Least–Squares Reduction of B–Spline Curves

David Eberly
Magic Software, Inc.
http://www.magic-software.com

Created: June 29, 2003
Modified: June 30, 2003 (the equation for $\mathbf{Q}_i$ in terms of $\mathbf{P}_j$ was corrected)

Consider an open, uniform B–spline curve whose control points are $\mathbf{P}_j$ for $0 \le j \le n$ and whose degree is d. The B–spline basis functions are $N_{j,d}(t)$ where $t \in [0,1]$ and depend on a set of knots s_j for $0 \le j \le n+d+1$. The curve is defined by

$$\mathbf{X}(t) = \sum_{j=0}^{n} N_{j,d}(t)\mathbf{P}_j$$

The goal is to construct another open, uniform B–spline curve of degree d that approximates $\mathbf{X}(t)$ but has fewer control points. Let the approximating curve have control points $\mathbf{Q}_j$ for $0 \le j \le m$ where $m < n$. The B–spline basis functions for this curve are $M_{j,d}(t)$ and depend on a set of knots t_j for $0 \le j \le m+d+1$. The curve is defined by

$$\mathbf{Y}(t) = \sum_{j=0}^{m} M_{j,d}(t)\mathbf{Q}_j$$

The approximation curve is chosen to minimize the integral of the squared distance between $\mathbf{X}(t)$ and $\mathbf{Y}(t)$, a least–squares fitting algorithm.

The control points of $\mathbf{Y}(t)$ are the unknown quantities to be determined by the algorithm. The least–squares error function is

$$E(\mathbf{Q}_0, \ldots, \mathbf{Q}_m) = \int_0^1 \left| \sum_{j=0}^{m} M_{j,d}(t)\mathbf{Q}_j - \sum_{j=0}^{n} N_{j,d}(t)\mathbf{P}_j \right|^2 dt$$

The error function is nonnegative and has a global minimum that occurs when its gradient vector is zero. We may establish these conditions by setting the derivatives with respect to the control points to zero (as compared to setting the derivatives with respect to the *components* of the control points). That is,

$$\mathbf{0} = \frac{\partial E}{\partial \mathbf{Q}_i} = \int_0^1 \left(\sum_{j=0}^{m} M_{j,d}(t)\mathbf{Q}_j - \sum_{j=0}^{n} N_{j,d}(t)\mathbf{P}_j \right) M_{i,d}(t)\, dt$$

This simplifies to

$$\begin{array}{rcl} \sum_{j=0}^{m} \left(\int_0^1 M_{i,d}(t) M_{j,d}(t) \right) \mathbf{Q}_j & = & \sum_{j=0}^{n} \left(\int_0^1 M_{i,d}(t) N_{j,d}(t) \right) \mathbf{P}_j \\ \sum_{j=0}^{m} a_{ij}\mathbf{Q}_j & = & \sum_{j=0}^{n} b_{ij}\mathbf{P}_j \\ A\mathbf{Q} & = & B\mathbf{P} \end{array}$$

where the matrix $A = [a_{ij}]$ is $(m+1) \times (m+1)$ and the matrix $B = [b_{ij}]$ is $(m+1) \times (n+1)$. The vector $\mathbf{Q}$ is an $(m+1) \times 1$ block–column vector whose rows are the unknown control points $\mathbf{Q}_k$ for $0 \le k \le m$.

Similarly the vector $\mathbf{P}$ is an $(n+1) \times 1$ block–column vector whose rows are the known control points $\mathbf{P}_k$ for $0 \leq k \leq n$. If A is an invertible matrix, then we may solve for the unknown control points: $\mathbf{Q} = A^{-1}B\mathbf{P}$. If $A^{-1}B = [c_{ij}]$, this equation reduces to

$$\mathbf{Q}_i = \sum_{j=0}^{n} c_{ij}\mathbf{P}_j, \quad 0 \leq i \leq m$$

The problem of computing a curve $\mathbf{Y}(t)$ that approximates $\mathbf{X}(t)$ and has fewer control points reduces to computing the matrices A and B, inverting A, computing the product $A^{-1}B$, and finally computing the products $\mathbf{Q}_i = \sum_{j=0}^{m} c_{ij}\mathbf{P}_j$.

The entry a_{ij} of the matrix A is an integral of the product of basis functions $M_{i,d}(t)M_{j,d}(t)$. The *support* of $M_{i,d}(t)$ is the closed interval $[t_i, t_{i+d+1}]$ and has the property that $M_{i,d}(t)$ is not zero on the open interval (t_i, t_{i+d+1}). The basis function is zero everywhere outside the closed interval. What this means to us is that the product $M_{i,d}(t)M_{j,d}(t)$ only contributes to the integral of a_{ij} when it is nonzero. And it can only be nonzero when the support of $M_{i,d}(t)$, namely $[t_i, t_{i+d+1}]$, and the support of $M_{j,d}(t)$, namely $[t_j, t_{j+d+1}]$, are intervals that overlap on some interval of positive length (overlap at a single point is not relevant). The supports *do not overlap* when $t_{i+d+1} \leq tt_j$ or $t_{j+d+1} \leq t_i$. Equivalently, no overlap occurs when $i+d+1 \leq j$ or $j+d+1 \leq i$. Thus, the supports *do overlap* when $|i-j| \leq d$. The elements a_{ij} are

$$a_{ij} = \begin{cases} \int_{t_j}^{t_{i+d+1}} M_{i,d}(t)M_{j,d}(t)\,dt, & 0 \leq i-j \leq d \\ \int_{t_i}^{t_{j+d+1}} M_{i,d}(t)M_{j,d}(t)\,dt, & -d \leq i-j \leq 0 \\ 0, & |i-j| > 0 \end{cases}$$

Consequently A is a symmetric and *banded* matrix with $2d-1$ bands, each band starting in the first row or first column of the matrix and proceeding diagonally downwards in the matrix. The matrix elements on the bands are the only (potentially) nonzero elements. All elements off the bands are zero. You are certainly familiar with diagonal matrices (one band) and tridiagonal matrices (three bands). A banded matrix generalizes these concepts. For example, if $m = 4$ and $d = 2$, the matrix A has three bands,

$$A = \begin{bmatrix} a_{00} & a_{01} & 0 & 0 & 0 \\ a_{01} & a_{11} & a_{12} & 0 & 0 \\ 0 & a_{12} & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{23} & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{34} & a_{44} \end{bmatrix}$$

The main diagonal is one band, a band starts in row 0 and column 1, and a band starts in row 1 and column 0. If $m = 5$ and $d = 3$, the matrix A has five bands,

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & 0 & 0 & 0 \\ a_{01} & a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{02} & a_{12} & a_{22} & a_{23} & a_{24} & 0 \\ 0 & a_{13} & a_{23} & a_{33} & a_{34} & a_{35} \\ 0 & 0 & a_{24} & a_{34} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{35} & a_{45} & a_{55} \end{bmatrix}$$

The main diagonal is one band. Two bands start in row 0, one at column 1 and one at column 2. Two bands start in column 0, one at row 1 and one at row 2.

A similar analysis applies to matrix B. The support for $M_{i,d}(t)$ is $[t_i, t_{i+d+1}]$ and the support for $N_{j,d}(t)$ is $[s_j, s_{j+d+1}]$. The integral for b_{ij} is nonzero only when the two supports have positive overlap. The support interval lengths are not the same, so the test for overlap is not as easily reduced to comparisons of the knot indices. However, this is irrelevant for the application since we can compute only those b_{ij} that are not zero by testing the supports themselves for overlap.

Clearly the computational effort lies in the inversion of the matrix A. The band entries a_{ij} are integrals of polynomials and may be computed in closed form. Rather than determining the closed form equations, the implementation uses a Romberg integration to compute the integrals. Inverting a general $N \times N$ matrix requires on the order of N^3 operations. However, a banded matrix requires less computation time because of the presence of all those zeros. A diagonal matrix with all nonzero diagonal entries can be inverted with N operations (one division per diagonal element). Inversion of a tridiagonal matrix is covered in standard undergraduate text books on numerical methods and requires on the order of N operations. Our banded matrix can be inverted quickly as long as the degree d is much smaller than m, which for most practical applications it is. The inversion algorithm itself is similar to the one used for tridiagonal matrices.

NOTE. The implementation is not currently available at the web site. It is based on the next version of Wild Magic which should appear in August 2003. An illustrative application is provided to show how effective an approximation you can obtain with a greatly reduced set of control points (the example uses $m = n/10$).