finite element method to approximate the manifold, with collocation enforced at Gaussian points for the sake of faster convergence (as suggested by [1]). In both cases, Newton's method was used to find a solution to the resulting nonlinear system of equations.

In the following sections we will consider only the $n = 3$ case. The techniques used, however, work equally well in higher dimensions.

2 The Finite Difference Approach

This approach relies on approximating the partial derivatives in (2) in the interior of $[0,1]^2$ by finite differences.

We impose a uniform grid $\{u_i, v_j\}$, $i = 0, ..., N_1 + 1, j = 0, ..., N_2 + 1$ on the unit square, with u_0, u_{N_1+1}, v_0 and v_{N_2+1} lying on the boundary. We make use of the central difference schemes for both the first and second partial derivatives:

$$\begin{aligned} x_u(u_i, v_j) &= \frac{x(u_{i+1}, v_j) - x(u_{i-1}, v_j)}{2h_1} \\ x_{uu}(u_i, v_j) &= \frac{x(u_{i+1}, v_j) - 2x(u_i, v_j) + x(u_{i-1}, v_j)}{h_1^2} \end{aligned}$$

where $h_1 = 1/N_1$ (with analogous equations for x_v and x_{vv}).

The partial differential equations in (2) then become

$$\begin{array}{l}(x(u_{i+1},v_j)-x(u_{i-1},v_j))\cdot(x(u_{i+1},v_j)-2x(u_i,v_j)+x(u_{i-1},v_j))=0\\(x(u_i,v_{j+1})-x(u_i,v_{j-1}))\cdot(x(u_i,v_{j+1})-2x(u_i,v_j)+x(u_i,v_{j-1}))=0\end{array}$$

for $i=0,...,N+1, j=0,...,M+1$.

From now on, we will write $x_{i,j}$ for $x(u_i,v_j)$. We introduce the difference operators

$$\begin{array}{rcl}Du(x_{i,j}) & = & (x_{i+1,j}-x_{i-1,j})\cdot(x_{i+1,j}-2x_{i,j}+x_{i-1,j})\\Dv(x_{i,j}) & = & (x_{i,j+1}-x_{i,j-1})\cdot(x_{i,j+1}-2x_{i,j}+x_{i,j-1}).\end{array}$$

and define the function $F:\mathbb{R}^{3MN}\to\mathbb{R}^{3MN}$ by

$$F\begin{pmatrix}x_{1,1}\\x_{1,2}\\x_{1,3}\\\vdots\\x_{1,M}\\x_{2,1}\\\vdots\\x_{N,M}\end{pmatrix}=\begin{pmatrix}f(x_{1,1})\\Du(x_{1,1})\\Dv(x_{1,1})\\f(x_{1,2})\\Du(x_{1,2})\\Dv(x_{1,2})\\\vdots\\f(x_{1,M})\\Du(x_{1,M})\\Dv(x_{1,M})\\f(x_{2,1})\\Du(x_{2,1})\\Dv(x_{2,1})\\\vdots\\f(x_{N,M})\\Du(x_{N,M})\\Dv(x_{N,M})\end{pmatrix},$$

where the domain of the function F is arranged to provide a nicely banded Jacobian.

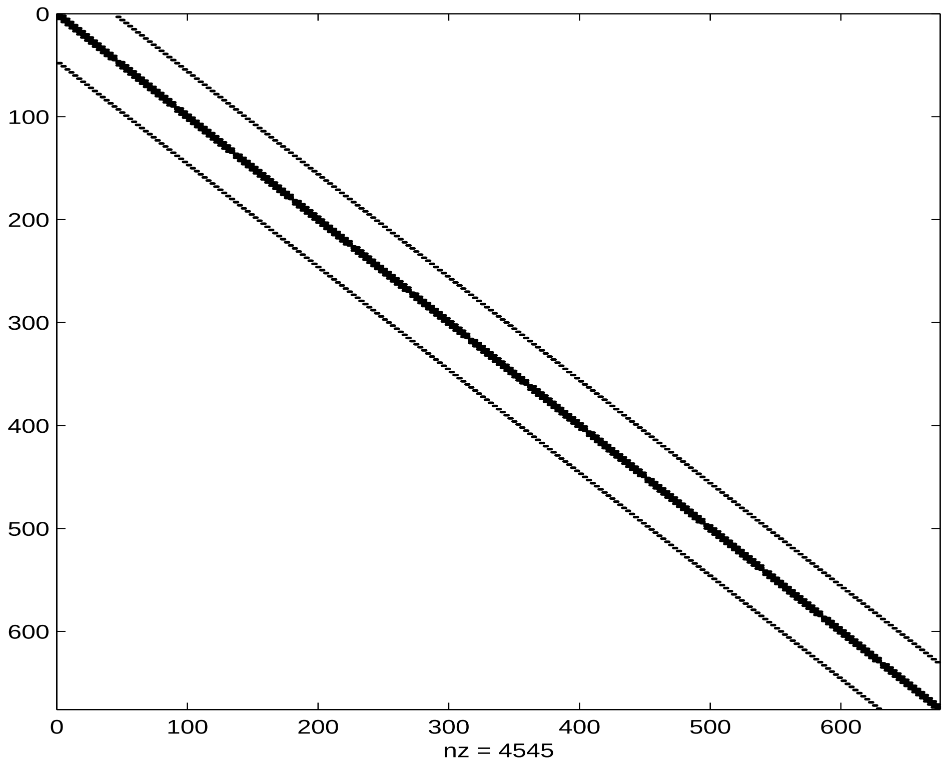
We use Newton's method to solve the nonlinear equation $F=0$. As known, this procedure is sensitive to a good initial estimate x_0 for the vector $x\in\mathbb{R}^{3MN}$ at the grid points. In our examples, we used the x_0 outcome from Coons' patch, which is the surface S_0 obtained by interpolating the given boundary data:

$$\begin{array}{rcl}S_0(u,v) & = & (1-v)x^{bottom}(u)+vx^{top}(u)+(1-u)x^{left}(v)+ux^{right}(v)\\ & & -(1-u)(1-v)x^{bottom}(0)-uvx^{top}(1)-u(1-v)x^{bottom}(1)-(1-u)vx^{top}(0).\end{array}$$

Finally, we iteratively solve

$$x_{n+1}=x_n-[J_F(x_n)]^{-1}F(x_n),$$

where J_F is the Jacobian of F. This Jacobian is a $3MN$ by $3MN$ banded matrix with bandwidth of about $6N$. Its structure is of the form



It is of interest to note that the band width of J_F is dependent only on h_1. This may allow a significant improvement of resolution for some problems without the addition of commonly expected computational expense.

3 The Finite Element Approach

The finite element approach attempts to compute the projection of the solution onto a chosen finite dimensional function space. We chose this space to be the one spanned by the tensor product of fourth order B-splines built on the sets of knots

$$\{0, 0, 0, 0, u_1, u_1, u_2, u_2, ..., u_{N_1}, u_{N_1}, 1, 1, 1, 1\}$$

and

$$\{0, 0, 0, 0, v_1, v_1, v_2, v_2, ..., v_{N_2}, v_{N_2}, 1, 1, 1, 1\},$$

where all u_i's and v_i's have the same meaning as in the previous section.

If $y \in \mathbb{R}^{3(2N_1+4)(2N_2+4)}$ denotes the vector formed by all the coefficients of the solution in this function space, we have that

$$x^s(u, v) = \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} y_{3((i-1)M_2+j-1)+s} C_i(u) D_j(v),$$

where $M_1 = 2N_1 + 4$ and $M_2 = 2N_2 + 4$ are the dimensions of the B-spline function spaces in the u and v directions, respectively.

We force the equations in (2) to hold on the collocation points $(\tilde{u}_p, \tilde{v}_q)$ $(1 \leq p \leq M_1 - 2, 1 \leq M_2 - 2)$, where $\tilde{u}_p$ and $\tilde{v}_q$ are the roots of the quadratic Legendre polynomial, appropriately translated and scaled in the $[u_i, u_{i+1}]$ and $[v_j, v_{j+1}]$ intervals. This has proven successful in the one dimensional case, as mentioned in the first section.

The functional equalities in (2), evaluated on these points, yield the following sets of non-linear algebraic equations in y:

$$f(\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} Y_{3((i-1)M_2+j-1)} C_i(\tilde{u}_p) D_j(\tilde{v}_q)) = 0, \quad \forall 1 \leq p \leq M_1 - 2, \forall 1 \leq q \leq M_2 - 2, \tag{3}$$

$$\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\sum_{l=1}^{M_1}\sum_{m=1}^{M_2} Y_{3((i-1)M_2+j-1)} \cdot Y_{3((l-1)M_2+m-1)} C_i'(\tilde{u}_p) C_l''(\tilde{u}_p) D_j(\tilde{v}_q) D_m(\tilde{v}_q) = 0, \quad \forall 1 \leq p \leq M_1 - 2, \forall 1 \leq q \leq M_2 - 2 \tag{4}$$

and

$$\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\sum_{l=1}^{M_1}\sum_{m=1}^{M_2} Y_{3((i-1)M_2+j-1)} \cdot Y_{3(l-1)M_2+m-1)} C_i(\tilde{u}_p) C_l(\tilde{u}_p) D_j'(\tilde{v}_q) D_m''(\tilde{v}_q) = 0, \quad \forall 1 \leq p \leq M_1 - 2, \forall 1 \leq q \leq M_2 - 2, \tag{5}$$

where $Y_k = (y_{k+1}, y_{k+2}, y_{k+3}) \in \mathbb{R}^3$.

To these $3(M_1 - 2)(M_2 - 2)$ equations we add the equations which result from the boundary conditions:

$$\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} y_{3((i-1)M_2+j-1)+s} C_i(0) D_j(\tilde{v}_q) - x_s^{left}(\tilde{v}_q) = 0, \quad \forall 0 \leq q \leq M_2 - 1, \forall 1 \leq s \leq 3, \tag{6}$$

$$\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} y_{3((i-1)M_2+j-1)+s} C_i(1) D_j(\tilde{v}_q) - x_s^{right}(\tilde{v}_q) = 0, \quad \forall 0 \leq q \leq M_2 - 1, \forall 1 \leq s \leq 3, \tag{7}$$

$$\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} y_{3((i-1)M_2+j-1)+s} C_i(\tilde{u}_p) D_j(0) - x_s^{bottom}(\tilde{u}_p) = 0, \quad \forall 1 \leq p \leq M_1 - 2, \forall 1 \leq s \leq 3, \tag{8}$$

$$\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} y_{3((i-1)M_2+j-1)+s} C_i(\tilde{u}_p) D_j(1) - x_s^{top}(\tilde{u}_p) = 0, \quad \forall 1 \leq p \leq M_1 - 2, \forall 1 \leq s \leq 3, \tag{9}$$

where $\tilde{u}_0 = \tilde{v}_0 = 0$ and $\tilde{u}_{M_1-1} = \tilde{v}_{M_2-2} = 0$. This is now a closed system of $3M_1M_2$ equations with as many variables.

As in the previous section, we attempt to solve this system with Newton's method. For this, we first need to compute the derivatives of the left-hand side of the system with respect to y_k, $k = 1 \cdots 3M_1M_2$.

Differentiation of the left hand side of (3), (4), (5), (6), (7), (8) and (9) with respect to $y_{3((i-1)M_2+j-1)+t}$ respectively yields

$$C_i(\tilde{u}_p)D_j(\tilde{v}_q)f_{,t}(\sum_{l=1}^{M_1}\sum_{m=1}^{M_2} Y_{3((l-1)M_2+m-1)}C_l(\tilde{u}_p)D_m(\tilde{v}_q)) \quad \text{for} \quad 1 \leq p \leq M_1-2, 1 \leq q \leq M_2-2, 1 \leq t \leq 3$$

$$\sum_{l=1}^{M_1}\sum_{m=1}^{M_2} y_{3((l-1)M_2+m-1)+t}(C_i'(\tilde{u}_p)C_l''(\tilde{u}_p) + C_l'(\tilde{u}_p)C_i''(\tilde{u}_p))D_j(\tilde{v}_q)D_m(\tilde{v}_q) \quad \text{for} \quad 1 \leq p \leq M_1-2, 1 \leq q \leq M_2-2, 1 \leq t \leq 3$$

$$\sum_{l=1}^{M_1}\sum_{m=1}^{M_2} y_{3((l-1)M_2+m-1)+t}C_i(\tilde{u}_p)C_l(\tilde{u}_p)(D_j'(\tilde{v}_q)D_m''(\tilde{v}_q) + D_m'(\tilde{v}_q)D_j''(\tilde{v}_q)) \quad \text{for} \quad 1 \leq p \leq M_1-2, 1 \leq q \leq M_2-2, 1 \leq t \leq 3$$

$$\delta_{ts}C_i(0)D_j(\tilde{v}_q) \quad \text{for} \quad 0 \leq q \leq M_2-1, 1 \leq s \leq 3,$$

$$\delta_{ts}C_i(1)D_j(\tilde{v}_q) \quad \text{for} \quad 0 \leq q \leq M_2-1, 1 \leq s \leq 3,$$

$$\delta_{ts}C_i(\tilde{u}_p)D_j(0) \quad \text{for} \quad 1 \leq p \leq M_1-2, 1 \leq s \leq 3$$

and

$$\delta_{ts}C_i(\tilde{u}_p)D_j(1) \quad \text{for} \quad 1 \leq p \leq M_1-2, 1 \leq s \leq 3.$$

4 Creating Boundary Conditions

We have formulated a partial differential algebraic equation as a boundary value problem. In addition, two numerical methods coupled with Newton's method have been presented. In order to generate the initial surface from Coons' patch, we need to find the boundary curves.

Let Ω be the bounded domain of interest and $\phi : \Omega \to R^2$ defines $\partial\Omega$ implicitly as its zero level set. The boundary curves may be found by finding the intersection of the $\{f = 0\}$ and $\{\phi = 0\}$. As mentioned in the earlier section, this can be formulated into a DAE and solved by the method described in [2].

Another approach follows from the observation that the boundary curves are tangent to the normals of both surfaces. This condition can be written mathematically as

$$\dot{x} = \frac{\nabla f \times \nabla \phi}{|\nabla f \times \nabla \phi|}.$$

We can either formulate the above equation as a initial value problem or a boundary value problem, and solve it with an appropriate ODE scheme.

5 Error Estimation

An appropriate criteria for error estimation is the distances from the points on the numerically obtained surface to the zero level set of f. This can be achieved by first calculating the distance function $\phi : \Omega \rightarrow \mathbb{R}$ to the zero level set of f. We do so by using the results in [4].

Consider the partial differential equation

$$\phi_t = sgn(f)(1 - \|\nabla \phi\|),$$

where $sgn(f)$ gives the sign of f. Solving the above equation to steady state provides a function ϕ with the property that $\|\nabla \phi\| = 1$, since convergence occurs when the right hand side is zero. The sign function controls the flow of information in the above; if ϕ is negative, information flows one way and if ϕ is positive, then information flows the other way. The net effect is to straighten out the level sets on either side of the zero level set and produce a ϕ function with $\|\nabla \phi\| = 1$ corresponding to the signed distance function.

We measure the error of our approximation surface $\tilde{x}(u, v)$ as

$$\int_{[0,1]^2} |\phi(\tilde{x}(u, v))| dS,$$

where dS is the surface element. Geometrically, the error measure the L^1 distant between the exact surface and the approximation.

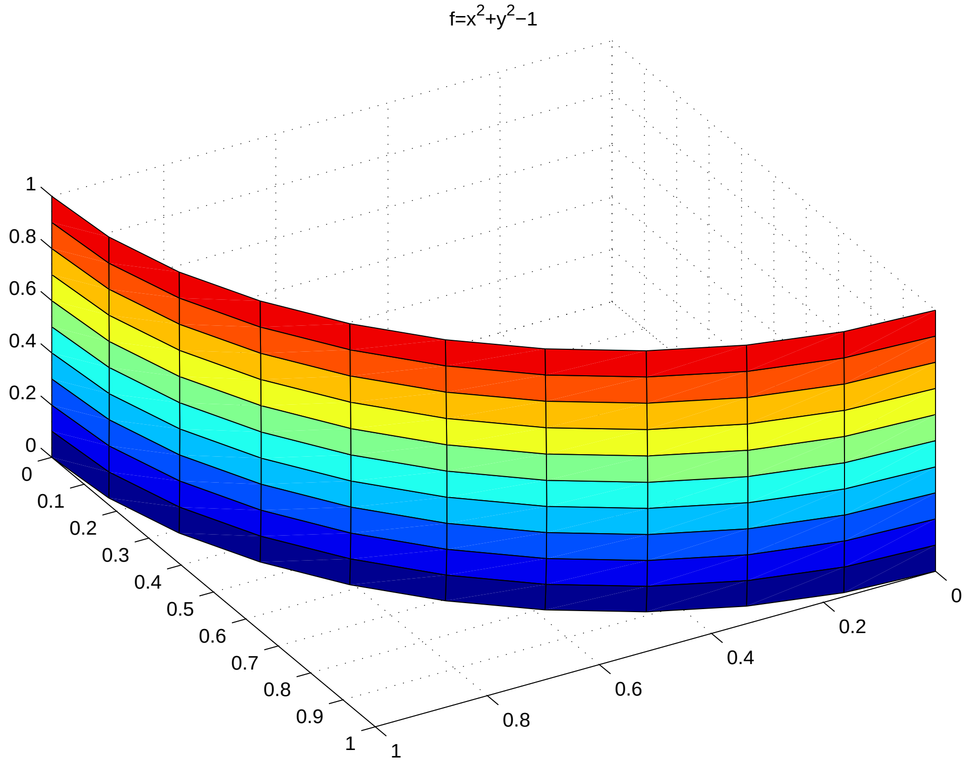
6 Examples

We started with a simple example, for testing purposes. Consider

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad f(x, y, z) = x^2 + y^2 - 1,$$

along with the following boundary data:

$$\begin{aligned} x = 0, y = 1, z = v \qquad & \text{for } v \in [0, 1] \\ x = 1, y = 0, z = v \qquad & \text{for } v \in [0, 1] \\ x = \cos(\frac{\pi}{2}u), y = \sin(\frac{\pi}{2}u), z = 0 \qquad & \text{for } u \in [0, 1] \\ x = \cos(\frac{\pi}{2}u), y = \sin(\frac{\pi}{2}u), z = 1 \qquad & \text{for } u \in [0, 1]. \end{aligned}$$



The surface solution to this problem is a quarter cylinder of radius one, between the planes $z = 0$ and $z = 1$. By using code built on either the finite difference scheme or the finite element idea, we were able to find out approximations of the following parameterization of this surface:

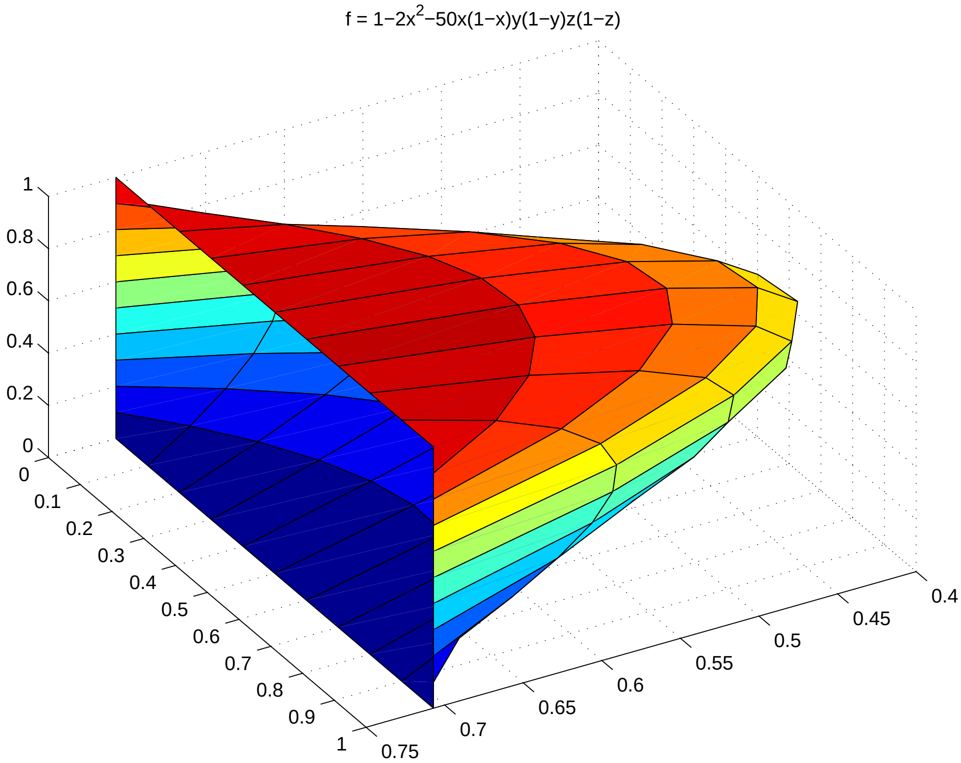
$$x = \cos(\frac{\pi}{2}u), y = \sin(\frac{\pi}{2}u), z = v \quad \text{for } u \in [0, 1], v \in [0, 1],$$

pictured in the figure above.

We next attempted calculating more difficult examples, like:

$$f : \mathbb{R}^3 \ \rightarrow \ \mathbb{R}, \quad f(x, y, z) = 1 - 2x^2 - 50x(1 - x)y(1 - y)z(1 - z).$$

In just a few iterations, our code calculated the parameterization pictured in the figure below. In the initial running, we found damping was needed in the first few Newton iterations, in order for the solution to converge.



Finally, we present a practical application, that of computing the surface envelope $\mathcal{E}$ of a swept surface, which is defined as the boundary of the set

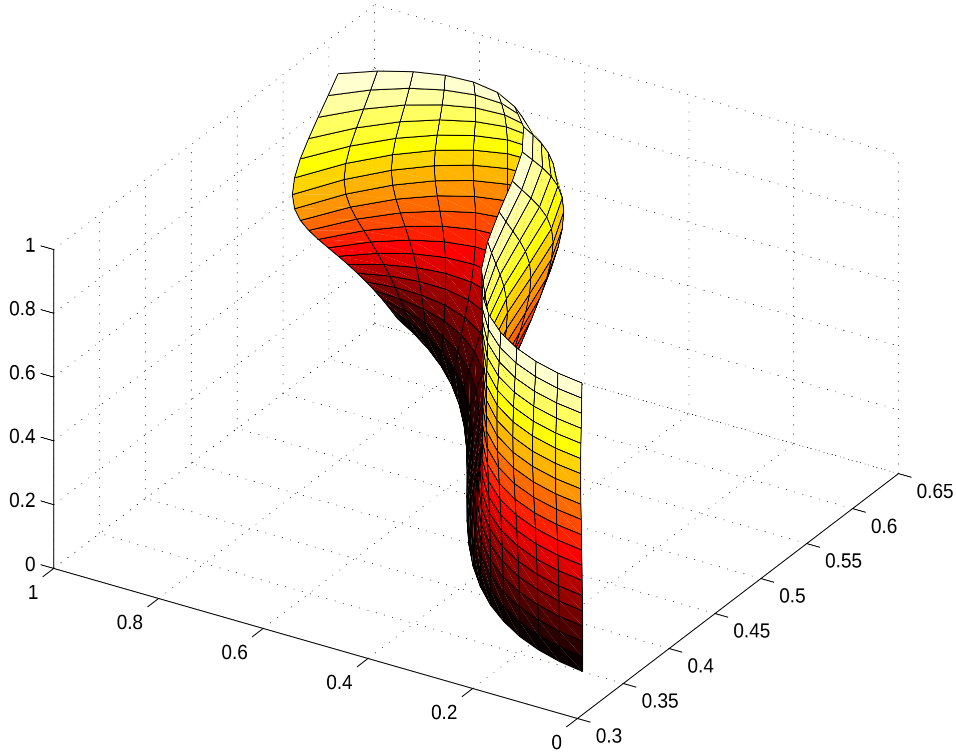
$$\{x \in \mathbb{R}^3 : \exists t \in [0, T] \text{ with } x \in S(t)\},$$

where $S(t)$ is our time evolving surface. A characterization for all the points situated on $\mathcal{E}$ is that the velocity S_t should be perpendicular to the normal at the surface. Thus, if the surface $S(t)$ is given parametrically by u and v, the appropriate condition is

$$(S_u \times S_v) \cdot S_t = 0.$$

This can be interpreted as the zero level set of a real function defined on the (u, v, t) space and computed using either one of our codes.

Pictured below is the representation in the (u, v, t) space of a surface envelope for a certain $S(t)$ described by its spline coefficients. Using the results of our code, we were able to build up a three dimensional movie (see attached *envelope.wrl* file), a snapshot of which is shown on the last page.



References

[1] de Boor C., Swartz B., *Collocation at Gaussian points*, SIAM J. Numer. Anal. **10** (1973), 582-606.

[2] Grandine T. A., *Applications of contouring*, SIAM Review **42** (2000), 297-316.

[3] Grandine T. A., Klein F. W., *A new approach to the surface intersection problem*, Comput. Aided Geom. Design **14** (1997), 111-134.

[4] Sussman M., Smereka P., Osher S. J., *A level set method for computing solutions to incompressible two-phase flow*, J. Comput. Phys. **114** (1994), 146-159.