

MA 323 Geometric Modelling

Course Notes: Day 28

Data Fitting to Surfaces

David L. Finn

Today, we want to exam interpolation and data fitting problems for surface patches. Our general method is the same, solve a system of linear equations. In fact, the structure of a tensor product surface allows us to proceed much the same for surfaces as for curves. The difficulty for curves of choosing the correct parameter values is again the problem for surfaces. But now that problem is exasperated by the fact that we need two parameters.

28.1 Data Fitting to a Surface

The generic problem that we want to consider today is the problem of fitting a surface to data. This problem includes the interpolation problem and may be stated as

Given a set of data points. Find a surface that approximates the data points.

For our purposes a surface today will be a bicubic Bezier patch. Thus to solve this data fitting problem, we will need at least sixteen data points. We can use more points if we use least squares method to fit the data, and we can use fewer points if we add extra conditions to fill out the conditions so we can determine the 16 control points of a bicubic Bezier patch.

We remark that the method that we will consider today extends to higher order Bezier patches and to spline patches (to be introduced over the next few days. Our interest today is in the mechanics of the interpolation problem and some aspects of how to choose points to interpolate and the parameter values to use. We note that in general the problem is very similar to the data fitting problem for curves. The major difference is that we have the added complexity of fitting data to a surface.

28.2 Interpolation Problem for a Bezier Patch

The interpolation problem for a bicubic Bezier patch is

Given sixteen data points. Find a bicubic Bezier patch that passes through the data points.

We can solve this by choosing appropriate parameter values, and solving a system of 16 equations and 16 unknowns. [The equations are vector valued equations, and the unknowns are vectors. So, we really have 48 linear equations and 48 unknowns. The nice thing is that the system of the 48 equations decompose naturally into solving the same system 16 general equations in 16 unknowns three different times.] The main problem is how to choose the parameter values (u, v) for each data point. We need to choose 16 pairs of (u, v) values, one for each point.

Suppose for the moment that the parameter values have been chosen for each data point x_k , we have a parameter values (u_k, v_k) . The problem is then to solve the system of equations

$$p_k = \sum_{i=0}^3 \sum_{j=0}^3 B_i^3(u_k) B_j^3(v_k) p_{ij}.$$

We can simplify the problem by using the tensor product structure of a Bezier patch. To take advantage of the tensor product structure of the patch, we need to arrange the data into a 4×4 grid, then we can use curve methods to choose the parameter values. What do we mean by this.

Suppose we can arrange the data into a 4×4 grid with each horizontal row can be thought of as defining a curve in the u variable (each horizontal row has the same value of the parameter v), and thus the u parameter values can be chosen via curve methods either uniform spacing or chord length spacing. Also, each vertical row can be thought of as defining a curve in the v variable (each vertical row has the same value of the parameter u), and thus the v parameter values can be chosen via curve methods either uniform spacing or chord length spacing. [See additional handouts for Maple examples.] Choosing the parameter values in this manner makes the problem computationally easier. Instead of solving system of 16 equations in 16 unknowns, we need to solve several systems of 4 equations in 4 unknowns.

With this method of choosing the parameter values, we have the problem in the following form. We are given a matrix P of data points,

$$Q = \begin{bmatrix} q_{00} & q_{01} & q_{02} & q_{03} \\ q_{10} & q_{11} & q_{12} & q_{13} \\ q_{20} & q_{21} & q_{22} & q_{23} \\ q_{30} & q_{31} & q_{32} & q_{33} \end{bmatrix},$$

and our task is to find a matrix of control points B ,

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{bmatrix},$$

such that

$$Q = U^T P V,$$

where U is the matrix of Bernstein polynomials evaluated in u and V is the matrix of Bernstein polynomials in v . The matrix U is

$$U = \begin{bmatrix} B_0^3(u_0) & B_0^3(u_1) & B_0^3(u_2) & B_0^3(u_3) \\ B_1^3(u_0) & B_1^3(u_1) & B_1^3(u_2) & B_1^3(u_3) \\ B_2^3(u_0) & B_2^3(u_1) & B_2^3(u_2) & B_2^3(u_3) \\ B_3^3(u_0) & B_3^3(u_1) & B_3^3(u_2) & B_3^3(u_3) \end{bmatrix}$$

and the matrix V is

$$V = \begin{bmatrix} B_0^3(v_0) & B_0^3(v_1) & B_0^3(v_2) & B_0^3(v_3) \\ B_1^3(v_0) & B_1^3(v_1) & B_1^3(v_2) & B_1^3(v_3) \\ B_2^3(v_0) & B_2^3(v_1) & B_2^3(v_2) & B_2^3(v_3) \\ B_3^3(v_0) & B_3^3(v_1) & B_3^3(v_2) & B_3^3(v_3) \end{bmatrix}.$$

This set-up is a direct result of the tensor product formulation of the patch. Recall we can write a Bezier patch as

$$p(u, v) = \begin{bmatrix} B_0^3(u) & B_1^3(u) & B_2^3(u) & B_3^3(u) \end{bmatrix} \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} B_0^3(v) \\ B_1^3(v) \\ B_2^3(v) \\ B_3^3(v) \end{bmatrix},$$

the product of the basis functions in u times the matrix of control points times the basis functions in v . From this formulation, the solution of the interpolation problem is given by

$$P = (U^T)^{-1} Q V^{-1}.$$

This is a considerable savings in computing power, we only need to solve two 4×4 systems, that is invert two 4×4 matrices, or apply Gaussian elimination to solve two 4×4 systems if one wants better numerical stability.

28.3 Data Fitting with the Least Squares Fit

The standard data fitting problem for a bicubic Bezier patch is

Suppose we are given more than sixteen data points. Find a bicubic Bezier patch that approximates the data points.

We can solve this in a similar manner to solving the interpolation problem provided we have the data arranged in a grid, so we can easily choose the parameter values. In this case, once we have determined parameter values, we will have more than 16 equations in 16 unknowns, and we can apply least squares methods to find the control points for the patch. If we set up the data, to exploit the tensor product structure we again have a much simpler problem.

We need to set-up the data into a $(m+1) \times (n+1)$ grid of data points (an $m \times n$ matrix X) where $m \geq 3$ and $n \geq 3$. Then find the parameter values $u_0 < u_2 < \dots < u_m$ and $v_0 < v_1 < \dots < v_n$. We form the matrices

$$U = \begin{bmatrix} B_0^3(u_0) & B_0^3(u_1) & \dots & B_0^3(u_m) \\ B_1^3(u_0) & B_1^3(u_1) & \dots & B_1^3(u_m) \\ B_2^3(u_0) & B_2^3(u_1) & \dots & B_2^3(u_m) \\ B_3^3(u_0) & B_3^3(u_1) & \dots & B_3^3(u_m) \end{bmatrix}$$

and

$$V = \begin{bmatrix} B_0^3(v_0) & B_0^3(v_1) & \dots & B_0^3(v_n) \\ B_1^3(v_0) & B_1^3(v_1) & \dots & B_1^3(v_n) \\ B_2^3(v_0) & B_2^3(v_1) & \dots & B_2^3(v_n) \\ B_3^3(v_0) & B_3^3(v_1) & \dots & B_3^3(v_n) \end{bmatrix},$$

so U is an $(m+1) \times 4$ matrix and V is an $(n+1) \times 4$ matrix. We then want the best solution to

$$X = U^T P V$$

where B is the matrix of control points. The solution is given by

$$P = (U U^T)^{-1} U X V^T (V V^T)^{-1}.$$

The matrices $U U^T$ and $V V^T$ are 4×4 matrices.

The generic problem does not provide the data in such a grid fashion. We can still solve the problem using least squares method, however the choice of parameter values is then a

very difficult problem, and the solution of data fitting problem can be more strange. In fact, this is a current research problem, generating a surface from a point cloud. Luckily, in a lot of applications the data will be generated in a grid like fashion, i.e. laser scanning, medical scanning, et cetera. Most sensing equipment generate data in a grid like fashion. If we do not have a grid of data points or can arrange the data in a grid, it may be more useful to consider another method to model the surface. Tensor product surfaces (Bezier patches) are not always the most appropriate method. They are appropriate when the data or the control structure to use is in a grid structure. For the generic problem, other surface modelling techniques are primarily used first to create a first approximation of the surfaces, then if you desire one can use a Bezier or B-Spline techniques to approximate the first model. One can sample the data from one model of the surface to create a grid of data from which a Bezier or B-Spline approximation is more easily obtained and an appropriate choice

Bezier approximations or more generally spline approximations are designed primarily to use a grid structure. This can be avoided by using triangular patches. A triangular structure can be more naturally adapted to a point cloud, as it is easy to determine *small triangles* in a point cloud. Three points that are close together should determine a triangle in a triangular grid, but it is hard to determine if four points that are close together should form a rectangle in a rectangular grid. This is just one problem with only using rectangular patches to construct a model of a surface.

28.4 Choosing the Data Points

In some applications, notably computer graphics and computer aided geometric design, one can choose the points used to data points used. How should one do that? Where should one concentrate points? For curves, we decided that one should use more data points to approximate points of high curvature. With surfaces, the choice of data points is more subtle. For there are more notions of curvature to a surface. The generic answer is still the same. We use more data points where there is higher curvature, the rate of geometric change is large. Problem is how much data to use. This again is a current research problem. There is not one specific answer.

EXERCISES

1. Given the data points

$$X = \begin{bmatrix} [0, 0, 0] & [1, 0, 0] & [1, 1, 1] & [2, 1, 0] \\ [0, 1, 2] & [1, 2, 1] & [2, 0, 1] & [3, 1, 0] \\ [-1, 3, 1] & [2, 3, 0] & [3, 3, 0] & [3, 3, 2] \\ [1, 5, 1] & [2, 4, 1] & [3, 4, 1] & [4, 3, -1] \end{bmatrix},$$

find an interpolating bicubic Bezier patch, please use the code in the Maple Worksheet for Interpolating and Data Fitting.

- (a) using uniform spacing in both the u and v directions
- (b) using $u_0 = 0$, $u_1 = 0.3$, $u_2 = 0.6$, $u_3 = 1$, and $v_0 = 0$, $v_1 = 0.5$, $v_2 = 0.7$, $v_3 = 1$.
- (c) using parameter values of your own choice.
- (d) what is the effect of using different parameter variables?

2. Given the set of 36 data points arranged in the grid

$$X = \begin{bmatrix} [0, 0, 0] & [1, 0, 0] & [1, 1, 1] & [2, 1, 0] & [3, 1, 0] & [4, 1, 1] \\ [0, 1, 2] & [1, 2, 1] & [2, 1, 1] & [3, 1, 1] & [4, 2, 1] & [5, 2, 1] \\ [-1, 2, 1] & [1, 3, 0] & [2, 3, 0] & [3, 3, 2] & [4, 3, 3] & [5, 3, 2] \\ [0, 3, 0] & [1, 4, 0] & [3, 3, 1] & [4, 3, 1] & [4, 4, 2] & [6, 3, 1] \\ [1, 4, 1] & [2, 4, 1] & [3, 4, 1] & [4, 3, 0] & [5, 4, 1] & [6, 4, 0] \end{bmatrix},$$

find a bicubic Bezier patch that approximates the data. You will have to use least squares to fit the data, please use the code in the Maple Worksheet for Interpolating and Data Fitting.

- (a) using uniform spacing in both the u and v directions ($u_0 = 0, u_1 = 0.2, u_2 = 0.4, u_3 = 0.6, u_4 = 0.8, u_5 = 1.0$ and $v_0 = 0, v_1 = 0.2, v_2 = 0.4, v_3 = 0.6, v_4 = 0.8, v_5 = 1.0$).
 - (b) using parameter values of your choice.
 - (c) what is the effect of using different parameter values?
3. Use the Interpolation and Data Fitting to Bezier Patches to examine the effect of interpolation and data fitting to surfaces using uniform spacing in both the u and v variables.