Communications in Information and Systems
Volume 16, Number 3, 147–183, 2016

Trivariate and n-variate optimal smoothing splines with dynamic shape modeling of deforming object

Hiroyuki Kano and Hiroyuki Fujioka

We develop a method of constructing multi-variate optimal smoothing splines using normalized uniform B-spline as the basis functions. First we consider trivariate splines in details, which are useful particularly for modeling dynamic shape of 3-dimensional deformable object by using two variables for 3D shape and one for time evolution. The splines are constructed as a tensor product of three B-splines, and an optimal smoothing spline problem is solved together with typical examples of constraints as periodicity and boundary conditions. The algorithms are developed so that various types of constraints can be incorporated easily and existing numerical solvers for convex quadratic programming (QP) can be readily applicable for numerical solutions. The theory and algorithms are then extended to the general n-variate case. Using trivariate splines, we demonstrate usefulness of the method by two examples; vibration of rectangular membrane and dynamic 3D shape modeling of red blood cell. We will see that relatively small number of observation data with noises yield satisfactory results.

1 Introduction **148**

2 Trivariate optimal smoothing splines **150**

3 Periodicity and constraints on splines **156**

4 Generalization to n-variate splines **162**

5 Numerical experiments on trivariate splines **168**

Key words and phrases: splines, smoothing, multi-variate splines, vibration of rectangular membrane, dynamic shape modeling, red blood cell.

6 Concluding remarks **180**

References **180**

1. Introduction

We consider optimal multi-variate splines and develop practical algorithm for numerical computations. There have been numerous studies on splines in single variable and their theories and algorithms as interpolation and approximation or smoothing, have been extended to splines in two variables (bivariate splines). They have been shown very useful in various fields of science and engineering as computer graphics, numerical analysis, image processing, trajectory planning, statistical data analysis, etc. (see e.g. [4, 15, 23, 26, 31]). Obviously, one is interested in extending those theories and algorithms to the case of three variables and moreover to the general case of n-variate splines. Such extensions are desired in many applications as aforementioned area in general, and here we consider two examples, namely vibration of rectangular membrane and modeling of 3D shape of deformable objects.

Typical objects with such deformable motions may be wet material object – such as jellyfish, red blood cell and amoeba, etc. One of important issues in their studies is to analyze and understand the motions of such objects from the observational data, e.g. image frames in a movie file. Therein the contour modeling of objects plays key roles and have been studied in the field of image processing, and various techniques, e.g. active contour model [5, 7], have been developed. The approach in these studies is to treat the contours independently at each sampling time and hence is not suitable to analyze and understand a whole motion of the deformable objects continuously in time.

Another applications related to our present study are the spline-based solid modeling of the human organs from a set of tomogram data obtained, for instance, by the magnetic range imaging (MRI) [1, 2]. Human lungs are modeled by periodic smoothing spline surface [16], where a set of contours are designed first from tomograph data and then a spline surface (bivariate spline) is designed from the contours. The design method of smoothing spline surfaces is thus viewed as a two step procedure, and we may point out that the whole set of tomograph data is not used effectively to construct the spline surface. Also, a similar idea has been applied to the 3-dimensional shape modeling of cell nucleus [29].

As for the theories and algorithms of multi-variable splines, there are some literatures: Trivariate splines are used to model 3D shapes using triangle mesh of geometric shapes as the input data [28]. Also an interpolation problem is considered for multi-dimensional splines, but restricting to only cubic splines [14]. Another approach to splines is by the so-called dynamic splines, where linear control systems are used as spline generator. The book [8] contains the state of the art of dynamic splines, and the authors developed a theory of periodic splines based on Hilbert space optimization [19]. This method has the advantage in that various types of functions can be used as basis functions as exponential functions, trigonometric functions, polynomials, and their combinations by choosing appropriate system matrix. However, extension to multi-variable case is not easy by this approach.

Here we develop a method for constructing optimal multi-variate smoothing splines using tensor product of B-splines as the basis elements. In particular, normalized uniform B-splines [17] are used, which enable us to derive concise representation of optimal smoothing splines and to incorporate various types of constraints more easily. Included in the constraints are periodicity and boundary conditions as treated in this paper as well as the so-called shape preserving splines as monotone splines, convex splines, constraints on integral values, and so forth [11, 13, 20]. This paper extends our B-spline based studies on optimal splines, in particular those in [10, 18, 21], and the theories and algorithms are developed first for trivariate case in details and the results are then extended to the n-variate case. The splines treated are of arbitrary degree. The problem is formulated as convex quadratic programming (QP) problem in such a way that a MATLAB QP solver is readily applicable for numerical solutions. Trivariate smoothing splines are useful to dynamic shape modeling of deforming objects, where two variables are used to model the 3D shape imposing periodicity and the other to represent time evolution. Here we consider vibration of rectangular membrane as solution of 2-dimensional wave equation and modeling red blood cell [22, 30]. We will see that relatively small number of observation data consisting of time sequence of 2D data sets yield satisfactory results.

Note that, by restricting to tensor product construction of B-splines with uniform knot point distribution (cf. ([3, 24]), we could derive concise expressions for multi-variate smoothing splines with various constraints, readily implementable as computational algorithms. Moreover, they can be used systematically and effectively in problems as dynamic 3D shape modeling, which is novel to the authors knowledge.

This paper is organized as follows. In Section 2, we formulate and solve the problems of optimal design of trivariate smoothing splines. Periodicity and boundary constraints are considered in Section 3. The results in Sections 2 and 3 are extended to n-variate case in Section 4. In Section 5, the results are applied to vibration of rectangular membrane and dynamic shape modeling of red blood cell and shown. Concluding remarks are given in Section 6.

2. Trivariate optimal smoothing splines

We describe a problem of optimal design of trivariate smoothing splines and derive the solution.

2.1. Trivariate spline by B-splines

For designing trivariate splines $x(t),\ t=(t_1,t_2,t_3)$, we employ normalized, uniform B-spline function $B_k(s)$ of degree k as the basis functions,

$$x(t)=\sum_{i_1=-k}^{m_1-1}\sum_{i_2=-k}^{m_2-1}\sum_{i_3=-k}^{m_3-1}\tau_{i_1,i_2,i_3}B_{k,i_1}^{[1]}(t_1)B_{k,i_2}^{[2]}(t_2)B_{k,i_3}^{[3]}(t_3), \tag{1}$$

where $m_i(>2)$ are integers, τ_{i_1,i_2,i_3} are the control points, and $B_{k,i}^{[j]}(\cdot)$ for $j=1,2,3$ are the scaled and shifted B-splines of degree k with knot points $u_i^{[j]}$ defined by

$$B_{k,i}^{[j]}(s)=B_k(\alpha_j(s-u_i^{[j]})). \tag{2}$$

Here $\alpha_j(>0)$ are the scaling factors, and specify the intervals of equally spaced knot points as

$$u_{i+1}^{[j]}-u_i^{[j]}=\frac{1}{\alpha_j}. \tag{3}$$

Note that, here and hereafter, the superscript $[j]$ is used to refer j-th variable t_j for $j=1,2,3$.

For convenience, the definition of normalized, uniform B-spline functions $B_k(s)$ used in above is given.

$$B_k(s)=\begin{cases}N_{k-j,k}(s-j) & j\le s<j+1\quad j=0,\dots,k\\ 0 & s<0,\ k+1\le s.\end{cases} \tag{4}$$

Table 1: Cubic B-spline $B_3(s)$ and its derivatives $B_3^{(l)}(s)$.

	$B_3(s)$	$B_3^{(1)}(s)$	$B_3^{(2)}(s)$	$B_3^{(3)}(s)$
$0 \le s < 1$	$\frac{1}{6}s^3$	$\frac{1}{2}s^2$	s	1
$1 \le s < 2$	$\frac{1}{6}\left(-3s^3+12s^2-12s+4\right)$	$\frac{1}{2}\left(-3s^2+8s-4\right)$	$-3s+4$	-3
$2 \le s < 3$	$\frac{1}{6}\left(3s^3-24s^2+60s-44\right)$	$\frac{1}{2}\left(3s^2-16s+20\right)$	$3s-8$	3
$3 \le s < 4$	$\frac{1}{6}(4-s)^3$	$-\frac{1}{2}(4-s)^2$	$-s+4$	-1
$s < 0,\ s \ge 4$	0	0	0	0

Here the basis elements $N_{j,k}(s)$ $(j = 0, 1, \dots, k)$ are obtained recursively by the following algorithm [6]. Let $N_{0,0}(s) \equiv 1$ and, for $i = 1, 2, \dots, k$, compute

$$(5)\quad \begin{cases} N_{0,i}(s) = \frac{1-s}{i} N_{0,i-1}(s) \\ N_{j,i}(s) = \frac{i-j+s}{i} N_{j-1,i-1}(s) + \frac{1+j-s}{i} N_{j,i-1}(s), \quad j = 1, \dots, i-1 \\ N_{i,i}(s) = \frac{s}{i} N_{i-1,i-1}(s). \end{cases}$$

Thus, $B_k(s)$ is a piece-wise polynomial of degree k with integer knot points and is $k-1$ times continuously differentiable, and it holds that

$$\sum_{j=0}^{k} N_{j,k}(s) = 1, \quad 0 \le s \le 1.$$

For reference, we show here the cubic B-spline $B_3(s)$ and its derivatives in Table 1.

2.2. Optimal smoothing splines

First we formulate a problem of trivariate optimal smoothing splines and then present its solution. We develop the theory and algorithms in the form that they may readily be used in trivariate problems and that further extensions to general n-variate cases becomes clear as we continue in Section 4. Also note that the problem described in this section is usually solved in conjunction with the constraints as given in Section 3.

By choosing appropriate control points τ_{i_1,i_2,i_3} $(-k \le i_j \le m_j - 1; j = 1, 2, 3)$, the function $x(t)$ in (1) represents trivariate spline of degree k on the domain $\mathcal{S} = \mathcal{I}_1 \times \mathcal{I}_2 \times \mathcal{I}_3 \subset \mathbf{R}^3$ where $\mathcal{I}_j = [u_0^{[j]}, u_{m_j}^{[j]}]$. Now suppose that a set of data

$$(6) \qquad \mathcal{D} = \left\{ (v_i; d_i) : v_i = (v_i^{[1]}, v_i^{[2]}, v_i^{[3]}) \in \mathcal{S},\ d_i \in \mathbf{R},\ i = 1, 2, \ldots, N \right\}$$

is given where $v_i \in \mathbf{R}^3$ is the data point and $d_i \in \mathbf{R}$ is the data, and let $\tau = [\tau_{i_1,i_2,i_3}] \in \mathbf{R}^{M_1 \times M_2 \times M_3}$ with $M_i = m_i + k$.

Then, a standard problem of designing optimal trivariate smoothing splines is to find a function $x(t)$, or equivalently an array τ, minimizing the following cost function consisting of a smoothness term and approximation error term.

Problem 1. Construct the spline $x(t)$ in (1) such that

$$\min_{\tau \in \mathbf{R}^{M_1 \times M_2 \times M_3}} J(\tau),$$

where

$$(7) \qquad J(\tau) = \lambda \int_{\mathcal{S}} \left(\nabla^2 x(t) \right)^2 dt + \sum_{i=1}^{N} w_i (x(v_i) - d_i)^2.$$

In (7), $\lambda (> 0)$ is a smoothing parameter, $\nabla^2 = \frac{\partial^2}{\partial t_1^2} + \frac{\partial^2}{\partial t_2^2} + \frac{\partial^2}{\partial t_3^2}$ and w_i $(0 < w_i \leq 1)$ denotes weights for approximation errors.

This problem can be solved as follows. Let $b_j(s) \in \mathbf{R}^{M_j}$ $(j = 1, 2, 3)$ be

$$(8) \qquad \begin{aligned} b_j(s) &= \left[B_{k,-k}^{[j]}(s) \quad B_{k,-k+1}^{[j]}(s) \quad \cdots \quad B_{k,m_j-1}^{[j]}(s) \right]^T \\ &= \Big[\ B_k(\alpha_j(s - u_{-k}^{[j]})) \quad B_k(\alpha_j(s - u_{-k+1}^{[j]})) \quad \cdots \\ &\qquad\qquad \cdots \quad B_k(\alpha_j(s - u_{m_j-1}^{[j]})) \ \Big]^T \end{aligned}$$

and first, the three dimensional array τ is reshaped into a one dimensional, column vector form $\hat{\tau} \in \mathbf{R}^M$ with $M = M_1 M_2 M_3$ as follows.

Recall that two dimensional array, say a matrix $A = [a_1\ a_2 \cdots\ a_{M_2}] \in \mathbf{R}^{M_1 \times M_2}$ with $a_i \in \mathbf{R}^{M_1}$, can be reshaped to a column vector $\hat{A} \in \mathbf{R}^M$ $(M = M_1 M_2)$ by the so-called vec-function for matrices as

$$\hat{A} = \left[\ a_1^T \quad a_2^T \quad \cdots \quad a_{M_2}^T \ \right]^T,$$

and this construction is denoted as $\hat{A} = \text{vec } A$ (see e.g. [25]). Natural extension to the three dimensional case is to apply this vec-function to each matrix $\tau_i = [\tau_{i_1,i_2,i}] \in \mathbf{R}^{M_1 \times M_2}$, obtained from the three dimensional array τ by fixing the last index i_3 at $i_3 = i$, as $\hat{\tau}_i = \text{vec}\ \tau_i$ and then construct

$$\hat{\tau} = \left[\begin{array}{cccc} \hat{\tau}_{-k}^T & \hat{\tau}_{-k+1}^T & \cdots & \hat{\tau}_{m_3-1}^T \end{array}\right]^T. \tag{9}$$

This construction of the vector $\hat{\tau}$ from the three dimensional array τ will be denoted as

$$\hat{\tau} = \text{vec}\ \tau. \tag{10}$$

Remark 1. Using MATLAB function 'reshape', the vector $\hat{\tau}$ is obtained from the array τ by $\hat{\tau} = \text{reshape}(\tau, M, 1)$, and conversely $\tau = \text{reshape}(\hat{\tau}, M_1, M_2, M_3)$.

Now we get the following formula for $x(t)$, where $\otimes$ denotes Kronecker product.

Proposition 1. *Trivariate spline $x(t)$ in (1) is expressed as*

$$x(t) = b^T(t)\hat{\tau}, \tag{11}$$

where $b(t) \in \mathbf{R}^M$ is defined by

$$b(t) = b_3(t_3) \otimes b_2(t_2) \otimes b_1(t_1). \tag{12}$$

Proof. In (1), we rewrite $x(t_1, t_2, t_3)$ as

$$x(t_1, t_2, t_3) = \sum_{i=-k}^{m_3-1} x_i(t_1, t_2) B_{k,i}^{[3]}(t_3), \tag{13}$$

where $x_i(t_1, t_2)$ is defined by

$$x_i(t_1, t_2) = \sum_{i_1=-k}^{m_1-1} \sum_{i_2=-k}^{m_2-1} \tau_{i_1,i_2,i} B_{k,i_1}^{[1]}(t_1) B_{k,i_2}^{[2]}(t_2). \tag{14}$$

Letting $\tau_i = [\tau_{i_1,i_2,i}] \in \mathbf{R}^{M_1 \times M_2}$ as in (9) and using (8), we can express $x_i(t_1, t_2)$ as

$$\begin{aligned} x_i(t_1, t_2) &= b_1^T(t_1)\tau_i b_2(t_2) \\ &= \left(b_2^T(t_2) \otimes b_1^T(t_1)\right) \hat{\tau}_i \end{aligned} \tag{15}$$

where $\hat{\tau}_i \in \mathbf{R}^{M_{12}}$ $(M_{12} = M_1 M_2)$ is the column vector expression of matrix τ_i, namely $\hat{\tau}_i = \text{vec}\ \tau_i$. In (15), we used the relation vec $AXB = (B^T \otimes$

A)vec X for matrices A, B, X of compatible dimensions and $c = \text{vec}\ c$ for any scalar c. Substituting (15) into (13) yields

$$\begin{aligned} x(t_1, t_2, t_3) &= \sum_{i=-k}^{m_3-1} B_{k,i}^{[3]}(t_3) \left(b_2^T(t_2) \otimes b_1^T(t_1) \right) \hat{\tau}_i \\ &= \left(b_3^T(t_3) \otimes b_2^T(t_2) \otimes b_1^T(t_1) \right) \hat{\tau} \end{aligned} \tag{16}$$

where $\hat{\tau} = \begin{bmatrix} \hat{\tau}_{-k}^T & \hat{\tau}_{-k+1}^T & \cdots & \hat{\tau}_{m_3-1}^T \end{bmatrix}^T$, and it coincides with the vector $\hat{\tau}$ generated by (10). Thus (11) follows. □

The cost function in (7) is then obtained in terms of $\hat{\tau}$ as

$$J(\hat{\tau}) = \lambda \hat{\tau}^T Q \hat{\tau} + \left(B^T \hat{\tau} - d \right)^T W \left(B^T \hat{\tau} - d \right). \tag{17}$$

Here, $Q \in \mathbf{R}^{M \times M}$ is a Gram matrix defined by

$$Q = \int_{\mathcal{S}} \left(\nabla^2 b(t)) \right) \left(\nabla^2 b(t) \right)^T dt, \tag{18}$$

and obviously $Q = Q^T \geq 0$. The Laplace operator ∇^2 on vector $b(t)$ should be understood as operating on each element of the vector. Moreover, in (17), matrix $B \in \mathbf{R}^{M \times N}$ is defined by

$$\begin{aligned} B &= \begin{bmatrix} b(v_1) & b(v_2) & \cdots & b(v_N) \end{bmatrix} \\ &= \begin{bmatrix} b_3(v_1^{[3]}) \otimes b_2(v_1^{[2]}) \otimes b_1(v_1^{[1]}) & \cdots & b_3(v_N^{[3]}) \otimes b_2(v_N^{[2]}) \otimes b_1(v_N^{[1]}) \end{bmatrix}, \end{aligned} \tag{19}$$

and $W \in \mathbf{R}^{N \times N}$ and $d \in \mathbf{R}^N$ by

$$\begin{aligned} W &= \text{diag} \left\{ \begin{matrix} w_1 & w_2 & \cdots & w_N \end{matrix} \right\} \\ d &= \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}^T. \end{aligned} \tag{20}$$

Thus, the cost function in (17) is expressed as

$$J(\hat{\tau}) = \hat{\tau}^T G \hat{\tau} - 2 g^T \hat{\tau} + c, \tag{21}$$

where

$$G = \lambda Q + B W B^T, \quad g = B W d, \quad c = d^T W d, \tag{22}$$

and the optimal smoothing spline is obtained as a solution of

$$G \hat{\tau} = g. \tag{23}$$

Finally in this section, we consider the Gram matrix Q in (18) from its computational point of view. Noting that

$$\nabla^2 b(t) = \left(\frac{\partial^2}{\partial t_1^2} + \frac{\partial^2}{\partial t_2^2} + \frac{\partial^2}{\partial t_3^2}\right)(b_3(t_3) \otimes b_2(t_2) \otimes b_1(t_1)) \tag{24}$$

and using the properties of Kronecker products $(A \otimes B \otimes C)^T = A^T \otimes B^T \otimes C^T$ and $(A \otimes B \otimes C)(A' \otimes B' \otimes C') = (AA') \otimes (BB') \otimes (CC')$ for matrices of compatible dimensions, we obtain

$$\begin{aligned} Q = {} & Q_3^{(22)} \otimes Q_2^{(00)} \otimes Q_1^{(00)} + Q_3^{(20)} \otimes Q_2^{(02)} \otimes Q_1^{(00)} \\ & + Q_3^{(20)} \otimes Q_2^{(00)} \otimes Q_1^{(02)} + Q_3^{(02)} \otimes Q_2^{(20)} \otimes Q_1^{(00)} \\ & + Q_3^{(00)} \otimes Q_2^{(22)} \otimes Q_1^{(00)} + Q_3^{(00)} \otimes Q_2^{(20)} \otimes Q_1^{(02)} \\ & + Q_3^{(02)} \otimes Q_2^{(00)} \otimes Q_1^{(20)} + Q_3^{(00)} \otimes Q_2^{(02)} \otimes Q_1^{(20)} \\ & + Q_3^{(00)} \otimes Q_2^{(00)} \otimes Q_1^{(22)}. \end{aligned} \tag{25}$$

Here $Q_l^{(ij)} \in \mathbf{R}^{M_l \times M_l}$ $(l = 1, 2, 3;\ i, j = 0, 1, 2)$ are defined by

$$Q_l^{(ij)} = \int_{\mathcal{I}_l} \frac{d^i b_l(s)}{ds^i} \frac{d^j b_l^T(s)}{ds^j} ds \tag{26}$$

and hence $Q_l^{(ji)} = (Q_l^{(ij)})^T$ holds. The elements of $Q_l^{(ij)}$ can be precomputed from the B-splines once the parameters k and m_l are determined (see [9] for the cubic spline case $k = 3$).

For convenience, we summarize an algorithm for constructing the trivariate smoothing spline $x(t)(= x(t_1, t_2, t_3))$ described above.

Algorithm 1. (Trivariate smoothing spline without constraints) Suppose that we are given a set of data in (6). Then, $x(t)$ is constructed in the following steps (S1)-(S6).

(S1) Set k, α_i, m_i in (1), and let $M_i = m_i + k$ for $i = 1, 2, 3$.

(S2) Set λ and w_i for $i = 1, 2, \ldots, N$ in (7).

(S3) Compute the matrices Q in (25) and B in (19), and set up the matrix W and the vector d in (20).

(S4) Compute the matrix G and the vector g in (22).

(S5) Solve (23) to obtain the control point vector $\hat{\tau}$.

(S6) Compute the spline $x(t)$ in (11).

3. Periodicity and constraints on splines

There are various types of constraints that we would like to impose on smoothing splines $x(t)$. Here we consider the cases of periodicity and boundary constraints, where, as we see in the numerical examples, the former can be used to construct closed surface and the latter to specify boundary conditions of surfaces.

3.1. t_3-periodicity

For convenience of description, we start with the spline $x(t_1, t_2, t_3)$ which is periodic in the third variable t_3, referring it as t_3-periodic. Specifically, recalling that the spline $x(t_1, t_2, t_3)$ is defined for $t_j \in \mathcal{I}_j = [u_0^{[j]}, u_{m_j}^{[j]}]$ $(j = 1, 2, 3)$, the spline $x(t_1, t_2, t_3)$ is t_3-periodic if it satisfies the constraint

$$\frac{\partial^l}{\partial t_3^l} x(t_1, t_2, u_0^{[3]}) = \frac{\partial^l}{\partial t_3^l} x(t_1, t_2, u_{m_3}^{[3]}) \tag{27}$$

for $\forall t_1 \in \mathcal{I}_1, \forall t_2 \in \mathcal{I}_2$, and $\forall l = 0, 1, \ldots, k-1$. This t_3-periodicity of $x(t_1, t_2, t_3)$ should be understood as that it is periodic in t_3 with the period $u_{m_3}^{[3]} - u_0^{[3]}$ if its domain $\mathcal{I}_1 \times \mathcal{I}_2 \times \mathcal{I}_3$ is extended in the entire t_3-axis, namely in $\mathcal{I}_1 \times \mathcal{I}_2 \times (-\infty, +\infty)$.

Proposition 2. *The spline $x(t_1, t_2, t_3)$ is t_3-periodic if and only if*

$$\hat{\tau}_i = \hat{\tau}_{m_3+i}, \quad \forall i = -k, -k+1, \ldots, -1 \tag{28}$$

holds for vectors $\hat{\tau}_i$ defined in (9).

Proof. By (16) and (2), we get

$$\begin{aligned} x(t_1, t_2, t_3) &= \sum_{i=-k}^{m_3-1} B_k(\alpha_3(t_3 - u_i^{[3]})) \left(b_2^T(t_2) \otimes b_1^T(t_1)\right) \hat{\tau}_i \\ &= \left(b_2^T(t_2) \otimes b_1^T(t_1)\right) \sum_{i=-k}^{m_3-1} B_k(\alpha_3(t_3 - u_i^{[3]}))\hat{\tau}_i, \end{aligned} \tag{29}$$

yielding

$$\begin{aligned}(30)\qquad \frac{\partial^l}{\partial t_3^l}x(t_1,t_2,u_0^{[3]}) &= \left(b_2^T(t_2)\otimes b_1^T(t_1)\right)\sum_{i=-k}^{m_3-1}\alpha_3^l B_k^{(l)}(\alpha_3(u_0^{[3]}-u_i^{[3]}))\hat{\tau}_i \\ &= \alpha_3^l\left(b_2^T(t_2)\otimes b_1^T(t_1)\right)\sum_{i=-k}^{-1}B_k^{(l)}(-i)\hat{\tau}_i\end{aligned}$$

for $l=0,1,\ldots,k-1$, where we used (3) and the fact that $B_k^{(l)}(i)$ vanishes unless $1\le i\le k$. Similarly, for $t_3=u_{m_3}^{[3]}$, we get

$$(31)\qquad \frac{\partial^l}{\partial t_3^l}x(t_1,t_2,u_{m_3}^{[3]}) = \alpha_3^l\left(b_2^T(t_2)\otimes b_1^T(t_1)\right)\sum_{i=-k}^{-1}B_k^{(l)}(-i)\hat{\tau}_{m_3+i},$$

and hence we see that the constraint (27) holds if and only if

$$(32)\qquad \sum_{i=-k}^{-1}B_k^{(l)}(-i)(\hat{\tau}_i-\hat{\tau}_{m_3+i})=0$$

for $\forall l=0,1,\ldots,k-1$. Noting that the matrix

$$(33)\qquad \bar{B}=\begin{bmatrix} B_k^{(0)}(k) & B_k^{(1)}(k) & \cdots & B_k^{(k-1)}(k) \\ B_k^{(0)}(k-1) & B_k^{(1)}(k-1) & \cdots & B_k^{(k-1)}(k-1) \\ \vdots & \vdots & & \vdots \\ B_k^{(0)}(1) & B_k^{(1)}(1) & \cdots & B_k^{(k-1)}(1)\end{bmatrix}$$

is nonsingular (see [10]), we see that (32) is equivalent to $\hat{\tau}_i-\hat{\tau}_{m_3+i}=0$ for $i=-k,-k+1,\ldots,-1$ and the assertion holds. □

Note that, in terms of the entire vector $\hat{\tau}\in\mathbf{R}^M$, the condition (28) is expressed from (9) as

$$(34)\qquad C_3\hat{\tau}=0$$

where $C_3\in\mathbf{R}^{kM_{12}\times M}$ ($M_{12}=M_1M_2$) is defined by

$$(35)\qquad C_3=\begin{bmatrix} I_{kM_{12}} & O_{kM_{12},M-2kM_{12}} & -I_{kM_{12}}\end{bmatrix}.$$

Thus t_3-periodic smoothing spline is obtained by minimizing the cost (21) under the constraint (34).

3.2. t_2- and t_1-periodicity

For two vectors $p \in \mathbf{R}^m$ and $q \in \mathbf{R}^n$, we define a permutation matrix $K_{n,m} \in \mathbf{R}^{nm \times nm}$ such that

$$K_{n,m}(p \otimes q) = q \otimes p. \tag{36}$$

Then it holds that $K_{n,m}^T = K_{n,m}^{-1} = K_{m,n}$ (see e.g. [27]), and it is straightforward to verify that $K_{n,m}$ is given by

$$K_{n,m} = \begin{bmatrix} e_{1,1} & e_{2,1} & \cdots & e_{m,1} \\ e_{1,2} & e_{2,2} & \cdots & e_{m,2} \\ \vdots & \vdots & & \vdots \\ e_{1,n} & e_{2,n} & \cdots & e_{m,n} \end{bmatrix} \tag{37}$$

where $e_{i,j}$ is the $m \times n$ matrix with 1 at the ij-th position and 0 elsewhere.

Now, t_2- and t_1-periodicity are defined similarly as in (27), and we get the following conditions. The latter case is put in parentheses.

Proposition 3. *The spline $x(t_1, t_2, t_3)$ is t_2-periodic (t_1-periodic) if and only if*

$$\hat{\tau}'_i = \hat{\tau}'_{m_2+i} \quad (\hat{\tau}''_i = \hat{\tau}''_{m_1+i}) \quad \forall i = -k, -k+1, \ldots, -1 \tag{38}$$

holds, where the vectors $\hat{\tau}'_i \in \mathbf{R}^{M_{13}}$ with $M_{13} = M_1 M_3$ (the vectors $\hat{\tau}''_i \in \mathbf{R}^{M_{23}}$ with $M_{23} = M_2 M_3$) are defined by partitioning a permuted $\hat{\tau}$ vector as

$$\hat{\tau}' = K_{M_{12},M_3}\hat{\tau} = \begin{bmatrix} \hat{\tau}'_{-k} \\ \hat{\tau}'_{-k+1} \\ \vdots \\ \hat{\tau}'_{m_2-1} \end{bmatrix} \quad \left(\hat{\tau}'' = K_{M_1,M_{23}}\hat{\tau} = \begin{bmatrix} \hat{\tau}''_{-k} \\ \hat{\tau}''_{-k+1} \\ \vdots \\ \hat{\tau}''_{m_1-1} \end{bmatrix} \right). \tag{39}$$

Outline of proof. The case of t_2-periodicity: Using (36), the vector $b(t)$ in (12) is arranged as

$$\begin{aligned} b(t) &= b_3(t_3) \otimes (b_2(t_2) \otimes b_1(t_1)) \\ &= K_{M_3,M_{12}}((b_2(t_2) \otimes b_1(t_1)) \otimes b_3(t_3)) \\ &= K_{M_3,M_{12}}(b_2(t_2) \otimes b_1(t_1) \otimes b_3(t_3)). \end{aligned} \tag{40}$$

Thus we get

$$x(t_1,t_2,t_3) = b^T(t)\hat{\tau} = (b_2^T(t_2)\otimes b_1^T(t_1)\otimes b_3^T(t_3))\hat{\tau}' \tag{41}$$

where $\hat{\tau}' = K^T_{M_3,M_{12}}\hat{\tau} = K_{M_{12},M_3}\hat{\tau}$, and similarly as (29)

$$\begin{aligned} x(t_1,t_2,t_3) &= \sum_{i=-k}^{m_2-1} B_k(\alpha_2(t_2-u_i^{[2]}))\left(b_1^T(t_1)\otimes b_3^T(t_3)\right)\hat{\tau}'_i \\ &= \left(b_1^T(t_1)\otimes b_3^T(t_3)\right)\sum_{i=-k}^{m_2-1} B_k(\alpha_2(t_2-u_i^{[2]}))\hat{\tau}'_i. \end{aligned} \tag{42}$$

Then $\frac{\partial^l}{\partial t_2^l}x(t_1,u_0^{[2]},t_3) = \frac{\partial^l}{\partial t_2^l}x(t_1,u_{m_2}^{[2]},t_3)$ $\forall t_1,t_3,l$ yields

$$\sum_{i=-k}^{-1} B_k^{(l)}(-i)(\hat{\tau}'_i - \hat{\tau}'_{m_2+i}) = 0 \tag{43}$$

for $\forall l = 0,1,\ldots,k-1$, and (38) follows as in the case of (32).

The case of t_1-periodicity: As in (29) and (41), here we rearrange $b(t)$ using (36) as

$$b(t) = K_{M_{23},M_1}(b_1(t_1)\otimes b_3(t_3)\otimes b_2(t_2)) \tag{44}$$

so that the term $b_1(t_1)$ is placed at the leftmost position in the expression of $x(t_1,t_2,t_3)$, namely

$$x(t_1,t_2,t_3) = b^T(t)\hat{\tau} = (b_1^T(t_1)\otimes b_3^T(t_3)\otimes b_2^T(t_2))\hat{\tau}'' \tag{45}$$

where $\hat{\tau}'' = K^T_{M_{23},M_1}\hat{\tau} = K_{M_1,M_{23}}\hat{\tau}$. Similar process as in (42)–(43) yields the desired result (38). □

In terms of the vector $\hat{\tau}$, the condition (38) for t_2-periodicity (t_1-periodicity) is given as

$$C_2\hat{\tau} = 0 \quad (C_1\hat{\tau} = 0) \tag{46}$$

where $C_2 \in \mathbf{R}^{kM_{13}\times M}$ ($C_1 \in \mathbf{R}^{kM_{23}\times M}$) is defined by

$$\begin{aligned} &C_2 = \left[\begin{array}{ccc} I_{kM_{13}} & O_{kM_{13},M-2kM_{13}} & -I_{kM_{13}} \end{array}\right] K_{M_{12},M_3} \\ &\left(C_1 = \left[\begin{array}{ccc} I_{kM_{23}} & O_{kM_{23},M-2kM_{23}} & -I_{kM_{23}} \end{array}\right] K_{M_1,M_{23}}\right) \end{aligned} \tag{47}$$

Remark 2. From Proposition 2 and (9), we see that the condition for t_3-periodicity may be stated as $\tau_i = \tau_{m_3+i}$ $\forall i$ $(-k \leq i \leq -1)$. Likewise we can verify that t_2- and t_1-periodicity are equivalent to $\tau'_i = \tau'_{m_2+i}$ and $\tau''_i = \tau''_{m_1+i}$ $\forall i$ $(-k \leq i \leq -1)$, respectively, where τ'_i and τ''_i are matrices obtained from 3D array $\tau = [\tau_{i_1,i_2,i_3}]$ by fixing i_2 and i_1 as i, namely

$$\tau'_i = [\tau_{i_1,i,i_3}] \in \mathbf{R}^{M_1 \times M_3}, \quad \tau''_i = [\tau_{i,i_2,i_3}] \in \mathbf{R}^{M_2 \times M_3} \tag{48}$$

In other words, the t_j-periodicity is achieved by equating the first and last k matrices constructed from τ_{i_1,i_2,i_3} with the j-th index fixed. Propositions 2 and 3 are employed so that the conditions can be used directly in connection with the cost $J(\hat{\tau})$ in (21).

3.3. Boundary constraints

As an another constraint on trivariate splines $x(t_1, t_2, t_3)$, we consider to restrict the spline at the boundary of the domain $\mathcal{S} = \mathcal{I}_1 \times \mathcal{I}_2 \times \mathcal{I}_3$ $(\mathcal{I}_j = [u_0^{[j]}, u_{m_j}^{[j]}],\ j = 1, 2, 3)$. First, for a given integer l $(0 \leq l < k)$, consider the following boundary constraint at $t_1 = u_0^{[1]}$,

$$\frac{\partial^l}{\partial t_1^l} x(u_0^{[1]}, t_2, t_3) = 0 \quad \forall (t_2, t_3) \in \mathcal{I}_2 \times \mathcal{I}_3. \tag{49}$$

This constraint is easily realized as follows. Similarly as (30), the left hand side of (49) is obtained from (45) as

$$\frac{\partial^l}{\partial t_1^l} x(u_0^{[1]}, t_2, t_3) = \alpha_1^l \left(b_3^T(t_3) \otimes b_2^T(t_2) \right) \sum_{i=-k}^{-1} B_k^{(l)}(-i) \hat{\tau}''_i. \tag{50}$$

Hence (49) is equivalent to

$$\sum_{i=-k}^{-1} B_k^{(l)}(-i) \hat{\tau}''_i = 0 \tag{51}$$

or $C'_1 \hat{\tau}'' = 0$ where $C'_1 \in \mathbf{R}^{M_{23} \times M}$ is given by

$$\begin{aligned} C'_1 = \Big[\; & B_k^{(l)}(k) I_{M_{23}} \quad B_k^{(l)}(k-1) I_{M_{23}} \quad \cdots \\ & \cdots \quad B_k^{(l)}(1) I_{M_{23}} \quad O_{M_{23}, M-kM_{23}} \; \Big]. \end{aligned} \tag{52}$$

Substituting the relation $\hat{\tau}'' = K_{M_1,M_{23}}\hat{\tau}$ to $C_1'\hat{\tau}'' = 0$ gives the desired condition in terms of the vector $\hat{\tau}$, namely

$$\bar{C}_1'\hat{\tau} = 0 \tag{53}$$

with $\bar{C}_1' = C_1'K_{M_1,M_{23}}$. When $k = 3$, the matrix C_1' in (52) may be constructed using Table 1. For example, if $l = 0$, we get $B_3(3) = \frac{1}{6}, B_3(2) = \frac{4}{6}$ and $B_3(1) = \frac{1}{6}$.

The constraint at another boundary $t_1 = u_{m_1}^{[1]}$ is treated similarly by noting

$$\frac{\partial^l}{\partial t_1^l}x(u_{m_1}^{[1]}, t_2, t_3) = \alpha_1^l \left(b_3^T(t_3) \otimes b_2^T(t_2)\right) \sum_{i=-k}^{-1} B_k^{(l)}(-i)\hat{\tau}''_{m_1+i} = 0. \tag{54}$$

In this case the boundary constraint is realized by $C_1''\hat{\tau}' = 0$ where $C_1'' \in \mathbf{R}^{M_{23}\times M}$ is given by

$$C_1'' = \Big[\; O_{M_{23},M-kM_{23}} \quad B_k^{(l)}(k)I_{M_{23}} \quad B_k^{(l)}(k-1)I_{M_{23}} \quad \cdots \quad \cdots \quad B_k^{(l)}(1)I_{M_{23}} \;\Big]. \tag{55}$$

On the other hand, t_2-boundary constraints

$$\frac{\partial^l}{\partial t_2^l}x(t_1, t_2, t_3) = 0 \quad \forall (t_1, t_3) \in \mathcal{I}_1 \times \mathcal{I}_3 \tag{56}$$

for $t_2 = u_0^{[2]}$ and $t_2 = u_{m_2}^{[2]}$ are realized in terms of $\hat{\tau}'_i \in \mathbf{R}^{M_{13}}$ in (39) by

$$\sum_{i=-k}^{-1} B_k^{(l)}(-i)\hat{\tau}'_i = 0 \quad \text{and} \quad \sum_{i=-k}^{-1} B_k^{(l)}(-i)\hat{\tau}'_{m_2+i} = 0 \tag{57}$$

respectively. As in (53), these conditions are rewritten in terms of $\hat{\tau}$ using $\hat{\tau}' = K_{M_{12},M_3}\hat{\tau}$.

Finally, in the case of t_3-boundary, we simply replace $\hat{\tau}'_i$ in (57) by $\hat{\tau}_i \in \mathbf{R}^{M_{12}}$ and m_2 by m_3.

Remark 3. As in (49), we considered the zero boundary conditions here. However the above results can be generalized to nonzero case as

$$\frac{\partial^l}{\partial t_1^l} x(u_0^{[1]}, t_2, t_3) = f(t_2, t_3) \ \ \forall (t_2, t_3) \in \mathcal{I}_2 \times \mathcal{I}_3. \tag{58}$$

as long as the given function $f(t_2, t_3)$ can be expressed by the same basis functions $B_{k,i_2}^{[2]}(t_2)$ and $B_{k,i_3}^{[3]}(t_3)$ as in (1).

Remark 4. All the constraints considered in this section are described in terms of the control point vector $\hat{\tau}$ in the form $C\hat{\tau} = 0$, and hence the smoothing spline with these constraints can be obtained by minimizing the cost $J(\hat{\tau})$ in (21) with the equality constraint $C\hat{\tau} = 0$. For numerical solution, Lagrange multiplier method can be employed or convex quadratic programming solver as 'quadprog' in MATLAB may be used in Step (S5) in Algorithm 1.

4. Generalization to n-variate splines

In this section, we generalize the theory and algorithms on trivariate smoothing splines developed in the previous sections to the general n-variate case.

Recalling the trivariate case in (1), an n-variate spline function $x(t)$, $t = (t_1, t_2, \dots, t_n)$ is constructed using $B_k(s)$ by

$$x(t) = \sum_{i_1=-k}^{m_1-1} \cdots \sum_{i_n=-k}^{m_n-1} \tau_{i_1, i_2, \dots, i_n} \prod_{j=1}^{n} B_{k,i_j}^{[j]}(t_j). \tag{59}$$

Here $m_i (> 2)$ are integers, $\tau_{i_1, i_2, \dots, i_n}$ are the control points, $B_{k,i}^{[j]}(\cdot)$ are the scaled B-splines of degree k with knot points $u_i^{[j]}$ defined by

$$B_{k,i}^{[j]}(s) = B_k(\alpha_j (s - u_i^{[j]})), \tag{60}$$

and $\alpha_j (> 0)$ are the scaling factors adjusting the knot point intervals by

$$u_{i+1}^{[j]} - u_i^{[j]} = \frac{1}{\alpha_j}. \tag{61}$$

Also the vectors $b_j(s) \in \mathbf{R}^{M_j}$ in (8) are cited again for convenience.

$$b_j(s) = \begin{bmatrix} B_{k,-k}^{[j]}(s) & B_{k,-k+1}^{[j]}(s) & \cdots & B_{k,m_j-1}^{[j]}(s) \end{bmatrix}^T. \tag{62}$$

4.1. Optimal smoothing splines

The domain of this n-variate spline function $x(t)$ is $\mathcal{S} = \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_n \subset \mathbf{R}^n$ where $\mathcal{I}_j = [u_0^{[j]}, u_{m_j}^{[j]}]$, and smoothing splines are constructed assuming that we are given the following set of data:

$$(63)\quad \mathcal{D} = \left\{ (v_i; d_i) : v_i = (v_i^{[1]}, v_i^{[2]}, \ldots, v_i^{[n]}) \in \mathcal{S},\ d_i \in \mathbf{R},\ i = 1, 2, \ldots, N \right\}.$$

Here $v_i \in \mathbf{R}^n$ is the data point and $d_i \in \mathbf{R}$ is the data.

As in Problem 1, the problem of optimal smoothing spline is to construct the spline $x(t)$, or equivalently to find the n-dimensional control point array $\tau = [\tau_{i_1, i_2, \ldots, i_n}]$ $(\in \mathbf{R}^{M_1 \times M_2 \times \cdots \times M_n})$ $(M_i = m_i + k)$ so that the following cost function is minimized.

$$(64)\qquad J(\tau) = \lambda \int_{\mathcal{S}} \left(\nabla^2 x(t) \right)^2 dt + \sum_{i=1}^{N} w_i (x(v_i) - d_i)^2.$$

Here $\nabla^2 = \frac{\partial^2}{\partial t_1^2} + \frac{\partial^2}{\partial t_2^2} + \cdots + \frac{\partial^2}{\partial t_n^2}$, and the parameters $\lambda (> 0)$ and w_i $(0 < w_i \leq 1)$ are the same as before.

We convert the cost function in the form such that its minimum can be computed by applying a MATLAB solver of QP problems. Specifically we vectorize the n-dimensional array τ as a column vector $\hat{\tau} \in \mathbf{R}^M$ $(M = M_1 M_2 \cdots M_n)$ as denoted by

$$(65)\qquad \hat{\tau} = \text{vec } \tau.$$

This construction is a straightforward extension of the case of $n = 3$ described in Section 2, and is extended recursively as $n = 2, 3, \ldots$. Specifically, letting τ_i be an $(n-1)$-dimensional array obtained from n-dimensional array τ by fixing the last index as $i_n = i$ $(-k \leq i \leq m_n - 1)$, namely $\tau_i = \left[\tau_{i_1, i_2, \ldots, i_{n-1}, i} \right]$ $(\in \mathbf{R}^{M_1 \times M_2 \times \cdots \times M_{n-1}})$, and with $\hat{\tau}_i = \text{vec } \tau_i$, the vector $\hat{\tau}$ is constructed by

$$(66)\qquad \hat{\tau} = \left[\begin{array}{cccc} \hat{\tau}_{-k}^T & \hat{\tau}_{-k+1}^T & \cdots & \hat{\tau}_{m_n - 1}^T \end{array} \right]^T.$$

With this definition of $\hat{\tau}$, we get the following expression of $x(t)$.

Proposition 4. *n-variate spline $x(t)$ in (59) is expressed as*

$$x(t) = b^T(t)\hat{\tau}, \tag{67}$$

where $b(t) \in \mathbf{R}^M$ is defined by

$$b(t) = b_n(t_n) \otimes b_{n-1}(t_{n-1}) \otimes \cdots \otimes b_1(t_1). \tag{68}$$

Proof. We prove this by mathematical induction method. When $n = 1$, the assertion obviously holds since $x(t) = \sum_{i_1=-k}^{m_1-1} \tau_{i_1} B_{k,i_1}^{[1]}(t) = b_1^T(t)\tau$ in (59), $b(t) = b_1(t)$ and $\hat{\tau} = \tau$.

Now assume that the assertion holds for $n = n - 1$, and write (59) as

$$x(t) = \sum_{i=-k}^{m_n-1} x_i(\bar{t}) B_{k,i}^{[n]}(t_n) \tag{69}$$

where $\bar{t} = (t_1, t_2, \ldots, t_{n-1})$ and

$$x_i(\bar{t}) = \sum_{i_1=-k}^{m_1-1} \cdots \sum_{i_{n-1}=-k}^{m_{n-1}-1} \tau_{i_1,i_2,\ldots,i_{n-1},i} \prod_{j=1}^{n-1} B_{k,i_j}^{[j]}(t_j). \tag{70}$$

By the present assumption, we can write

$$x_i(\bar{t}) = \bar{b}^T(\bar{t})\hat{\tau}_i \tag{71}$$

for $\bar{b}(\bar{t}) = b_{n-1}(t_{n-1}) \otimes \cdots \otimes b_1(t_1)$. Thus (69) is rewritten as

$$\begin{aligned} x(t) &= \sum_{i=-k}^{m_n-1} \bar{b}^T(\bar{t})\hat{\tau}_i B_{k,i}^{[n]}(t_n) \\ &= \begin{bmatrix} B_{k,-k}^{[n]}(t_n)\bar{b}^T(\bar{t}) & \cdots & B_{k,m_n-1}^{[n]}(t_n)\bar{b}^T(\bar{t}) \end{bmatrix} \begin{bmatrix} \hat{\tau}_{-k} \\ \vdots \\ \hat{\tau}_{m_n-1} \end{bmatrix} \\ &= \left(b_n^T(t_n) \otimes \bar{b}^T(\bar{t}) \right) \hat{\tau} \end{aligned} \tag{72}$$

where we used (66). Noting that $b_n^T(t_n) \otimes \bar{b}^T(\bar{t}) = b^T(t)$ given in (68), we see that (67) holds for the case $n = n$, and the Proposition is proved. □

Once we have this expression for $x(t)$, we see that the cost function (64) is rewritten in terms of the vector $\hat{\tau}$ completely in the same fashion as described in Section 2.2. The relevant equations are listed again as

$$J(\hat{\tau}) = \hat{\tau}^T G \hat{\tau} - 2g^T \hat{\tau} + c, \tag{73}$$

where

$$G = \lambda Q + BWB^T, \quad g = BWd, \quad c = d^T W d, \tag{74}$$

and the definitions of matrices and vectors are analogous to the case of (22) with $b(t)$ given by (68). In particular, the Gram matrix Q is given as

$$Q = \int_{\mathcal{S}} \left(\nabla^2 b(t)\right)\right) \left(\nabla^2 b(t)\right)^T dt = \sum_{\substack{i_1,\dots,i_n \in \bar{I} \\ i_1+\cdots+i_n=22}} Q_n^{(i_n)} \otimes \cdots \otimes Q_1^{(i_1)} \tag{75}$$

where $\bar{I} = \{00, 02, 20, 22\}$ and $Q_i^{(j)}$ is as defined in (26). This sum should be understood as taken, for example when $n = 2$, over $(i_1, i_2) = (00, 22)$, $(02, 20)$, $(20, 02)$, $(22, 00)$, and hence

$$Q = Q_2^{(22)} \otimes Q_1^{(00)} + Q_2^{(20)} \otimes Q_1^{(02)} + Q_2^{(02)} \otimes Q_1^{(20)} + Q_2^{(00)} \otimes Q_1^{(22)}. \tag{76}$$

Similarly, when $n = 3$, this yields the matrix Q given in (25).

4.2. Periodicity and boundary constraints

For given p $(1 \le p \le n)$, we call $x(t) = x(t_1, t_2, \dots, t_n)$ to be t_p-periodic if all the partial derivatives of $x(t)$ with respect to t_p agree at its boundaries $u_0^{[p]}$ and $u_{m_p}^{[p]}$ namely

$$\left.\frac{\partial^l x(t)}{\partial t_p^l}\right|_{t_p=u_0^{[p]}} = \left.\frac{\partial^l x(t)}{\partial t_p^l}\right|_{t_p=u_{m_p}^{[p]}}, \quad \forall t_i \in \mathcal{I}_i (i \neq p), \quad \forall l = 0, 1, \dots, k-1. \tag{77}$$

As we have seen in Sections 3.1 and 3.2, the basic idea for expressing the condition (77) in terms of the vector $\hat{\tau}$ is to rewrite (67) so that $b_p(t_p)$ is in the leftmost position as $b_2(t_2)$ in (41) for the t_2-periodic case.

Using a permutation matrix K in (36), we have an expression for $b(t)$ as

$$\begin{aligned} b(t) &= b_n(t_n) \otimes b_{n-1}(t_{n-1}) \otimes \cdots \otimes b_1(t_1) \\ &= (b_n(t_n) \otimes \cdots \otimes b_{p+1}(t_{p+1})) \otimes (b_p(t_p) \otimes \cdots \otimes b_1(t_1)) \\ &= K_{\bar{M}',\bar{M}} (b_p(t_p) \otimes \cdots \otimes b_1(t_1)) \otimes (b_n(t_n) \otimes \cdots \otimes b_{p+1}(t_{p+1})) \end{aligned} \tag{78}$$

where $\bar{M} = M_1 \cdots M_p$ and $\bar{M}' = M_{p+1} \cdots M_n$. Thus by (67) we get

$$\begin{aligned}(79)\quad x(t) &= b^T(t)\hat{\tau} \\ &= (b_p(t_p) \otimes \cdots \otimes b_1(t_1))^T \otimes (b_n(t_n) \otimes \cdots \otimes b_{p+1}(t_{p+1}))^T K^T_{\bar{M}',\bar{M}} \hat{\tau} \\ &= (b_p^T(t_p) \otimes \bar{b}^T(\bar{t}_p))\hat{\tau}'\end{aligned}$$

where $\bar{t}_p = (t_1, \ldots, t_{p-1}, t_{p+1}, \ldots, t_n)$,

$$\bar{b}(\bar{t}_p) = b_{p-1}(t_{p-1}) \otimes \cdots \otimes b_1(t_1) \otimes b_n(t_n) \otimes \cdots \otimes b_{p+1}(t_{p+1}), \tag{80}$$

and

$$\hat{\tau}' = K^T_{\bar{M}',\bar{M}}\hat{\tau} = K_{\bar{M},\bar{M}'}\hat{\tau}. \tag{81}$$

Noting that the vector $\bar{b}(\bar{t}_p)$ is of size $\hat{M} \times 1$ with $\hat{M} = M/M_p$, we partition the vector $\hat{\tau}' \in \mathbf{R}^M$ into M_p vectors $\hat{\tau}'_i \in \mathbf{R}^{\hat{M}}$ as

$$\hat{\tau}' = \begin{bmatrix} \hat{\tau}'_{-k} \\ \hat{\tau}'_{-k+1} \\ \vdots \\ \hat{\tau}'_{m_p-1} \end{bmatrix}. \tag{82}$$

Then in view of (62), $x(t)$ in (79) is rewritten as

$$\begin{aligned}(83)\quad x(t) &= \begin{bmatrix} B^{[p]}_{k,-k}(t_p)\bar{b}^T(\bar{t}_p) & \cdots & B^{[p]}_{k,m_p-1}(t_p)\bar{b}^T(\bar{t}_p)\end{bmatrix} \begin{bmatrix} \hat{\tau}'_{-k} \\ \hat{\tau}'_{-k+1} \\ \vdots \\ \hat{\tau}'_{m_p-1} \end{bmatrix} \\ &= \sum_{i=-k}^{m_p-1} B^{[p]}_{k,i}(t_p)\bar{b}^T(\bar{t}_p)\hat{\tau}'_i \\ &= \bar{b}^T(\bar{t}_p) \sum_{i=-k}^{m_p-1} B^{[p]}_{k,i}(t_p)\hat{\tau}'_i\end{aligned}$$

Using (60), we get as in the case of (30)

$$\left.\frac{\partial^l x(t)}{\partial t_p^l}\right|_{t_p = u_0^{[p]}} = \alpha_p^l \bar{b}^T(\bar{t}_p) \sum_{i=-k}^{-1} B_k^{(l)}(-i)\hat{\tau}'_i. \tag{84}$$

Similarly we get the following expression at the another boundary,

$$\left.\frac{\partial^l x(t)}{\partial t_p^l}\right|_{t_p=u^{[p]}_{m_p}} = \alpha_p^l \bar{b}^T(\bar{t}_p)\sum_{i=-k}^{-1} B_k^{(l)}(-i)\hat{\tau}'_{m_p+i}. \tag{85}$$

Hence the periodicity constraint (77) is written as

$$\alpha_p^l \bar{b}^T(\bar{t}_p)\sum_{i=-k}^{-1} B_k^{(l)}(-i)(\hat{\tau}'_i - \hat{\tau}'_{m_p+i}) = 0 \tag{86}$$

which must hold for all $\bar{t}_p (\in \mathcal{I}_1 \times \cdots \times \mathcal{I}_{p-1} \times \mathcal{I}_{p+1} \times \cdots \times \mathcal{I}_n)$ and all l $(0 \leq l < k)$. Analogous arguments as for (32) leads to the following conclusion as generalization of Propositions 2 and 3.

Proposition 5. *n-variate spline $x(t)$ in (59) is t_p-periodic if and only if*

$$\hat{\tau}'_i = \hat{\tau}'_{m_p+i} \quad \forall i = -k, -k+1, \ldots, -1 \tag{87}$$

where the vectors $\hat{\tau}'_i$ are defined from $\hat{\tau}$ by (81) and (82).

Noting (81), this condition is rewritten in terms of $\hat{\tau}$ as

$$C\hat{\tau} = 0, \; C = \begin{bmatrix} I_{k\hat{M}} & O_{k\hat{M},M-2k\hat{M}} & -I_{k\hat{M}} \end{bmatrix} K_{\bar{M},\bar{M}'} \tag{88}$$

Finally we consider boundary conditions for variable t_p as follows,

$$\left.\frac{\partial^l x(t)}{\partial t_p^l}\right|_{t_p=u^{[p]}_0} = 0, \quad \left.\frac{\partial^l x(t)}{\partial t_p^l}\right|_{t_p=u^{[p]}_{m_p}} = 0 \;\; \forall t_i \in \mathcal{I}_i (i \neq p) \tag{89}$$

for given l $(0 \leq l < k)$. From (84) and (85), we see that the boundary conditions in (89) are satisfied if and only if

$$\sum_{i=-k}^{-1} B_k^{(l)}(-i)\hat{\tau}'_i = 0, \quad \sum_{i=-k}^{-1} B_k^{(l)}(-i)\hat{\tau}'_{m_p+i} = 0 \tag{90}$$

for $t_p = u_0^{[p]}$ and $t_p = u_{m_p}^{[p]}$, respectively. These conditions can be expressed in terms of the vector $\hat{\tau}$ as $CK_{\bar{M},\bar{M}'}\hat{\tau} = 0$ with C defined similarly as (52) and (55).

5. Numerical experiments on trivariate splines

We examine the performances of multi-variate optimal smoothing splines numerically for the case $n = 3$. Two examples are considered: a vibration of rectangular membrane and a deformation of red blood cell.

First we summarize the numerical procedure for constructing the splines. The problem of optimal smoothing splines is formulated as a convex quadratic programming (QP) problem with the cost function given in the form

$$J(\hat{\tau}) = \hat{\tau}^T G \hat{\tau} - 2g^T \hat{\tau} + c, \tag{91}$$

together with the constraints

$$C\hat{\tau} = 0. \tag{92}$$

This constraint contains all the necessary equality constraints such as the periodicity and boundary conditions as developed in the previous sections, and it can be assumed without loss of generality that the constraint matrix C is of row full rank. Such a QP problem is solved numerically by using MATLAB function 'quadprog'.

Another important issue in optimal smoothing splines is on the choice of the smoothing parameter λ in the cost function (7). For this purpose, we employ the so-called generalized cross validation (GCV) method [12]. Under the assumption of $G > 0$ and $W = \frac{1}{N}I$, an optimal λ is obtained as the minimizer of the following GCV function $V(\lambda)$,

$$V(\lambda) = \frac{\frac{1}{N}\left\|(I - A(\lambda))\, d\right\|^2}{\left(\frac{1}{N}\mathrm{tr.}\,(I - A(\lambda))\right)^2}. \tag{93}$$

Here, $A(\lambda) \in \mathbf{R}^{N\times N}$ denotes the so-called 'influence matrix' defined by

$$\begin{bmatrix} x_\lambda(v_1) \\ x_\lambda(v_2) \\ \vdots \\ x_\lambda(v_N) \end{bmatrix} = A(\lambda) \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}, \tag{94}$$

where $x_\lambda(t)$ is used to denote the optimal spline for (91) with (92) under the parameter value λ, and $(v_i; d_i)$ are the given data in (6). It can be shown that $A(\lambda)$ is computed by

$$A(\lambda) = \frac{1}{N} B^T \left(G^{-1} + G^{-1} C^T \Delta^{-1} C G^{-1}\right) B \tag{95}$$

where $\Delta = -CG^{-1}C^T$ and G depends on λ. Note that Δ is invertible since C is assumed to be of row full rank.

In the following experiments, cubic splines ($k = 3$) are used.

5.1. Vibration of rectangular membrane

Here, trivariate splines $x(t_1, t_2, t_3)$ are denoted as $x(r, s, t)$ with the first two variable denoting spatial variables and the third time variable. We construct optimal smoothing spline $x(r, s, t)$ from the data $\mathcal{D}$ in (6) obtained by sampling the following function

$$f(r, s, t) = \cos\left(\frac{\sqrt{5}\pi}{2} t\right) \sin\left(\frac{\pi}{2} r\right) \sin\left(\pi s\right) \tag{96}$$

where $r \in \mathcal{I}_1 = [0, 2]$, $s \in \mathcal{I}_2 = [0, 1]$ and $t \in \mathcal{I}_3 = [0, T]$ with $T = 4/\sqrt{5}$.

This function is taken from a motion of a rectangular membrane of size $a \times b$ (see e.g. [32]) . Specifically $f(r, s, t)$ denotes a vertical displacement of a membrane at position $(r, s) \in [0, a] \times [0, b]$, at time $t(\geq 0)$, and is obtained as a solution of the following two-dimensional wave equation

$$\frac{\partial^2 f}{\partial t^2} = c^2 \left(\frac{\partial^2 f}{\partial r^2} + \frac{\partial^2 f}{\partial s^2}\right), \quad 0 \leq r \leq a,\ 0 \leq s \leq b,\ 0 \leq t \tag{97}$$

with the boundary conditions

$$\begin{aligned} f(0, s, t) = f(a, s, t) = 0, &\quad 0 \leq s \leq b,\ 0 \leq t \\ f(r, 0, t) = f(r, b, t) = 0, &\quad 0 \leq r \leq a,\ 0 \leq t \end{aligned} \tag{98}$$

and initial conditions,

$$\begin{aligned} f(r, s, 0) &= \sin\left(\frac{\pi}{a} r\right) \sin\left(\frac{\pi}{b} s\right) \\ \frac{\partial f}{\partial t}(r, s, 0) &= 0, \quad 0 \leq r \leq a,\ 0 \leq s \leq b \end{aligned} \tag{99}$$

The function in (96) corresponds to the periodic solution for the case $a = 2,\ b = 1, c = 1$ with the period $T = 2/(c\sqrt{1/a^2 + 1/b^2}) = 4/\sqrt{5}$.

The data points $v_i = (v_i^{[1]}, v_i^{[2]}, v_i^{[3]})$ are taken as 3-dimensional lattice points, equally spaced in each axis with the number $N_1 = 10,\ N_2 = 5,\ N_3 = 10$ in r, s, t axes respectively. Hence the total number of data is $N = N_1 N_2 N_3 = 500$, and the function is sampled to generate the data d_i with 20% additive white Gaussian noise.

For constructing an optimal smoothing spline $x(r, s, t)$, we impose the boundary constraints on r, s corresponding to those for the original problem (98)

$$\begin{aligned} &x(0, s, t) = x(2, s, t) = 0, \quad \forall s \in [0, 1],\ \forall t \in [0, T] \\ &x(r, 0, t) = x(r, 1, t) = 0, \quad \forall r \in [0, 2],\ \forall t \in [0, T] \end{aligned} \tag{100}$$

by the method described in Section 3.3. Moreover, using the method in Section 3.1, we constrain $x(r, s, t)$ to be t-periodic with the period T $(= 4/\sqrt{5})$, namely

$$\frac{\partial^l x(r, s, 0)}{\partial t^l} = \frac{\partial^l x(r, s, T)}{\partial t^l}, \quad \forall r \in [0, 2], \forall s \in [0, 1], \quad l = 0, 1, 2. \tag{101}$$

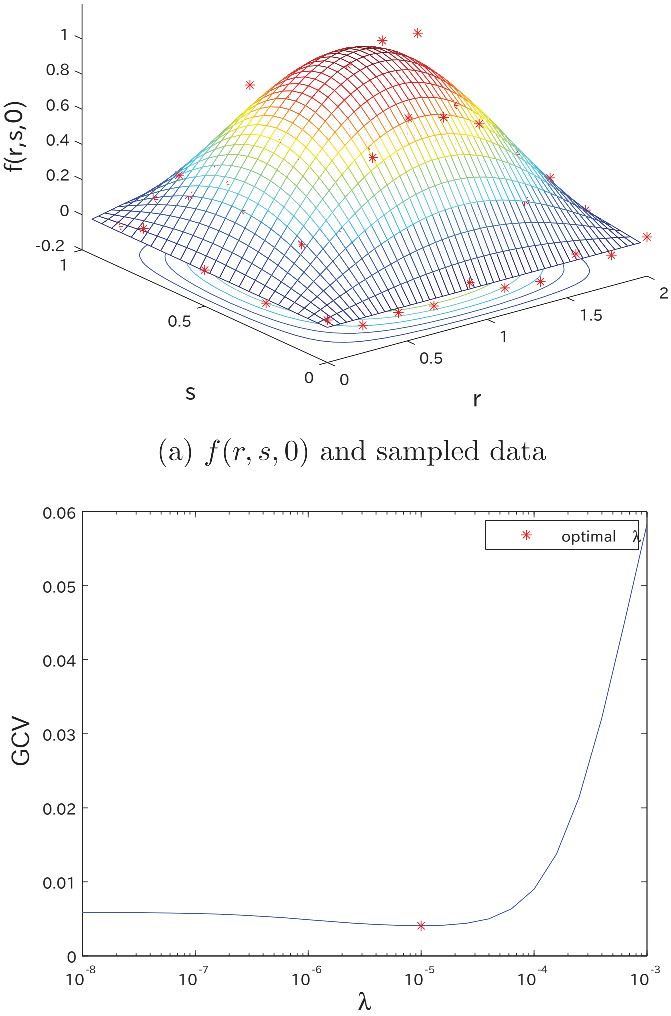
The number of control points are set as $m_1 = 10, m_2 = 5, m_3 = 10$, hence the knot point intervals along the r, s, t axes are $a/m_1 = 0.2 (= 1/\alpha_1)$, $b/m_2 = 0.2 (= 1/\alpha_2)$ and $T/m_3 = 4/\sqrt{5}/10 \approx 0.179 (= 1/\alpha_3)$, respectively. The weights w_i for approximation errors are set as $w_i = 1/N\ \forall i$.

Figure 1 (a) shows the function $f(r, s, t)$ at initial time $t = 0$ and sampled data with additive noises at $t = 0$. The number of data at $t = 0$ is $N_1 N_2 = 10 \times 5 = 50$. The generalized cross validation function $V(\lambda)$ is computed based on the 500 data d_i and plotted in (b). The optimal value of the smoothing parameter is obtained as $\lambda^* = 1.0 \times 10^{-5}$ (red asterisk in Figure 1 (b)).

Using this optimal λ^*, we constructed the optimal smoothing splines $x(r, s, t)$ as shown in Figure 2 for $t = 0, T/4, 2T/4$ and $t = 3T/4$. Although the number of used data is relatively small as $N = 500$ and the data contains 20% noise, we see that the 3D shapes $x(r, s, t)$ during the entire motion period $[0, T]$ resembles those of original function $f(r, s, t)$ [1].

In Figure 3 (a), the approximation error $f(r, s, t) - x(r, s, t)$ is plotted for $t = 0$ and the error for time $t \in [0, T]$ is evaluated as matrix 2-norm $\|f(r, s, t) - x(r, s, t)\|$ as shown in (b) (blue line). In the same figure, the norm of $f(r, s, t)$ is also plotted (dotted line, right scale), and we may judge

[1] Animation videos are available for both original and its spline approximated vibration motions.

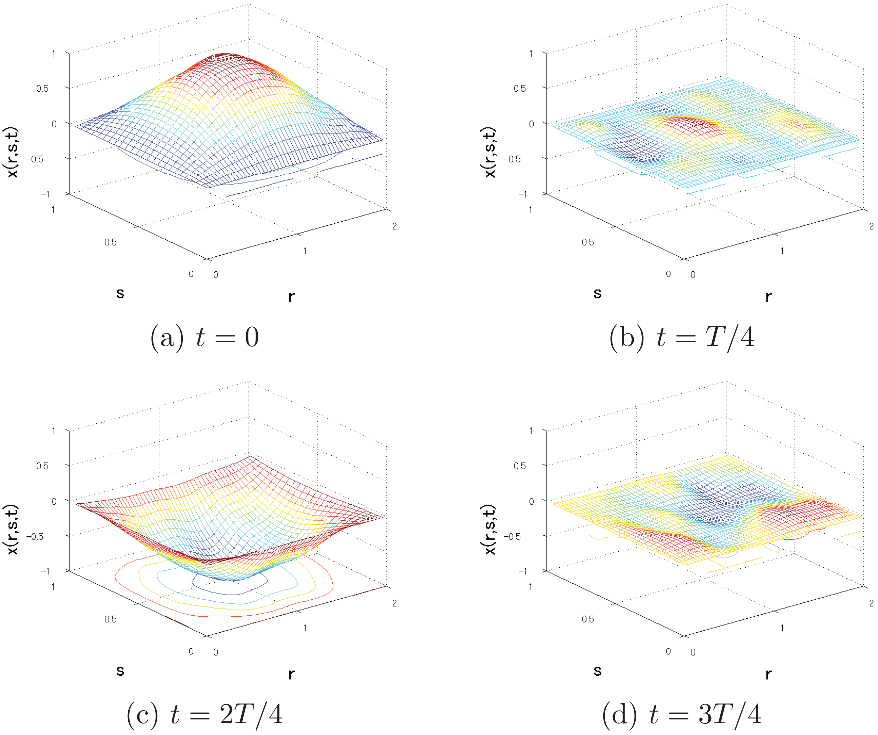


(a) $f(r, s, 0)$ and sampled data

(b) GCV function

Figure 1: True function $f(r, s, t)$ and sampled noisy data (denoted by *) at time $t = 0$ (a), and computed generalized cross validation function (b).

that the error is in the 10% range of original function $f(r, s, t)$. The same figure also includes the errors for the case of $N_1 = 20, N_2 = 10, N_3 = 20$ ($N =$



(a) $t = 0$

(b) $t = T/4$

(c) $t = 2T/4$

(d) $t = 3T/4$

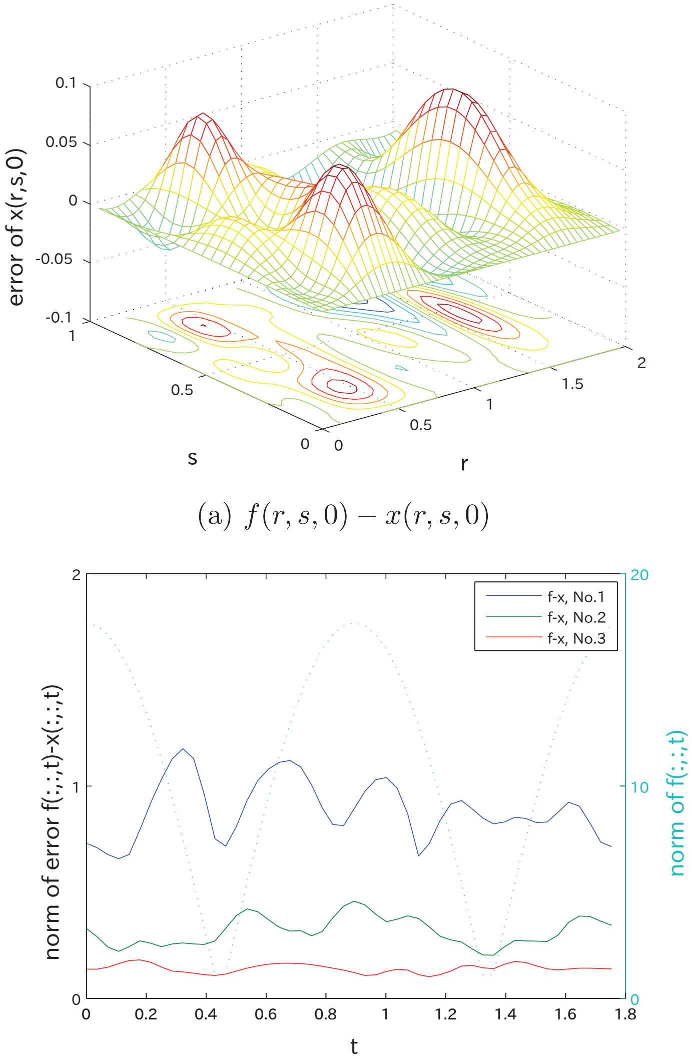
Figure 2: Sample plots of 3D shapes at time $t = 0, T/4, 2T/4, 3T/4$ ($T = 4/\sqrt{5}$) from vibration motion of rectangular membrane constructed by optimal periodic smoothing splines $x(r, s, t)$.

4000, green line), and $N_1 = 40, N_2 = 20, N_3 = 40$ ($N = 32000$, red line). We observe that the error decreases as the number of data N increases.

5.2. Deformation of red blood cell

We examine 3D deformation motion of red blood cells by trivariate smoothing splines. The data $\mathcal{D}$ is obtained by sampling the following function

$$f(\theta, \phi, t) = \sqrt{h_1^2(\theta, \phi, t) + h_2^2(\theta, \phi, t) + h_3^2(\theta, \phi, t)} \tag{102}$$



(a) $f(r, s, 0) - x(r, s, 0)$

(b) $\|f(r, s, t) - x(r, s, t)\|$ for each t

Figure 3: (a) Approximation error $f(r, s, t) - x(r, s, t)$ at time $t = 0$ and (b) the error norms evaluated for each t (blue line). Also shown in (b) are error norms by another experiments with $N_1 = 20, N_2 = 10, N_3 = 20$ (green line) and $N_1 = 40, N_2 = 20, N_3 = 40$ (red line) and norm of $f(r, s, t)$ (dotted line, right scale).

where $\theta \in \mathcal{I}_1 = [0, 2\pi]$ and $\phi \in \mathcal{I}_2 = [0, 2\pi]$ denote spacial variables, $t \in \mathcal{I}_3 = [0, T]$ with $T = 10$ denotes time, and

$$\begin{aligned} h_1(\theta, \phi, t) &= (a + 0.5\cos(\omega t))\gamma \sin\theta\cos\phi \\ h_2(\theta, \phi, t) &= a\gamma \sin\theta \sin\phi \\ h_3(\theta, \phi, t) &= (a + 0.5\cos(\omega t))\frac{\gamma}{2} \\ &\quad \times (0.207 + 2.003\sin^2\theta - 1.123\sin^4\theta)\cos\theta \end{aligned} \tag{103}$$

with $a = 2.8$, $\gamma = 1.38581894$ and $\omega = \pi/5$. This model is taken from [30], which we modified so as to include the additional parameter t in order to generate deforming motion. Note that $f(\theta, \phi, t)$ is periodic in all the three variables. In Figure 4, the function $f(\theta, \phi, t)$ at time $t = 0$ is plotted in (a), and the corresponding 3D shape of red blood cell in $O - XYZ$ space is constructed in (b) by the following transformation of polar coordinate system

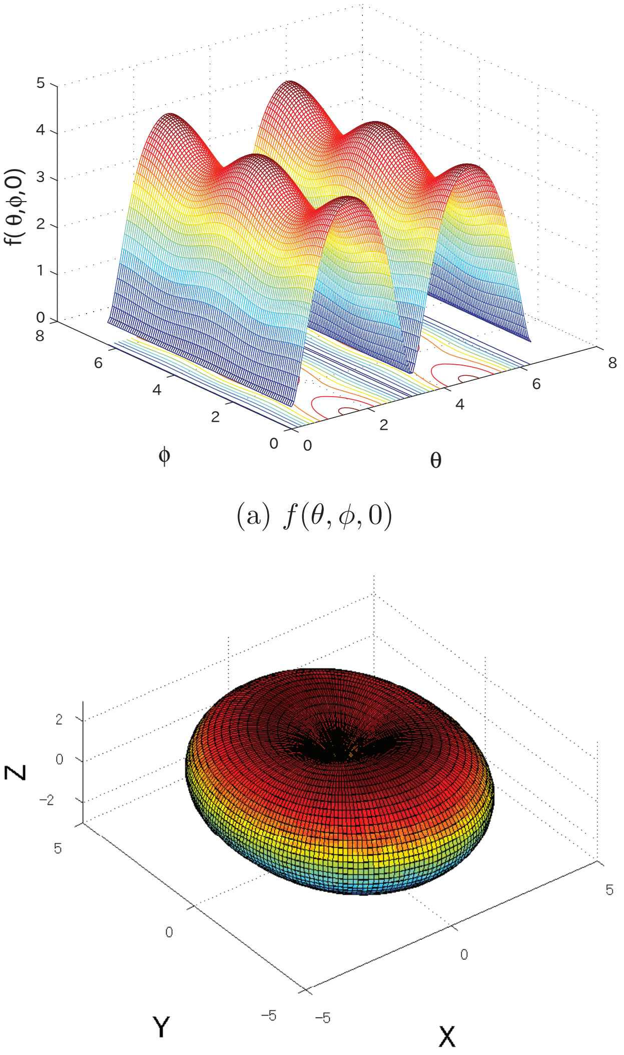
$$\begin{aligned} X(\theta, \phi, t) &= f(\theta, \phi, t)\sin\theta\sin\phi \\ Y(\theta, \phi, t) &= f(\theta, \phi, t)\sin\theta\cos\phi \\ Z(\theta, \phi, t) &= f(\theta, \phi, t)\cos\theta. \end{aligned} \tag{104}$$

The data points $v_i = (v_i^{[1]}, v_i^{[2]}, v_i^{[3]})$ are taken as 3-dimensional lattice points, equally spaced in each axis with the number $N_1 = 5,\ N_2 = 5,\ N_3 = 10$ in θ, ϕ, t axes respectively. Hence the total number of data is $N = 250$, and the function is sampled to generate the data $d_i = f(v_i) + \epsilon_i$ with ϵ_i being 10% additive white Gaussian noise.

A trivariate optimal smoothing spline $x(\theta, \phi, t)$ is constructed under the periodicity constraints on all the variables θ, ϕ, t with the periods $2\pi, 2\pi$ and $T (= 10)$ respectively, where the method described in Section 3 is used. The number of control points are set as $m_1 = 10, m_2 = 10, m_3 = 15$, hence the knot point intervals along the θ, ϕ, t axes are $2\pi/m_1 \approx 0.628 (= 1/\alpha_1)$, $2\pi/m_2 \approx 0.628 (= 1/\alpha_2)$ and $10/m_3 \approx 0.667 (= 1/\alpha_3)$, respectively. The weights w_i for approximation errors are set as $w_i = 1/N\ \forall i$.

Figure 5 (a) shows a set of $N_1 \times N_2 = 25$ noisy data sampled at $t = 0$ is shown and the generalized cross validation function computed based on 250 data is plotted in (b). The optimal value of the smoothing parameter is obtained as $\lambda^* = 6.3096 \times 10^{-6}$ (red asterisk in Figure 5 (b)).

We constructed trivariate optimal smoothing spline $x(\theta, \phi, t)$. The results are shown in Figure 6 as the 3D shapes of the red blood cell reconstructed in the $O - XYZ$ space by a polar coordinate transformation defined similarly

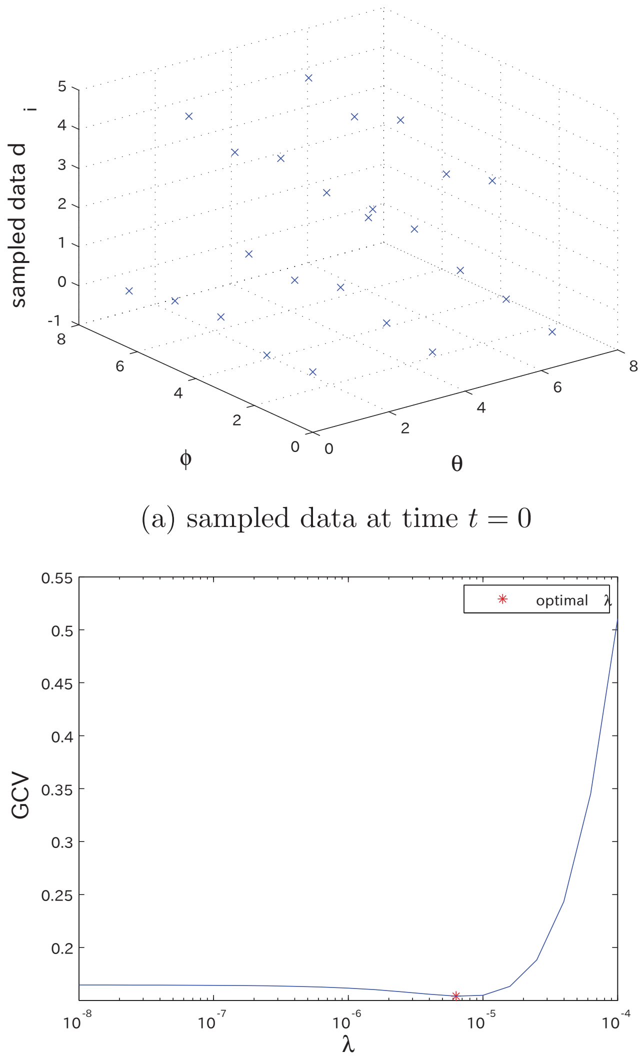


(a) $f(\theta, \phi, 0)$

(b) 3D shape of red blood cell at $t = 0$

Figure 4: True function $f(\theta, \phi, t)$ at time $t = 0$ (a), and the corresponding 3D shape of red blood cell.

as in (104). The shapes are plotted at the four time instants of continuous, periodic deforming motion reconstructed from $x(\theta, \phi, t)$. Also, by animation videos, we can observe that the 3D shapes are reconstructed for t $(0 \le t \le$

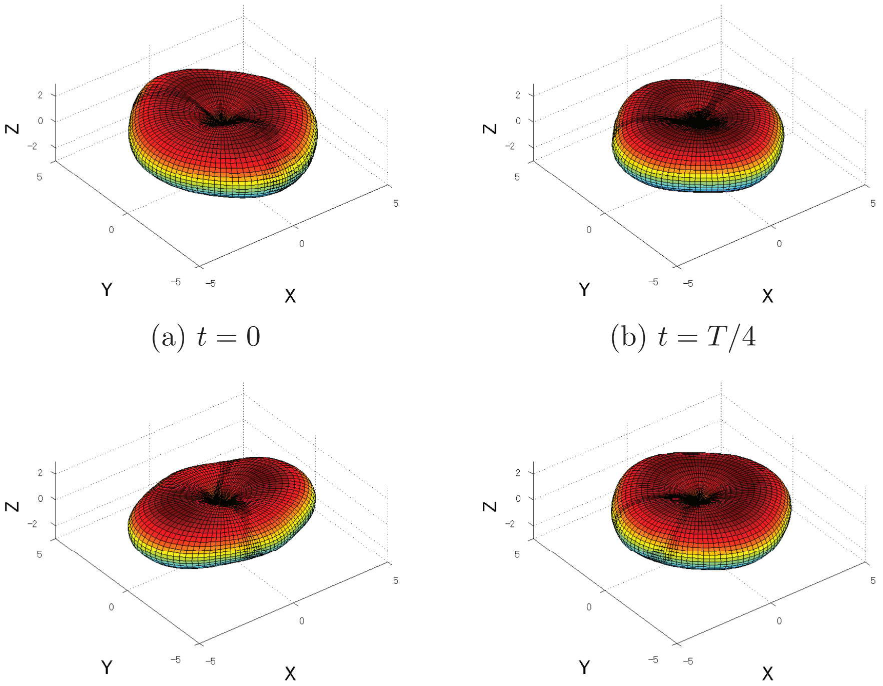


(a) sampled data at time $t = 0$

(b) generalized cross validation function

Figure 5: An example of sampled noisy data at time $t = 0$ (a), and computed generalized cross validation function $V(\lambda)$ (b).

10) fairly accurately although a small number of sampled data is used with noises.



(a) $t = 0$

(b) $t = T/4$

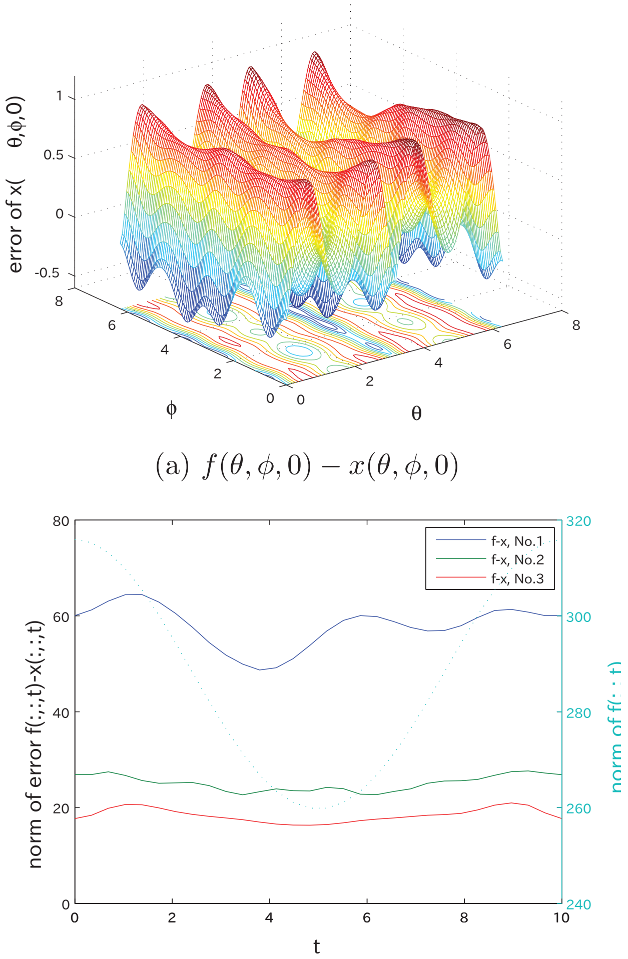
(c) $t = 2T/4$

(d) $t = 3T/4$

Figure 6: 3D shapes of red blood cell sampled at time $t = 0$, $T/4$, $2T/4$, $3T/4$ ($T = 10$) from its deforming motion reconstructed by optimal periodic smoothing spline $x(\theta, \phi, t)$.

We examined the approximation errors. Figure 7 (a) shows the error between the original function and constructed spline function $f(\theta, \phi, t) - x(\theta, \phi, t)$ at time $t = 0$, and error norms $\|f(\theta, \phi, t) - x(\theta, \phi, t)\|$ computed for each t are plotted in (b) (blue line), where matrix 2-norm is used. Also shown in (b) are the error norms for the case of increased number of data points with $N_1 = N_2 = 10, N_3 = 20$ ($N = 2000$, green line) and $N_1 = N_2 = 20, N_3 = 40$ ($N = 32000$, red line), and the norm of original function $f(\theta, \phi, t)$ (dotted line, right scale). Clearly increasing the number of data decreases approximation errors.

Figure 8 shows XY-plane profiles of 3D shapes of the red blood cell plotted at four time instants $t = 0, T/4, 2T/4$ and $t = 3T/4$ for two cases of data points: (a) $N_1 = N_2 = 5, N_3 = 10$ and (b) $N_1 = N_2 = 20, N_3 = 40$. The profiles obtained from the optimal spline $x(\theta, \phi, t)$ are shown in solid

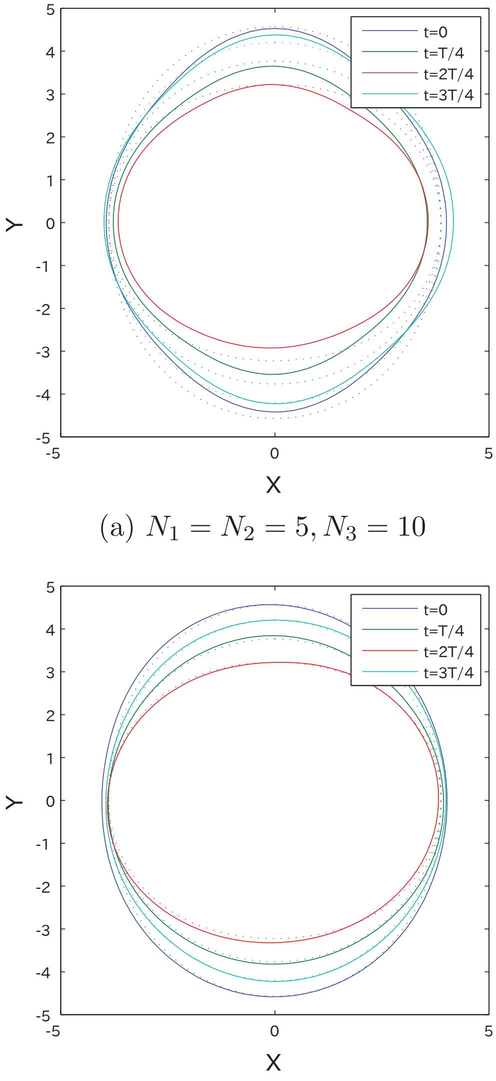


(a) $f(\theta, \phi, 0) - x(\theta, \phi, 0)$

(b) $\|f(\theta, \phi, t) - x(\theta, \phi, t)\|$ for each t

Figure 7: (a) Approximation error $f(\theta, \phi, t) - x(\theta, \phi, t)$ at time $t = 0$ and (b) the error norms evaluated for each t (blue line). Also shown in (b) are error norms by another experiments with $N_1 = N_2 = 10, N_3 = 20$ (green line) and $N_1 = N_2 = 20, N_3 = 40$ (red line) and norm of $f(\theta, \phi, t)$ (dotted line, right scale).

lines and those by original function $f(\theta, \phi, t)$ in dotted lines. We see that the profiles by splines agree quite well in (a) although the number of sampled data is relatively small, and agree very well in (b).



(a) $N_1 = N_2 = 5, N_3 = 10$

(b) $N_1 = N_2 = 20, N_3 = 40$

Figure 8: XY-plane profile of constructed 3D shape of the red blood cell plotted at time $t = 0, T/4, 2T/4, 3T/4$: (a) The case of $N_1 = N_2 = 5, N_3 = 10$ and (b) $N_1 = N_2 = 20, N_3 = 40$. Solid lines show profiles by spline $x(\theta, \phi, t)$ and dotted lines true profiles by $f(\theta, \phi, t)$.

6. Concluding remarks

We developed a method of constructing trivariate and n-variate optimal smoothing splines using normalized uniform B-spline as the basis functions. Trivariate case is described in detail and then extended to general case. Trivariate splines, in particular, are useful for modeling dynamic shape of deformable objects by using two variables for 3D shape and one for time evolution. The splines are constructed as a tensor product of B-splines, and an optimal smoothing spline problem is solved together with typical examples of constraints as periodicity and boundary conditions. Note that the splines are optimized for fixed knot points but the smoothing parameter will be adjusted by employing the generalized cross validation method. The problem is formulated as convex QP problem in such a way that, by converting n-dimensional control point array to a vector, Lagrange multiplier method or QP solver is applicable readily for computations of numerical solution. Accordingly, the periodicity and boundary constraints are established in terms of the control point vector. We demonstrated the usefulness of the method by two numerical examples, namely, vibration modeling of rectangular membrane and dynamic shape modeling of red blood cell. In either cases, we see that relatively small number of observation data with noises yield satisfactory results. Further studies are needed to derive algorithms for shape preserving splines and splines incorporating various types of other constraints.

References

[1] E. B. Ameur, D. Sbibih, and C. Leger, *New spline quasi-interpolant for fitting* 3*-D data on the sphere: applications to medical imaging*, IEEE Signal Processing Letters **14** (2007), no. 5, 333–336.

[2] A. A. Amini, Y. Chen, M. Elayyadi, and P. Radeva, *Tag surface reconstruction and tracking of myocardial beads from SPAMM-MRI with parametric B-spline surfaces*, IEEE Trans. Medical Imaging **20** (2001), no. 2, 94–103.

[3] I. J. Anderson, M. G. Cox, and J. C. Mason, *Tensor-product spline interpolation to data on or near a family of lines*, Numerical Algorithms **5** (1993), 193–204.

[4] L. Biagiotti and C. Melchiorri, *Trajectory planning for automatic machines and robots*, Springer, 2008.

[5] A. Blake and M. Isard, *Active contours*, Springer, 2000.

[6] C. de Boor, *A practical guide to splines*, revised edition, Springer-Verlag, New York, 2001.

[7] P. Brigger, J. Hoeg, and M. Unser, *B-spline snakes: A flexible tool for parametric contour detection*, IEEE Trans. Image Processing **9** (2000), no. 9, 1484–1496.

[8] M. Egerstedt and C. Martin, *Control theoretic splines: Optimal control, statistics and path planning*, Princeton University Press, Princeton, NJ, 2010.

[9] H. Fujioka, H. Kano, M. Egerstedt, and C. Martin, *Smoothing spline curves and surfaces for sampled data*, Int. J. of Innovative Computing, Information and Control **1** (2005), no. 3, 429–449.

[10] H. Fujioka and H. Kano, *Periodic smoothing spline surface and its application to dynamic contour modeling of wet material objects*, IEEE Trans. Systems, Man, and Cybernetics, Part A **39** (2009), no. 1, 251–261.

[11] H. Fujioka and H. Kano, *Recursive construction of optimal smoothing spline surfaces with constraints*, Preprints of The 18th IFAC World Congress, pp. 2278–2283, Milan, Italy, Aug. 28–Sept. 2, 2011.

[12] H. Fujioka and H. Kano, *Dynamic contour modeling of wet material objects by periodic smoothing splines*, Proc. of the 59th ISI World Statistics Congress (WSC), Special Topics Session on 'Two views of smoothing splines and related tools', pp. 1149–1154, Hong Kong, August 25–30, 2013.

[13] H. Fujioka, H. Kano, and C. Martin, *Constrained smoothing and interpolating spline surfaces using normalized uniform B-splines*, Communications in Information and Systems **14** (2014), no. 1, 23–56.

[14] C. Habermann and F. Kindermann, *Multidimensional spline interpolation: Theory and applications*, Computational Economics **30** (2007), no. 2, 153–169.

[15] M. Hosaka, *Modeling of curves and surfaces in CAD/CAM*, Springer-Verlag, 1992.

[16] F. Jaillet, B. Shariat, and D. Vandorpe, *Periodic B-spline surface skinning of anatomic shapes*, Proc. of the 9th Canadian Conf. on Computational Geometry, pp. 199–210, Kingston, Ontario, Canada, Aug. 11–14, 1997.

[17] H. Kano, H. Nakata, and C. Martin, *Optimal curve fitting and smoothing using normalized uniform B-splines: A tool for studying complex systems*, Applied Mathematics and Computation **169** (2005), no. 1, 96–128.

[18] H. Kano, H. Fujioka, M. Egerstedt, and C. F. Martin, *Optimal smoothing spline curves and contour synthesis*, Proc. of the 16th IFAC World Congress, Prague, Czech Republic, July 4–8, 2005.

[19] H. Kano, M. Egerstedt, H. Fujioka, S. Takahashi, and C. F. Martin, *Periodic smoothing spline*, Automatica **44** (2008), no. 1, 185–192.

[20] H. Kano, H. Fujioka, and C. F. Martin, *Optimal smoothing and interpolating splines with constraints*, Applied Mathematics and Computation **218** (2011), no. 5, 1831–1844.

[21] H. Kano and H. Fujioka, *Trivariate optimal smoothing splines with dynamic shape modeling of deforming object*, Preprints of the 19th IFAC World Congress, Cape Town, South Africa, August 24–29, pp. 9165–9170, 2014.

[22] A. Karlsson, J. He, J. Swartling, and S. Anderson, *Numerical simulations of light scattering by red blood cells*, IEEE Trans. Biomedical Engineering **52** (2005), no. 1, 13–18.

[23] R. Kress, *Numerical analysis*, Springer-Verlag, 1998.

[24] M. J. Lai and L. L. Schumaker, *Spline functions on triangulations*, Cambridge University Press, 2007.

[25] P. Lancaster and M. Tismenetsky, *The theory of matrices*, second edition, Academic Press, 1985.

[26] K-H Liang, T. Tjahjadi, and Y. Yee-Hong, *Bounded diffusion for multiscale edge detection using regularized cubic B-spline fitting*, IEEE Trans. Systems, Man, and Cybernetics, Part B **29** (1999), no. 2, 291–297.

[27] J. R. Magnus and H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics*, revised edition, John Wiley and Sons, 1999.

[28] T. Martin, E. Cohen, R. M. Kirby, *Volumetric parametrization and trivariate B-spline fitting using harmonic functions*, Computer Aided Geometric Design **26** (2009), 648–664.

[29] T. Peng and R. F. Murphy, *Image-derived, three-dimensional generative models of cellular organization*, Cytometry Part-A **79** (2011), no. 5, 383–391.

[30] C. Pozrikidis, *Numerical simulation of the flow-induced deformation of red blood cells*, Annals of Biomedical Engineering **31** (2003), 1194–1205.

[31] G. Wahba, *Spline models for observational data*, CBMS-NSF Regional Conference Series in Applied Mathematics **59**, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

[32] K. Yoshida and J. Watanabe, *Methods of mathematical physics* (in Japanese), Iwanami, 1969 (translated from L. Schwartz, Methodes mathematiques pour les sciences physiques, 1961).

School of Science and Engineering, Tokyo Denki University
Saitama 350-0394, Japan
E-mail address: `kano@mail.dendai.ac.jp`

Department of System Management
Fukuoka Institute of Technology
Fukuoka 811-0295, Japan
E-mail address: `fujioka@fit.ac.jp`

Received November 27, 2016
Accepted November 21, 2017