Parametric Surfaces

These notes describe how to make some specific kinds of surfaces and how to invent more. It is best to study this material in conjunction with displaying surfaces on a computer.

1. Parametric curves and surfaces

Recall that a parametric *curve* in $\mathbb{E}^3$ is the image of a function $\mathbf{P}: \mathbb{R} \to \mathbb{E}^3$, in other words, $\mathbf{P}(t) = \begin{bmatrix} f(t) \\ g(t) \\ h(t) \end{bmatrix}$, or $x = f(t), y = g(t), z = h(t)$. You can think of the curve as being a distorted image of $\mathbb{R}$, produced by the function $\mathbf{P}$. If only the values $a \le t \le b$ are used, then $\mathbf{P}: [a, b] \to \mathbb{E}^3$ and the curve is a distorted image of the interval $[a, b]$, as in Figure 1.

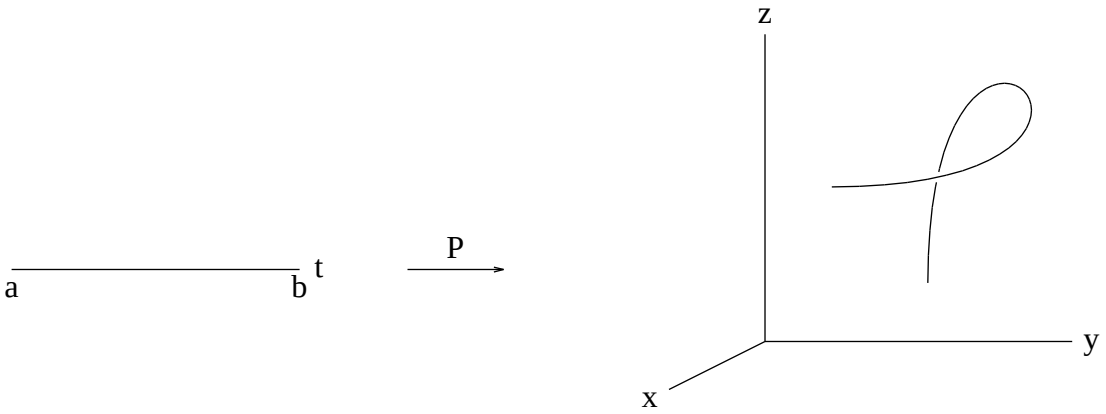

Figure 1: A parametric curve segment as a distorted interval

A parametric *surface* is the image of a function $\mathbf{P}(t, u)$ with two parameters as arguments and with values in $\mathbb{E}^3$. If t, u are allowed to take on any values, then $\mathbf{P}: \mathbb{R}^2 \to \mathbb{E}^3$, and the surface is a distorted image of all of $\mathbb{R}^2$. If $a \le t \le b$ and $c \le u \le d$ then the domain is a rectangle and the surface is a distorted rectangle—a "surface patch". In either case, $\mathbf{P}(t, u) = \begin{bmatrix} f(t, u) \\ g(t, u) \\ h(t, u) \end{bmatrix}$, or equivalently, $x = f(t, u)$, $y = g(t, u)$, $z = h(t, u)$, as in Figure 2.

Example 1. A sphere: Regard t, u as latitude and longitude and let $x = \cos t \cos u$, $y = \cos t \sin u$, $z = \sin t$, for $-\frac{\pi}{2} \le t \le \frac{\pi}{2}$, $0 \le u \le 2\pi$.

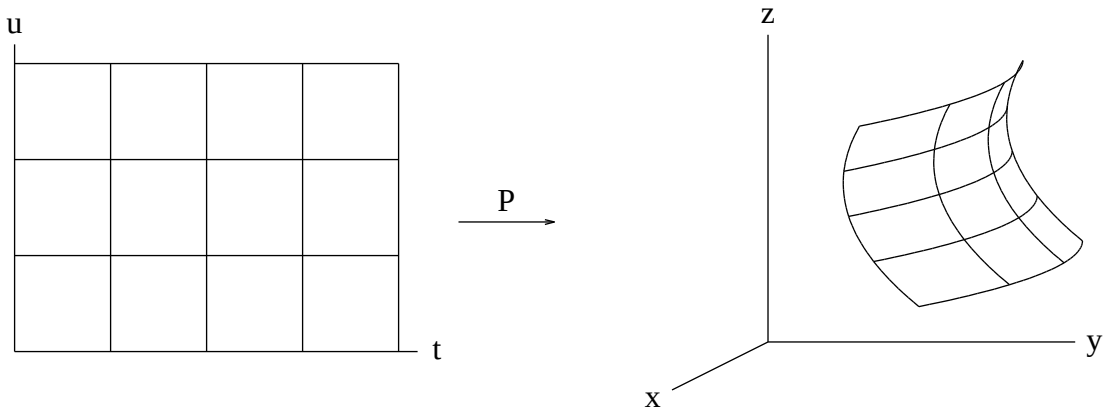

Figure 2: A parametric surface patch as a distorted rectangle

Example 2. The graph of a function F of two variables: Let $x = t$, $y = u$, $z = F(t, u)$.

Example 3. A twisted ribbon: Let $x = -u \sin t$, $y = t$, $z = u \cos t$, for $-\pi \leq t \leq \pi$, $-1 \leq u \leq 1$.

On a computer display using wire-frame graphics, surfaces are indicated by drawing some curves in which one of t and u is held fixed while the other varies. These are *isoparametric* curves.

2. Inventing Surfaces

It is possible to invent surfaces of your own if you make them step-by-step. It helps especially to sweep them out by motions.

Example 1, revisited. To make a sphere of radius 1, first take an easy point, such as $(1, 0, 0)$. Make it into a vertical semicircle by applying $R_t^{x \to z}$, where t varies from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$; you get $(1, 0, 0)R_t^{x \to z}$. Then twist the semicircle around the z-axis by applying $R_u^{x \to y}$, where u varies from 0 to 2π. You get

$$\mathbf{P}(t, u) = R_u^{x \to y} R_t^{x \to z} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \text{ which reduces to}$$

$$\mathbf{P}(t, u) = \begin{bmatrix} \cos t \cos u \\ \cos t \sin u \\ \sin t \end{bmatrix} \text{ for } -\tfrac{\pi}{2} \leq t \leq \tfrac{\pi}{2}, 0 \leq u \leq 2\pi.$$

Example 3, revisited. Pick an axis along which to twist the ribbon, say the y-axis. Imagine the ribbon as swept out by a moving line segment, as the time t changes. Pick a line segment for time 0, say a segment of length

2 along the z-axis: $\begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix}$ for $-1 \le u \le 1$. At time t, the segment should be rotated by $R_t^{x\to z}$ and translated along the y-axis a distance t (a distance proportional to t would also do), by adding $(0, t, 0)$. The time interval is arbitrary, but $-\pi \le t \le \pi$ is nice, since it corresponds to one full twist. You get

$\mathbf{P}(t,u) = R_t^{x\to z} \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ t \\ 0 \end{bmatrix}$, which reduces to

$$\mathbf{P}(t,u) = \begin{bmatrix} -u\sin t \\ t \\ u\cos t \end{bmatrix}, \text{ for } -\pi \le t \le \pi, -1 \le u \le 1.$$

Example 4. Any graph $\begin{bmatrix} f(t,u) \\ g(t,u) \\ h(t,u) \end{bmatrix}$. In fact, this is the most general parametric surface. You might think you would get an interesting surface by just choosing f, g, h to be unrelated messy functions, but such a method often gives disappointing results.

Example 5. A curve rotated about the x-axis, with radius some function $r(x)$, $a \le x \le b$:

The curve drawn in the x, z-plane would be $z = r(x)$. To make this parametric in one variable, use $x = t, y = 0, z = r(t)$, so that the points are $\begin{bmatrix} t \\ 0 \\ r(t) \end{bmatrix}$. Now rotate about the x-axis by multiplying by $R_u^{y\to z}$, for $0 \le u \le 2\pi$. You get

$$\mathbf{P}(t,u) = R_u^{y\to z} \begin{bmatrix} t \\ 0 \\ r(t) \end{bmatrix} = \begin{bmatrix} t \\ -r(t)\sin u \\ r(t)\cos u \end{bmatrix} \text{ for } a \le t \le b,\ 0 \le u \le 2\pi.$$

Example 6. A torus (doughnut): Pick a small radius r and a large radius s, say 1 and 3. Start with one point, $\begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix}$. Rotate it in the x, z-plane by $R_t^{x\to z}$, to sweep out a circle. Then move the circle by a translation along the x-axis so that its center is moved to $(s, 0, 0)$. Then sweep out the surface by rotating around a vertical axis, using $R_u^{x\to y}$ for $0 \le u \le 2\pi$. You get

$\mathbf{P}(t,u) = R_u^{x\to y}\left(R_t^{x\to z}\begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix}\right)$, which you can simplify, or program as is.

Example 7. A Möbius strip: This is something like a combination of Examples 3 and 6. You need to move a line segment around a circle with time, turning it as you go. Start with a vertical line segment $\begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix}$, $-1 \le u \le 1$. Choose a radius $R > 1$. At time t, the segment should be rotated in the x,z-plane, translated by $\begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix}$, and then rotated by $R_t^{x\to y}$, where $0 \le t \le 2\pi$. But how fast should the segment be rotated in the x,z-plane? It should get a half-twist during the time it sweeps around the circle of radius R, so that the rotation should be $R_{t/2}^{x\to z}$. You get

$\mathbf{P}(t,u) = R_t^{x\to y}\left(R_{t/2}^{x\to z}\begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} + \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix}\right)$ which you can simplify, or program as is. (Be careful about parentheses in this formula, lists versus grouping.)

3. Ruled surfaces

A *ruled surface* is a surface that is swept out by a moving straight line, for example, the ribbon and the Möbius strip. One simple way to get a ruled surface is to take two parametric curves in $\mathbb{E}^3$, say $\mathbf{A}(t)$ and $\mathbf{B}(t)$, for the same t-interval $[a,b]$, and at each time t draw a line between them using a new parameter u, to make a "linear blending" of $\mathbf{A}(t)$ and $\mathbf{B}(t)$.

$\mathbf{P}(t,u) = (1-u)\mathbf{A}(t) + u\mathbf{B}(t)$, for $0 \le u \le 1$ and $a \le t \le b$.

Example 8. A hyperboloid of one sheet: Let $\mathbf{A}(t)$ be a circle in the $z=-1$ plane, and let $\mathbf{B}(t)$ be a circle in the $z=1$ plane, traced out 90° out of phase with the first one. Thus

$\mathbf{A}(t) = \begin{bmatrix} \cos t \\ \sin t \\ -1 \end{bmatrix}$ and

$\mathbf{B}(t) = \begin{bmatrix} \cos(t+\frac{\pi}{2}) \\ \sin(t+\frac{\pi}{2}) \\ 1 \end{bmatrix}$, for $0 \le t \le 2\pi$.

Example 9. A self-intersecting surface: Let $\mathbf{A}(t)$ be a circle in the $z=-1$ plane, and let $\mathbf{B}(t)$ be a circle in the $z=1$ plane, swept out twice as the first

circle is swept out once. Thus $\mathbf{A}(t) = \begin{bmatrix} \cos t \\ \sin t \\ -1 \end{bmatrix}$ and $\mathbf{B}(t) = \begin{bmatrix} \cos 2t \\ \sin 2t \\ 1 \end{bmatrix}$, for $0 \le t \le 2\pi$. Horizontal cross sections are limaçons; one is a cardioid.

Example 10. A bilinear patch: Let $P_{00}, P_{01}, P_{10}, P_{11}$ be four points in $\mathbf{E}^3$. What is the simplest patch parameterized by a rectangle that has these points as corners? Just let $\mathbf{A}(t)$ be the line segment from P_{00} to P_{10} and let $\mathbf{B}(t)$ be the line segment from P_{01} to P_{11}. After simplifying, you get

$$\mathbf{P}(t,u) = (1-t)(1-u)P_{00} + t(1-u)P_{10} + (1-t)uP_{01} + tuP_{11}.$$

This can be plotted for $0 \le t \le 1$ and $0 \le u \le 1$. You can check that this works. This function is quadratic, in that there are terms involving tu, but it is *bilinear* in the sense that if either parameter is held fixed the function in the other parameter is linear. $\mathbf{P}(t,u)$ is called a *bilinear patch*. A neater way to write it is to use matrices, allowing points (pairs of numbers) as entries:

$$\mathbf{P}(t,u) = \begin{bmatrix} (1-t) & t \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} \begin{bmatrix} (1-u) \\ u \end{bmatrix}$$

Even though the surface is curved, it contains two natural families of straight lines, namely the lines obtained by holding one of t and u constant and varying the other ("isoparametric curves", as in §6 below). Therefore the surface is ruled.

4. Other constructions of parametric surfaces

Example 11. A family of curves made into a surface: Take a family of planar curves expressible by a formula that has a constant that determines the individual curve. If you put z for the constant, you get a 3-dimensional surface that contains all the curves in the family as horizontal slices. For a parametric version, the family of curves should be made parametric with a parameter t; put u for the constant and set $z = u$.

For example, the parabola $y = x^2$ in $\mathbb{E}^2$ can be parameterized by $x = t, y = t^2$. Then $x = t + c, y = t^2 - c^2$ is a parabola with vertex at $\begin{bmatrix} c \\ -c^2 \end{bmatrix}$. As c changes, you get a family of parabolas. Then for a surface you get

$$\mathbf{P}(t,u) = \begin{bmatrix} t+u \\ t^2 - u^2 \\ u \end{bmatrix}, \text{ say for } -1 \le t \le 1, -1 \le u \le 1.$$

Example 12. A linearly blended Coons patch: Suppose that we have four curves in $\mathbf{E}^3$, parameterized over unit intervals and connected end-to-end, and we want to fill in a surface between them. We can think of these curves

as being given by a function $\mathbf{Q}(t,u)$ that is defined only on the *boundary* of the unit square in parameter space. In other words, $\mathbf{Q}(t,u)$ is defined only when one of t and u is 0 or 1 and the other is between 0 and 1; one boundary curve is $\mathbf{Q}(t,0)$, and other boundary curves are given similarly. We want to find a nice function $\mathbf{P}(t,u)$ defined on the *whole* unit square and agreeing with $\mathbf{Q}(t,u)$ on the boundary.

One idea is to take a linear blending of the two boundary curves $\mathbf{Q}(t,0)$ and $\mathbf{Q}(t,1)$. However, the resulting surface goes straight across between these two boundary curves and does not follow the other two boundary curves. Similarly, a linear blending of $\mathbf{Q}(0,u)$ and $\mathbf{Q}(1,u)$ does not work. Combining these two linear blendings by averaging them also does not work.

A solution turns out to be to add the two linear blendings and subtract off the bilinear patch determined by the corner values. Therefore, we let

$$\mathbf{P}(t,u) = ((1-u)\mathbf{Q}(t,0) + u\mathbf{Q}(t,1)) + ((1-t)\mathbf{Q}(0,u) + t\mathbf{Q}(1,u))$$
$$- ((1-t)(1-u)\mathbf{Q}(0,0) + (1-t)u\mathbf{Q}(0,1) + t(1-u)\mathbf{Q}(1,0) + tu\mathbf{Q}(1,1)).$$

A compact version written using a matrix of point-valued functions is

$$\mathbf{P}(t,u) = \begin{bmatrix} (1-t) & t & 1 \end{bmatrix} \begin{bmatrix} -\mathbf{Q}(0,0) & -\mathbf{Q}(0,1) & \mathbf{Q}(0,u) \\ -\mathbf{Q}(1,0) & -\mathbf{Q}(1,1) & \mathbf{Q}(1,u) \\ \mathbf{Q}(t,0) & \mathbf{Q}(t,1) & 0 \end{bmatrix} \begin{bmatrix} (1-u) \\ u \\ 1 \end{bmatrix}.$$

5. Tensor bases

For curves, you have studied *interpolation* (using the Lagrange method—splines will come later) and *controlled design* (Bézier—B-splines will come later). It is possible to use analogous methods for surfaces. The key is to find a good way of making basis functions; the rest is easy.

Let's concentrate on the case of Lagrange interpolation and look first at the nonparametric case. To ease the later transition to the parametric case, let's look at z as a function of t and u rather than of x and y.

Suppose, then, that we are considering functions $z(t,u)$ with $0 \le t \le 4$ and $0 \le u \le 4$, and that at integer grid points we have given heights to interpolate, say z_{ij} at $t = i, u = j$. Thus $t_i = i$, $u_i = i$ for each $i = 0\ldots, 4$.

Step 1. Construct Lagrange basis functions $f_0(t), \ldots, f_4(t)$. Thus $f_i(j) = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$.

From these make basis functions in two variables by simply taking $f_i(t)f_j(u)$ for $i = 0, \ldots, 4$ and $j = 0, \ldots, 4$. The graph of $f_2(t)f_3(u)$ is shown in Figure 3. Notice that this function has value 1 *only* at *one* grid point, namely at $(2,3)$, and it has value 0 at all other grid points—a nice property. These

basis functions are said to form a *tensor basis*, a term that refers to a basis made of products.

Figure 3: One tensor-basis function in two variables

Step 2. Let $z(t,u) = \sum_{ij} z_{ij} f_i(t) f_j(u)$. Then the graph is a surface that does interpolate at grid points as desired, i.e., $z(i,j) = z_{ij}$. (Why?)

Observe that since the $f_i(t)$ are cubic, the tensor-basis functions and the final result $z(t,u)$ is *bicubic* (cubic in each variable when the other variable is held fixed).

Now let's do a parametric version, where we need to interpolate given data points P_{ij}, and let's suppose that the parameter domain is $0 \le t \le m$, $0 \le u \le n$. Since m and n might not be the same, the first step is to construct basis functions *separately* for each parameter: $f_0(t), \ldots, f_m(t)$ and $g_0(u), \ldots, g_n(u)$. Then the tensor basis functions are $f_i(t)g_j(u)$, and the interpolating surface is given by

$\mathbf{P}(t,u) = \sum_{ij} f_i(t) g_j(u) P_{ij}$.

A compact way to write this is to make a matrix of the points:

$$\mathbf{P}(t,u) = \begin{bmatrix} f_0(t), \ldots, f_m(t) \end{bmatrix} \begin{bmatrix} P_{00} & \ldots & P_{0n} \\ \ldots & \ldots & \ldots \\ P_{m0} & \ldots & P_{mn} \end{bmatrix} \begin{bmatrix} g_0(u) \\ \ldots \\ g_n(u) \end{bmatrix}.$$

Applying the same idea in another context gives Bézier surfaces and B-spline surfaces. In each case you can find one-parameter basis functions (Bernstein polynomials in the case of Bézier curves), and then use them to make a tensor basis. The control points make a *control polyhedron*, although (just as for a

control polygon) it is not a solid object but rather consists of polygons with control points as vertices.

6. Drawing wire-frame surfaces with isoparametric curves

It is usually best to have the surface be the image of a rectangle, even if the equations would make sense for all t, u. Let's say the rectangle is $a \leq t \leq b$, $c \leq u \leq d$. The choice of a, b, c, d will depend upon the example.

A simple approach, the one used by Mathematica, is to imagine grid lines in the rectangle, compute the grid points where they cross, compute the images of these points under $\mathbf{P}$, and connect them the same way the grid is connected. If your grid points are dense enough, this gives a good picture, *provided* that you have the capability of surface shading, hidden-line removal, or both.

If you don't have surface shading or hidden-line removal, then you have a problem: If you use too many grid lines, the plot will be too dense and will obscure the shape of the surface. If you use too few grid lines, your surface will look jagged instead of smooth, as in Figure 4.

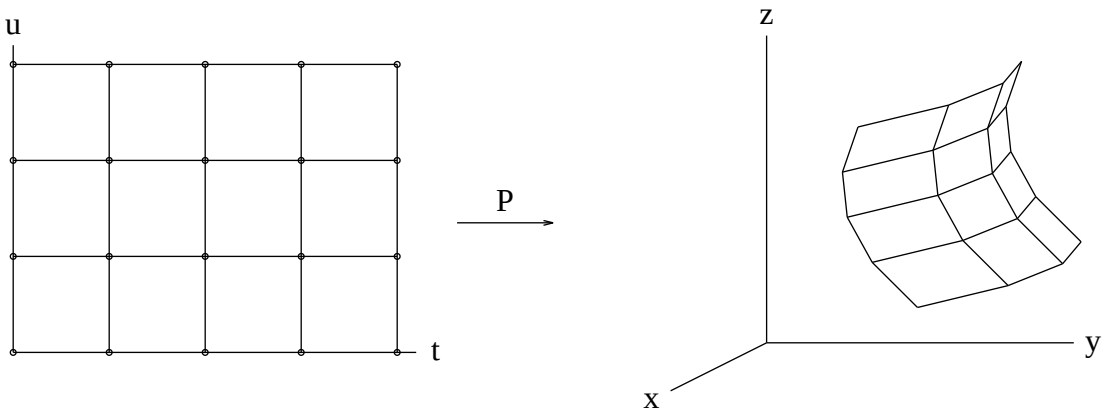

Figure 4: A surface drawn with coarse line segments

The solution is a modified method: Use only a few grid lines, but use more points on each line than just the grid intersections. Compute the images of all these points and connect them with line segments the same way they are connected in the grid. The image of each grid line will then look like a curve–an *isoparametric curve* (curve consisting of points all with one value of t or all with one value of u), as in Figure 5.

Observe the specifications in this example: In the t-direction there are four intervals, each subdivided into five parts; in the u direction, there are three intervals, each subdivided into four parts. In a program, these four numbers (4, 5, 3, 4) should be treated as four named constants or variables,

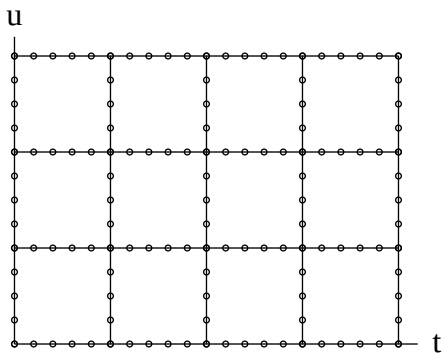

Figure 5: Recommended grid points to use

so that you can easily choose appropriate values and change them later. The grid boundaries should also be named constants or variables.

To find the actual points used, use this idea: To subdivide an interval $[a, b]$ into n pieces, the subdivision points are $a + \frac{k}{n}(b - a)$, for $k = 0, \ldots, n$. Notice that you must loop from 0 to n; if you go from 0 to $n - 1$ or 1 to n parts of your picture will be missing.

(To invent this formula, imitate the formula for a line segment: $a+t(b-a)$ for $0 \leq t \leq 1$ describes all numbers in the interval. Now take $t = \frac{k}{n}$ to step across the interval.)

If you are using a language such as PASCAL or C with a program that plots straight line segments, then to generate the points and connect them, use two double loops. For the horizontal grid lines, you'll need an outer loop for u values corresponding to the heights of the lines and an inner loop for t values corresponding to all the points on a horizontal line. For the vertical grid lines, you'll need another double loop of the same sort.

For each (t, u) point you compute, you can apply $\mathbf{P}$ to it immediately and plot the value; there is no need to store values computed. (Points where grid lines meet will be computed twice, but that's OK. By plotting the value is meant either writing out the value in $\mathbf{E}^3$ to send to a plotting program or else doing the projection to two dimensions yourself and sending the result to a plotting program.)

For some surfaces, the images of the horizontal grid lines will be straight lines. In this case, it often looks better to use just the horizontal grid lines (or just the vertical ones, if their images are straight). In any case, if the image of the grid line on the surface happens to be a straight line, the endpoints of the grid line are already enough; there is no need to use the fine subdivision.

If there is no easy way to remove hidden lines, the plot may look messy if your surfaces have large parts that should be hidden. An easy way to avoid such messiness is to show less of a surface than you could–say, a hemisphere

instead of a sphere. For some surfaces, it may be sufficient merely to view them from a good angle.

In Example 2, it can be effective to show not only the surface, but the original grid as well, drawn in the x, y-plane. Axes could also be included.

Of course, it isn't necessary to use the grid pattern. Sometimes some other pattern or picture in the parameter domain can be effective.

7. Exercises

Problem K-1. Consider the bilinear patch $z = f(t, u)$ with corner data $z_{11} = 1$, $z_{00} = z_{01} = z_{10} = 0$. Describe the horizontal cross-section (level curve) for $z = .5$, and the vertical cross-section for the plane given by $u = .5$.

Problem K-2. For the bilinear patch $\mathbf{P}(t, u)$ with corner data P_{00}, P_{01}, P_{10}, P_{11}, find a formula for $\mathbf{P}(\frac{1}{2}, \frac{1}{2})$.

Problem K-3. For a linearly blended Coons surface $z = f(t, u)$ consider these corner heights and boundary curves: $z_{00} = 0$, $z_{10} = 1$, $z_{01} = 0$, $z_{11} = -1$, $z_{t0} = t^2$ $z_{t1} = -t$, $z_{0u} = \sin \pi u$, $z_{1u} = 2(\cos \frac{\pi}{2} u) - 1$. (a) Is this data consistent? (b) If it is, find an equation for the surface. (The answer may be left as a product of matrices.)

Problem K-4. Explain in detail why the formula for constructing a parametric Lagrange surface does give the desired interpolation.

Problem K-5. (a) Explain why a Lagrange surface that has degree 1 in both t and u and has $t_0 = u_0 = 0$ and $t_1 = u_1 = 1$ is an instance of a bilinear patch (Example 10). (b) What about a relaxed cubic spline surface, still with $n = 1$ on both coordinates?

Problem K-6. Here are control points for a Bézier surface with tensor-product basis, for $m = n = 3$. Describe what the surface should look like, either in words or by an intuitive sketch. The "middle four" control points are the corners of a square in the $z = 1$ plane; specifically, $P_{11} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$, $P_{12} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$, $P_{21} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$, $P_{22} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. All the other control points are at the origin.

Problem K-7. Consider the parametric Bézier surface $P(t,u)$ with $m = n = 2$ and control points P_{ij} given by $P_{ij} = \begin{bmatrix} i \\ j \\ 0 \end{bmatrix}$ except that $P_{11} = \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix}$. Since $m = n = 2$, you will need the Bernstein polynomials $J_{2,0}(t) = (1-t)^2$, $J_{2,1}(t) = 2(1-t)t$, $J_{2,2}(t) = t^2$.

(a) Find the coordinate functions of $P(t,u)$. (Use tensor basis; simplify algebraically.)

(b) Sketch $P(t,u)$. (Just sketch the six isoparametric curves for t and u values $0, .5, 1$.)

Problem K-8. Write down an expression for a Lagrange surface with $m = n = 2$ for the data points $P_{ij} = \begin{bmatrix} i \\ j \\ 0 \end{bmatrix}$, with $t_i = i$, $u_j = j$, for $i = 0, 1, 2$ and $j = 0, 1, 2$, except that $P_{11} = \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix}$. Then simplify the expression. (Method: Simplify one coordinate at a time. It can help to use any properties you know about linear combinations of Lagrange basis functions. Of course, if you can guess a function that fits the data, then by the uniqueness of the Lagrange solution that must be correct, so the algebra is avoided.)

Problem K-9. Recall that a partial derivative with respect to one variable is just the derivative when all other variables are held fixed. For a parametric surface $\mathbf{P}(t,u)$, observe that the two partial derivatives $\frac{\partial \mathbf{P}}{\partial t}(t_0,u_0)$ and $\frac{\partial \mathbf{P}}{\partial u}(t_0,u_0)$ give velocity vectors of the two isoparametric curves at $\mathbf{P}(t_0,u_0)$. These two vectors are therefore tangent to the surface, and their cross product is a normal (a perpendicular) to the surface.

(a) For the sphere parameterized as $\mathbf{P}(t,u) = R_u^{x\to y} R_t^{x\to z} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, or the equivalent written coordinatewise, find a formula for a *unit* normal at $\mathbf{P}(t,u)$.

(b) Does your normal point inwards or outwards? (Either is OK.)

(c) What happens at the poles?

Problem K-10. (a) Using the method of Problem K-9, for the Möbius strip described in the handout on parametric surfaces find a normal vector to the surface for $t = 0, u = 0$ and again for $t = 2\pi, u = 0$, using the cross product of partial derivatives.

(b) Isn't it inconsistent that the two answers are different even though both parameter pairs give the same point on the surface?

Problem K-11. Using the idea of Problem K-9, find a formula for a perpendicular vector at the $t = 0, u = 0$ corner of a Bézier surface patch of degree 2 with control points P_{ij}, where $i, j = 0, 1, 2$.

Problem K-12. Invent the concept of a quadratic blending of three curves and give an example. (Use Lagrange interpolation.)

Problem K-13. Describe surfaces that have the form

$$\mathbf{P}(t,u) = \begin{bmatrix} t \\ u \\ f(\sqrt{t^2+u^2}) \end{bmatrix}.$$

Problem K-14. Consider a Bézier tensor product surface. Explain how the isoparametric curves are actually Bézier curves in $\mathbf{R}^3$. With what control points?