3 Polynomial Interpolation

Definition: A polynomial of degree n in x is the function

$$P_n(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,$$

where a_i are the coefficients of the polynomial (in our case, they are real numbers). Note that there are $n+1$ coefficients.

Calculating a polynomial involves additions, multiplications, and exponentiations, but there are two methods that greatly simplify this calculation. They are the following:

1. Horner's rule. A degree-3 polynomial can be written in the form

$$P(x) = \big((a_3 x + a_2)x + a_1\big)x + a_0,$$

thereby eliminating all exponentiations.

2. Forward differences. This is one of Newton's many contributions to mathematics and it is described in some detail in Section 1.5.1. Only the first step requires multiplications. All other steps are performed with additions and assignments only.

Given a set of points, it is possible to construct a polynomial that when plotted passes through the points. When fully computed and displayed, such a polynomial becomes a curve that's referred to as a *polynomial interpolation* of the points. The first part of this chapter discusses methods for polynomial interpolation and shows their limitations. The second part extends the discussion to a two-dimensional grid of points, and shows how to compute a two-parameter polynomial that passes through the points. When fully computed and displayed, such a polynomial becomes a surface. The methods described here apply the algebra of polynomials to the geometry of curves and surfaces, but this application is limited, because high-degree polynomials tend to oscillate. Section 1.5, and especially Exercise 1.20 show why this is so. Still, there are cases where high-degree polynomials are useful.

This chapter starts with a simple example where four points are given and a cubic polynomial that passes through them is derived from first principles. Following this, the Lagrange and Newton polynomial interpolation methods are introduced. The chapter continues with a description of several simple surface algorithms based on polynomials. It concludes with the Coons and Gordon surfaces, which also employ polynomials.

3.1 Four Points

Four points (two-dimensional or three-dimensional) $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$ are given. We are looking for a PC curve that passes through these points and has the form

$$\mathbf{P}(t) = \mathbf{a}t^3 + \mathbf{b}t^2 + \mathbf{c}t + \mathbf{d} = (t^3, t^2, t, 1)(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})^T = \mathbf{T}(t)\mathbf{A} \quad \text{for} \quad 0 \le t \le 1, \tag{3.1}$$

where each of the four coefficients $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$ is a pair (or a triplet), $\mathbf{T}(t)$ is the row vector $(t^3, t^2, t, 1)$, and $\mathbf{A}$ is the column vector $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})^T$. The only unknowns are $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$.

Since the four points can be located anywhere, we cannot assume anything about their positions and we make the general assumption that $\mathbf{P}_1$ and $\mathbf{P}_4$ are the two endpoints $\mathbf{P}(0)$ and $\mathbf{P}(1)$ of the curve, and that $\mathbf{P}_2$ and $\mathbf{P}_3$ are the two interior points $\mathbf{P}(1/3)$ and $\mathbf{P}(2/3)$. (Having no information about the locations of the points, the best we can do is to use equi-distant values of the parameter t.) We therefore write the four equations $\mathbf{P}(0) = \mathbf{P}_1$, $\mathbf{P}(1/3) = \mathbf{P}_2$, $\mathbf{P}(2/3) = \mathbf{P}_3$, and $\mathbf{P}(1) = \mathbf{P}_4$, or explicitly

$$\begin{aligned}
\mathbf{a}(0)^3 + \mathbf{b}(0)^2 + \mathbf{c}(0) + \mathbf{d} &= \mathbf{P}_1,\\
\mathbf{a}(1/3)^3 + \mathbf{b}(1/3)^2 + \mathbf{c}(1/3) + \mathbf{d} &= \mathbf{P}_2,\\
\mathbf{a}(2/3)^3 + \mathbf{b}(2/3)^2 + \mathbf{c}(2/3) + \mathbf{d} &= \mathbf{P}_3,\\
\mathbf{a}(1)^3 + \mathbf{b}(1)^2 + \mathbf{c}(1) + \mathbf{d} &= \mathbf{P}_4.
\end{aligned} \tag{3.2}$$

The solutions of this system of equations are

$$\begin{aligned}
\mathbf{a} &= -(9/2)\mathbf{P}_1 + (27/2)\mathbf{P}_2 - (27/2)\mathbf{P}_3 + (9/2)\mathbf{P}_4,\\
\mathbf{b} &= 9\mathbf{P}_1 - (45/2)\mathbf{P}_2 + 18\mathbf{P}_3 - (9/2)\mathbf{P}_4,\\
\mathbf{c} &= -(11/2)\mathbf{P}_1 + 9\mathbf{P}_2 - (9/2)\mathbf{P}_3 + \mathbf{P}_4,\\
\mathbf{d} &= \mathbf{P}_1.
\end{aligned} \tag{3.3}$$

Substituting these solutions into Equation (3.1) gives

$$\begin{aligned}
\mathbf{P}(t) =& \big(-(9/2)\mathbf{P}_1 + (27/2)\mathbf{P}_2 - (27/2)\mathbf{P}_3 + (9/2)\mathbf{P}_4\big)t^3\\
&+ \big(9\mathbf{P}_1 - (45/2)\mathbf{P}_2 + 18\mathbf{P}_3 - (9/2)\mathbf{P}_4\big)t^2\\
&+ \big(-(11/2)\mathbf{P}_1 + 9\mathbf{P}_2 - (9/2)\mathbf{P}_3 + \mathbf{P}_4\big)t + \mathbf{P}_1.
\end{aligned}$$

After rearranging, this becomes

$$\begin{aligned}\mathbf{P}(t) =&(-4.5t^3+9t^2-5.5t+1)\mathbf{P}_1+(13.5t^3-22.5t^2+9t)\mathbf{P}_2\\ &+(-13.5t^3+18t^2-4.5t)\mathbf{P}_3+(4.5t^3-4.5t^2+t)\mathbf{P}_4\\ =&G_1(t)\mathbf{P}_1+G_2(t)\mathbf{P}_2+G_3(t)\mathbf{P}_3+G_4(t)\mathbf{P}_4\\ =&\mathbf{G}(t)\mathbf{P},\end{aligned} \tag{3.4}$$

where the four functions $G_i(t)$ are cubic polynomials in t

$$\begin{aligned}G_1(t)&=(-4.5t^3+9t^2-5.5t+1), & G_3(t)&=(-13.5t^3+18t^2-4.5t),\\ G_2(t)&=(13.5t^3-22.5t^2+9t), & G_4(t)&=(4.5t^3-4.5t^2+t),\end{aligned} \tag{3.5}$$

$\mathbf{P}$ is the column $(\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3,\mathbf{P}_4)^T$ and $\mathbf{G}(t)$ is the row $(G_1(t),G_2(t),G_3(t),G_4(t))$ (see also Exercise 3.8 for a different approach to this polynomial).

The functions $G_i(t)$ are called blending functions because they represent any point on the curve as a blend of the four given points. Note that they are barycentric (they should be, since they blend points, and this is shown in the next paragraph). We can also write

$$G_1(t)=(t^3,t^2,t,1)(-4.5,9,-5.5,1)^T$$

and similarly for $G_2(t)$, $G_3(t)$, and $G_4(t)$. The curve can now be expressed as

$$\mathbf{P}(t)=\mathbf{G}(t)\mathbf{P}=(t^3,t^2,t,1)\begin{bmatrix}-4.5 & 13.5 & -13.5 & 4.5\\ 9.0 & -22.5 & 18 & -4.5\\ -5.5 & 9.0 & -4.5 & 1.0\\ 1.0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\\ \mathbf{P}_4\end{bmatrix}=\mathbf{T}(t)\,\mathbf{N}\,\mathbf{P}. \tag{3.6}$$

Matrix $\mathbf{N}$ is called the basis matrix and $\mathbf{P}$ is the geometry vector. Equation (3.1) tells us that $\mathbf{P}(t)=\mathbf{T}(t)\,\mathbf{A}$, so we conclude that $\mathbf{A}=\mathbf{N}\,\mathbf{P}$.

The four functions $G_i(t)$ are barycentric because of the nature of Equation (3.2), not because of the special choice of the four t values. To see why this is so, we write Equation (3.2) for four different, arbitrary values t_1, t_2, t_3, and t_4 (they have to be different, otherwise two or more equations would be contradictory).

$$\begin{aligned}at_1^3+bt_1^2+ct_1+d&=P_1,\\ at_2^3+bt_2^2+ct_2+d&=P_2,\\ at_3^3+bt_3^2+ct_3+d&=P_3,\\ at_4^3+bt_4^2+ct_4+d&=P_4,\end{aligned} \tag{3.7}$$

(where we treat the four values P_i as numbers, not points, and as a result, a, b, c, and d are also numbers). The solutions are of the form

$$\begin{aligned}a&=c_{11}P_1+c_{12}P_2+c_{13}P_3+c_{14}P_4,\\ b&=c_{21}P_1+c_{22}P_2+c_{23}P_3+c_{24}P_4,\\ c&=c_{31}P_1+c_{32}P_2+c_{33}P_3+c_{34}P_4,\\ d&=c_{41}P_1+c_{42}P_2+c_{43}P_3+c_{44}P_4.\end{aligned} \tag{3.8}$$

Comparing Equation (3.8) to Equations (3.3) and (3.5) shows that the four functions $G_i(t)$ can be expressed in terms of the c_{ij} in the form

$$G_i(t) = (c_{1i}t^3 + c_{2i}t^2 + c_{3i}t + c_{4i}). \tag{3.9}$$

The point is that the 16 coefficients c_{ij} do not depend on the four values P_i. They are the same for any choice of the P_i. As a special case, we now select $P_1 = P_2 = P_3 = P_4 = 1$ which reduces Equation (3.8) to

$$\begin{aligned} at_1^3 + bt_1^2 + ct_1 + d = 1, \quad at_2^3 + bt_2^2 + ct_2 + d = 1,\\ at_3^3 + bt_3^2 + ct_3 + d = 1, \quad at_4^3 + bt_4^2 + ct_4 + d = 1. \end{aligned}$$

Because the four values t_i are arbitrary, the four equations above can be written as the single equation $at^3 + bt^2 + ct + d = 1$, that holds for any t. Its solutions must therefore be $a = b = c = 0$ and $d = 1$.

Thus, we conclude that when all four values P_i are 1, a must be zero. In general, $a = c_{11}P_1 + c_{12}P_2 + c_{13}P_3 + c_{14}P_4$, which implies that $c_{11} + c_{12} + c_{13} + c_{14}$ must be zero. Similar arguments show that $c_{21} + c_{22} + c_{23} + c_{24} = 0$, $c_{31} + c_{32} + c_{33} + c_{34} = 0$, and $c_{41} + c_{42} + c_{43} + c_{44} = 1$. These relations, combined with Equation (3.9), show that the four $G_i(t)$ are barycentric.

To calculate the curve, we only need to calculate the four quantities **a**, **b**, **c**, and **d** (that constitute vector **A**), and write Equation (3.1) using the numerical values of **a**, **b**, **c**, and **d**.

Example: (This example is in two dimensions, each of the four points $\mathbf{P}_i$ along with the four coefficients $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ form a pair. For three-dimensional curves the method is the same except that triplets are used instead of pairs.) Given the four two-dimensional points $\mathbf{P}_1 = (0,0)$, $\mathbf{P}_2 = (1,0)$, $\mathbf{P}_3 = (1,1)$, and $\mathbf{P}_4 = (0,1)$, we set up the equation

$$\begin{pmatrix}\mathbf{a}\\ \mathbf{b}\\ \mathbf{c}\\ \mathbf{d}\end{pmatrix} = \mathbf{A} = \mathbf{NP} = \begin{pmatrix} -4.5 & 13.5 & -13.5 & 4.5\\ 9.0 & -22.5 & 18 & -4.5\\ -5.5 & 9.0 & -4.5 & 1.0\\ 1.0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix}(0,0)\\ (1,0)\\ (1,1)\\ (0,1)\end{pmatrix}.$$

Its solutions are

$$\begin{aligned} \mathbf{a} &= -4.5(0,0) + 13.5(1,0) - 13.5(1,1) + 4.5(0,1) = (0,-9),\\ \mathbf{b} &= 19(0,0) - 22.5(1,0) + 18(1,1) - 4.5(0,1) = (-4.5, 13.5),\\ \mathbf{c} &= -5.5(0,0) + 9(1,0) - 4.5(1,1) + 1(0,1) = (4.5, -3.5),\\ \mathbf{d} &= 1(0,0) - 0(1,0) + 0(1,1) - 0(0,1) = (0,0). \end{aligned}$$

So the curve $\mathbf{P}(t)$ that passes through the given points is

$$\mathbf{P}(t) = \mathbf{T}(t)\,\mathbf{A} = (0,-9)t^3 + (-4.5, 13.5)t^2 + (4.5, -3.5)t.$$

It is now easy to calculate and verify that $\mathbf{P}(0) = (0,0) = \mathbf{P}_1$, and

$$\begin{aligned} \mathbf{P}(1/3) &= (0,-9)(1/27) + (-4.5,13.5)(1/9) + (4.5,-3.5)(1/3) = (1,0) = \mathbf{P}_2,\\ \mathbf{P}(1) &= (0,-9)1^3 + (-4.5,13.5)1^2 + (4.5,-3.5)1 = (0,1) = \mathbf{P}_4. \end{aligned}$$

◇ **Exercise 3.1:** Calculate $\mathbf{P}(2/3)$ and verify that it equals $\mathbf{P}_3$.

◇ **Exercise 3.2:** Imagine the circular arc of radius 1 in the first quadrant (a quarter circle). Write the coordinates of the four points that are equally spaced on this arc. Use the coordinates to calculate a PC approximating this arc. Calculate point $\mathbf{P}(1/2)$. How far does it deviate from the midpoint of the true quarter circle?

◇ **Exercise 3.3:** Calculate the PC that passes through the four points $\mathbf{P}_1$ through $\mathbf{P}_4$ assuming that only the three relative coordinates $\Delta_1 = \mathbf{P}_2 - \mathbf{P}_1$, $\Delta_2 = \mathbf{P}_3 - \mathbf{P}_2$, and $\Delta_3 = \mathbf{P}_4 - \mathbf{P}_3$ are given. Show a numeric example.

The main advantage of this method is its simplicity. Given the four points, it is easy to calculate the PC that passes through them. This, however, is also the reason for the downside of the method. It produces only *one* PC that passes through four given points. If that PC does not have the required shape, there is nothing the user can do. This simple curve method is not interactive.

Even though this method is not very useful for curve drawing, it may be useful for interpolation. Given two points $\mathbf{P}_1$ and $\mathbf{P}_2$, we know that the point midway between them is their average, $(\mathbf{P}_1+\mathbf{P}_2)/2$. A natural question is: given four points $\mathbf{P}_1$ through $\mathbf{P}_4$, what point is located midway between them? We can answer this question by calculating the average, $(\mathbf{P}_1+\mathbf{P}_2+\mathbf{P}_3+\mathbf{P}_4)/4$, but this weighted sum assigns the same weight to each of the four points. If we want to assign more weight to the interior points $\mathbf{P}_2$ and $\mathbf{P}_3$, we can calculate the PC that passes through the points and compute $\mathbf{P}(0.5)$ from Equation (3.6). The result is

$$\mathbf{P}(0.5) = -0.0625\mathbf{P}_1 + 0.5625\mathbf{P}_2 + 0.5625\mathbf{P}_3 - 0.0625\mathbf{P}_4.$$

This is a weighted sum that assigns more weight to the interior points. Notice that the weights are barycentric. Exercise 3.13 provides a hint as to why the two extreme weights are negative. This method can be extended to a two-dimensional grid of points (Section 3.6.1).

A precisian professor had the habit of saying: "...quartic polynomial $ax^4 + bx^3 + cx^2 + dx + e$, where e need not be the base of the natural logarithms."
—J. E. Littlewood, *A Mathematician's Miscellany*

◇ **Exercise 3.4:** The preceding method makes sense if the four points are (approximately) equally spaced along the curve. If they are not, the following approach may be taken. Instead of using 1/3 and 2/3 as the intermediate values, the user may specify values α and β, both in the interval $(0, 1)$, such that $\mathbf{P}_2 = \mathbf{P}(\alpha)$ and $\mathbf{P}_3 = \mathbf{P}(\beta)$. Generalize Equation (3.6) such that it depends on α and β.

3.2 The Lagrange Polynomial

The preceding section shows how a cubic interpolating polynomial can be derived for a set of four given points. This section discusses the Lagrange polynomial, a general approach to the problem of polynomial interpolation.

Given the $n+1$ data points $\mathbf{P}_0 = (x_0, y_0)$, $\mathbf{P}_1 = (x_1, y_1), \ldots, \mathbf{P}_n = (x_n, y_n)$, the problem is to find a function $y = f(x)$ that will pass through all of them. We first try an expression of the form $y = \sum_{i=0}^{n} y_i L_i^n(x)$. This is a weighted sum of the individual y_i coordinates where the weights depend on the x_i coordinates. This sum will pass through the points if

$$L_i^n(x) = \begin{cases} 1, & x = x_i, \\ 0, & \text{otherwise.} \end{cases}$$

A good mathematician can easily guess that such functions are given by

$$L_i^n(x) = \frac{\Pi_{j\neq i}(x - x_j)}{\Pi_{j\neq i}(x_i - x_j)} = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}.$$

(Note that $(x - x_i)$ is missing from the numerator and $(x_i - x_i)$ is missing from the denominator.) The function $y = \sum_{i=0}^{n} y_i L_i^n(x)$ is called the Lagrange polynomial because it was originally developed by Lagrange [Lagrange 77] and it is a polynomial of degree n. It is denoted by LP.

Horner's rule and the method of forward differences make polynomials very desirable to use. In practice, however, polynomials are used in parametric form as illustrated in Section 1.5, since any explicit function $y = f(x)$ is limited in the shapes of curves it can generate (note that the explicit form $y = \sum_{i=0}^{n} y_i L_i^n(x)$ of the LP cannot be calculated if two of the $n+1$ given data points have the same x coordinate).

The LP has two properties that make it impractical for interactive curve design, it is of a high degree and it is unique.

1. Writing $P_n(x) = 0$ creates an equation of degree n in x. It has n solutions (some may be complex numbers), so when plotted as a curve it intercepts the x axis n times. For large n, such a curve may be loose because it tends to oscillate wildly. In practice, we normally prefer tight curves.

2. It is easy to show that the LP is unique (see below). There are infinitely many curves that pass through any given set of points and the one we are looking for may not be the LP. Any useful, practical mathematical method for curve design should make it easy for the designer to change the shape of the curve by varying the values of parameters.

It's easy to show that there is only one polynomial of degree n that passes through any given set of $n+1$ points.

A root of the polynomial $P_n(x)$ is a value x_r such that $P_n(x_r) = 0$. A polynomial $P_n(x)$ can have at most n distinct roots (unless it is the zero polynomial). Suppose that there is another polynomial $Q_n(x)$ that passes through the same $n+1$ data points. At the points, we would have $P_n(x_i) = Q_n(x_i) = y_i$ or $(P_n - Q_n)(x_i) = 0$. The difference $(P_n - Q_n)$ is a polynomial whose degree must be $\le n$, so it cannot have more than n distinct roots. On the other hand, this difference is 0 at the $n+1$ data points, so it has

$n+1$ roots. We conclude that it must be the zero polynomial, which implies that $P_n(x)$ and $Q_n(x)$ are identical.

This uniqueness theorem can also be employed to show that the Lagrange weights $L_i^n(x)$ are barycentric. Given a function $f(x)$, select $n+1$ distinct values x_0 through x_n, and consider the $n+1$ *support* points $(x_0, f(x_0))$ through $(x_n, f(x_n))$. The uniqueness theorem states that there is a unique polynomial $p(x)$ of degree n or less that passes through the points, i.e., $p(x_k) = f(x_k)$ for $k = 0, 1, \ldots, n$. We say that this polynomial interpolates the points. Now consider the constant function $f(x) \equiv 1$. The Lagrange polynomial that interpolates its points is

$$\mathrm{LP}(x) = \sum_{i=0}^{n} y_i L_i^n(x) = \sum_{i=0}^{n} 1 \times L_i^n(x) = \sum_{i=0}^{n} L_i^n(x).$$

On the other hand, $\mathrm{LP}(x)$ must be identical to 1, because $\mathrm{LP}(x_k) = f(x_k)$ and $f(x_k) = 1$ for any point x_k. Thus, we conclude that $\sum_{i=0}^{n} L_i^n(x) = 1$ for any x.

Because of these two properties, we conclude that a practical curve design method should be based on polynomials of low degree and should depend on parameters that control the shape of the curve. Such methods are discussed in the chapters that follow. Still, polynomial interpolation may be useful in special situations, which is why it is discussed in the remainder of this chapter.

◇ **Exercise 3.5:** Calculate the LP between the two points $\mathbf{P}_0 = (x_0, y_0)$ and $\mathbf{P}_1 = (x_1, y_1)$. What kind of a curve is it?

I have another method not yet communicated... a convenient, rapid and general solution of this problem, *To draw a geometrical curve which shall pass through any number of given points...* These things are done at once geometrically with no calculation intervening... Though at first glance it looks unmanageable, yet the matter turns out otherwise. For it ranks among the most beautiful of all that I could wish to solve.
(Isaac Newton in a letter to Henry Oldenburg, October 24, 1676, quoted in [Turnbull 59], vol. II, p 188.)

—James Gleick, *Isaac Newton* (2003).

The LP can also be expressed in parametric form. Given the $n+1$ data points $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$, we need to construct a polynomial $\mathbf{P}(t)$ that passes through all of them, such that $\mathbf{P}(t_0) = \mathbf{P}_0$, $\mathbf{P}(t_1) = \mathbf{P}_1$, $\ldots$, $\mathbf{P}(t_n) = \mathbf{P}_n$, where $t_0 = 0$, $t_n = 1$, and t_1 through t_{n-1} are certain values between 0 and 1 (the t_i are called *knot* values). The LP has the form $\mathbf{P}(t) = \sum_{i=0}^{n} \mathbf{P}_i L_i^n(t)$. This is a weighted sum of the individual points where the weights (or basis functions) are given by

$$L_i^n(t) = \frac{\Pi_{j\neq i}^{n}(t - t_j)}{\Pi_{j\neq i}^{n}(t_i - t_j)}. \tag{3.10}$$

Note that $\sum_{i=0}^{n} L_i^n(t) = 1$, so these weights are barycentric.

◇ **Exercise 3.6:** Calculate the parametric LP between the two general points $\mathbf{P}_0$ and $\mathbf{P}_1$.

◇ **Exercise 3.7:** Calculate the parametric LP for the three points $\mathbf{P}_0 = (0,0)$, $\mathbf{P}_1 = (0,1)$, and $\mathbf{P}_2 = (1,1)$.

◇ **Exercise 3.8:** Calculate the parametric LP for the four equally-spaced points $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$ and show that it is identical to the interpolating PC given by Equation (3.4).

The parametric LP is also mentioned on page 109, in connection with Gordon surfaces.

The LP has another disadvantage. If the resulting curve is not satisfactory, the user may want to fine-tune it by adding one more point. However, all the basis functions $L_i^n(t)$ will have to be recalculated in such a case, since they also depend on the points, not only on the knot values. This disadvantage makes the LP slow to use in practice, which is why the Newton polynomial (Section 3.3) is sometimes used instead.

3.2.1 The Quadratic Lagrange Polynomial

Equation (3.10) can easily be employed to obtain the Lagrange polynomial for three points $\mathbf{P}_0$, $\mathbf{P}_1$, and $\mathbf{P}_2$. The weights in this case are

$$\begin{aligned}
L_0^2(t) &= \frac{\prod_{j\neq 0}^2 (t-t_j)}{\prod_{j\neq 0}^2 (t_0-t_j)} = \frac{(t-t_1)(t-t_2)}{(t_0-t_1)(t_0-t_2)},\\
L_1^2(t) &= \frac{\prod_{j\neq 1}^2 (t-t_j)}{\prod_{j\neq 1}^2 (t_1-t_j)} = \frac{(t-t_0)(t-t_2)}{(t_1-t_0)(t_1-t_2)},\\
L_2^2(t) &= \frac{\prod_{j\neq 2}^2 (t-t_j)}{\prod_{j\neq 2}^2 (t_2-t_j)} = \frac{(t-t_0)(t-t_1)}{(t_2-t_0)(t_2-t_1)},
\end{aligned} \tag{3.11}$$

and the polynomial $\mathbf{P}_2(t) = \sum_{i=0}^2 \mathbf{P}_i L_i^2(t)$ is easy to calculate once the values of t_0, t_1, and t_2 have been determined.

The Uniform Quadratic Lagrange Polynomial is obtained when $t_0 = 0$, $t_1 = 1$, and $t_2 = 2$. (See discussion of uniform and nonuniform parametric curves in Section 1.4.1.) Equation (3.11) yields

$$\begin{aligned}
\mathbf{P}_{2u}(t) &= \frac{t^2-3t+2}{2}\mathbf{P}_0 - (t^2-2t)\mathbf{P}_1 + \frac{t^2-t}{2}\mathbf{P}_2\\
&= (t^2, t, 1)\begin{pmatrix} 1/2 & -1 & 1/2\\ -3/2 & 2 & -1/2\\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix}\mathbf{P}_0\\ \mathbf{P}_1\\ \mathbf{P}_2\end{pmatrix}.
\end{aligned} \tag{3.12}$$

The sums of three rows of the matrix of Equation (3.12) are (from top to bottom) 0, 0, and 1, showing that the three basis functions are barycentric, as they should be.

The Nonuniform Quadratic Lagrange Polynomial is obtained when $t_0 = 0$, $t_1 = t_0 + \Delta_0 = \Delta_0$, and $t_2 = t_1 + \Delta_1 = \Delta_0 + \Delta_1$ for some positive Δ_0 and Δ_1. Equation (3.11) gives

$$L_0^2(t) = \frac{(t-\Delta_0)(t-\Delta_0-\Delta_1)}{(-\Delta_0)(-\Delta_0-\Delta_1)},\ L_1^2(t) = \frac{(t-0)(t-\Delta_0-\Delta_1)}{\Delta_0(-\Delta_1)},\ L_2^2(t) = \frac{(t-0)(t-\Delta_0)}{(\Delta_0+\Delta_1)\Delta_1},$$

and the nonuniform polynomial is

$$\mathbf{P}_{2nu}(t) = (t^2, t, 1)\begin{bmatrix} \dfrac{1}{\Delta_0(\Delta_0+\Delta_1)} & -\dfrac{1}{\Delta_0\Delta_1} & \dfrac{1}{(\Delta_0+\Delta_1)\Delta_1} \\ \dfrac{-1}{\Delta_0+\Delta_1}-\dfrac{1}{\Delta_0} & \dfrac{1}{\Delta_0}+\dfrac{1}{\Delta_1} & -\dfrac{1}{\Delta_1}+\dfrac{1}{\Delta_0+\Delta_1} \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix}\mathbf{P}_0\\ \mathbf{P}_1\\ \mathbf{P}_2\end{bmatrix}. \quad (3.13)$$

For $\Delta_0 = \Delta_1 = 1$, Equation (3.13) reduces to the uniform polynomial, Equation (3.12). For $\Delta_0 = \Delta_1 = 1/2$, the parameter t varies in the "standard" range $[0, 1]$ and Equation (3.13) becomes

$$\mathbf{P}_{2std}(t) = (t^2, t, 1)\begin{pmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix}\mathbf{P}_0\\ \mathbf{P}_1\\ \mathbf{P}_2\end{pmatrix}. \quad (3.14)$$

(Notice that the three rows again sum to 0, 0, and 1, to produce three barycentric basis functions.) In most cases, Δ_0 and Δ_1 should be set to the chord lengths $|\mathbf{P}_1 - \mathbf{P}_0|$ and $|\mathbf{P}_2 - \mathbf{P}_1|$, respectively.

◇ **Exercise 3.9:** Use Cartesian product to generalize Equation (3.14) to a surface patch that passes through nine given points.

Example: The three points $\mathbf{P}_0 = (1, 0)$, $\mathbf{P}_1 = (1.3, .5)$, and $\mathbf{P}_2 = (4, 0)$ are given. The uniform LP is obtained when $\Delta_0 = \Delta_1 = 1$ and it equals

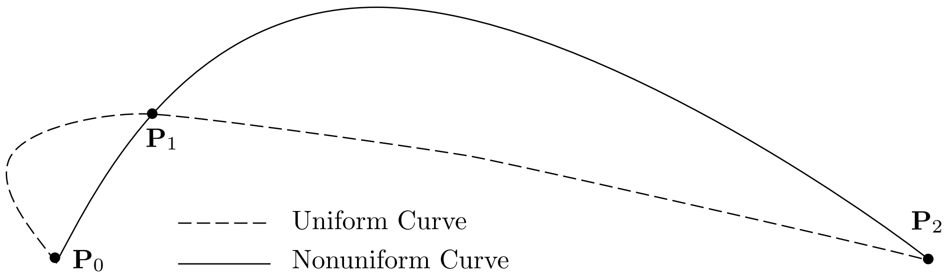
$$\mathbf{P}_{2u}(t) = \big(1 - 0.9t + 1.2t^2, 0.5(2-t)t\big).$$

Many nonuniform polynomials are possible. We select the one that's obtained when the Δ values are the chord lengths between the points. In our case, they are $\Delta_0 = |\mathbf{P}_1 - \mathbf{P}_0| \approx 0.583$ and $\Delta_1 = |\mathbf{P}_2 - \mathbf{P}_1| \approx 2.75$. This polynomial is

$$\mathbf{P}_{2nu}(t) = (1 + 0.433t + 0.14t^2, 1.04t - 0.312t^2).$$

These uniform and nonuniform polynomials are shown in Figure 3.1. The figure illustrates how the nonuniform curve based on the chord lengths between the points is tighter (features smaller overall curvature). Such a curve is generally considered a better interpolation of the three points.

Figure 3.2 shows three examples of nonuniform Lagrange polynomials that pass through the three points $\mathbf{P}_0 = (1, 1)$, $\mathbf{P}_1 = (2, 2)$, and $\mathbf{P}_2 = (4, 0)$. The value of Δ_0 is 1.414, the chord length between $\mathbf{P}_0$ and $\mathbf{P}_1$. The chord length between $\mathbf{P}_1$ and $\mathbf{P}_2$ is 2.83 and Δ_1 is first assigned this value, then half this value, and finally twice it. The three



```
(* 3-point Lagrange polynomial (uniform and nonunif) *)
Clear[T,H,B,d0,d1];
d0=1; d1=1;
T={t^2,t,1};
H={{1/(d0(d0+d1)),-1/(d0 d1),1/(d1(d0+d1))},
{-1/(d0+d1)-1/d0,1/d0+1/d1,-1/d1+1/(d0+d1)},{1,0,0}};
B={{1,0},{1.3,.5},{4,0}};
Simplify[T.H.B];
C1=ParametricPlot[T.H.B,{t,0,d0+d1},PlotRange->All, Compiled->False,
 PlotStyle->AbsoluteDashing[{2,2}], DisplayFunction->Identity];
d0=.583; d1=2.75;
H={{1/(d0(d0+d1)),-1/(d0 d1),1/(d1(d0+d1))},
{-1/(d0+d1)-1/d0,1/d0+1/d1,-1/d1+1/(d0+d1)},{1,0,0}};
Simplify[T.H.B];
C2=ParametricPlot[T.H.B,{t,0,d0+d1},PlotRange->All, Compiled->False,
 DisplayFunction->Identity];
Show[C1, C2, AspectRatio->Automatic, DefaultFont->{"cmr10", 10},
 DisplayFunction->$DisplayFunction];
```

Figure 3.1: Three-Point Lagrange Polynomials.

resulting curves illustrate how the Lagrange polynomial can be reshaped by modifying the Δ_i parameters. The three polynomials in this case are

$$(1 + 0.354231t + 0.249634t^2, 1 + 1.76716t - 0.749608t^2),$$
$$(1 + 0.70738t - 0.000117766t^2, 1 + 1.1783t - 0.333159t^2),$$
$$(1 + 0.777945t - 0.0500221t^2, 1 + 0.919208t - 0.149925t^2).$$

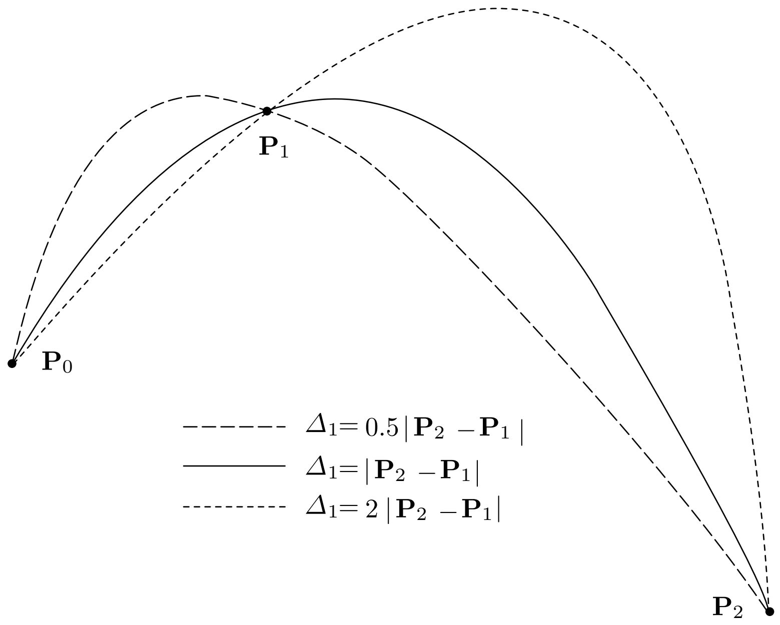
3.2.2 The Cubic Lagrange Polynomial

Equation (3.10) is now applied to the cubic Lagrange polynomial that interpolates the four points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$. The weights in this case are

$$L_0^3(t) = \frac{\prod_{j\neq 0}^3 (t - t_j)}{\prod_{j\neq 0}^3 (t_0 - t_j)} = \frac{(t - t_1)(t - t_2)(t - t_3)}{(t_0 - t_1)(t_0 - t_2)(t_0 - t_3)},$$

$$L_1^3(t) = \frac{\prod_{j\neq 1}^3 (t - t_j)}{\prod_{j\neq 1}^3 (t_1 - t_j)} = \frac{(t - t_0)(t - t_2)(t - t_3)}{(t_1 - t_0)(t_1 - t_2)(t_1 - t_3)},$$

$$L_2^3(t) = \frac{\prod_{j\neq 2}^3 (t - t_j)}{\prod_{j\neq 2}^3 (t_2 - t_j)} = \frac{(t - t_0)(t - t_1)(t - t_3)}{(t_2 - t_0)(t_2 - t_1)(t_2 - t_3)},$$



```
(* 3-point Lagrange polynomial (3 examples of nonuniform) *)
Clear[T,H,B,d0,d1,C1,C2,C3];
d0=1.414; d1=1.415; (* d1=0.5|P2-P1| *)
T={t^2,t,1};
H={{1/(d0(d0+d1)),-1/(d0 d1),1/(d1(d0+d1))},
{-1/(d0+d1)-1/d0,1/d0+1/d1,-1/d1+1/(d0+d1)},{1,0,0}};
B={{1,1},{2,2},{4,0}};
Simplify[T.H.B]
C1=ParametricPlot[T.H.B,{t,0,d0+d1},PlotRange->All, Compiled->False,
 DisplayFunction->Identity];
d1=2.83; (* d1=|P2-P1| *)
H={{1/(d0(d0+d1)),-1/(d0 d1),1/(d1(d0+d1))},
{-1/(d0+d1)-1/d0,1/d0+1/d1,-1/d1+1/(d0+d1)},{1,0,0}};
Simplify[T.H.B]
C2=ParametricPlot[T.H.B,{t,0,d0+d1},PlotRange->All, Compiled->False,
 DisplayFunction->Identity];
d1=5.66; (* d1=2|P2-P1| *)
H={{1/(d0(d0+d1)),-1/(d0 d1),1/(d1(d0+d1))},
{-1/(d0+d1)-1/d0,1/d0+1/d1,-1/d1+1/(d0+d1)},{1,0,0}};
Simplify[T.H.B]
C3=ParametricPlot[T.H.B,{t,0,d0+d1},PlotRange->All, Compiled->False,
 DisplayFunction->Identity];
Show[C1,C2,C3, AspectRatio->Automatic, DefaultFont->{"cmr10", 10},
 DisplayFunction->$DisplayFunction];
(* (1/24,-1/8)t^3+(-1/3,3/4)t^2+(1,-1)t *)
```

Figure 3.2: Three-Point Nonuniform Lagrange Polynomials.

$$L_3^3(t) = \frac{\prod_{j\neq 3}^{3}(t-t_j)}{\prod_{j\neq 3}^{3}(t_3-t_j)} = \frac{(t-t_0)(t-t_1)(t-t_2)}{(t_3-t_0)(t_3-t_1)(t_3-t_2)}, \tag{3.15}$$

and the polynomial $\mathbf{P}_3(t) = \sum_{i=0}^{3} \mathbf{P}_i L_i^3(t)$ is easy to calculate once the values of t_0, t_1, t_2, and t_3 have been determined.

The Nonuniform Cubic Lagrange Polynomial is obtained when $t_0 = 0$, $t_1 = t_0 + \Delta_0 = \Delta_0$, $t_2 = t_1 + \Delta_1 = \Delta_0 + \Delta_1$, and $t_3 = t_2 + \Delta_2 = \Delta_0 + \Delta_1 + \Delta_2$ for positive Δ_i. The expression for the polynomial is

$$\mathbf{P}_{3nu}(t) = (t^3, t^2, t, 1)\mathbf{Q}\begin{pmatrix}\mathbf{P}_0\\ \mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\end{pmatrix}, \tag{3.16}$$

where $\mathbf{Q}$ is the matrix

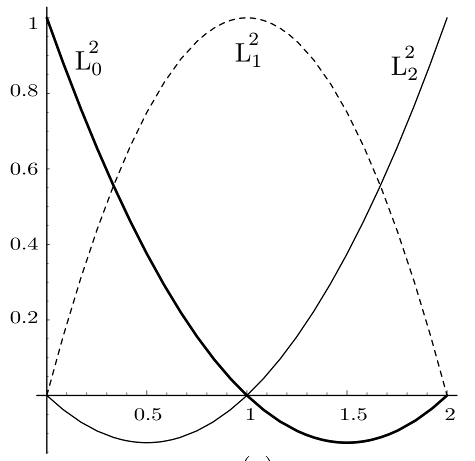
$$\mathbf{Q} = \left(\begin{matrix} \frac{1}{(-\Delta_0)(-\Delta_0-\Delta_1)(-\Delta_0-\Delta_1-\Delta_2)} & \frac{1}{\Delta_0(-\Delta_1)(-\Delta_1-\Delta_2)} \\ -\frac{3\Delta_0+2\Delta_1+\Delta_2}{(-\Delta_0)(-\Delta_0-\Delta_1)(-\Delta_0-\Delta_1-\Delta_2)} & -\frac{2\Delta_0+2\Delta_1+\Delta_2}{\Delta_0(-\Delta_1)(-\Delta_1-\Delta_2)} \\ \frac{\Delta_0(\Delta_0+\Delta_1)+(\Delta_0+\Delta_1)(\Delta_0+\Delta_1+\Delta_2)+(\Delta_0+\Delta_1+\Delta_2)\Delta_0}{(-\Delta_0)(-\Delta_0-\Delta_1)(-\Delta_0-\Delta_1-\Delta_2)} & \frac{(\Delta_0+\Delta_1)(\Delta_0+\Delta_1+\Delta_2)}{\Delta_0(-\Delta_1)(-\Delta_1-\Delta_2)} \\ -\frac{\Delta_0(\Delta_0+\Delta_1)(\Delta_0+\Delta_1+\Delta_2)}{(-\Delta_0)(-\Delta_0-\Delta_1)(-\Delta_0-\Delta_1-\Delta_2)} & 0 \end{matrix}\right.$$

$$\left.\begin{matrix} \frac{1}{(\Delta_0+\Delta_1)\Delta_1(-\Delta_2)} & \frac{1}{(\Delta_0+\Delta_1+\Delta_2)(\Delta_1+\Delta_2)\Delta_2} \\ -\frac{2\Delta_0+\Delta_1+\Delta_2}{(\Delta_0+\Delta_1)\Delta_1(-\Delta_2)} & -\frac{2\Delta_0+\Delta_1}{(\Delta_0+\Delta_1+\Delta_2)(\Delta_1+\Delta_2)\Delta_2} \\ \frac{\Delta_0(\Delta_0+\Delta_1+\Delta_2)}{(\Delta_0+\Delta_1)\Delta_1(-\Delta_2)} & \frac{\Delta_0(\Delta_0+\Delta_1)}{(\Delta_0+\Delta_1+\Delta_2)(\Delta_1+\Delta_2)\Delta_2} \\ 0 & 0 \end{matrix}\right).$$

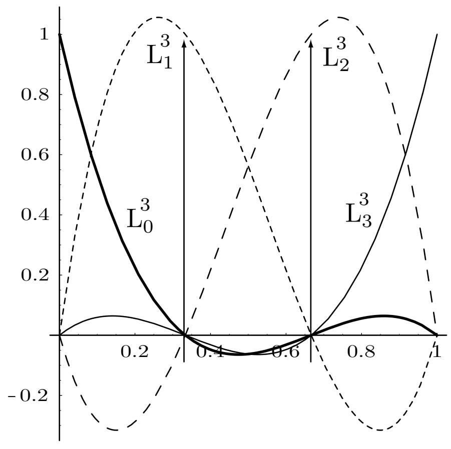
The Uniform Cubic Lagrange Polynomial. We construct the "standard" case, where t varies from 0 to 1. This implies $t_0 = 0$, $t_1 = 1/3$, $t_2 = 2/3$, and $t_3 = 1$. Equation (3.16) reduces to

$$\mathbf{P}_{3u}(t) = (t^3, t^2, t, 1)\begin{pmatrix} -9/2 & 27/2 & -27/2 & 9/2\\ 9 & -45/2 & 18 & -9/2\\ -11/2 & 9 & -9/2 & 1\\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix}\mathbf{P}_0\\ \mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\end{pmatrix}. \tag{3.17}$$

Figure 3.3 shows the quadratic and cubic Lagrange basis functions. It is easy to see that there are values of t (indicated by arrows) for which one of the basis functions is 1 and the others are zeros. This is how the curve (which is a weighted sum of the functions) passes through a point. The functions add up to 1, but most climb above 1 and are negative in certain regions. In the nonuniform case, the particular choice of the various Δ_i reshapes the basis functions in such a way that a function still retains its basic shape, but its areas above and below the t axis may increase or decrease significantly. Those willing to experiment can copy Matrix $\mathbf{Q}$ of Equation (3.16) into



(a)



(b)

```
(* Plot quadratic and cubic Lagrange basis functions *)
lagq={t^2,t,1}.{{1/2,-1,1/2},{-3/2,2,-1/2},{1,0,0}};
Plot[{lagq[[1]],lagq[[2]],lagq[[3]]},{t,0,2},
 PlotRange->All, AspectRatio->Automatic, DefaultFont->{"cmr10", 10}];
lagc={t^3,t^2,t,1}.{{-9/2,27/2,-27/2,9/2},{9,-45/2,18,-9/2},
 2{-11/2,9,-9/2,1},{1,0,0,0}}
Plot[{lagc[[1]],lagc[[2]],lagc[[3]],lagc[[4]]},{t,0,1},
 PlotRange->All, AspectRatio->Automatic, DefaultFont->{"cmr10", 10}];
```

Figure 3.3: (a) Quadratic and (b) Cubic Lagrange Basis Functions.

appropriate mathematical software and use code similar to that of Figure 3.3 to plot the basis functions for various values of Δ_i.

It should be noted that the basis functions of the Bézier curve (Section 6.2) are more intuitive and provide easier control of the shape of the curve, which is why Lagrange interpolation is not popular and is used in special cases only.

3.2.3 Barycentric Lagrange Interpolation

Given the $n+1$ data points $\mathbf{P}_0 = (x_0, y_0)$ through $\mathbf{P}_n = (x_n, y_n)$, the explicit (nonparametric) Lagrange polynomial that interpolates them is $\text{LP}(x) = \sum_{i=0}^{n} y_i L_i^n(x)$, where

$$L_i^n(x) = \frac{\Pi_{j\neq i}^n (x - x_j)}{\Pi_{j\neq i}^n (x_i - x_j)} = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}.$$

This representation of the Lagrange polynomial has the following disadvantages:

1. The denominator of $L_i^n(x)$ requires n subtractions and $n-1$ multiplications, for a total of $O(n)$ operations. The denominators of the $n+1$ weights therefore require $O(n^2)$ operations. The numerators also require $O(n^2)$ operations, but have to be recomputed for each value of x.

2. Adding a new point $\mathbf{P}_{n+1}$ requires the computation of a new weight $L_{n+1}^{n+1}(x)$

and a recomputation of all the original weights $L_i^n(x)$, because

$$L_i^{n+1}(x) = L_i^n(x)\frac{x - x_{n+1}}{x_i - x_{n+1}}, \quad \text{for} \quad i = 0, 1, \ldots, n.$$

3. The computations are numerically unstable. A small change in any of the data points may cause a large change in $\text{LP}(x)$.

Numerical analysts have long believed that these reasons make the Newton polynomial (Section 3.3) more attractive for practical work. However, recent research has resulted in a new, barycentric form of the LP, that makes Lagrange interpolation more attractive. This section is based on [Berrut and Trefethen 04].

The barycentric form of the LP is

$$\begin{aligned}\text{LP}(x) &= \sum_{i=0}^{n} y_i L_i^n(x) = \sum_{i=0}^{n} y_i \frac{\Pi_{j\neq i}^n (x - x_j)}{\Pi_{j\neq i}^n (x_i - x_j)}\\ &= \sum_{i=0}^{n} y_i \frac{w_i}{x - x_i}\left[\Pi_{j=0}^n (x - x_j)\right] = \Pi_{j=0}^n (x - x_j)\sum_{i=0}^{n} y_i \frac{w_i}{x - x_i}\\ &= L(x)\sum_{i=0}^{n} y_i \frac{w_i}{x - x_i},\end{aligned} \tag{3.18}$$

where

$$w_i = \frac{1}{\Pi_{j\neq i}^n (x_i - x_j)}, \quad \text{for} \quad i = 0, 1, \ldots, n.$$

Each weight w_i requires $O(n)$ operations, for a total of $O(n^2)$, but these weights no longer depend on x and consequently have to be computed just once! The only quantity that depends on x is $L(x)$ and it requires only $O(n)$ operations. Also, when a new point is added, the only operations required are (1) divide each w_i by $(x_i - x_{n+1})$ and (2) compute w_{n+1}. These require $O(n)$ steps.

A better form of Equation (3.18), one that's more numerically stable, is obtained when we consider the case $y_i = 1$. If all the data points are of the form $(x_i, 1)$, then the interpolating LP should satisfy $\text{LP}(x) \equiv 1$, which brings Equation (3.18) to the form

$$1 = L(x)\sum_{i=0}^{n} \frac{w_i}{x - x_i}, \tag{3.19}$$

We can now divide Equation (3.18) by Equation (3.19) to obtain

$$\text{LP}(x) = \left[\sum_{i=0}^{n} y_i \frac{w_i}{x - x_i}\right] \Bigg/ \left[\sum_{j=0}^{n} \frac{w_j}{x - x_j}\right]. \tag{3.20}$$

The weights of Equation (3.20) are

$$\frac{\dfrac{w_i}{x - x_i}}{\sum_{j=0}^{n} w_j/(x - x_j)}, \quad i = 0, 1, \ldots, n,$$

and it's easy to see that they are barycentric. Also, any common factors in the weights can now be cancelled out. For example, it can be shown that in the case of data points that are uniformly distributed in the interval $[-1,+1]$

$$\mathbf{P}_0=(-1,y_0),\quad \mathbf{P}_1=(-1+h,y_1),\quad \mathbf{P}_2=(-1+2h,y_2),\ldots,\mathbf{P}_n=(+1,y_n)$$

(where $h=2/n$), the weights become $w_i=(-1)^{n-i}\binom{n}{i}/(h^n n!)$. The common factors are those that do not depend on i. When they are cancelled out, the weights become the simple expressions

$$w_i=(-1)^i\binom{n}{i}.$$

(This is also true for points that are equidistant in any interval $[a,b]$. Incidentally, it can be shown that the case of equidistant data points is ill conditioned and the LP, in any form, can change its value wildly in response to even small changes in the data points.)

3.3 The Newton Polynomial

The Newton polynomial offers an alternative approach to the problem of polynomial interpolation. The final interpolating polynomial is identical to the LP, but the derivation is different. It allows the user to easily add more points and thereby provide fine control over the shape of the curve. We again assume that $n+1$ data points $\mathbf{P}_0$, $\mathbf{P}_1,\ldots,\mathbf{P}_n$ are given and are assigned knot values

$$t_0=0<t_1<\cdots<t_{n-1}<t_n=1.$$

We are looking for a curve expressed by the degree-n parametric polynomial

$$\mathbf{P}(t)=\sum_{i=0}^{n}N_i(t)\mathbf{A}_i,$$

where the basis functions $N_i(t)$ depend only on the knot values and not on the data points. Only the (unknown) coefficients $\mathbf{A}_i$ depend on the points. This definition (originally proposed by Newton) is useful because each coefficient $\mathbf{A}_i$ depends only on points $\mathbf{P}_0$ through $\mathbf{P}_i$. If the user decides to add a point $\mathbf{P}_{n+1}$, only one coefficient, $\mathbf{A}_{n+1}$, and one basis function, $N_{n+1}(t)$, need be recomputed.

The definition of the basis functions is

$$N_0(t)=1\quad\text{and}\quad N_i(t)=(t-t_0)(t-t_1)\cdots(t-t_{i-1}),\quad\text{for}\quad i=1,\ldots,n.$$

To calculate the unknown coefficients, we write the equations

$$\begin{aligned}
\mathbf{P}_0&=\mathbf{P}(t_0)=\mathbf{A}_0,\\
\mathbf{P}_1&=\mathbf{P}(t_1)=\mathbf{A}_0+\mathbf{A}_1(t_1-t_0),\\
\mathbf{P}_2&=\mathbf{P}(t_2)=\mathbf{A}_0+\mathbf{A}_1(t_2-t_0)+\mathbf{A}_2(t_2-t_0)(t_2-t_1),\\
&\ \vdots\\
\mathbf{P}_n&=\mathbf{P}(t_n)=\mathbf{A}_0+\cdots.
\end{aligned}$$

These equations don't have to be solved simultaneously. Each can easily be solved after all its predecessors have been solved. The solutions are

$$\begin{aligned}
\mathbf{A}_0 &= \mathbf{P}_0,\\
\mathbf{A}_1 &= \frac{\mathbf{P}_1-\mathbf{P}_0}{t_1-t_0},\\
\mathbf{A}_2 &= \frac{\mathbf{P}_2-\mathbf{P}_0-\dfrac{(\mathbf{P}_1-\mathbf{P}_0)(t_2-t_0)}{t_1-t_0}}{(t_2-t_0)(t_2-t_1)} = \frac{\dfrac{\mathbf{P}_2-\mathbf{P}_1}{t_2-t_1}-\dfrac{\mathbf{P}_1-\mathbf{P}_0}{t_1-t_0}}{t_2-t_0}.
\end{aligned}$$

This obviously gets very complicated quickly, so we use the method of *divided differences* to express all the solutions in compact notation. The divided difference of the knots $t_i t_k$ is denoted $[t_i t_k]$ and is defined as

$$[t_i t_k] \overset{\text{def}}{=} \frac{\mathbf{P}_i-\mathbf{P}_k}{t_i-t_k}.$$

The solutions can now be expressed as

$$\begin{aligned}
\mathbf{A}_0 &= \mathbf{P}_0,\\
\mathbf{A}_1 &= \frac{\mathbf{P}_1-\mathbf{P}_0}{t_1-t_0} = [t_1t_0],\\
\mathbf{A}_2 &= [t_2t_1t_0] = \frac{[t_2t_1]-[t_1t_0]}{t_2-t_0},\\
\mathbf{A}_3 &= [t_3t_2t_1t_0] = \frac{[t_3t_2t_1]-[t_2t_1t_0]}{t_3-t_0},\\
&\vdots\\
\mathbf{A}_n &= [t_n\ldots t_1t_0] = \frac{[t_n\ldots t_1]-[t_{n-1}\ldots t_0]}{t_n-t_0}.
\end{aligned}$$

◇ **Exercise 3.10:** Given the same points and knot values as in Exercise 3.7, calculate the Newton polynomial that passes through the points.

◇ **Exercise 3.11:** The tangent vector to a curve $\mathbf{P}(t)$ is the derivative $\frac{d\mathbf{P}(t)}{dt}$, which we denote by $\mathbf{P}^t(t)$. Calculate the tangent vectors to the curve of Exercises 3.7 and 3.10 at the three points. Also calculate the slopes of the curve at the points.

3.4 Polynomial Surfaces

The polynomial $y = \sum a_i x^i$ is the explicit representation of a curve. Similarly, the parametric polynomial $\mathbf{P}(t) = \sum t^i \mathbf{P}_i$ and also $\mathbf{P}(t) = \sum a_i(t)\mathbf{P}_i$ (where $a_i(t)$ is a polynomial in t) are parametric representations of curves. These expressions can be extended to polynomials in two variables, which represent surfaces. Thus, the double polynomial $z = \sum_i \sum_j a_{ij} x^i y^j$ is the explicit representation of a surface patch, because it yields a z value for any pair of coordinates (x, y). Similarly, the double parametric polynomial $\mathbf{P}(u, w) = \sum_i \sum_j u^i w^j \mathbf{P}_{ij}$ is the parametric representation of a surface patch. For the cubic case (polynomials of degree 3), such a double polynomial can be expressed compactly in matrix notation

$$\mathbf{P}(u,w) = [u^3, u^2, u, 1]\mathbf{N} \begin{bmatrix} \mathbf{P}_{33} & \mathbf{P}_{32} & \mathbf{P}_{31} & \mathbf{P}_{30} \\ \mathbf{P}_{23} & \mathbf{P}_{22} & \mathbf{P}_{21} & \mathbf{P}_{20} \\ \mathbf{P}_{13} & \mathbf{P}_{12} & \mathbf{P}_{11} & \mathbf{P}_{10} \\ \mathbf{P}_{03} & \mathbf{P}_{02} & \mathbf{P}_{01} & \mathbf{P}_{00} \end{bmatrix} \mathbf{N}^T \begin{bmatrix} w^3 \\ w^2 \\ w \\ 1 \end{bmatrix}. \tag{3.21}$$

The corresponding surface patch is accordingly referred to as bicubic.

3.5 The Biquadratic Surface Patch

This section introduces the biquadratic surface patch and constructs this simple surface as a Cartesian product. Given the two quadratic (degree 2) polynomials

$$\mathbf{Q}(u) = \sum_{i=0}^{2} f_i(u)\mathbf{Q}_i \quad \text{and} \quad \mathbf{R}(w) = \sum_{j=0}^{2} g_j(w)\mathbf{R}_j$$

the biquadratic surface immediately follows from the principle of Cartesian product

$$\mathbf{P}(u,w) = \sum_{i=0}^{2}\sum_{j=0}^{2} f_i(u) g_j(w) \mathbf{P}_{ij}. \tag{3.22}$$

Different constructions are possible depending on the geometric meaning of the nine quantities $\mathbf{P}_{ij}$. The following section presents such a construction and Section 4.10 discusses another approach, based on points, tangent vectors, and twist vectors.

3.5.1 Nine Points

Equation (3.14), duplicated below, gives the quadratic standard Lagrange polynomial that interpolates three given points:

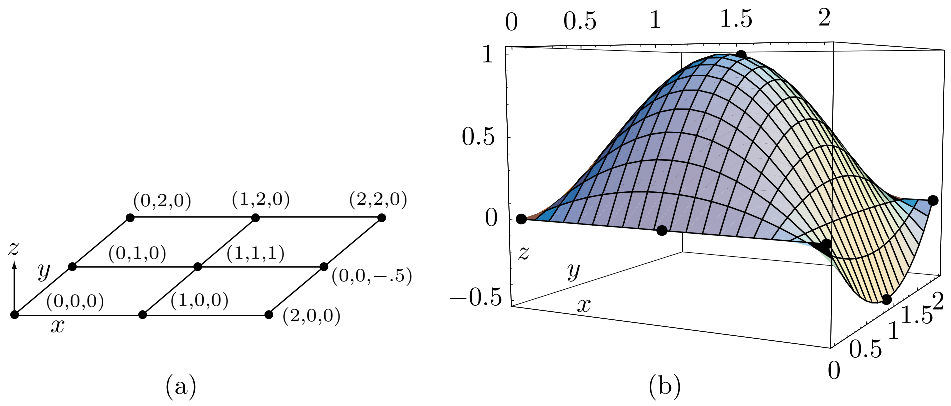
$$\mathbf{P}_{2std}(t) = (t^2, t, 1) \begin{pmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix}. \tag{3.14}$$

Cartesian product yields the corresponding biquadratic surface

$$\begin{aligned}\mathbf{P}(u,w) = (u^2,u,1)&\begin{pmatrix}2&-4&2\\-3&4&-1\\1&0&0\end{pmatrix}\begin{pmatrix}\mathbf{P}_{22}&\mathbf{P}_{21}&\mathbf{P}_{20}\\\mathbf{P}_{12}&\mathbf{P}_{11}&\mathbf{P}_{10}\\\mathbf{P}_{02}&\mathbf{P}_{01}&\mathbf{P}_{00}\end{pmatrix}\\&\times\begin{pmatrix}2&-4&2\\-3&4&-1\\1&0&0\end{pmatrix}^T\begin{pmatrix}w^2\\w\\1\end{pmatrix},\end{aligned}\tag{3.23}$$

where the nine quantities $\mathbf{P}_{ij}$ are points defining this surface patch. They should be roughly equally spaced over the surface.

Example: Given the nine points of Figure 3.4a, we compute and draw the biquadratic surface patch defined by them. The surface is shown in Figure 3.4b. The code is also listed.



```
<<:Graphics:ParametricPlot3D.m; (* Biquadratic patch for 9 points *)
Clear[T,pnt,M,g1,g2];
T[t_]:={t^2,t,1};
pnt={{{0,0,0},{1,0,0},{2,0,0}}, {{0,1,0},{1,1,1},{2,1,-.5}},
 {{0,2,0},{1,2,0},{2,2,0}}};
M={{2,-4,2},{-3,4,-1},{1,0,0}};
g2=Graphics3D[{AbsolutePointSize[4],
Table[Point[pnt[[i,j]]],{i,1,3},{j,1,3}] }];
comb[i_]:=(T[u].M.pnt)[[i]](Transpose[M].T[w])[[i]];
g1=ParametricPlot3D[comb[1]+comb[2]+comb[3], {u,0,1},{w,0,1},
Compiled->False, DisplayFunction->Identity];
Show[g1,g2, ViewPoint->{1.391, -2.776, 0.304}, DefaultFont->{"cmr10", 10},
DisplayFunction->$DisplayFunction]
```

Figure 3.4: A Biquadratic Surface Patch Example.

It is also possible to construct similar biquadratic surfaces from the expressions for the uniform and nonuniform quadratic Lagrange polynomials, Equations (3.12) and (3.13).

◇ **Exercise 3.12:** The geometry vector of Equation (3.14) has point $\mathbf{P}_0$ at the top, but the geometry matrix of Equation (3.23) has point $\mathbf{P}_{00}$ at its bottom-right instead of its top-left corner. Why is that?

3.6 The Bicubic Surface Patch

The parametric cubic (PC) curve, Equation (3.1), is useful, since it can be used when either four points, or two points and two tangent vectors, are known. The latter approach is the topic of Chapter 4. The PC curve can easily be extended to a bicubic surface patch by means of the Cartesian product.

A PC curve has the form $\mathbf{P}(t) = \sum_{i=0}^{3} \mathbf{a}_i t^i$. Two such curves, $\mathbf{P}(u)$ and $\mathbf{P}(w)$, can be combined to form the Cartesian product surface patch

$$\begin{aligned}
\mathbf{P}(u,w) &\\
&= \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{a}_{ij} u^i w^j\\
&= \mathbf{a}_{33}u^3w^3 + \mathbf{a}_{32}u^3w^2 + \mathbf{a}_{31}u^3w + \mathbf{a}_{30}u^3 + \mathbf{a}_{23}u^2w^3 + \mathbf{a}_{22}u^2w^2 + \mathbf{a}_{21}u^2w + \mathbf{a}_{20}u^2\\
&\quad + \mathbf{a}_{13}uw^3 + \mathbf{a}_{12}uw^2 + \mathbf{a}_{11}uw + \mathbf{a}_{10}u + \mathbf{a}_{03}w^3 + \mathbf{a}_{02}w^2 + \mathbf{a}_{01}w + \mathbf{a}_{00} \qquad (3.24)\\
&= (u^3, u^2, u, 1)\begin{pmatrix} \mathbf{a}_{33} & \mathbf{a}_{32} & \mathbf{a}_{31} & \mathbf{a}_{30}\\ \mathbf{a}_{23} & \mathbf{a}_{22} & \mathbf{a}_{21} & \mathbf{a}_{20}\\ \mathbf{a}_{13} & \mathbf{a}_{12} & \mathbf{a}_{11} & \mathbf{a}_{10}\\ \mathbf{a}_{03} & \mathbf{a}_{02} & \mathbf{a}_{01} & \mathbf{a}_{00} \end{pmatrix}\begin{pmatrix} w^3\\ w^2\\ w\\ 1\end{pmatrix}, \quad \text{where } 0 \le u, w \le 1. \qquad (3.25)
\end{aligned}$$

This is a double cubic polynomial (hence the name *bicubic*) with 16 terms, where each of the 16 coefficients $\mathbf{a}_{ij}$ is a triplet [compare with Equation (3.21)]. When w is set to a fixed value w_0, Equation (3.25) becomes $\mathbf{P}(u, w_0)$, which is a PC curve. The same is true for $\mathbf{P}(u_0, w)$. The conclusion is that curves that lie on this surface in the u or in the w directions are parametric cubics. The four boundary curves are consequently also PC curves.

Notice that the shape and location of the surface depend on all 16 coefficients. Any change in any of them produces a different surface patch. Equation (3.25) is the algebraic representation of the bicubic patch. In order to use it in practice, the 16 unknown coefficients have to be expressed in terms of known geometrical quantities, such as points, tangent vectors, or second derivatives.

Two types of bicubic surfaces are discussed here. The first is based on 16 data points and the second is constructed from four known curves. A third type—defined by four data points, eight tangent vectors, and four twist vectors—is the topic of Section 4.9.

Milo. . . glanced curiously at the strange circular room, where sixteen tiny arched windows corresponded exactly to the sixteen points of the compass. Around the entire circumference were numbers from zero to three hundred and sixty, marking the degrees of the circle, and on the floor, walls, tables, chairs, desks, cabinets, and ceiling were labels showing their heights, widths, depths, and distances to and from each other.

—Norton Juster, *The Phantom Tollbooth.*

3.6.1 Sixteen Points

We start with the sixteen given points

$$\begin{matrix} \mathbf{P}_{03} & \mathbf{P}_{13} & \mathbf{P}_{23} & \mathbf{P}_{33} \\ \mathbf{P}_{02} & \mathbf{P}_{12} & \mathbf{P}_{22} & \mathbf{P}_{32} \\ \mathbf{P}_{01} & \mathbf{P}_{11} & \mathbf{P}_{21} & \mathbf{P}_{31} \\ \mathbf{P}_{00} & \mathbf{P}_{10} & \mathbf{P}_{20} & \mathbf{P}_{30}. \end{matrix}$$

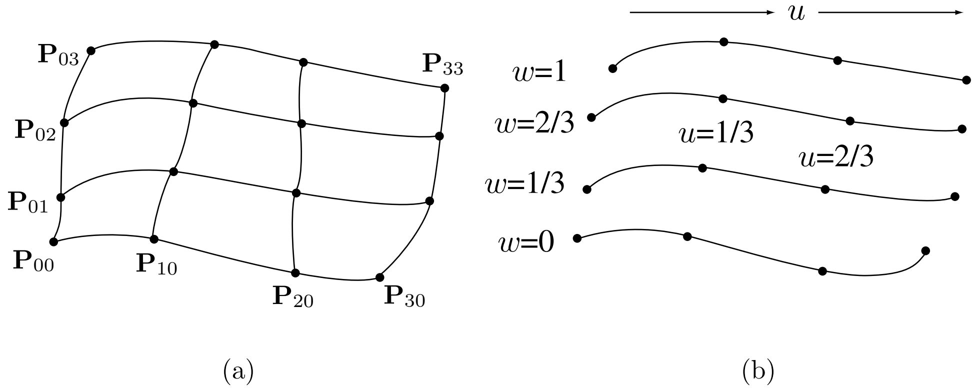


Figure 3.5: (a) Sixteen Points. (b) Four Curves.

We assume that the points are (roughly) equally spaced on the rectangular surface patch as shown in Figure 3.5a. We know that the bicubic surface has the form

$$\mathbf{P}(u,w) = \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{a}_{ij} u^i w^j, \tag{3.26}$$

where each of the 16 coefficients $\mathbf{a}_{ij}$ is a triplet. To calculate the 16 unknown coefficients, we write 16 equations, each based on one of the given points

$$\begin{matrix} \mathbf{P}(0,0) = \mathbf{P}_{00}, & \mathbf{P}(0,1/3) = \mathbf{P}_{01}, & \mathbf{P}(0,2/3) = \mathbf{P}_{02}, & \mathbf{P}(0,1) = \mathbf{P}_{03}, \\ \mathbf{P}(1/3,0) = \mathbf{P}_{10}, & \mathbf{P}(1/3,1/3) = \mathbf{P}_{11}, & \mathbf{P}(1/3,2/3) = \mathbf{P}_{12}, & \mathbf{P}(1/3,1) = \mathbf{P}_{13}, \\ \mathbf{P}(2/3,0) = \mathbf{P}_{20}, & \mathbf{P}(2/3,1/3) = \mathbf{P}_{21}, & \mathbf{P}(2/3,2/3) = \mathbf{P}_{22}, & \mathbf{P}(2/3,1) = \mathbf{P}_{23}, \\ \mathbf{P}(1,0) = \mathbf{P}_{30}, & \mathbf{P}(1,1/3) = \mathbf{P}_{31}, & \mathbf{P}(1,2/3) = \mathbf{P}_{32}, & \mathbf{P}(1,1) = \mathbf{P}_{33}. \end{matrix}$$

After solving, the final expression for the surface patch becomes

$$\mathbf{P}(u,w)=(u^3,u^2,u,1)\mathbf{N}\begin{pmatrix}\mathbf{P}_{00}&\mathbf{P}_{10}&\mathbf{P}_{20}&\mathbf{P}_{30}\\ \mathbf{P}_{01}&\mathbf{P}_{11}&\mathbf{P}_{21}&\mathbf{P}_{31}\\ \mathbf{P}_{02}&\mathbf{P}_{12}&\mathbf{P}_{22}&\mathbf{P}_{32}\\ \mathbf{P}_{03}&\mathbf{P}_{13}&\mathbf{P}_{23}&\mathbf{P}_{33}\end{pmatrix}\mathbf{N}^T\begin{pmatrix}w^3\\ w^2\\ w\\ 1\end{pmatrix}, \tag{3.27}$$

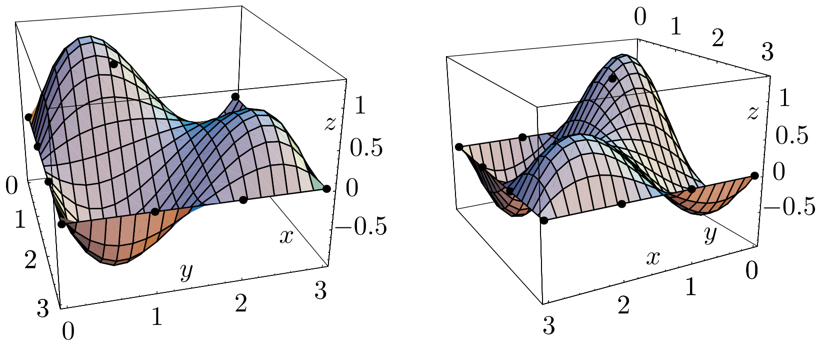
where

$$\mathbf{N}=\begin{pmatrix}-4.5&13.5&-13.5&4.5\\ 9.0&-22.5&18&-4.5\\ -5.5&9.0&-4.5&1.0\\ 1.0&0&0&0\end{pmatrix}.$$

is the basis matrix used to blend four points in a PC [Equation (3.6)]. As mentioned, this type of surface patch has only limited use because it cannot have a very complex shape. A larger surface, made up of a number of such patches, can be constructed, but it is difficult to connect the individual patches smoothly.

(This type of surface is also derived in Section 1.9 as a Cartesian product.)

Example: Given the 16 points listed in Figure 3.6, we compute and plot the bicubic surface patch defined by them. The figure shows two views of this surface.



```
<<:Graphics:ParametricPlot3D.m; (* BiCubic patch for 16 points *)
Clear[T,pnt,M,g1,g2];
T[t_]:={t^3,t^2,t,1};
pnt={{{0,0,0},{1,0,0},{2,0,0},{3,0,0}}, {{0,1,0},{1,1,1},{2,1,-.5},{3,1,0}},
 {{0,2,-.5},{1,2,0},{2,2,.5},{3,2,0}},{{0,3,0},{1,3,0},{2,3,0},{3,3,0}}};
M={{-4.5,13.5,-13.5,4.5},{9,-22.5,18,-4.5},{-5.5,9,-4.5,1},{1,0,0,0}};
g2=Graphics3D[{AbsolutePointSize[3],
Table[Point[pnt[[i,j]]],{i,1,4},{j,1,4}] }];
comb[i_]:=(T[u].M.pnt)[[i]](Transpose[M].T[w])[[i]];
g1=ParametricPlot3D[comb[1]+comb[2]+comb[3]+comb[4], {u,0,1},{w,0,1},
Compiled->False, DisplayFunction->Identity];
Show[g1,g2, ViewPoint->{2.752, -0.750, 1.265}, DefaultFont->{"cmr10", 10},
(*  ViewPoint->{1.413, 2.605, 0.974}  for alt view *)
DisplayFunction->$DisplayFunction]
```

Figure 3.6: A Bicubic Surface Patch Example.

Even though this type of surface has limited use in graphics, it can be used for two-dimensional bicubic polynomial interpolation of points and numbers. Given a set of three-dimensional points arranged in a two-dimensional grid, the problem is to compute a weighted sum of the points and employ it to predict the value of a new point at the center of the grid. It makes sense to assign more weights to points that are closer to the center, and a natural way to achieve this is to calculate the surface patch $\mathbf{P}(u, w)$ that passes through all the points in the grid and use the value $\mathbf{P}(0.5, 0.5)$ as the interpolated value at the center of the grid.

The MLP image compression method [Salomon 04] is an example of the use of this approach. The problem is to interpolate the values of a group of 4×4 pixels in an image in order to predict the value of a pixel at the center of this group. The simple solution is to calculate the surface patch defined by the 16 pixels and to use the surface point $\mathbf{P}(0.5, 0.5)$ as the interpolated value of the pixel at the center of the group. Substituting $u = 0.5$ and $w = 0.5$ in Equation (3.27) produces

$$\begin{aligned}
&\mathbf{P}(0.5, 0.5)\\
&\quad = 0.00390625\mathbf{P}_{00} - 0.0351563\mathbf{P}_{01} - 0.0351563\mathbf{P}_{02} + 0.00390625\mathbf{P}_{03}\\
&\qquad - 0.0351563\mathbf{P}_{10} + 0.316406\mathbf{P}_{11} + 0.316406\mathbf{P}_{12} - 0.0351563\mathbf{P}_{13}\\
&\qquad - 0.0351563\mathbf{P}_{20} + 0.316406\mathbf{P}_{21} + 0.316406\mathbf{P}_{22} - 0.0351563\mathbf{P}_{23}\\
&\qquad + 0.00390625\mathbf{P}_{30} - 0.0351563\mathbf{P}_{31} - 0.0351563\mathbf{P}_{32} + 0.00390625\mathbf{P}_{33}.
\end{aligned}$$

The 16 coefficients are the ones used by MLP.

◇ **Exercise 3.13:** The center point of the surface is calculated as a weighted sum of the 16 equally-spaced data points (this technique is known as bicubic interpolation). It makes sense to assign small weights to points located away from the center, but our result assigns *negative* weights to eight of the 16 points. Explain the meaning of negative weights and show what role they play in interpolating the center of the surface.

Readers who find it tedious to follow the details above should compare the way two-dimensional bicubic polynomial interpolation is presented here to the way it is discussed by [Press and Flannery 88]; the following quotation is from their page 125: "...the formulas that obtain the c's from the function and derivative values are just a complicated linear transformation, with coefficients which, having been determined once, in the mists of numerical history, can be tabulated and forgotten."

Seated at his disorderly desk, caressed by a counterpane of drifting tobacco haze, he would pore over the manuscript, crossing out, interpolating, re-arguing, and then referring to volumes on his shelves.

—Christopher Morley, *The Haunted Bookshop* (1919).

3.6.2 Four Curves

A variant of the previous method starts with four curves (any curves, not just PCs), $\mathbf{P}_0(u)$, $\mathbf{P}_1(u)$, $\mathbf{P}_2(u)$, and $\mathbf{P}_3(u)$, roughly parallel, all going in the u direction (Figure 3.5b). It is possible to select four points $\mathbf{P}_i(0)$, $\mathbf{P}_i(1/3)$, $\mathbf{P}_i(2/3)$, and $\mathbf{P}_i(1)$ on each curve $\mathbf{P}_i(u)$, for a total of 16 points. The surface patch can then easily be constructed from Equation (3.27).

Example: The surface of Figure 3.7 is defined by the following four curves (shown in the diagram in an inset). All go along the x axis, at different y values, and are sine curves (with different phases) along the z axis.

$$\mathbf{P}_0(u) = (u, 0, \sin(\pi u)), \quad \mathbf{P}_1(u) = (u, 1 + u/10, \sin(\pi(u + 0.1))),$$
$$\mathbf{P}_2(u) = (u, 2, \sin(\pi(u + 0.2))), \quad \mathbf{P}_3(u) = (u, 3 + u/10, \sin(\pi(u + 0.3))),$$

The *Mathematica* code of Figure 3.7 shows how matrix `basis` is created with the 16 points

$$\begin{pmatrix} \mathbf{P}_0(0) & \mathbf{P}_0(.33) & \mathbf{P}_0(.67) & \mathbf{P}_0(1) \\ \mathbf{P}_1(0) & \mathbf{P}_1(.33) & \mathbf{P}_1(.67) & \mathbf{P}_1(1) \\ \mathbf{P}_2(0) & \mathbf{P}_2(.33) & \mathbf{P}_2(.67) & \mathbf{P}_2(1) \\ \mathbf{P}_3(0) & \mathbf{P}_3(.33) & \mathbf{P}_3(.67) & \mathbf{P}_3(1) \end{pmatrix}.$$

3.7 Coons Surfaces

This type of surface is based on the pioneering work of Steven Anson Coons at MIT in the 1960s. His efforts are summarized in [Coons 64] and [Coons 67].

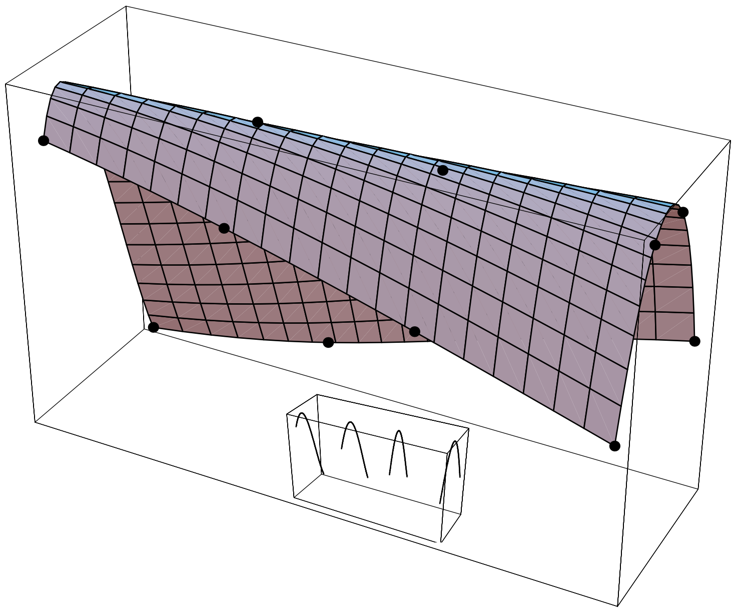
We start with the linear Coons surface, which is a generalization of lofted surfaces. This type of surface patch is defined by its four boundary curves. All four boundary curves are given, and none has to be a straight line. Naturally, the boundary curves have to meet at the corner points, so these points are implicitly known.

Coons decided to search for an expression $\mathbf{P}(u,w)$ of the surface that satisfies (1) it is symmetric in u and w and (2) it is an interpolation of $\mathbf{P}(u,0)$ and $\mathbf{P}(u,1)$ in one direction and of $\mathbf{P}(0,w)$ and $\mathbf{P}(1,w)$ in the other direction. He found a surprisingly simple, two-step solution.

The first step is to construct two lofted surfaces from the two sets of opposite boundary curves. They are $\mathbf{P}_a(u,w) = \mathbf{P}(0,w)(1-u) + \mathbf{P}(1,w)u$ and $\mathbf{P}_b(u,w) = \mathbf{P}(u,0)(1-w) + \mathbf{P}(u,1)w$.

The second step is to tentatively attempt to create the final surface $\mathbf{P}(u,w)$ as the sum $\mathbf{P}_a(u,w) + \mathbf{P}_b(u,w)$. It is clear that this is not the expression we are looking for because it does not converge to the right curves at the boundaries. For $u = 0$, for example, we want $\mathbf{P}(u,w)$ to converge to boundary curve $\mathbf{P}(0,w)$. The sum above, however, converges to $\mathbf{P}(0,w)+\mathbf{P}(0,0)(1-w)+\mathbf{P}(0,1)w$. We therefore have to subtract $\mathbf{P}(0,0)(1-w)+\mathbf{P}(0,1)w$. Similarly, for $u = 1$, the sum converges to $\mathbf{P}(1,w)+\mathbf{P}(1,0)(1-w) + \mathbf{P}(1,1)w$, so we have to subtract $\mathbf{P}(1,0)(1-w) + \mathbf{P}(1,1)w$. For $w = 0$, we have to subtract $\mathbf{P}(0,0)(1-u) + \mathbf{P}(1,0)u$, and for $w = 1$, we should subtract $\mathbf{P}(0,1)(1-u) + \mathbf{P}(1,1)u$.

Note that the expressions $\mathbf{P}(0,0)$, $\mathbf{P}(0,1)$, $\mathbf{P}(1,0)$, and $\mathbf{P}(1,1)$ are simply the four corner points. A better notation for them may be $\mathbf{P}_{00}$, $\mathbf{P}_{01}$, $\mathbf{P}_{10}$, and $\mathbf{P}_{11}$.



```
Clear[p0,p1,p2,p3,basis,fourP,g0,g1,g2,g3,g4,g5];
p0[u_]:={u,0,Sin[Pi u]}; p1[u_]:={u,1+u/10,Sin[Pi(u+.1)]};
p2[u_]:={u,2,Sin[Pi(u+.2)]}; p3[u_]:={u,3+u/10,Sin[Pi(u+.3)]};
(* matrix 'basis' has dimensions 4x4x3 *)
basis:={{p0[0],p0[.33],p0[.67],p0[1]},{p1[0],p1[.33],p1[.67],p1[1]},
{p2[0],p2[.33],p2[.67],p2[1]},{p3[0],p3[.33],p3[.67],p3[1]}};
fourP:= (* basis matrix for a 4-point curve *)
{{-4.5,13.5,-13.5,4.5},{9,-22.5,18,-4.5},{-5.5,9,-4.5,1},{1,0,0,0}};
prt[i_]:= (* extracts component i from the 3rd dimen of 'basis' *)
basis[[Range[1,4],Range[1,4],i]];
coord[i_]:= (* calc. the 3 parametric components of the surface *)
{u^3,u^2,u,1}.fourP.prt[i].Transpose[fourP].{w^3,w^2,w,1};
g0=ParametricPlot3D[p0[u], {u,0,1}]
g1=ParametricPlot3D[p1[u], {u,0,1}]
g2=ParametricPlot3D[p2[u], {u,0,1}]
g3=ParametricPlot3D[p3[u], {u,0,1}]
g4=Graphics3D[{AbsolutePointSize[4],
Table[Point[basis[[i,j]]],{i,1,4},{j,1,4}]}];
g5=ParametricPlot3D[{coord[1],coord[2],coord[3]},
{u,0,1,.05},{w,0,1,.05}, DisplayFunction->Identity];
Show[g0,g1,g2,g3, ViewPoint->{-2.576, -1.365, 1.718},
 Ticks->False, DisplayFunction->$DisplayFunction]
Show[g4,g5, ViewPoint->{-2.576, -1.365, 1.718},
 DisplayFunction->$DisplayFunction]
```

Figure 3.7: A Four-Curve Surface.

Today, this type of surface is known as the linear Coons surface. Its expression is $\mathbf{P}(u,w)=\mathbf{P}_a(u,w)+\mathbf{P}_b(u,w)-\mathbf{P}_{ab}(u,w)$, where

$$\mathbf{P}_{ab}(u,w)=\mathbf{P}_{00}(1-u)(1-w)+\mathbf{P}_{01}(1-u)w+\mathbf{P}_{10}u(1-w)+\mathbf{P}_{11}uw.$$

Note that $\mathbf{P}_a$ and $\mathbf{P}_b$ are lofted surfaces, whereas $\mathbf{P}_{ab}$ is a bilinear surface. The final expression is

$$\begin{aligned}\mathbf{P}(u,w)&=\mathbf{P}_a(u,w)+\mathbf{P}_b(u,w)-\mathbf{P}_{ab}(u,w)\\&=(1-u,u)\begin{pmatrix}\mathbf{P}(0,w)\\\mathbf{P}(1,w)\end{pmatrix}+(1-w,w)\begin{pmatrix}\mathbf{P}(u,0)\\\mathbf{P}(u,1)\end{pmatrix}\\&\quad-(1-u,u)\begin{pmatrix}\mathbf{P}_{00}&\mathbf{P}_{01}\\\mathbf{P}_{10}&\mathbf{P}_{11}\end{pmatrix}\begin{pmatrix}1-w\\w\end{pmatrix} \qquad (3.28)\\&=(1-u,u,1)\begin{pmatrix}-\mathbf{P}_{00}&-\mathbf{P}_{01}&\mathbf{P}(0,w)\\-\mathbf{P}_{10}&-\mathbf{P}_{11}&\mathbf{P}(1,w)\\\mathbf{P}(u,0)&\mathbf{P}(u,1)&(0,0,0)\end{pmatrix}\begin{pmatrix}1-w\\w\\1\end{pmatrix}. \qquad (3.29)\end{aligned}$$

Equation (3.28) is more useful than Equation (3.29) since it shows how the surface is defined in terms of the two barycentric pairs $(1-u,u)$ and $(1-w,w)$. They are the blending functions of the linear Coons surface. It turns out that many pairs of barycentric functions $\big(f_1(u),f_2(u)\big)$ and $\big(g_1(w),g_2(w)\big)$ can serve as blending functions, out of which more general Coons surfaces can be constructed. All that the blending functions have to satisfy is

$$\begin{aligned}&f_1(0)=1,\quad f_1(1)=0,\quad f_2(0)=0,\quad f_2(1)=1,\quad f_1(u)+f_2(u)=1,\\&g_1(0)=1,\quad g_1(1)=0,\quad g_2(0)=0,\quad g_2(1)=1,\quad g_1(w)+g_2(w)=1.\end{aligned} \qquad (3.30)$$

Example: We select the four (nonpolynomial) boundary curves

$$\begin{aligned}\mathbf{P}_{u0}&=(u,0,\sin(\pi u)),\quad \mathbf{P}_{u1}=(u,1,\sin(\pi u)),\\\mathbf{P}_{0w}&=(0,w,\sin(\pi w)),\quad \mathbf{P}_{1w}=(1,w,\sin(\pi w)).\end{aligned}$$

Each is one-half of a sine wave. The first two proceed along the x axis, and the other two go along the y axis. They meet at the four corner points $\mathbf{P}_{00}=(0,0,0)$, $\mathbf{P}_{01}=(0,1,0)$, $\mathbf{P}_{10}=(1,0,0)$, and $\mathbf{P}_{11}=(1,1,0)$. The surface and the *Mathematica* code that produced it are shown in Figure 3.8. Note the `Simplify` command, which displays the final, simplified expression of the surface `{u, w, Sin[Pi u] + Sin[Pi w]}`.

Example: Given the four corner points $\mathbf{P}_{00}=(-1,-1,0)$, $\mathbf{P}_{01}=(-1,1,0)$, $\mathbf{P}_{10}=(1,-1,0)$, and $\mathbf{P}_{11}=(1,1,0)$ (notice that they lie on the xy plane), we calculate the four boundary curves of a linear Coons surface patch as follows:

1. We select boundary curve $\mathbf{P}(0,w)$ as the straight line from $\mathbf{P}_{00}$ to $\mathbf{P}_{01}$:

$$\mathbf{P}(0,w)=\mathbf{P}_{00}(1-w)+\mathbf{P}_{01}w=(-1,2w-1,0).$$

```
<<:Graphics:ParametricPlot3D.m;
Clear[p00,p01,p10,p11,pu0,pu1,p0w,p1w];
p00:={0,0,0}; p01:={0,1,0};
p10:={1,0,0}; p11:={1,1,0};
pu0:={u,0,Sin[Pi u]};
pu1:={u,1,Sin[Pi u]};
p0w:={0,w,Sin[Pi w]};
p1w:={1,w,Sin[Pi w]};
Simplify[
{1-u,u}.{p0w,p1w}+{1-w,w}.{pu0,pu1}
-p00(1-u)(1-w)-p01(1-u)w
-p10(1-w)u-p11 u w]
ParametricPlot3D[%,
{u,0,1,.2},{w,0,1,.2},
PlotRange->All,
AspectRatio->Automatic,
RenderAll->False,
Ticks->{{1},{0,1},{0,1}},
Prolog->AbsoluteThickness[.4]]
```

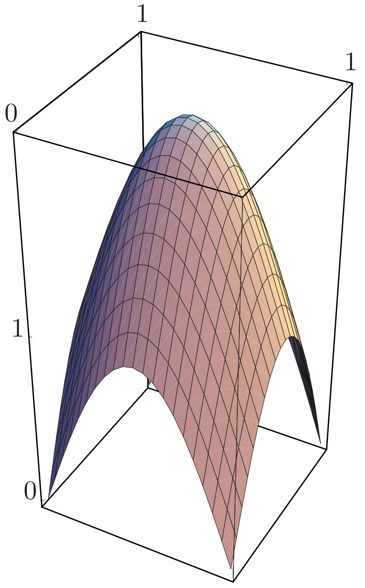


Figure 3.8: A Coons Surface.

2. We place the two points $(1,-0.5,0.5)$ and $(1,0.5,-0.5)$ between $\mathbf{P}_{10}$ and $\mathbf{P}_{11}$ and calculate boundary curve $\mathbf{P}(1,w)$ as the cubic Lagrange polynomial [Equation (3.17)] determined by these four points

$$\begin{aligned}\mathbf{P}(1,w)&=\frac{1}{2}(w^3,w^2,w,1)\begin{bmatrix}-9&-27&27&9\\18&-45&36&-9\\-11&18&-9&2\\2&0&0&0\end{bmatrix}\begin{bmatrix}(1,-1,0)\\(1,-0.5,0.5)\\(1,0.5,-0.5)\\(1,1,0)\end{bmatrix}\\&=\big(1,(-4-w+27w^2-18w^3)/4,27(w-3w^2+2w^3)/4\big).\end{aligned}$$

3. The single point $(0,-1,-0.5)$ is placed between points $\mathbf{P}_{00}$ and $\mathbf{P}_{10}$ and boundary curve $\mathbf{P}(u,0)$ is calculated as the quadratic Lagrange polynomial [Equation (3.14)] determined by these three points:

$$\mathbf{P}(u,0)=(u^2,u,1)\begin{bmatrix}2&-4&2\\-3&4&-1\\1&0&0\end{bmatrix}\begin{bmatrix}(-1,-1,0)\\(0,-1,-.5)\\(1,-1,0)\end{bmatrix}=(2u-1,-1,2u^2-2u).$$

4. Similarly, a new point $(0,1,.5)$ is placed between points $\mathbf{P}_{01}$ and $\mathbf{P}_{11}$, and boundary curve $\mathbf{P}(u,1)$ is calculated as the quadratic Lagrange polynomial determined

by these three points:

$$\mathbf{P}(u,1)=(u^2,u,1)\begin{bmatrix}2&-4&2\\-3&4&-1\\1&0&0\end{bmatrix}\begin{bmatrix}(-1,1,0)\\(0,1,.5)\\(1,1,0)\end{bmatrix}=(2u-1,1,-2u^2+2u).$$

The four boundary curves and the four corner points now become the linear Coons surface patch given by Equation (3.28):

$$\mathbf{P}(u,w)=(1-u,u,1)\begin{bmatrix}-(-1,-1,0)&-(-1,1,0)&(-1,2w-1,0)\\-(1,-1,0)&-(1,1,0)&(1,(-4-w+27w^2-18w^3)/4,27(w-3w^2+2w^3)/4)\\(2u-1,-1,2u^2-2u)&(2u-1,1,-2u^2+2u)&0\end{bmatrix}\begin{bmatrix}1-w\\w\\1\end{bmatrix}.$$

This is simplified with the help of appropriate software and becomes

$$\begin{aligned}\mathbf{P}(u,w)=\bigl(&-1+2u+(1-u)(1-w)-u(1-w)+(-1+2u)(1-w)\\&+(1-u)w-uw+(-1+2u)w,\\&-1+(1-u)(1-w)+u(1-w)+2w-(1-u)w\\&-uw+(1-u)(-1+2w)+u(-4-w+27w^2-18w^3)/4,\\&(-2u+2u^2)(1-w)+(2u-2u^2)w+27u(w-3w^2+2w^3)/4\bigr).\end{aligned}$$

The surface patch and the eight points involved are shown in Figure 3.9.

3.7.1 Translational Surfaces

Given two curves $\mathbf{P}(u,0)$ and $\mathbf{P}(0,w)$ that intersect at a point

$$\mathbf{P}(u,0)|_{u=0}=\mathbf{P}(0,w)|_{w=0}\overset{\text{def}}{=}\mathbf{P}_{00},$$

it is easy to construct the surface patch created by sliding one of the curves, say, $\mathbf{P}(u,0)$, along the other one (Figure 3.10).

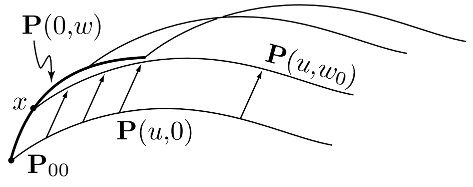
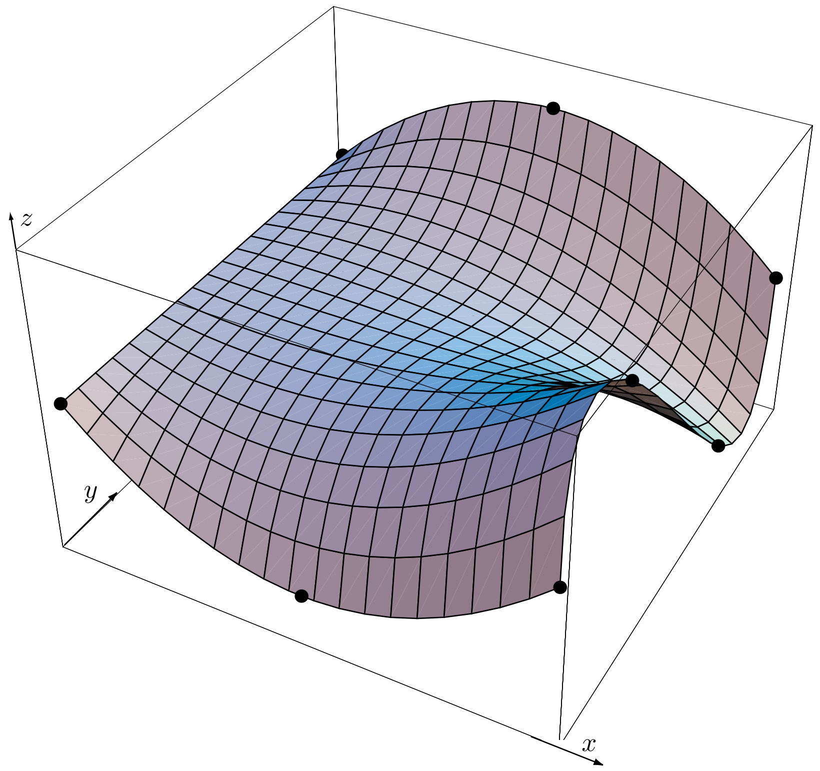


Figure 3.10: A Translational Surface.



```
p00={-1,-1,0}; p01={-1,1,0}; p10={1,-1,0}; p11={1,1,0};
pnts={p00,p01,p10,p11,{1,-1/2,1/2},{1,1/2,-1/2},
 {0,-1,-1/2},{0,1,1/2}};
p0w[w_]:={-1,2w-1,0};
p1w[w_]:={1,(-4-w+27w^2-18w^3)/4,27(w-3w^2+2w^3)/4};
pu0[u_]:={2u-1,-1,2u^2-2u};
pu1[u_]:={2u-1,1,-2u^2+2u};
p[u_,w_]:=(1-u)p0w[w]+u p1w[w]+(1-w)pu0[u]+w pu1[u] \
 -p00(1-u)(1-w)-p01(1-u)
w-p10 u(1-w)-p11 u w;
g1=Graphics3D[{AbsolutePointSize[5], Table[Point[pnts[[i]]],
{i,1,8}]}];
g2=ParametricPlot3D[p[u,w], {u,0,1},{w,0,1}, Compiled->False,
Ticks->{{-1,1},{-1,1},{-1,1}}, DisplayFunction->Identity];
Show[g1,g2]
```

Figure 3.9: A Coons Surface Patch and Code.

We fix w at a certain value w_0 and compute the vector from the intersection point $\mathbf{P}_{00}$ to point $\mathbf{P}(0,w_0)$ (marked with an x in the figure). This vector is the difference $\mathbf{P}(0,w_0)-\mathbf{P}_{00}$, implying that any point on the curve $\mathbf{P}(u,w_0)$ can be obtained by adding this vector to the corresponding point on curve $\mathbf{P}(u,0)$. The entire curve $\mathbf{P}(u,w_0)$ is therefore constructed as the sum $\mathbf{P}(u,0)+[\mathbf{P}(0,w_0)-\mathbf{P}_{00}]$ for $0\le u\le 1$. The resulting *translational surface* $\mathbf{P}(u,w)$ is obtained when w is released and is varied in the interval $[0,1]$

$$\mathbf{P}(u,w)=\mathbf{P}(u,0)+\mathbf{P}(0,w)-\mathbf{P}_{00}.$$

There is an interesting relation between the linear Coons surface and translational surfaces. The Coons patch is constructed from four intersecting curves. Consider a pair of such curves that intersect at a corner $\mathbf{P}_{ij}$ of the Coons patch. We can employ this pair and the corner to construct a translational surface $\mathbf{P}_{ij}(u,w)$. Once we construct the four translational surfaces for the four corners of the Coons patch, they can be used to express the entire Coons linear surface patch by a special version of Equation (3.29)

$$(1-u,u)\begin{bmatrix}\mathbf{P}_{00}(u,w) & \mathbf{P}_{01}(u,w)\\ \mathbf{P}_{10}(u,w) & \mathbf{P}_{11}(u,w)\end{bmatrix}\begin{bmatrix}1-w\\ w\end{bmatrix}.$$

This version expresses the Coons surface patch as a weighted combination of four translational surfaces

3.7.2 Higher-Degree Coons Surfaces

One possible pair of blending functions is the cubic Hermite polynomials, functions $F_1(t)$ and $F_2(t)$ of Equation (4.6)

$$\begin{aligned}H_{3,0}(t)&=B_{3,0}(t)+B_{3,1}(t)=(1-t)^3+3t(1-t)^2=1+2t^3-3t^2,\\ H_{3,3}(t)&=B_{3,2}(t)+B_{3,3}(t)=3t^2(1-t)+t^3=3t^2-2t^3,\end{aligned} \tag{3.31}$$

where $B_{n,i}(t)$ are the Bernstein polynomials, Equation (6.5). The sum $H_{3,0}(t)+H_{3,3}(t)$ is identically 1 (because the Bernstein polynomials are barycentric), so these functions can be used to construct the *bicubic Coons surface.* Its expression is

$$\begin{aligned}\mathbf{P}(u,w)&=(H_{3,0}(u),H_{3,3}(u),1)\begin{bmatrix}-\mathbf{P}_{00} & -\mathbf{P}_{01} & \mathbf{P}(0,w)\\ -\mathbf{P}_{10} & -\mathbf{P}_{11} & \mathbf{P}(1,w)\\ \mathbf{P}(u,0) & \mathbf{P}(u,1) & 0\end{bmatrix}\begin{bmatrix}H_{3,0}(w)\\ H_{3,3}(w)\\ 1\end{bmatrix} \qquad (3.32)\\ &=(1+2u^3-3u^2,3u^2-2u^3,1)\begin{bmatrix}-\mathbf{P}_{00} & -\mathbf{P}_{01} & \mathbf{P}(0,w)\\ -\mathbf{P}_{10} & -\mathbf{P}_{11} & \mathbf{P}(1,w)\\ \mathbf{P}(u,0) & \mathbf{P}(u,1) & (0,0,0)\end{bmatrix}\begin{bmatrix}1+2w^3-3w^2\\ 3w^2-2w^3\\ 1\end{bmatrix}.\end{aligned}$$

One advantage of the bicubic Coons surface patch is that it is especially easy to connect smoothly to other patches of the same type. This is because its blending functions satisfy

$$\left.\frac{dH_{3,0}(t)}{dt}\right|_{t=0}=0,\quad \left.\frac{dH_{3,0}(t)}{dt}\right|_{t=1}=0,\quad \left.\frac{dH_{3,3}(t)}{dt}\right|_{t=0}=0,\quad \left.\frac{dH_{3,3}(t)}{dt}\right|_{t=1}=0. \tag{3.33}$$

Figure 3.11 shows two bicubic Coons surface patches, $\mathbf{P}(u,w)$ and $\mathbf{Q}(u,w)$, connected along their boundary curves $\mathbf{P}(u,1)$ and $\mathbf{Q}(u,0)$, respectively. The condition for patch connection is, of course, $\mathbf{P}(u,1)=\mathbf{Q}(u,0)$. The condition for smooth connection is

$$\left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{w=1}=\left.\frac{\partial\mathbf{Q}(u,w)}{\partial w}\right|_{w=o} \tag{3.34}$$

(but see Section 1.10 for other, less restrictive conditions).

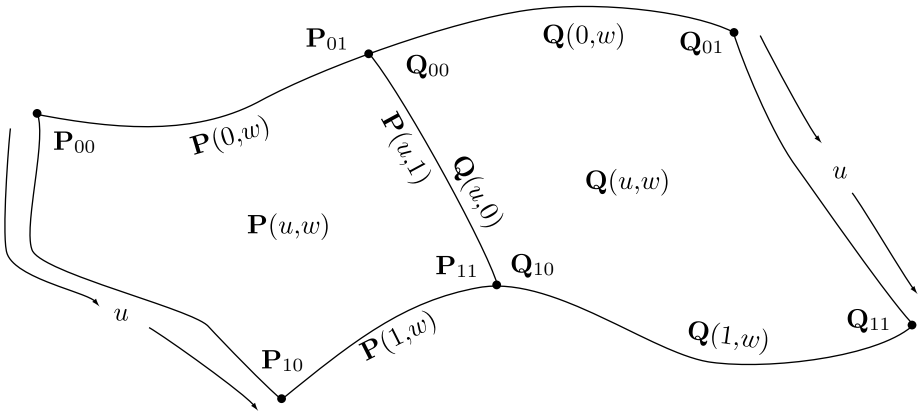


Figure 3.11: Smooth Connection of Bicubic Coons Surface Patches.

The partial derivatives of $\mathbf{P}(u,w)$ are easy to calculate from Equation (3.32). They are

$$\begin{aligned}\left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{w=1}&=H_{3,0}(u)\left.\frac{d\mathbf{P}(0,w)}{dw}\right|_{w=1}+H_{3,3}(u)\left.\frac{d\mathbf{P}(1,w)}{dw}\right|_{w=1},\\ \left.\frac{\partial\mathbf{Q}(u,w)}{\partial w}\right|_{w=0}&=H_{3,0}(u)\left.\frac{d\mathbf{Q}(0,w)}{dw}\right|_{w=0}+H_{3,3}(u)\left.\frac{d\mathbf{Q}(1,w)}{dw}\right|_{w=0}.\end{aligned} \tag{3.35}$$

[All other terms vanish because the blending functions satisfy Equation (3.33).] The condition for smooth connection, Equation (3.34), is therefore satisfied if

$$\left.\frac{d\mathbf{P}(0,w)}{dw}\right|_{w=1}=\left.\frac{d\mathbf{Q}(0,w)}{dw}\right|_{w=0}\quad\text{and}\quad\left.\frac{d\mathbf{P}(1,w)}{dw}\right|_{w=1}=\left.\frac{d\mathbf{Q}(1,w)}{dw}\right|_{w=0},$$

or, expressed in words, if the two boundary curves $\mathbf{P}(0,w)$ and $\mathbf{Q}(0,w)$ on the $u=0$ side of the patch connect smoothly, and the same for the two boundary curves $\mathbf{P}(1,w)$ and $\mathbf{Q}(1,w)$ on the $u=1$ side of the patch.

The reader should now find it easy to appreciate the advantage of the degree-5 Hermite blending functions [functions $F_1(t)$ and $F_2(t)$ of Equation (4.17)]

$$\begin{aligned}H_{5,0}(t)&=B_{5,0}(t)+B_{5,1}(t)+B_{5,2}(t)=1-10t^3+15t^4-6t^5,\\ H_{5,5}(t)&=B_{5,3}(t)+B_{5,4}(t)+B_{5,5}(t)=10t^3-15t^4+6t^5.\end{aligned} \tag{3.36}$$

They are based on the Bernstein polynomials $B_{5,i}(t)$ hence they satisfy the conditions of Equation (3.30). They further have the additional property that their first *and* second derivatives are zero for $t = 0$ and for $t = 1$. The degree-5 Coons surface constructed by them is

$$\mathbf{P}_5(u,w) = \bigl(H_{5,0}(u), H_{5,5}(u), 1\bigr) \begin{bmatrix} -\mathbf{P}_{00} & -\mathbf{P}_{01} & \mathbf{P}(0,w) \\ -\mathbf{P}_{10} & -\mathbf{P}_{11} & \mathbf{P}(1,w) \\ \mathbf{P}(u,0) & \mathbf{P}(u,1) & 0 \end{bmatrix} \begin{bmatrix} H_{5,0}(w) \\ H_{5,5}(w) \\ 1 \end{bmatrix}. \qquad (3.37)$$

Adjacent patches of this type of surface are easy to connect with G^2 continuity. All that's necessary is to have two pairs of boundary curves $\mathbf{P}(0,w)$, $\mathbf{Q}(0,w)$ and $\mathbf{P}(1,w)$, $\mathbf{Q}(1,w)$, where the two curves of each pair connect with G^2 continuity.

3.7.3 The Tangent Matching Coons Surface

The original aim of Coons was to construct a surface patch where all four boundary curves are specified by the user. Such patches are easy to compute and the conditions for connecting them smoothly are simple. It is possible to extend the original ideas of Coons to a surface patch where the user specifies the four boundary curves and also four functions that describe how (in what direction) this surface approaches its boundaries. Figure 3.12 illustrates the meaning of this statement. It shows a rectangular surface patch with some curves of the form $\mathbf{P}(u,w_i)$. Each of these curves goes from boundary curve $\mathbf{P}(0,w)$ to the opposite boundary curve $\mathbf{P}(1,w)$ by varying its parameter u from 0 to 1. Each has a different value of w_i. When such a curve reaches its end, it is moving in a certain, well-defined direction shown in the diagram. The end tangent vectors of these curves are different and we can imagine a function that yields these tangents as we move along the boundary curve $\mathbf{P}(1,w)$, varying w from 0 to 1. A good name for such a function is $\mathbf{P}_u(1,w)$, where the subscript u indicates that this tangent of the surface is in the u direction, the index 1 indicates the tangent at the end ($u = 1$), and the w indicates that this tangent vector is a function of w.

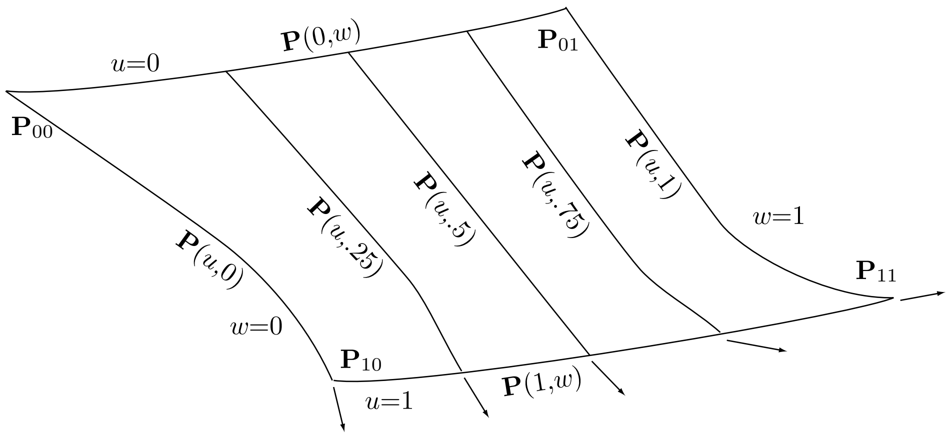


Figure 3.12: Tangent Matching in a Coons Surface.

There are four such functions, namely $\mathbf{P}_u(0,w)$, $\mathbf{P}_u(1,w)$, $\mathbf{P}_w(u,0)$, and $\mathbf{P}_w(u,1)$. Assuming that the user provides these functions, as well as the four boundary curves, our task is to obtain an expression $\mathbf{P}(u,w)$ for the surface that will satisfy the following:

1. When we substitute 0 or 1 for u and w in $\mathbf{P}(u,w)$, we get the four given corner points and the four given boundary curves. This condition can be expressed as the eight constraints

$$\mathbf{P}(0,0)=\mathbf{P}_{00},\quad \mathbf{P}(0,1)=\mathbf{P}_{01},\quad \mathbf{P}(1,0)=\mathbf{P}_{10},\quad \mathbf{P}(1,1)=\mathbf{P}_{11},$$
$$\mathbf{P}(0,w),\quad \mathbf{P}(1,w),\quad \mathbf{P}(u,0),\ \text{and}\ \mathbf{P}(u,1)\quad \text{are the given boundary curves.}$$

2. When we substitute 0 or 1 for u and w in the partial first derivatives of $\mathbf{P}(u,w)$, we get the four given tangent functions and their values at the four corner points. This condition can be expressed as the 12 constraints

$$\begin{aligned}
\left.\frac{\partial\mathbf{P}(u,w)}{\partial u}\right|_{u=0}&=\mathbf{P}_u(0,w), & \left.\frac{\partial\mathbf{P}(u,w)}{\partial u}\right|_{u=1}&=\mathbf{P}_u(1,w),\\
\left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{w=0}&=\mathbf{P}_w(u,0), & \left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{w=1}&=\mathbf{P}_w(u,1),\\
\left.\frac{\partial\mathbf{P}(u,w)}{\partial u}\right|_{u=0,w=0}&=\mathbf{P}_u(0,0), & \left.\frac{\partial\mathbf{P}(u,w)}{\partial u}\right|_{u=0,w=1}&=\mathbf{P}_u(0,1),\\
\left.\frac{\partial\mathbf{P}(u,w)}{\partial u}\right|_{u=1,w=0}&=\mathbf{P}_u(1,0), & \left.\frac{\partial\mathbf{P}(u,w)}{\partial u}\right|_{u=1,w=1}&=\mathbf{P}_u(1,1),\\
\left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{u=0,w=0}&=\mathbf{P}_w(0,0), & \left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{u=0,w=1}&=\mathbf{P}_w(0,1).\\
\left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{u=1,w=0}&=\mathbf{P}_w(1,0), & \left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{u=1,w=1}&=\mathbf{P}_w(1,1).
\end{aligned}$$

3. When we substitute 0 or 1 for u and w in the partial second derivatives of $\mathbf{P}(u,w)$, we get the four first derivatives of the given tangent functions at the four corner points. This condition can be expressed as the four constraints

$$\begin{aligned}
\left.\frac{\partial^2\mathbf{P}(u,w)}{\partial u\partial w}\right|_{u=0,w=0}&=\left.\frac{d\mathbf{P}_u(0,w)}{dw}\right|_{w=0}=\left.\frac{d\mathbf{P}_u(u,0)}{du}\right|_{u=0}\stackrel{\text{def}}{=}\mathbf{P}_{uw}(0,0),\\
\left.\frac{\partial^2\mathbf{P}(u,w)}{\partial u\partial w}\right|_{u=0,w=1}&=\left.\frac{d\mathbf{P}_u(0,w)}{dw}\right|_{w=1}=\left.\frac{d\mathbf{P}_u(u,1)}{du}\right|_{u=0}\stackrel{\text{def}}{=}\mathbf{P}_{uw}(0,1),\\
\left.\frac{\partial^2\mathbf{P}(u,w)}{\partial u\partial w}\right|_{u=1,w=0}&=\left.\frac{d\mathbf{P}_u(1,w)}{dw}\right|_{w=0}=\left.\frac{d\mathbf{P}_u(u,0)}{du}\right|_{u=1}\stackrel{\text{def}}{=}\mathbf{P}_{uw}(1,0),\\
\left.\frac{\partial^2\mathbf{P}(u,w)}{\partial u\partial w}\right|_{u=1,w=1}&=\left.\frac{d\mathbf{P}_u(1,w)}{dw}\right|_{w=1}=\left.\frac{d\mathbf{P}_u(u,1)}{du}\right|_{u=1}\stackrel{\text{def}}{=}\mathbf{P}_{uw}(1,1).
\end{aligned}$$

This is a total of 24 constraints. A derivation of this type of surface can be found in [Beach 91]. Here, we only quote the final result

$$\mathbf{P}(u,w) = \bigl(B_0(u), B_1(u), C_0(u), C_1(u), 1\bigr)\mathbf{M}\begin{bmatrix} B_0(w)\\ B_1(w)\\ C_0(w)\\ C_1(w)\\ 1 \end{bmatrix}, \tag{3.38}$$

where $\mathbf{M}$ is the 5×5 matrix

$$\mathbf{M} = \begin{bmatrix} -\mathbf{P}_{00} & -\mathbf{P}_{01} & -\mathbf{P}_w(0,0) & -\mathbf{P}_w(0,1) & \mathbf{P}(0,w)\\ -\mathbf{P}_{10} & -\mathbf{P}_{11} & -\mathbf{P}_w(1,0) & -\mathbf{P}_w(1,1) & \mathbf{P}(1,w)\\ -\mathbf{P}_u(0,0) & -\mathbf{P}_u(0,1) & -\mathbf{P}_{uw}(0,0) & -\mathbf{P}_{uw}(0,1) & \mathbf{P}_u(0,w)\\ -\mathbf{P}_u(1,0) & -\mathbf{P}_u(1,1) & -\mathbf{P}_{uw}(1,0) & -\mathbf{P}_{uw}(1,1) & \mathbf{P}_u(1,w)\\ \mathbf{P}(u,0) & \mathbf{P}(u,1) & \mathbf{P}_w(u,0) & \mathbf{P}_w(u,1) & (0,0,0) \end{bmatrix}. \tag{3.39}$$

The two blending functions $B_0(t)$ and $B_1(t)$ can be any functions satisfying conditions (3.30) and (3.33). Examples are the pairs $H_{3,0}(t)$, $H_{3,3}(t)$ and $H_{5,0}(t)$, $H_{5,5}(t)$ of Equations (3.31) and (3.36). The two blending functions $C_0(t)$ and $C_1(t)$ should satisfy

$$\begin{aligned} &C_0(0)=0, \quad C_0(1)=0, \quad C_0'(0)=1, \quad C_0'(1)=0,\\ &C_1(0)=0, \quad C_1(1)=0, \quad C_1'(0)=0, \quad C_1'(1)=1. \end{aligned}$$

One choice is the pair $C_0(t) = t - 2t^2 + t^3$ and $C_1(t) = -t^2 + t^3$.

Such a surface patch is difficult to specify. The user has to input the four boundary curves and four tangent functions, a total of eight functions. The user then has to calculate the coordinates of the four corner points and the other 12 quantities required by the matrix of Equation (3.39). The advantage of this type of surface is that once fully specified, such a surface patch is easy to connect smoothly to other patches of the same type since the tangents along the boundaries are fully specified by the user.

3.7.4 The Triangular Coons Surface

A triangular surface patch is bounded by three boundary curves and has three corner points. Such surface patches are handy in situations like the one depicted in Figure 3.15, where a triangular Coons patch is used to smoothly connect two perpendicular lofted surface patches. Section 6.23 discusses the triangular Bézier surface patch which is commonly used in practice. Our approach to constructing the triangular Coons surface is to merge two of the four corner points and explore the behavior of the resulting surface patch. We arbitrarily decide to set $\mathbf{P}_{01} = \mathbf{P}_{11}$, which reduces the boundary curve $\mathbf{P}(u,1)$ to a single point (Figure 3.13). The expression of this triangular surface patch is

$$\mathbf{P}(u,w) = \bigl(B_0(u), B_1(u), 1\bigr)\begin{pmatrix} -\mathbf{P}_{00} & -\mathbf{P}_{11} & \mathbf{P}(0,w)\\ -\mathbf{P}_{10} & -\mathbf{P}_{11} & \mathbf{P}(1,w)\\ \mathbf{P}(u,0) & \mathbf{P}_{11} & (0,0,0) \end{pmatrix}\begin{pmatrix} B_0(w)\\ B_1(w)\\ 1 \end{pmatrix}, \tag{3.40}$$

where the blending functions $B_0(t)$, $B_1(t)$ can be the pair $H_{3,0}$ and $H_{3,3}$, or the pair $H_{5,0}$ and $H_{5,5}$, or any other pair of blending functions satisfying Equations (3.30) and (3.33).

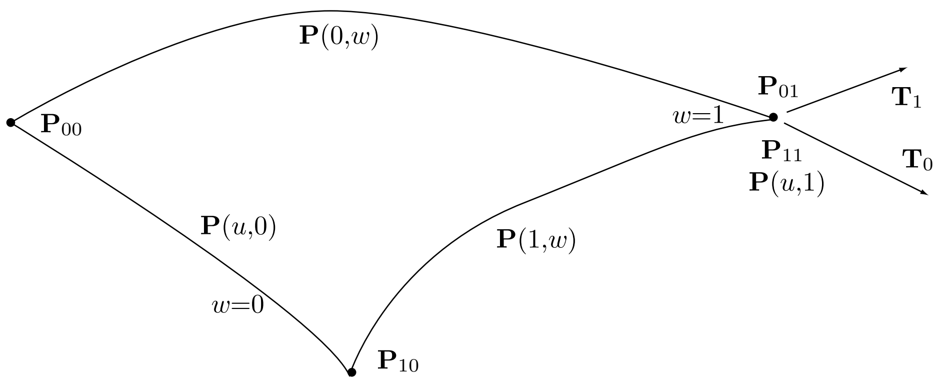


Figure 3.13: A Triangular Coons Surface Patch.

The tangent vector of the surface along the degenerate boundary curve $\mathbf{P}(u,1)$ is given by Equation (3.35):

$$\left.\frac{\partial\mathbf{P}(u,w)}{\partial w}\right|_{w=1} = B_0(u)\left.\frac{d\mathbf{P}(0,w)}{dw}\right|_{w=1} + B_1(u)\left.\frac{d\mathbf{P}(1,w)}{dw}\right|_{w=1}. \tag{3.41}$$

Thus, this tangent vector is a linear combination of the two tangents

$$\mathbf{T}_0 \overset{\text{def}}{=} \left.\frac{d\mathbf{P}(0,w)}{dw}\right|_{w=1} \quad\text{and}\quad \mathbf{T}_1 \overset{\text{def}}{=} \left.\frac{d\mathbf{P}(1,w)}{dw}\right|_{w=1},$$

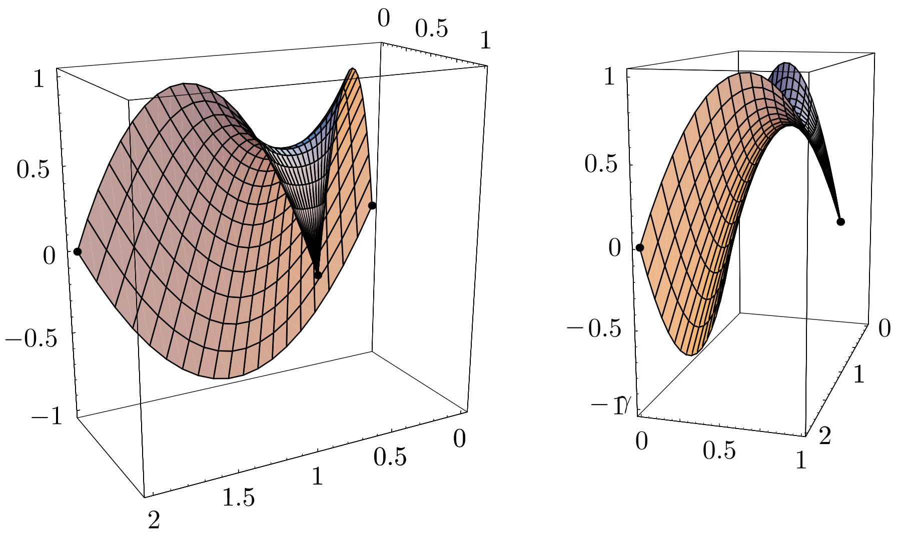
and therefore lies in the plane defined by them. As u varies from 0 to 1, this tangent vector swings from $\mathbf{T}_0$ to $\mathbf{T}_1$ while the curve $\mathbf{P}(u,1)$ stays at the common point $\mathbf{P}_{01} = \mathbf{P}_{11}$. Once this behavior is grasped, the reader should be able to accept the following statement: The triangular patch will be well behaved in the vicinity of the common point if this tangent vector does not reverse its movement while swinging from $\mathbf{T}_0$ to $\mathbf{T}_1$. If it starts moving toward $\mathbf{T}_1$, then reverses and goes back toward $\mathbf{T}_0$, then reverses again, the surface may have a *fold* close to the common point. To guarantee this smooth behavior of the tangent vector, the blending functions $B_0(t)$ and $B_1(t)$ must satisfy one more condition, namely $B_0(t)$ should be monotonically decreasing in t and $B_1(t)$ should be monotonically increasing in t. The two sets of blending functions $H_{3,0}$, $H_{3,3}$ and $H_{5,0}$, $H_{5,5}$ satisfy this condition and can therefore be used to construct triangular Coons surface patches.

Example: Given the three corners $\mathbf{P}_{00} = (0,0,0)$, $\mathbf{P}_{10} = (2,0,0)$, and $\mathbf{P}_{01} = \mathbf{P}_{11} = (1,1,0)$, we compute and plot the triangular Coons surface patch defined by them. The first step is to compute the three boundary curves. We assume that the "bottom" boundary curve $\mathbf{P}(u,0)$ goes from $\mathbf{P}_{00}$ through $(1,0,-1)$ to $\mathbf{P}_{10}$. We similarly require that the "left" boundary curve $\mathbf{P}(0,w)$ goes from $\mathbf{P}_{00}$ through $(0.5,0.5,1)$ to $\mathbf{P}_{01}$ and the "right" boundary curve $\mathbf{P}(1,w)$ goes from $\mathbf{P}_{10}$ through $(1.5,0.5,1)$ to $\mathbf{P}_{11}$. All three curves are computed as standard quadratic Lagrange polynomials from Equation (3.14).

They become

$$\mathbf{P}(u,0) = (2u, 0, 4u(u-1)),$$
$$\mathbf{P}(0,w) = (w, w, 4w(1-w)),$$
$$\mathbf{P}(1,w) = (2-w, w, 4w(w-1)).$$

Figure 3.14 shows two views of this surface and illustrates the downside of this type of surface. The technique of drawing a surface patch as a wireframe with two families of curves works well for rectangular surface patches but is unsuitable for triangular patches. The figure shows how one family of curves converges to the double corner point, thereby making the wireframe look unusually dense in the vicinity of the point. Section 6.23 presents a better approach to the display of a triangular surface patch as a wireframe.



```
<<:Graphics:ParametricPlot3D.m; (* Triangular Coons patch *)
Clear[T,pnt,M,g1,g2];
T[t_]:={1+2t^3-3t^2,3t^2-2t^3,1};
p00={0,0,0}; p10={2,0,0}; p11={1,1,0};
M={{-p00,-p11,{w,w,4w(1-w)}},{-p10,-p11,{2-w,w,4w(1-w)}},
 {{2u,0,4u(u-1)},p11,{0,0,0}}};
g2=Graphics3D[{AbsolutePointSize[3],Point[p00], Point[p10], Point[p11] }];
comb[i_]:=(T[u].M)[[i]] T[w][[i]];
g1=ParametricPlot3D[comb[1]+comb[2]+comb[3], {u,0,1},{w,0,1},
Compiled->False, DisplayFunction->Identity];
Show[g1,g2, ViewPoint->{2.933, 0.824, 0.673}, DefaultFont->{"cmr10", 10},
DisplayFunction->$DisplayFunction]
(*ViewPoint->{1.413, 2.605, 0.974} for alt view *)
```

Figure 3.14: A Triangular Coons Surface Patch Example.

◇ **Exercise 3.14:** What happens if the blending functions of the triangular Coons surface patch do not satisfy the condition of Equation (3.33)?

"Now, don't worry, my pet," Mrs. Whatsit said cheerfully. "We took care of that before we left. Your mother has had enough to worry her with you and Charles to cope with, and not knowing about your father, without our adding to her anxieties. We took a time wrinkle as well as a space wrinkle. It's very easy to do if you just know how."

—Madeleine L'Engle, *A Wrinkle in Time* (1962).

◇ **Exercise 3.15:** Given the four points $\mathbf{P}_{00} = (0,0,1)$, $\mathbf{P}_{10} = (1,0,0)$, $\mathbf{P}_{01} = (0.5,1,0)$, and $\mathbf{P}_{11} = (1,1,0)$, calculate the Coons surface defined by them, assuming straight lines as boundary curves. What type of a surface is this?

3.7.5 Summarizing Example

The surface shown in Figure 3.15 consists of four (intentionally separated) patches. A flat bilinear patch $\mathbf{B}$ at the top, two lofted patches $\mathbf{L}$ and $\mathbf{F}$ on both sides, and a triangular Coons patch $\mathbf{C}$ filling up the corner.

The bilinear patch is especially simple since it is defined by its four corner points. Its expression is

$$\begin{aligned}\mathbf{B}(u,w) &= (0,1/2,1)(1-u)(1-w) + (1,1/2,1)(1-u)w\\ &\quad + (0,3/2,1)(1-w)u + (1,3/2,1)uw\\ &= (w, 1/2+u, 1).\end{aligned}$$

The calculation of lofted patch $\mathbf{L}$ starts with the two boundary curves $\mathbf{L}(u,0)$ and $\mathbf{L}(u,1)$. Each is calculated using Hermite interpolation (Chapter 4) since its extreme tangents, as well as its endpoints, are easy to figure out from the diagram. The boundary curves are

$$\begin{aligned}\mathbf{L}(u,0) &= (u^3,u^2,u,1)\mathbf{H}\big((0,0,0),(0,1/2,1),(0,0,1),(0,1,0)\big)^T,\\ \mathbf{L}(u,1) &= (u^3,u^2,u,1)\mathbf{H}\big((1,0,0),(1,1/2,1),(0,0,1),(0,1,0)\big)^T,\end{aligned}$$

where $\mathbf{H}$ is the Hermite basis matrix, Equation (4.7). Surface patch L is thus

$$\mathbf{L}(u,w) = \mathbf{L}(u,0)(1-w) + \mathbf{L}(u,1)w = (w, u^2/2, u+u^2-u^3).$$

Lofted patch $\mathbf{F}$ is calculated similarly. Its boundary curves are

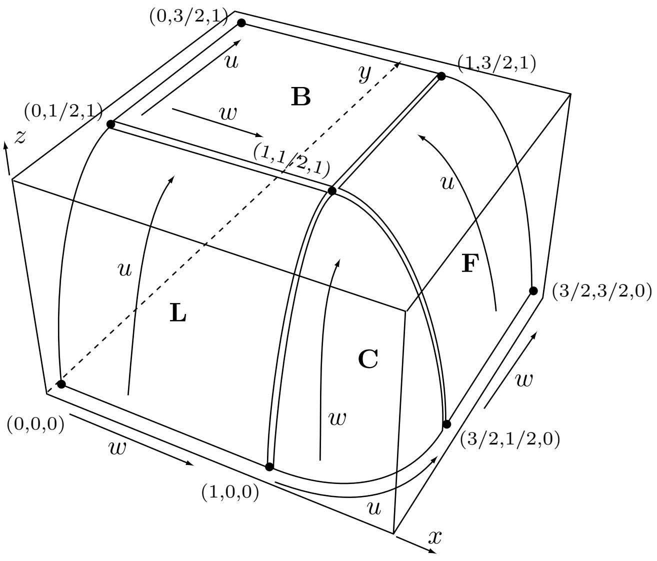
$$\begin{aligned}\mathbf{F}(u,0) &= (u^3,u^2,u,1)\mathbf{H}\big((3/2,1/2,0),(1,1/2,1),(0,0,1),(-1,0,0)\big)^T,\\ \mathbf{F}(u,1) &= (u^3,u^2,u,1)\mathbf{H}\big((3/2,3/2,0),(1,3/2,1),(0,0,1),(-1,0,0)\big)^T,\end{aligned}$$

and the patch itself is

$$\mathbf{F}(u,w) = \mathbf{F}(u,0)(1-w) + \mathbf{F}(u,1)w = ((3-u^2)/2, 1/2+w, u+u^2-u^3).$$

The triangular Coons surface $\mathbf{C}$ has corner points $\mathbf{C}_{00} = (1,0,0)$, $\mathbf{C}_{10} = (3/2,1/2,0)$, and $\mathbf{C}_{01} = \mathbf{C}_{11} = (1,1/2,1)$. Its bottom boundary curve is

$$\mathbf{C}(u,0) = (u^3,u^2,u,1)\mathbf{H}\big((1,0,0),(3/2,1/2,0),(1,0,0),(0,1,0)\big)^T,$$



```
b[u_,w_]:={0,1/2,1}(1-u)(1-w)+{1,1/2,1}(1-u)w
 +{0,3/2,1}(1-w)u+{1,3/2,1}u w;
H={{2,-2,1,1},{-3,3,-2,-1},{0,0,1,0},{1,0,0,0}};
lu0={u^3,u^2,u,1}.H.{{0,0,0},{0,1/2,1},{0,0,1},{0,1,0}};
lu1={u^3,u^2,u,1}.H.{{1,0,0},{1,1/2,1},{0,0,1},{0,1,0}};
l[u_,w_]:=lu0(1-w)+lu1 w;
fu0={u^3,u^2,u,1}.H.{{3/2,1/2,0},{1,1/2,1},{0,0,1},{-1,0,0}};
fu1={u^3,u^2,u,1}.H.{{3/2,3/2,0},{1,3/2,1},{0,0,1},{-1,0,0}};
f[u_,w_]:=fu0(1-w)+fu1 w;
cu0={u^3,u^2,u,1}.H.{{1,0,0},{3/2,1/2,0},{1,0,0},{0,1,0}};
cu1={1,1/2,1};
c0w={w^3,w^2,w,1}.H.{{1,0,0},{1,1/2,1},{0,0,1},{0,1,0}};
c1w={w^3,w^2,w,1}.H.{{3/2,1/2,0},{1,1/2,1},{0,0,1},{-1,0,0}};
c[u_,w_]:=(1-u)c0w+u c1w+(1-w)cu0+w cu1 \
 -(1-u)(1-w){1,0,0}-u(1-w){3/2,1/2,0}-w(1-u)cu1- u w cu1;
g1=ParametricPlot3D[b[u,w], {u,0,1},{w,0,1}]
g2=ParametricPlot3D[l[u,w], {u,0,1},{w,0,1}]
g3=ParametricPlot3D[f[u,w], {u,0,1},{w,0,1}]
g4=ParametricPlot3D[c[u,w], {u,0,1},{w,0,1}]
Show[g1,g2,g3,g4]
```

Figure 3.15: Bilinear, Lofted, and Coons Surface Patches.

and its top boundary curve $\mathbf{C}(u,1)$ is the multiple point $\mathbf{C}_{01}=\mathbf{C}_{11}$. The two boundary curves in the w direction are

$$\mathbf{C}(0,w)=(w^3,w^2,w,1)\mathbf{H}\big((1,0,0),(3/1,1/2,1),(0,0,1),(0,1,0)\big)^T,$$
$$\mathbf{C}(1,w)=(w^3,w^2,w,1)\mathbf{H}\big((3/1,1/2,0),(1,1/2,1),(0,0,1),(-1,0,0)\big)^T,$$

and the surface patch itself equals

$$\begin{aligned}\mathbf{C}(u,w)&=(1-u)\mathbf{C}(0,w)+u\mathbf{C}(1,w)+(1-w)\mathbf{C}(u,0)+w\mathbf{C}(u,1)\\&\quad-(1-u)(1-w)1,0,0-u(1-w)3/2,1/2,0-w(1-u)\mathbf{C}_{11}-uw\mathbf{C}_{11}\\&=((2+u^2(-1+w)-u(-2+w+w^2))/2,\\&\quad(-u^2(-1+w)-u(-1+w)w+w^2)/2,w+w^2-w^3).\end{aligned}$$

3.8 Gordon Surfaces

The Gordon surface is a generalization of Coons surfaces. A linear Coons surface is fully defined by means of four boundary curves, so its shape cannot be too complex. A Gordon surface (Figure 3.16) is defined by means of two families of curves, one in each of the u and w directions. It can have very complex shapes and is a good candidate for use in applications where realism is important.

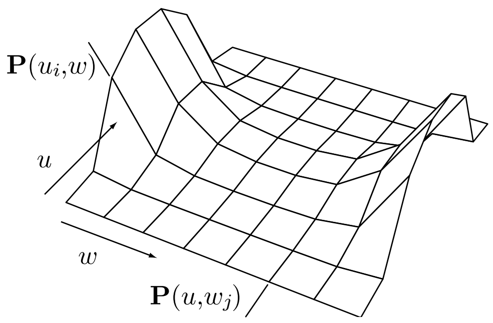


Figure 3.16: A Gordon Surface.

We denote the curves by $\mathbf{P}(u_i,w)$, where $i=0,\ldots,m$, and $\mathbf{P}(u,w_j)$, $j=0,\ldots,n$. The main idea is to find an expression for a surface $\mathbf{P}_a(u,w)$ that interpolates the first family of curves, add it to a similar expression for a surface $\mathbf{P}_b(u,w)$ that interpolates the second family of curves, and subtract a surface $\mathbf{P}_{ab}(u,w)$ that represents multiple contributions from $\mathbf{P}_a$ and $\mathbf{P}_b$.

The first surface, $\mathbf{P}_a(u,w)$, should interpolate the family of $m+1$ curves $\mathbf{P}(u_i,w)$. When moving on this surface in the u direction (fixed w), we want to intersect all $m+1$ curves. For a given, fixed w, we therefore need to find a curve that will pass through the $m+1$ points $\mathbf{P}(u_i, w)$. A natural (albeit not the only) candidate for such a curve is our old acquaintance the Lagrange polynomial (Section 3.2). We write it as $\mathbf{P}_a(u,w) = \sum_{i=o}^{m} \mathbf{P}(u_i, w) L_i^m(u)$, and it is valid for any value of w. Similarly, we can write the second surface as the Lagrange polynomial $\mathbf{P}_b(u,w) = \sum_{j=o}^{n} \mathbf{P}(u, w_j) L_j^n(w)$.

The surface representing multiple contributions is similar to the bilinear part of Equation (3.28). It is

$$\mathbf{P}_{ab}(u,w) = \sum_{i=o}^{m} \sum_{j=o}^{n} \mathbf{P}(u_i, w_j) L_i^m(u) L_j^n(w),$$

and the final expression of the Gordon surface is $\mathbf{P}(u,w) = \mathbf{P}_a(u,w) + \mathbf{P}_b(u,w) - \mathbf{P}_{ab}(u,w)$. Note that the $(m+1)\times(n+1)$ points $\mathbf{P}(u_i, w_j)$ should be located on *both* curves. For such a surface to make sense, the curves have to intersect.

A friend comes to you and asks if a particular polynomial $p(x)$ of degree 25 in $F_2[x]$ is irreducible. The friend explains that she has tried dividing $p(x)$ by every polynomial in $F_2[x]$ of degree from 1 to 18 and has found that $p(x)$ is not divisible by any of them. She is getting tired of doing all these divisions and wonders if there's an easier way to check whether or not $p(x)$ is irreducible. You surprise your friend with the statement that she need not do any more work: $p(x)$ is indeed irreducible!

—John Palmieri, *Introduction to Modern Algebra for Teachers*