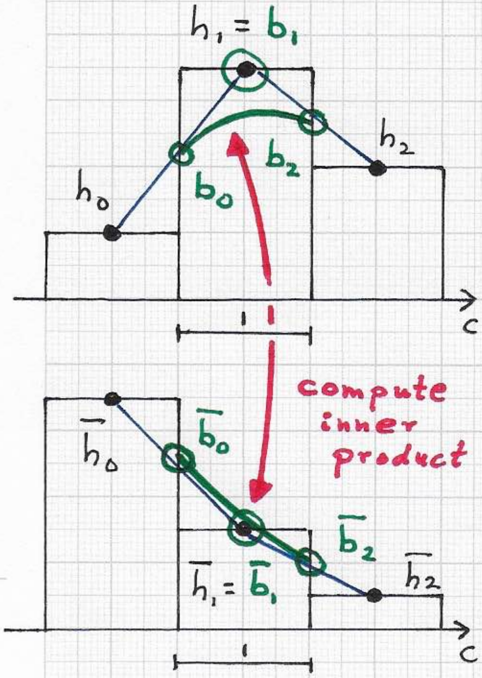
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions: ... Representing all discrete histogram data via (low-degree) B-spline representations makes possible the use of methods from approximation theory: best approximation; least-squares methods; normal equations; inner products of analytically defined piecewise polynomial functions etc. Quadratic B-splines are used to describe in detail the approximation of a new, given histogram as a linear combination of stored histograms of (material) samples. For simplicity, it is assumed that all histograms involved have the same number of bins (B), all of the same uniform width and together covering the same domain range of values. Further the number of bins is equal to the number of stored sample histograms (k): $B = k$. The crucial computation for best approximation is inner product calculation. Specifically, we consider inner products of quadratic B-splines. ...



Essential data involved in inner product computations (using local indices 0, 1 and 2):

- • original histogram data of a first and second distribution
- o 3 control points of quadratic B-spline segment used for Bernstein-Bézier representation
- — single analytically defined polynomial segment over a bin with width 1

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions: ... The "building block" of best approximation in our example and setting is the definition and computation of the inner product of two quadratic polynomials over the same interval, a specific bin in our case. The figures on the previous page illustrate the local data needed. (In the following, we perform integration over a local unit interval $[0,1]$; performing integration over the interval $[0,\Delta]$ would merely require one to multiple integral values for the unit interval by Δ.) In Bernstein-Bézier form the quadratic polynomials over the same bin/interval are:

$$h(c) = \sum_{i=0}^{2} b_i\, B_i^2(c) \quad \text{and}$$

$$\bar{h}(c) = \sum_{i=0}^{2} \bar{b}_i\, B_i^2(c), \quad c \in [0,1].$$

Their inner product is

$$\langle h(c), \bar{h}(c) \rangle = \int_0^1 h(c)\,\bar{h}(c)\, dc$$

$$= \int_0^1 \left(b_0 B_0^2(c) + b_1 B_1^2(c) + b_2 B_2^2(c) \right) \cdot \left(\bar{b}_0 B_0^2(c) + \bar{b}_1 B_1^2(c) + b_2 B_2^2(c) \right) dc$$

$$= \int_0^1 \left(\sum_{i=0}^{4} d_i\, B_i^4(c) \right) dc$$

$$= \frac{1}{5} \sum_{i=0}^{4} d_i \quad . \quad \ldots$$

$*$	$(1-c)^2$	$2(1-c)c$	c^2
$(1-c)^2$	$(1-c)^4$	$2(1-c)^3c$	$(1-c)^2c^2$
$2(1-c)c$	$2(1-c)^3c$	$4(1-c)^2c^2$	$2(1-c)c^3$
c^2	$(1-c)^2c^2$	$2(1-c)c^3$	c^4

The needed 9 products of Bernstein polynomials $B_i^2(c)$ and $B_j^2(c)$, $i,j=0,1,2$.

$$\Rightarrow h(c)\cdot\bar{h}(c) =$$

$$\begin{aligned}
= {} & b_0\bar{b}_0 && \cdot 1(1-c)^4 \\
& + (b_0\bar{b}_1 + b_1\bar{b}_0)/2 && \cdot 4(1-c)^3c \\
& + (b_0\bar{b}_2 + 4b_1\bar{b}_1 + b_2\bar{b}_0)/6 && \cdot 6(1-c)^2c^2 \\
& + (b_1\bar{b}_2 + b_2\bar{b}_1)/2 && \cdot 4(1-c)c^3 \\
& + b_2\bar{b}_2 && \cdot 1\,c^4
\end{aligned}$$

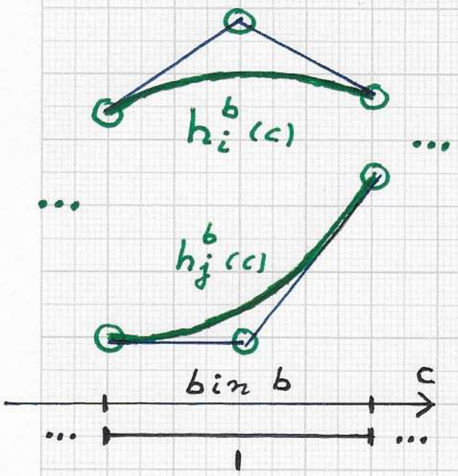
$$\begin{aligned}
= {} & d_0\, B_0^4(c) \\
& + d_1\, B_1^4(c) \\
& + d_2\, B_2^4(c) \\
& + d_3\, B_3^4(c) \\
& + d_4\, B_4^4(c)
\end{aligned}$$

$$= \sum_{i=0}^{4} d_i\, B_i^4(c) \quad .$$

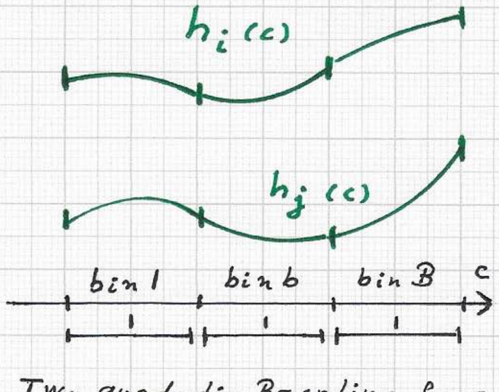
The needed product of $h(c)$ and $\bar{h}(c)$ in Bernstein-Bézier form. The coefficients of this quartic polynomial are the d_i-coefficients.

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions: ...



The functions $h_i^b(c)$ and $h_j^b(c)$ are two quadratic B-spline segments over the same bin b, $b=1\ldots k$. In fact, the complete domain of these two functions is the union of all bins, from bin 1 to bin k. For simplicity, only one of their k segments is shown. (Note: $B=k$, i.e., number of bins = number of stored sample histograms.)



Two quadratic B-spline functions shown over their entire shared domain, from bin 1 to bin B. Functions $h_i(c)$ and $h_j(c)$ are elements of a set of "basis" functions to represent a new $h(c)$.

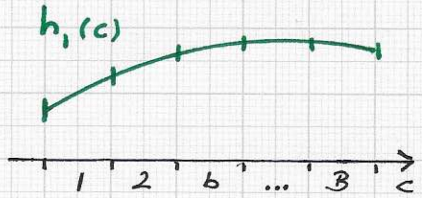
... We have used the notation $h(c)$ and $\bar{h}(c)$, $c \in [0,1]$, for two quadratic polynomials in Bernstein-Bézier form and determined the formula for their inner product. The value of their inner product is $P = \frac{1}{5}\sum_{i=0}^{4} d_i$. (See previous page.) We must compute inner products for pairs of quadratic B-spline functions defined over B $(=k)$ bins, bin 1 to bin B, representing our k stored sample histogram functions and the one new, given histogram function to be expressed as an expansion of the k stored functions. The function $h_i(c)$ is the quadratic B-spline ("basis") function having all bins as its domain, i.e., its domain is $\bigcup_{b=1}^{B}$ bin b. The new, given function $h(c)$ is the quadratic B-spline function to be expanded (exactly or as a best approximation) via the "basis" functions $h_i(c)$:

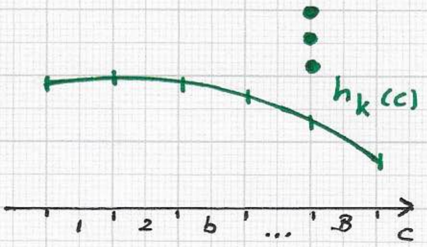
$$h(c) = \sum_{i=1}^{k} h_i(c) \quad .$$

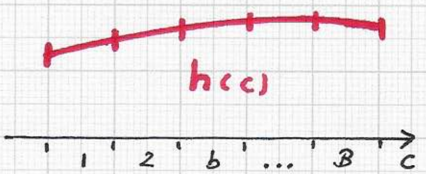
• Note: At this point, we have a set-up where we have enforced that $B=k$ and the segments b of all B-spline functions $h(c)$ and $h_i(c)$, $i=1\ldots B$, have the same domain bins: bin b. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

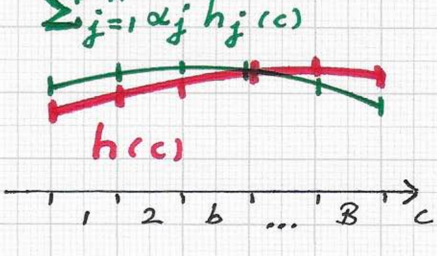
• Laplacian eigenfunctions: ...







Best approximation of $h(c)$. The stored k quadratic histogram B-spline "basis" functions are $h_1(c), \dots, h_k(c)$. All functions are defined over the same domain, i.e., the union of bins $1 \dots B$. Bin width is uniform; its value is $\Delta = 1$. Note: $B = k$.



Final best approximation of $h(c)$: the linear combination of stored sample histogram functions $h_j(c)$ minimizing error in the least-squares sense.

To simplify the description of the best approximation method, we use the following notations: The inner product of two histogram functions $h_i(c)$ and $h_j(c)$ is written as

$$\langle h_i(c), h_j(c) \rangle = \int_{\text{bin } 1}^{\text{bin } B} h_i(c)\, h_j(c)\, dc$$

$$= \sum_{\text{all bins } b} \int_{\text{bin } b} h_i(c)\, h_j(c)\, dc$$

$$= \sum_{\text{all bins } b} P_{i,j}^{b} = \sum_{b=1}^{B} P_{i,j}^{b} .$$

Similarly, we write the inner product of the new, given histogram function $h(c)$ and a histogram function $h_i(c)$ as

$$\langle h(c), h_i(c) \rangle = \int_{\text{bin } 1}^{\text{bin } B} h(c)\, h_i(c)\, dc$$

$$= \sum_{\text{all bins } b} \int_{\text{bin } b} h(c)\, h_i(c)\, dc$$

$$= \sum_{\text{all bins } b} P_{i}^{b} = \sum_{b=1}^{B} P_{i}^{b} .$$

(The previous pages define the values of $P_{i,j}^{b}$ and P_{i}^{b}, which are inner product values of two quadratic B-spline segments.)

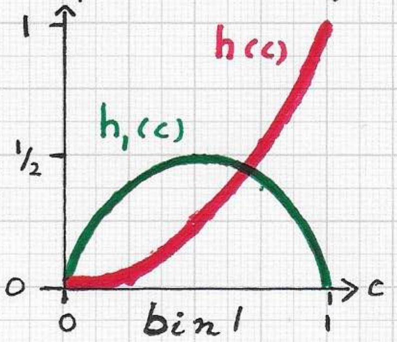
...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions: … One must solve the normal equations to obtain the coefficients α_j of the best approximation $\sum_{j=1}^{k} \alpha_j h_j(c)$ that minimizes the squared error $\left(h(c) - \sum_{j=1}^{k} \alpha_j h_j(c) \right)^2$ over the shared, common domain of all histogram functions involved, i.e., the union of all bins, $\bigcup_{b=1}^{B} \text{bin } b$. The normal equations are

$$\begin{bmatrix} \sum_{b=1}^{B} P_{1,1}^{b} & \cdots & \sum_{b=1}^{B} P_{1,k}^{b} \\ \vdots & & \vdots \\ \sum_{b=1}^{B} P_{k,1}^{b} & \cdots & \sum_{b=1}^{B} P_{k,k}^{b} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} \sum_{b=1}^{B} P_{1}^{b} \\ \vdots \\ \sum_{b=1}^{B} P_{k}^{b} \end{bmatrix}.$$

• Simple example: best approximation of a single quadratic segment $h(c)$ by a "linear combination" of a single "basis" function $h_1(c)$, a quadratic function, $c \in [0,1]$:



$$h(c) = c^2$$

$$h_1(c) = 2(c - c^2)$$

⇒ best app. of $h(c)$:

$$\sum_{j=1}^{1} \alpha_j h_j(c)$$

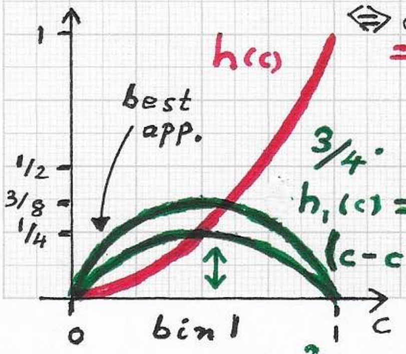
normal equations:

$$\langle h_1(c), h_1(c) \rangle \alpha_1 = \langle h(c), h_1(c) \rangle$$

$$\Leftrightarrow \alpha_1 \int_0^1 h_1^2 \, dc = \int_0^1 2c^2 (c - c^2) \, dc$$

$$\Leftrightarrow \alpha_1 \cdot \frac{2}{15} = 2 \int_0^1 c^3 - c^4 \, dc = \frac{1}{10}$$

$$\Leftrightarrow \alpha_1 = \frac{3}{4}$$



The expansion $\frac{3}{4} h_1(c)$ is the best approximation.

The matrix consists of B (=k) rows and B (=k) columns; it generally does not have a "special structure"; it is not a diagonal or block-diagonal matrix. Thus, this matrix should be inverted, or factorized, when a coefficient tuple $(\alpha_1, \ldots, \alpha_k)$ must be computed efficiently; this is possible since all matrix values only depend on stored histogram data, generated during "training," during sample histogram creation.

Thus, matrix inversion is done IN ADVANCE.

…