Interactive Computer Graphics: Lecture 13

Introduction to Surface Construction

Teapot Subdivision: Russ Fish

Non Parametric Surface

- Surfaces can be constructed from Cartesian equations directly, and this is acceptable for specific applications, usually involving interpolation.

- As before, using a simple polynomial surface is a quick and easy approach.

Non Parametric Polynomial Surface

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & j \\ d & g & j & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = 0$$

- which multiplies out to:

$$ax^2 + ey^2 + hz^2 + 2bxy + 2cxz + 2fyz + 2dx + 2gy + 2jz + 1 = 0$$

- Because of the symmetry there are 9 scalar unknowns in the equation
- So we need to specify nine points through which the surface will pass

As Before

- This formulation suffers the same problems as the non-parametric spline curve. It is a fixed surface for a given set of nine points.

- We need more flexibility for the design of surfaces.

Simple Parametric surfaces

- We can extend the formulation to simple parametric surfaces using the vector equation:

$$\mathbf{P}(\mu,\nu) = (\mu,\nu,1)\begin{pmatrix}\mathbf{a} & \mathbf{b} & \mathbf{c}\\ \mathbf{b} & \mathbf{d} & \mathbf{e}\\ \mathbf{c} & \mathbf{e} & \mathbf{f}\end{pmatrix}\begin{pmatrix}\mu\\ \nu\\ 1\end{pmatrix}$$

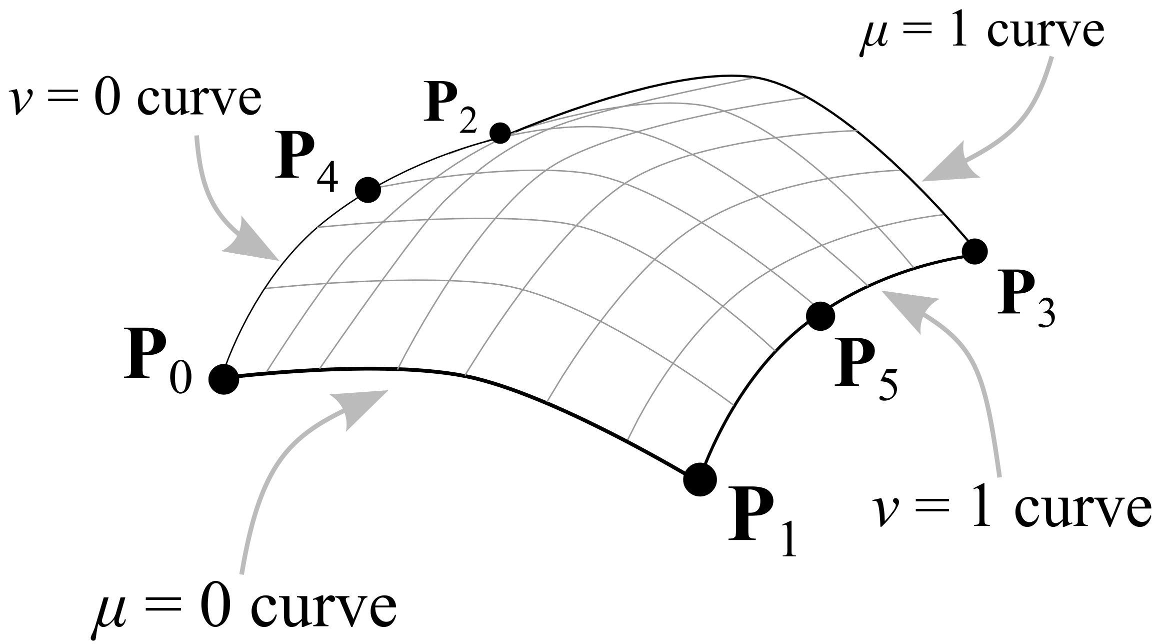
$$\mathbf{P}(\mu,\nu) = \mathbf{a}\mu^2 + \mathbf{d}\nu^2 + 2\mathbf{b}\mu\nu + 2\mathbf{c}\mu + 2\mathbf{e}\nu + \mathbf{f}$$

- There are six unknown parameter vectors $\{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{e},\mathbf{f}\}$

Associating points and parameters

- We can solve for the six vector unknowns by substituting in six points at known values of μ and v.
- We might have an arrangement such as:

	μ	ν
$\mathbf{P}_0$	0	0
$\mathbf{P}_1$	0	1
$\mathbf{P}_2$	1	0
$\mathbf{P}_3$	1	1
$\mathbf{P}_4$	$1/2$	0
$\mathbf{P}_5$	$1/2$	1



Surface parameter equations

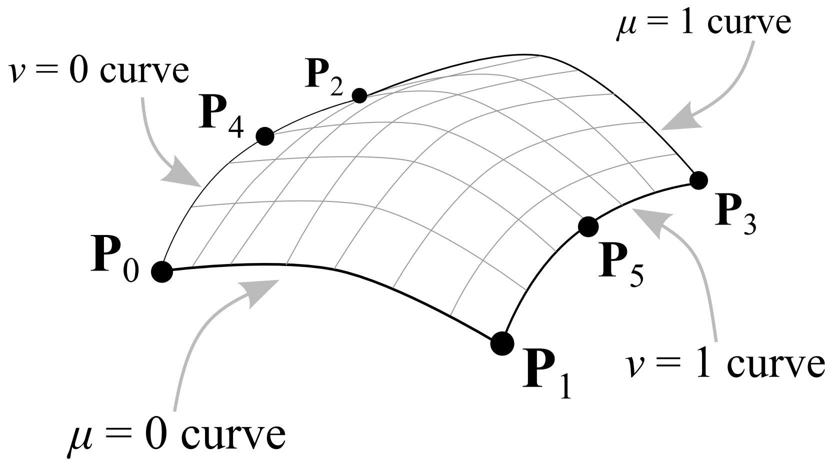
- Substituting these values of μ and v into the patch equation gives us these six equations

$$
\begin{aligned}
\mathbf{P}_0 &= \mathbf{f} \\
\mathbf{P}_1 &= \mathbf{d} + 2\mathbf{e} + \mathbf{f} \\
\mathbf{P}_2 &= \mathbf{a} + 2\mathbf{c} + \mathbf{f} \\
\mathbf{P}_3 &= \mathbf{a} + 2\mathbf{b} + 2\mathbf{c} + \mathbf{d} + 2\mathbf{e} + \mathbf{f} \\
\mathbf{P}_4 &= \mathbf{a}/4 + \mathbf{c} + \mathbf{f} \\
\mathbf{P}_5 &= \mathbf{a}/4 + \mathbf{b} + \mathbf{c} + \mathbf{d} + 2\mathbf{e} + \mathbf{f}
\end{aligned}
$$

- The $\mathbf{P}$'s are known and we can solve for the unknowns $\{\mathbf{a}, \ldots, \mathbf{f}\}$ using standard methods

Getting the edges from the surface equation

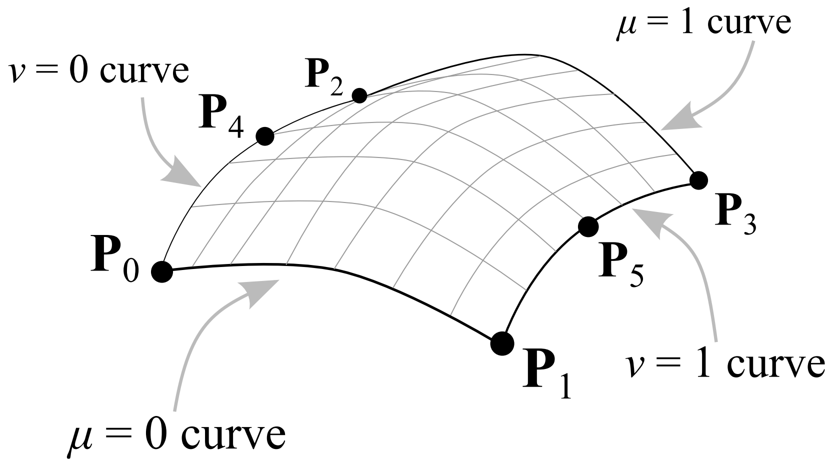
μ and ν are in the range [0, 1]. Thus the contours that bound the patch can be found by substituting 0 or 1 for one of μ or ν in the patch equation.



$$
\begin{aligned}
\mathbf{P}(0,\nu) &= \mathbf{d}\nu^2 + 2\mathbf{e}\nu + \mathbf{f} \\
\mathbf{P}(1,\nu) &= \mathbf{a} + 2(\mathbf{b}+\mathbf{e})\nu + 2\mathbf{c} + \mathbf{d}\nu^2 + \mathbf{f} \\
\mathbf{P}(\mu,0) &= \mathbf{a}\mu^2 + 2\mathbf{c}\mu + \mathbf{f} \\
\mathbf{P}(\mu,1) &= \mathbf{a}\mu^2 + 2(\mathbf{b}+\mathbf{c})\mu + \mathbf{d} + 2\mathbf{e} + \mathbf{f}
\end{aligned}
$$

The resulting surface

The boundaries are all second order curves and so will be nice and smooth



There is quite a lot of flexibility in this formulation, but it is still only suitable for simple surfaces.

We can use higher orders

E.g. using the tensor product

$$\mathbf{P}(\mu,\nu) = \begin{pmatrix}\mu^3 & \mu^2 & \mu & 1\end{pmatrix} \begin{pmatrix}\mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} \\ \mathbf{b} & \mathbf{e} & \mathbf{f} & \mathbf{g} \\ \mathbf{c} & \mathbf{f} & \mathbf{h} & \mathbf{j} \\ \mathbf{d} & \mathbf{g} & \mathbf{j} & \mathbf{k}\end{pmatrix} \begin{pmatrix}\nu^3 \\ \nu^2 \\ \nu \\ 1\end{pmatrix}$$

Using higher orders gives more variety in shape and better control

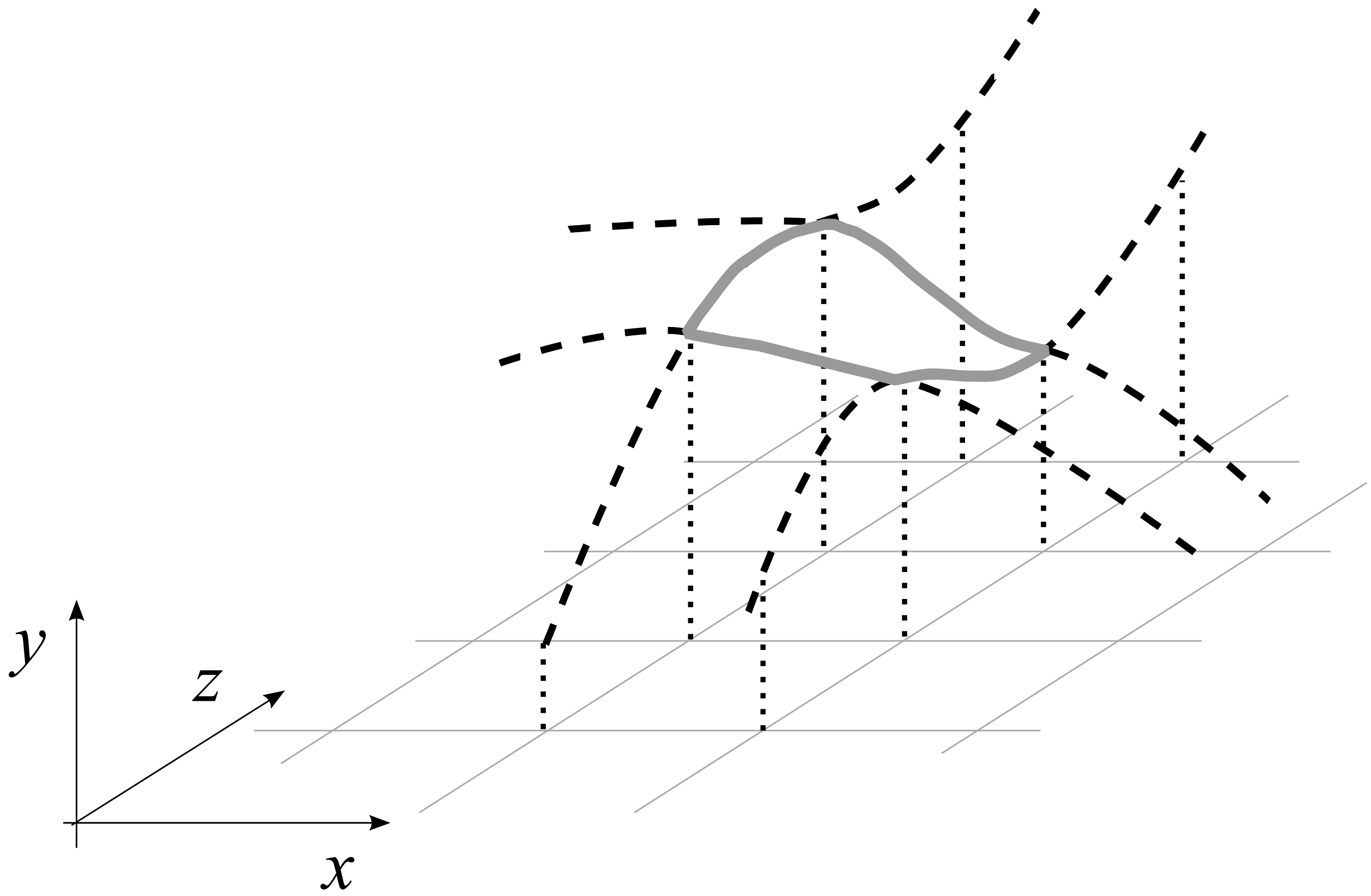
But the method is hard to apply and generalise, and so is not usually done

Cubic Spline Patches

- The patch method is generally effective in creating more complex surfaces.

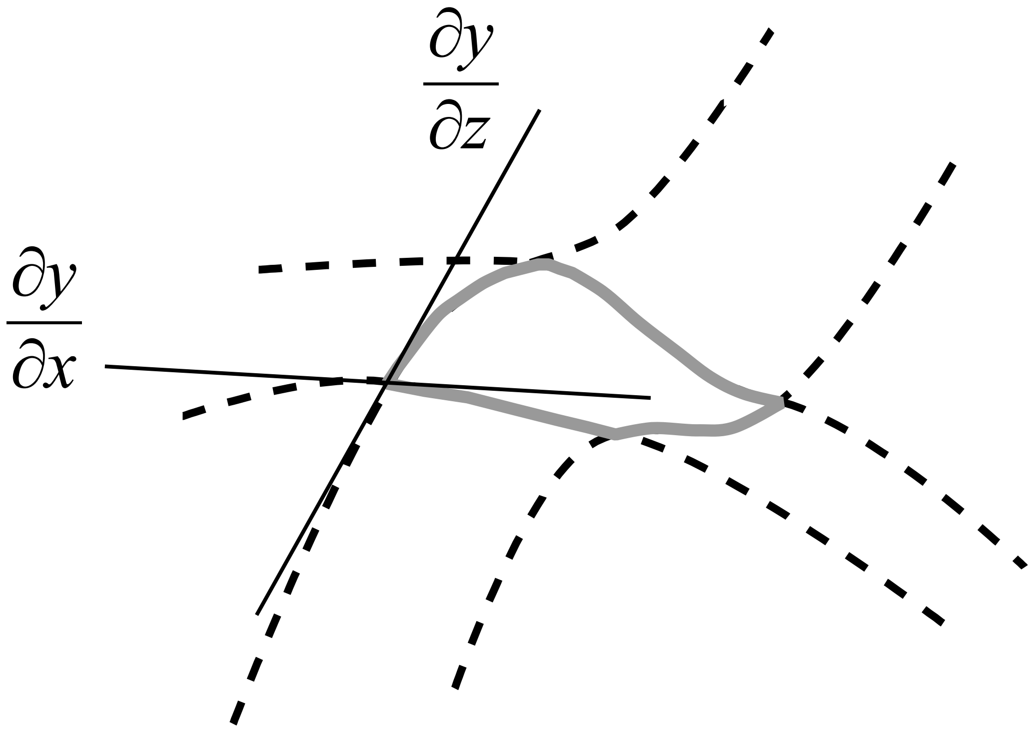
- The idea is, as in the case of the curves, to create a surface by joining a lot of simple surfaces continuously.

Cartesian surface patches - terrain map



Points and Gradients

- At each corner of the patch we need to interpolate the points and set the gradients to match the adjacent patch.
- There are two gradients



Parametric patches

- In practice we use the more general parametric patch formulation with two parameters μ and ν.
- The terrain map can be modelled with parametric patches.
- We need to match three values at each corner

$$\mathbf{P}(\mu,\nu) \qquad \frac{\partial \mathbf{P}(\mu,\nu)}{\partial \mu} \qquad \frac{\partial \mathbf{P}(\mu,\nu)}{\partial \nu}$$

Corners

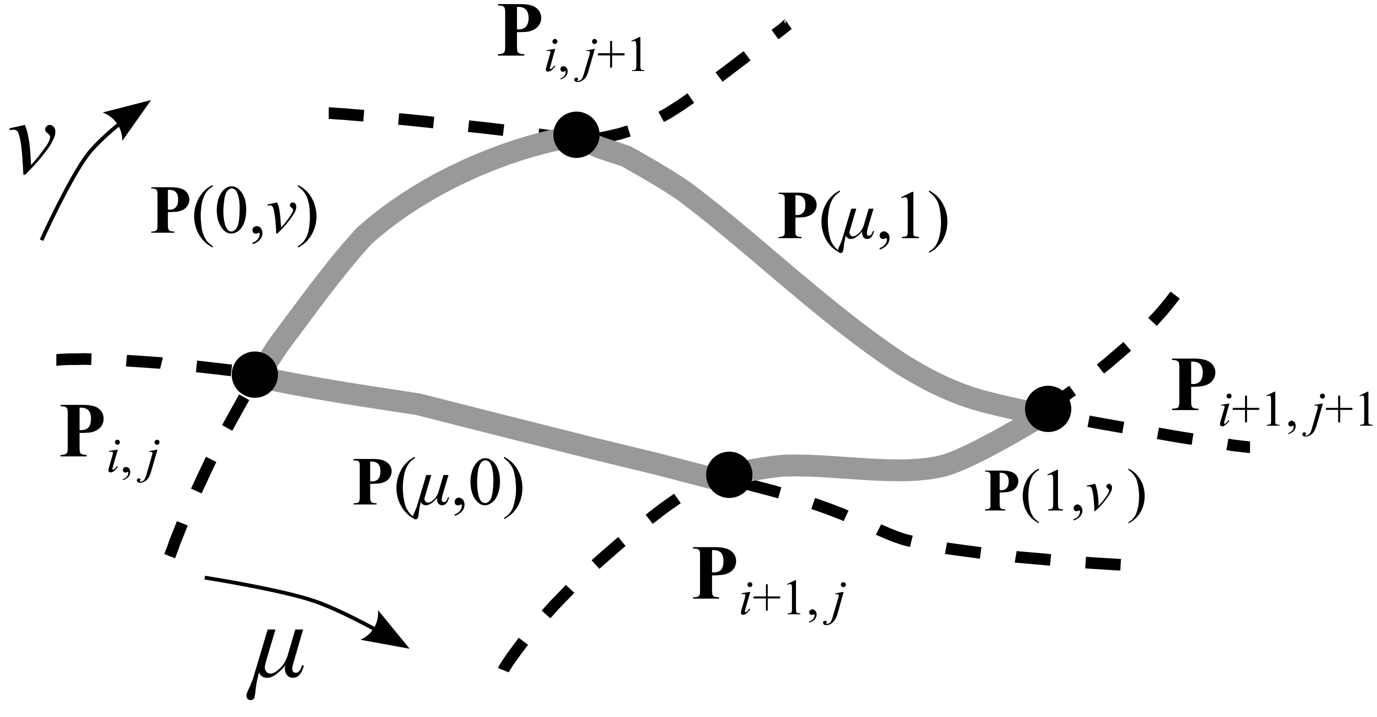
- As usual we adopt the convention that the corners are at parameter values (0, 0), (0, 1), (1, 0) and (1, 1)
- We need to ensure that the patch joins its neighbours exactly at the edges.
- Hence we ensure that the edge contours are the same on adjacent patches

Edges

- We do this by designing the edge curves in an identical manner to the cubic spline curve patch.

Edge curve	Points joined	
$\mathbf{P}(0,\nu)$	$\mathbf{P}_{i,j}$	$\mathbf{P}_{i,j+1}$
$\mathbf{P}(1,\nu)$	$\mathbf{P}_{i+1,j}$	$\mathbf{P}_{i+1,j+1}$
$\mathbf{P}(\mu,0)$	$\mathbf{P}_{i,j}$	$\mathbf{P}_{i+1,j}$
$\mathbf{P}(\mu,1)$	$\mathbf{P}_{i,j+1}$	$\mathbf{P}_{i+1,j+1}$

A parametric spline patch



As long as the gradients are the same for the four patches that meet at a point the surface will join seamlessly

The Coons patch

To define the internal points we linearly interpolate the edge curves:

$$\begin{aligned}\mathbf{P}(\mu,\nu) &= \mathbf{P}(\mu,0)(1-\nu)+\mathbf{P}(\mu,1)\nu + \\ &\quad \mathbf{P}(0,\nu)(1-\mu)+\mathbf{P}(1,\nu)\mu - \\ &\quad \mathbf{P}(0,1)(1-\mu)\nu-\mathbf{P}(1,0)\mu(1-\nu) - \\ &\quad \mathbf{P}(0,0)(1-\mu)(1-\nu)-\mathbf{P}(1,1)\mu\nu\end{aligned}$$

Substituting values of 0 or 1 for μ and/or ν we can easily verify that the equation fits the edge curves.

Rendering a patch: Polygonisation

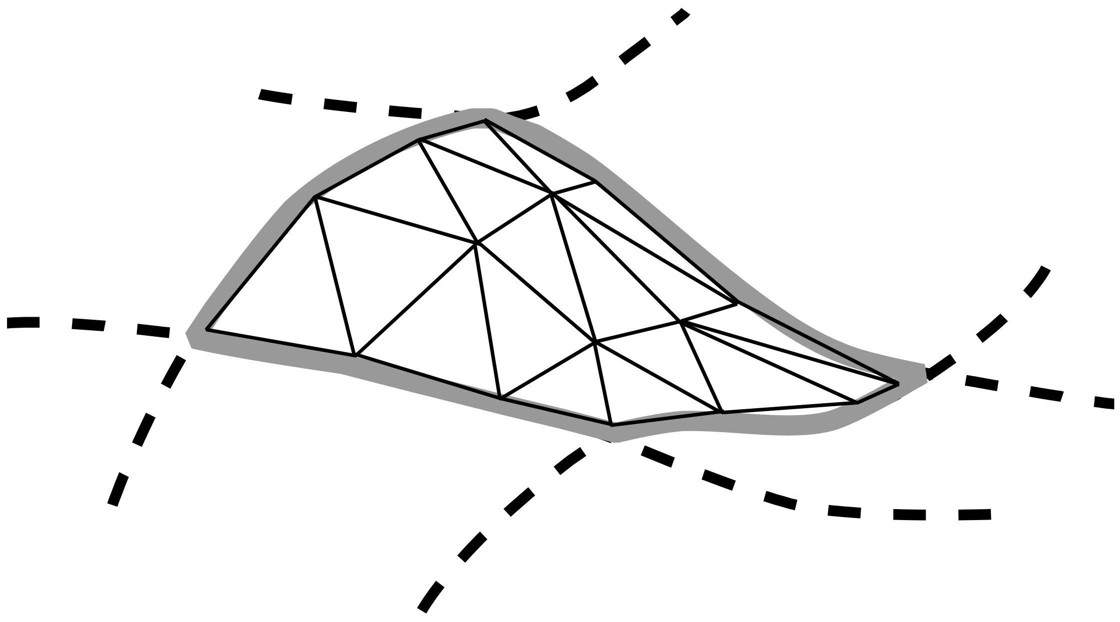
To render (draw) a spline patch we can simply transform it into polygons.

We select a grid of points, e.g.:

$$\begin{aligned} \mu &= \{0.0, 0.1, 0.2, \ldots 1.0\} \\ \nu &= \{0.0, 0.1, 0.2, \ldots 1.0\} \end{aligned}$$

and triangulate to that grid.

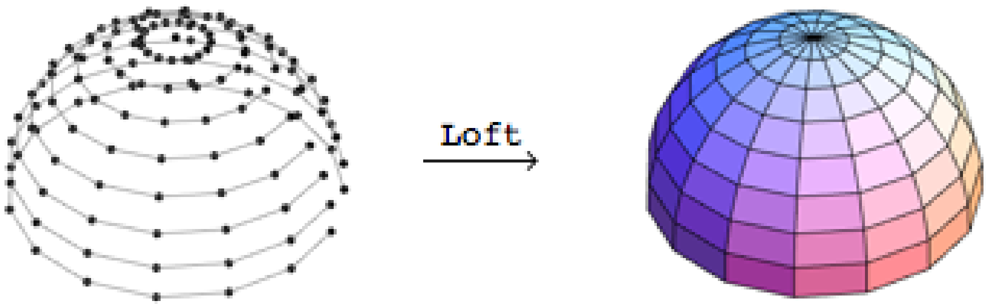
Rendering a patch: Polygonisation



- For speed we can use large polygons with Gouraud or Phong shading.
- For accuracy we use small polygons, chosen to match the pixel size.

Rendering a patch: Lofting

- Surfaces can also be drawn by a technique called lofting (now really obsolete).
- This means drawing contours of constant μ and/or of constant ν
- Algorithms for eliminating the hidden parts have been devised.



Rendering a patch: Ray tracing

- The patch equation is fourth order

$$
\begin{aligned}
\mathbf{P}(\mu,\nu) \;\; = \;\; & \mathbf{P}(\mu,0)(1-\nu) + \mathbf{P}(\mu,1)\nu + \\
& \mathbf{P}(0,\nu)(1-\mu) + \mathbf{P}(1,\nu)\mu - \\
& \mathbf{P}(0,1)(1-\mu)\nu - \mathbf{P}(1,0)\mu(1-\nu) - \\
& \mathbf{P}(0,0)(1-\mu)(1-\nu) - \mathbf{P}(1,1)\mu\nu
\end{aligned}
$$

- Hence no closed form solution exists for a ray patch intersection
- Can use numeric algorithms but computation can be costly

Rendering a patch: Ray tracing

- Numerical Ray-Patch algorithm

1. Polygonise the patch at a low resolution (say 4 x 4)
2. Calculate the ray intersection with the 32 triangles and find the nearest intersection.
3. Polygonise the immediate area of the intersection and calculate a better estimate of the intersection
4. Continue until the best estimate is found

Rendering a patch: Ray tracing

- Numerical Ray-Patch algorithm
  - May be multiple intersections between the ray and the surface
  - Algorithm will find an intersection, but not necessarily the nearest.
  - If the object is relatively smooth it should work well in most cases.
  - Note that it will be necessary to do a ray intersection with each patch of the object to find the nearest intersection.

Example of Using a Coons Patch

- Part of a terrain map defined on a regular x-y grid is as follows:

$y,\ \nu \longrightarrow$

$x,\ \mu\ \downarrow$	2	3	4	5	6	7
7	·	·	·	·	·	·
8	·	·	10	9	·	·
9	·	14	12	11	10	·
10	·	15	13	14	10	·
11	·	·	10	11	·	·

- Find the Coons patch on the centre four points

Corners

- The corners at μ, $\nu = 0, 1$ are defined directly in the question:

$$\mathbf{P}(0,0) = (9,4,12) \qquad \mathbf{P}(1,0) = (10,4,13)$$
$$\mathbf{P}(0,1) = (9,5,11) \qquad \mathbf{P}(1,1) = (10,5,14)$$

y, ν $\longrightarrow$

x, μ $\downarrow$

	2	3	4	5	6	7
7	·	·	·	·	·	·
8	·	·	10	9	·	·
9	·	14	12	11	10	·
10	·	15	13	14	10	·
11	·	·	10	11	·	·

Gradients in the x / μ direction

Example

$$\left.\frac{\partial \mathbf{P}}{\partial \mu}\right|_{(0,0)} = \frac{(10,4,13)^T - (8,4,10)^T}{2} = \begin{pmatrix} 1 \\ 0 \\ 1.5 \end{pmatrix}$$

$y,\ \nu \longrightarrow$

$x,\ \mu\ \downarrow$	2	3	4	5	6	7
7	·	·	·	·	·	·
8	·	·	10	9	·	·
9	·	14	12	11	10	·
10	·	15	13	14	10	·
11	·	·	10	11	·	·

Gradients in the x / μ direction

$$\left.\frac{\partial \mathbf{P}}{\partial \mu}\right|_{(0,0)} = \frac{(10,4,13)^T - (8,4,10)^T}{2} = (1,0,1.5)^T$$

$$\left.\frac{\partial \mathbf{P}}{\partial \mu}\right|_{(1,0)} = \frac{(11,4,10)^T - (9,4,12)^T}{2} = (1,0,-1)^T$$

$$\left.\frac{\partial \mathbf{P}}{\partial \mu}\right|_{(0,1)} = \frac{(10,5,14)^T - (8,5,9)^T}{2} = (1,0,2.5)^T$$

$$\left.\frac{\partial \mathbf{P}}{\partial \mu}\right|_{(1,1)} = \frac{(11,5,11)^T - (9,5,11)^T}{2} = (1,0,0)^T$$

Gradients in the y / v direction

$$\left.\frac{\partial \mathbf{P}}{\partial \nu}\right|_{(0,0)} = \frac{(9,5,11)^T - (9,3,14)^T}{2} = (0,1,-1.5)^T$$

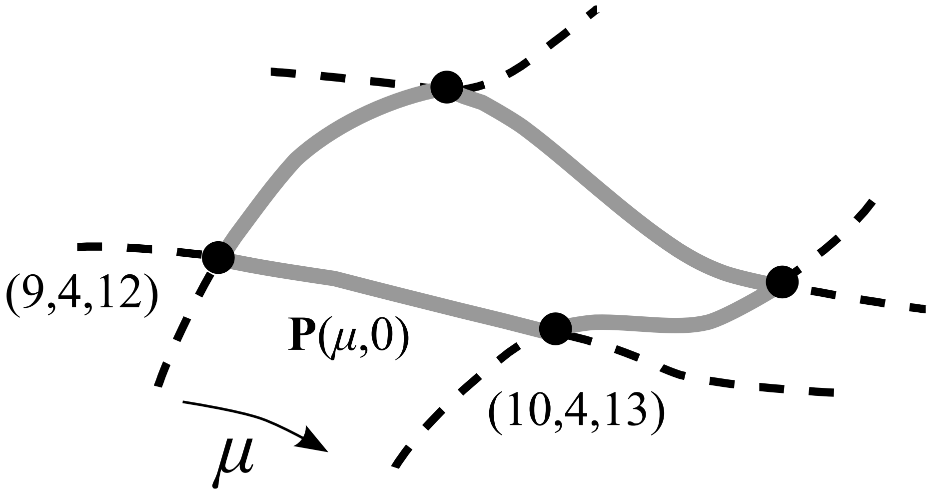
$$\left.\frac{\partial \mathbf{P}}{\partial \nu}\right|_{(1,0)} = \frac{(10,5,14)^T - (10,3,15)^T}{2} = (0,1,-0.5)^T$$

$$\left.\frac{\partial \mathbf{P}}{\partial \nu}\right|_{(0,1)} = \frac{(9,6,10)^T - (9,4,12)^T}{2} = (0,1,-1)^T$$

$$\left.\frac{\partial \mathbf{P}}{\partial \nu}\right|_{(1,1)} = \frac{(10,6,10)^T - (10,4,13)^T}{2} = (0,1,-1.5)^T$$

Finding the boundary curves

E.g. Finding curve $\mathbf{P}(\mu, 0)$



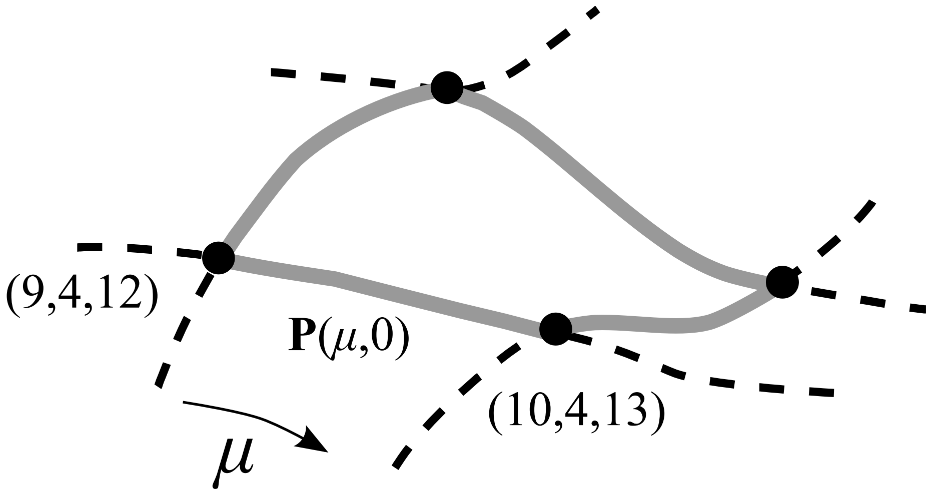
$$\mathbf{P}(\mu, 0) = \mathbf{a}_3\mu^3 + \mathbf{a}_2\mu^2 + \mathbf{a}_1\mu + \mathbf{a}_0$$

$$\begin{pmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3 & -2 & 3 & -1 \\ 2 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}'_0 \\ \mathbf{P}_1 \\ \mathbf{P}'_1 \end{pmatrix}$$

- see cubic spline patch equation (previous lecture)

Finding the boundary curves

E.g. Finding curve $\mathbf{P}(\mu, 0)$



$$\mathbf{P}(\mu, 0) = \mathbf{a}_3\mu^3 + \mathbf{a}_2\mu^2 + \mathbf{a}_1\mu + \mathbf{a}_0$$

$$\begin{pmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3 & -2 & 3 & -1 \\ 2 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 9 & 4 & 12 \\ 1 & 0 & 1.5 \\ 10 & 4 & 13 \\ 1 & 0 & -1 \end{pmatrix}$$

After substituting in $\mathbf{P}(0,0)$, $\left.\frac{\partial \mathbf{P}}{\partial \mu}\right|_{(0,0)}$, $\mathbf{P}(1,0)$, $\left.\frac{\partial \mathbf{P}}{\partial \mu}\right|_{(1,0)}$

Finding the boundary curve $\mathbf{P}(\mu, 0)$

- Calculating the constant vectors $\mathbf{a}_3$, $\mathbf{a}_2$, $\mathbf{a}_1$ and $\mathbf{a}_0$

$$
\begin{aligned}
\mathbf{a}_0 &= \mathbf{P}_0 = (9, 4, 12) \\
\mathbf{a}_1 &= \mathbf{P}_0' = (1, 0, 1.5) \\
\mathbf{a}_2 &= -3\mathbf{P}_0 - 2\mathbf{P}_0' - 3\mathbf{P}_1 - \mathbf{P}_1' \\
&= -3 \times (9, 4, 12) - 2 \times (1, 0, 1.5) + 3 \times (10, 4, 13) - (1, 0, 1) \\
&= (0, 0, 1) \\
\mathbf{a}_3 &= 2\mathbf{P}_0 + \mathbf{P}_0' - 2\mathbf{P}_1 + \mathbf{P}_1' \\
&= 2 \times (9, 4, 12) + (1, 0, 1.5) - 2 \times (10, 4, 13) + (1, 0, 1) \\
&= (0, 0, 0.5)
\end{aligned}
$$

Finding the boundary curves $\mathbf{P}(\mu, 1)$, $\mathbf{P}(0, v)$, $\mathbf{P}(1, v)$

- These curves are found identically to $\mathbf{P}(\mu, 0)$.

- We now have all the individual bits:

  $\mathbf{P}(\mu, 0)$: a cubic polynomial in μ
  $\mathbf{P}(\mu, 1)$: a cubic polynomial in μ
  $\mathbf{P}(0, v)$: a cubic polynomial in v
  $\mathbf{P}(1, v)$: a cubic polynomial in v
  $\mathbf{P}(0, 0)$, $\mathbf{P}(0, 1)$, $\mathbf{P}(1, 0)$ and $\mathbf{P}(1, 1)$: the corner points

- Given values of μ and v, we can calculate each of these eight points

So, for any given value for μ and ν …

. . . we can evaluate the coordinate on the Coons patch:

$$
\begin{aligned}
\mathbf{P}(\mu,\nu) \quad = \quad & \mathbf{P}(\mu,0)(1-\nu)+\mathbf{P}(\mu,1)\nu+ \\
& \mathbf{P}(0,\nu)(1-\mu)+\mathbf{P}(1,\nu)\mu- \\
& \mathbf{P}(0,1)(1-\mu)\nu-\mathbf{P}(1,0)\mu(1-\nu)- \\
& \mathbf{P}(0,0)(1-\mu)(1-\nu)-\mathbf{P}(1,1)\mu\nu
\end{aligned}
$$