Hiroyuki Kano
Tokyo Denki University
Japan

Constrained Smoothing Splines
Optimal Spline Surfaces and Extensions

We would like to design optimal spline curves and surfaces, but, in general, there are various types of constraints that we need to take into consideration.
**Constrained Splines**

What types of constraints arise, and where?

(a) **isolated point constraints**
   - (initial condition, interval interpolation)
   - \( x(t_0) = \dot{x}(t_0) = \ddot{x}(t_0) = 0 \)
   - \( a < x(0.25) < b \)

(b) **constraints over interval**
   - (trajectory planning)
   - \( x(t) \geq c, \forall t \in [t_j, t_{j+1}] \)

(c) **periodicity**
   - (periodic curves)
   - \( x(t_0) = x(t_m) \)
   - \( \dot{x}(t_0) = \dot{x}(t_m) \)
   - \( \ddot{x}(t_0) = \ddot{x}(t_m) \)

(d) **integral value constraint**
   - (probability density function)
   - \( \int_{t_0}^{t_m} x(t) \, dt = c \)

(e) **monotonicity**
   - (probability distribution function)
   - \( \dot{x}(t) \geq 0 \)

(f) **convexity**
   - (CAD, shape design)
   - \( x^{(2)}(t) \leq 0 \)

Note: Periodic splines can be used to construct closed curves.
How can we formulate these constraints, equality or inequality, and incorporate to optimal splines?

Review of Optimal Splines

◆ We constitute splines by employing B-splines as the basis functions

\[ x(t) = \sum_{i=-k}^{m-1} \tau_i B_k(\alpha(t - t_i)) \]

Here \( B_k(\cdot) \) is a normalized uniform B-spline of degree \( k \) defined as

\[
B_k(t) = \begin{cases} 
N_{k,k}(t) & 0 \leq t < 1 \\
N_{k-1,k}(t-1) & 1 \leq t < 2 \\
\vdots & \vdots \\
N_{0,k}(t-k) & k \leq t < k + 1 \\
0 & t < 0 \ \text{or} \ t \geq k + 1 
\end{cases}
\]

\( N_{j,k}(t) \ (j = 0, 1, \cdots, k) \) are the basis elements constituting the B-spline
◆ There is a recursive algorithm for generating the basis elements \( N_{j,k}(t) \)

**De Boor Algorithm for generating** \( N_{j,k}(t) \) \((j = 0, 1, \ldots, k)\)

Let \( N_{0,0}(t) = 1 \) and compute for \( i = 0, 1, \ldots, k \)

\[
\begin{align*}
N_{0,i}(t) & = \frac{1-t}{i} N_{0,i-1}(t) \\
N_{j,i}(t) & = \frac{i-j+1}{i} N_{j-1,i-1}(t) + \frac{1+j-t}{i} N_{j,i-1}(t), j = 1, \ldots, i - 1 \\
N_{i,i}(t) & = \frac{i}{i} N_{i-1,i-1}(t).
\end{align*}
\]

◆ B-splines are normalized in the sense that

\[
\sum_{j=0}^{k} N_{j,k}(t) = 1, \quad 0 \leq t \leq 1
\]

And it holds that

\[
\int_{-\infty}^{\infty} B_k(t) dt = \int_{0}^{k+1} B_k(t) dt = 1
\]

---

**Optimal Design of Splines**

■ **Given data**

Let an interval \([t_0, t_m]\) and a set of \(N\) data points be given as

\[ D = \{(s_i; d_i) : s_i \in [t_0, t_m], \; d_i \in \mathbb{R}, \; i = 1, \ldots, N\} \]

■ **Smoothing splines**

The standard optimal smoothing spline problem is stated as follows.

**Problem 1:** Find the control point vector \( \tau \) minimizing the cost function

\[
J(\tau) = \lambda \int_{t_0}^{t_m} (x''(t))^2 dt + \sum_{i=1}^{N} w_i (x(s_i) - d_i)^2
\]

\[
\lambda \ (> 0) : \text{smoothing parameter} \\
w_i(0 < w_i \leq 1) : \text{weights for approximation errors}
\]

\[
\tau = \begin{bmatrix} \tau_{-k} & \tau_{-k+1} & \cdots & \tau_{m-1} \end{bmatrix}^T \in \mathbb{R}^M \ (M = m + k)
\]
The cost function was quadratic in $\tau$ as

**Cost function in Problem 1:**

$$J(\tau) = \lambda \int_{t_0}^{t_m} (x^{(2)}(t))^2 \, dt + \sum_{i=1}^{N} w_i (x(s_i) - d_i)^2$$

$$= \lambda \tau^T Q \tau + (B^T \tau - d)^T W (B^T \tau - d)$$

$$= \tau^T (\lambda Q + BWB^T) \tau - 2 \tau^T BWd + d^T Wd$$

Here

$$Q = \int_{t_0}^{t_m} \frac{d^2 b(t)}{dt^2} \frac{d^2 b^T(t)}{dt^2} \, dt$$

$$B = \begin{bmatrix} b(s_1) & b(s_2) & \cdots & b(s_N) \end{bmatrix}$$

$$b(t) = \begin{bmatrix} B_k(a(t-t_{k-1})) & B_k(a(t-t_{k+1})) & \cdots & B_k(a(t-t_{m-1})) \end{bmatrix}^T$$

$$W = \text{diag}\{w_1, w_2, \cdots, w_N\}$$

$$d = \begin{bmatrix} d_1 \ d_2 \ \cdots \ d_N \end{bmatrix}^T$$

**Summary of optimal solution (continued)**

Thus if there are no constraints, the optimal solution is obtained as follows.

**Optimal solution to Problem 1:**

Optimal $\tau$ is a solution of the following linear algebraic equation:

$$(\lambda Q + BWB^T) \tau = BWd$$

For convenience of later use, the cost function is rewritten as follows.

**Cost function in Problem 1:**

$$J(\tau) = \tau^T (\lambda Q + BWB^T) \tau - 2 \tau^T BWd + d^T Wd$$

$$\Rightarrow \quad J(\tau) = \frac{1}{2} \tau^T G \tau + g^T \tau$$

where

$$G = 2(\lambda Q + BWB^T), \quad g = -2BWd$$

**Note:** In the last expression, the constant term is dropped for simplicity.
Realization of constraints

Previous constraints can be incorporated as linear constraints on the control point vector \( \tau \). For example, for the cubic case of \( k=3 \),

**Ex1:** initial condition (equality constraints)

\[
x(t_0) = x_0, \quad \dot{x}(t_0) = \dot{x}_0, \quad x(t_0) = x_0 \quad \Rightarrow \quad A\tau = d
\]

\[
A = \begin{bmatrix}
\frac{t_0}{6} & \frac{t_0^2}{2} & \frac{t_0^3}{6} & \cdots & 0 \\
\frac{t_0^2}{2} & \frac{t_0^3}{3} & \frac{t_0^4}{4} & \cdots & 0 \\
\frac{t_0^3}{6} & \frac{t_0^4}{4} & \frac{t_0^5}{5} & \cdots & 0 \\
\frac{t_0^4}{4} & \frac{t_0^5}{5} & \frac{t_0^6}{6} & \cdots & 0
\end{bmatrix}, \quad d = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ 0 \end{bmatrix}
\]

**Ex2:** interval inequality constraints

\[
x(t) \leq c, \quad \forall t \in [t_j, t_{j+1}] \quad \Rightarrow \quad E_j \tau \leq c_4
\]

\[
c_4 = [c \ c \ c \ c]^T \in \mathbb{R}^4, \quad E_j = [I_4 \ 0] \in \mathbb{R}^{4 \times 4}
\]

**Ex3:** integral value constraints

\[
\int_{t_0}^{t_m} x(t) dt = c \quad \Rightarrow \quad a^T \tau = c
\]

\[
a = \frac{1}{24\alpha}[1 \ 12 \ 23 \ 24 \ \cdots \ 24 \ 23 \ 12 \ 1]^T \in \mathbb{R}^M
\]

Outline of derivation

Recall that, when \( k=3 \) (cubic spline), the spline \( x(t) \) is expressed in terms of the control point vector \( \tau \) as

\[
x(t) = \sum_{i=-3}^{m-1} \tau_i B_3(\alpha(t-t_i)) = b^T(t) \tau
\]

where \( b(t) \) is a vector of time-shifted B-splines defined by

\[
b(t) = \left[ B_3(\alpha(t-t_{-3})) \ B_3(\alpha(t-t_{-2})) \ \cdots \ B_3(\alpha(t-t_{m-1})) \right]^T
\]

■ **pointwise constraints on function value**

Any constraint at a particular point, say at \( t=s \), can be realized as

\[
x(s) = c \Rightarrow b^T(s) \tau = c \Rightarrow a^T \tau = c
\]

\[
x(s) \geq c \Rightarrow b^T(s) \tau \geq c \Rightarrow a^T \tau \geq c
\]

etc., where \( a = b(s) \)

In general, we need to compute \( b(t) \) for \( t=s \).
**Initial and terminal conditions**

For example, in the case of initial condition, the vector $a$ is computed as

\[
a = b(t_0)
\]

\[
= \begin{bmatrix}
B_3(\alpha(t_0 - t_{-3})) & B_3(\alpha(t_0 - t_{-2})) & \cdots & B_3(\alpha(t_0 - t_{m-1}))
\end{bmatrix}^T
\]

\[
= \begin{bmatrix}
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0
\end{bmatrix}^T
\]

since the knot point interval is uniform with

\[t_{i+1} - t_i = \frac{1}{\alpha}\] hence \[t_0 - t_i = \frac{-i}{\alpha}\] and \[\alpha(t_0 - t_i) = -i\]

Similarly, in the case of terminal condition, we have

\[
a = b(t_m)
\]

\[
= \begin{bmatrix}
B_3(\alpha(t_m - t_{-3})) & B_3(\alpha(t_m - t_{-2})) & \cdots & B_3(\alpha(t_m - t_{m-1}))
\end{bmatrix}^T
\]

\[
= \begin{bmatrix}
0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{bmatrix}^T
\]

**Constraints on integral value**

An constraint on the integral value of $x(t)$ over $[t_0, t_m]$ can be realized as

\[
\int_{t_0}^{t_m} x(t)dt = c \Rightarrow \int_{t_0}^{t_m} b^T(t)\tau dt = c \Rightarrow a^T\tau = c
\]

\[
\int_{t_0}^{t_m} x(t)dt \geq c \Rightarrow \int_{t_0}^{t_m} b^T(t)\tau dt \geq c \Rightarrow a^T\tau \geq c
\]

e etc., where \[a = \int_{t_0}^{t_m} b(t)dt\] , we can compute the vector $a$ as

\[
a = \int_{t_0}^{t_m} b(t)dt
\]

\[
= \int_{t_0}^{t_m} \begin{bmatrix}
B_3(\alpha(t - t_{-3})) & B_3(\alpha(t - t_{-2})) & \cdots & B_3(\alpha(t - t_{m-1}))
\end{bmatrix}^T dt
\]

\[
= \begin{bmatrix}
\frac{1}{2\alpha} & \frac{12}{24\alpha} & \frac{23}{24\alpha} & \frac{1}{\alpha} & \cdots & \frac{23}{24\alpha} & \frac{12}{24\alpha} & \frac{1}{2\alpha}
\end{bmatrix}^T
\]

Note that the integration interval can be taken as arbitrary knot point interval as $[t_i, t_j]$
Constraints on Derivatives

Constraints on derivatives are needed in many cases as

(i) **trajectory planning**: constraints on velocity and acceleration, e.g.
\[ c_1 \leq x^{(1)}(t) \leq c_2, \quad c_3 \leq x^{(2)}(t) \leq c_4 \quad \forall t \in [t_0, t_m] \]

(ii) **monotone splines**: constraints on the first derivative, e.g.
\[ x^{(1)}(t) \geq 0 \quad \forall t \in [t_0, t_m] \]

(iii) **convex splines**: constraints on the second derivative, e.g.
\[ x^{(2)}(t) \geq 0 \quad \forall t \in [t_0, t_m] \]

An import point in these constraints is that the inequality must hold for all \( t \) in the interval \([t_0, t_m]\), not at specific time.

Constraints on Derivatives (continued)

If the constraint is imposed on some point \( s \), then the problem is easy to solve.

Again noting that
\[ x(t) = b^T(t)\tau \]
\[ b(t) = \begin{bmatrix} B_k(o(t-t_{-k})) & B_k(o(t-t_{-k+1})) & \cdots & B_k(o(t-t_{m-1})) \end{bmatrix}^T \]

an **constraint at an isolated point**, say for \( t = s \in [t_0, t_m] \), and
\[ c_1 \leq x^{(1)}(s) \leq c_2 \]
is realized as the constraint on the control point vector \( \tau \) as
\[ c_1 \leq a^T\tau \leq c_2 \]
where \( a = b^{(1)}(s) \) and similarly for higher derivatives.
■ Constraints on Derivatives (continued)

Next we consider constraints over interval, for example

\[ x^{(l)}(t) \geq c \quad \forall t \in [t_0, t_m] \quad \text{for} \quad x(t) = \sum_{i=-k}^{m-1} \tau_i B_k(\alpha(t-t_i)) \]

The following result holds for derivatives of \( x(t) \).

**Lemma:** The \( l \)-th derivative of spline \( x(t) \) of degree \( k \) is expressed in the same form \( x(t) \) as

\[ x^{(l)}(t) = \sum_{i=-k}^{m-1} \tau_i^{(l)} B_{k-l}(\alpha(t-t_i)), \quad l = 1, 2, \ldots \]

where the new control points \( \tau_i^{(l)} \) are obtained recursively as a difference of the original control points as

\[ \tau_i^{(l)} = \alpha (\tau_i^{(l-1)} - \tau_{i-1}^{(l-1)}), \quad \tau_0^{(0)} = \tau_i \]

**Note:** In terms of the original control points, we get

\[
\begin{align*}
\tau_i^{(1)} &= \alpha (\tau_i - \tau_{i-1}) \\
\tau_i^{(2)} &= \alpha (\tau_i^{(1)} - \tau_{i-1}^{(1)}) = \alpha^2 (\tau_i - 2\tau_{i-1} + \tau_{i-2}) 
\end{align*}
\]

■ Constraints on Derivatives (continued)

We have the following sufficient condition for constraints over interval.

**Lemma 1:** (constraints over interval)

Regarding the \( l \)-th order derivative of splines \( x^{(l)}(t) \), it holds that

\[ \tau_i^{(l)} \geq c \quad \forall i \quad \Rightarrow \quad x^{(l)}(t) \geq c \quad \forall t \in [t_0, t_m] \]

**(Proof):** In general, when \( \tau_i = 1 \quad \forall i \), then it holds that

\[ x(t) = \sum_{i=-k}^{m-1} \tau_i B_k(\alpha(t-t_i)) = \sum_{i=-k}^{m-1} B_k(\alpha(t-t_i)) \equiv 1 \quad t \in [t_0, t_m] \]

Noting that this holds for any \( k \) and that B-splines are all nonnegative,

\[
\begin{align*}
x^{(l)}(t) &= \sum_{i=-k(l)}^{m-1} \tau_i^{(l)} B_{k-l}(\alpha(t-t_i)) \geq \sum_{i=-k(l)}^{m-1} c B_{k-l}(\alpha(t-t_i)) \\
&= c \sum_{i=-k(l)}^{m-1} B_{k-l}(\alpha(t-t_i)) = c \quad \text{(QED)}
\end{align*}
\]
Constraints on Derivatives (continued)

Two cases are considered in more details for the assertion in Lemma 1.

\[ \tau_i^{(l)} \geq c \ \forall i \implies x_i^{(l)}(t) \geq c \ \forall t \in [t_0, t_m] \]

(i) First derivative: Since \( \tau_i^{(1)} = \alpha (\tau_i - \tau_{i-1}) \), this is written as

\[ \alpha(\tau_i - \tau_{i-1}) \geq c \ \forall i \implies x^{(1)}(t) \geq c \ \forall t \in [t_0, t_m] \]

The condition on the left is written more compactly as

\[ \alpha \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ \vdots \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \tau_{-k} \\ \tau_{-k+1} \\ \vdots \\ \tau_{m-1} \end{bmatrix} \geq \begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \]

i.e. where

or, in terms of the control point vector \( \tau \), in the form of

\[ A\tau \geq c \]

where \( A = \alpha D^{(1)}, \ c = c_{M-1} \)

\[ D^{(1)} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ \vdots \\ -1 & 1 \end{bmatrix} : (M-1) \times (M-1) \quad c_{M-1} = \begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} : (M-1) \times 1 \]

(ii) Second derivative: Since \( \tau_i^{(2)} = \alpha^2 (\tau_i - 2\tau_{i-1} + \tau_{i-2}) \), this is written as

\[ \alpha^2 (\tau_i - 2\tau_{i-1} + \tau_{i-2}) \geq c \ \forall i \implies x^{(2)}(t) \geq c \ \forall t \in [t_0, t_m] \]

The condition on the left is written as

\[ \alpha^2 \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ \vdots & \ddots & \ddots \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} \tau_{-h} \\ \tau_{-h+1} \\ \vdots \\ \tau_{m-1} \end{bmatrix} \geq \begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \]

and hence, in the form of

\[ A\tau \geq c \]

where \( A = \alpha D^{(2)}, \ c = c_{M-2} \)

\[ D^{(2)} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ \vdots & \ddots & \ddots \\ 1 & -2 & 1 \end{bmatrix} : (M-2) \times (M-2) \]
Constraints on Derivatives (continued)

Summary:

**Corollary 1:** (constraints over interval)

(i) If $\tau$ satisfies $\alpha D^{(1)} \tau \geq c_{M-1}$, then $x^{(1)}(t) \geq c \forall t \in [t_0, t_m]$

(ii) If $\tau$ satisfies $\alpha^2 D^{(2)} \tau \geq c_{M-2}$, then $x^{(2)}(t) \geq c \forall t \in [t_0, t_m]$

Note:

(i) The above inequality “$\geq$” is readily replaced by “$\leq$”

(ii) The inequality can be imposed to arbitrary knot point interval as $x^{(l)}(t) \geq c \forall t \in [t_j, t_{j+1}]$

(iii) The higher derivatives can be dealt with similarly. In fact if $D^{(1)} = \Delta_{M-1}$, then $D^{(2)} = \Delta_{M-2}D^{(1)}$

Formulation of Constrained Splines

Combination of any constraints can be introduced to optimal splines, as long as they are consistent.

**The case of equality constraints**

In particular, if the constraints are all equality, and written as $A\tau = d$

then the process of deriving the optimal solution is straightforward.

**Problem 2:** Minimize the cost function with respect to $\tau$

$$J(\tau) = \frac{1}{2} \tau^T G \tau + g^T \tau \quad (\tau \in \mathbb{R}^M)$$

subject to the equality constraint $A\tau = d$. 
The case of equality constraints (continued)

We can employ the Lagrange multiplier method.

Let the Lagrange function be

\[ L(\tau, \mu) = J(\tau) + \mu^T(A\tau - d) = \frac{1}{2}\tau^T G\tau + g^T \tau + \mu^T(A\tau - d) \]

(\( \mu \): M-dimensional Lagrange multiplier)

Then the necessary condition for optimality is that the gradients vanish.

\[ \nabla_\tau L(\tau, \mu) = G\tau + g + A^T \mu = 0 \]
\[ \nabla_\mu L(\tau, \mu) = A\tau - d = 0 \]

which is written as

\[
\begin{bmatrix}
G & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
\tau \\
\mu
\end{bmatrix}
=
\begin{bmatrix}
-g \\
d
\end{bmatrix}
\]

The case of inequality and/or equality constraints

If the constraints include inequalities, then the problem is the so-called Quadratic Programming Problem (QP), and an efficient software is available for numerical solution, e.g. quadprog in MATLAB.

Our problem is

\[ \text{cost function: } J(\tau) = \frac{1}{2}\tau^T G\tau + g^T \tau \quad + \quad \text{constraints: } A\tau = d, \quad E\tau \leq c_4 \quad \text{etc.} \]

Thus the general form is written as follows.

**Problem 3:** (QP Problem) Minimize the cost function with respect to \( \tau \)

\[ J(\tau) = \frac{1}{2}\tau^T G\tau + g^T \tau \quad (\tau \in \mathbb{R}^M) \]

subject to the constraints of the form

\[ A\tau = d, \quad f_1 \leq E\tau \leq f_2, \]

for some matrices and vectors of appropriate dimensions.
Numerical Examples

■ Trajectory planning

Plan $x(t)$ to pass through the following intervals at 3 time instants:

- $0.3 \leq x(0.25) \leq 0.5$
- $0.1 \leq x(0.5) \leq 0.3$
- $0.8 \leq x(0.8) \leq 1.0$

- Parameters
  - $k = 5$, $\alpha = 20$, $m = 20$, $N = 3$
  - $\lambda = 10^{-5}$, $w_i = 1/N$

- Constraints
  - $|x^{(1)}(t)| \leq 2| x^{(2)}(t) | \leq 20$, $\forall t \in [0, 1]$

(a) Planned trajectory $x(t)$

(b) Acceleration $x^{(2)}(t)$

■ Approximation of probability density function

Histogram of 100 Gaussian random numbers

Constrained smoothing splines for histogram from Gaussian probability density function

Pointwise constraints:
- $x(-5) = 0$, $x(5) = 0$
- $x(t) \geq 0$, $\forall t \in [-5, +5]$

Integral constraint:
- $\int_{-5}^{+5} x(t) dt = 1$
Approximation of probability distribution function

The number of data used for constructing splines is (only) five, obtained by sampling a Gaussian probability distribution function. By the constraints, we get remarkably good approximation.

(a) Equally spaced data points

(b) Randomly spaced data points

Constraints

\[ x(-5) = 0, \ x(5) = 1, \ x^{(1)}(t) \geq 0 \ \forall t \in [-5, 5] \]

Approximation of discontinuous function

We tested the performance of constrained smoothing splines for approximating a discontinuous function \( f(t) \):

Sampled function:

\[ f(t) = \begin{cases} 
1 & 0 \leq t \leq 1 \\
0 & \text{otherwise}
\end{cases} \]

Constraints over interval:

\[ 0 \leq x(t) \leq 1, \ \forall t \in [-1, +2] \]

Gibbs phenomenon suppressed
References

■ spline curve


■ spline surface

◆ H. Fujioka and H. Kano, Recursive Construction of Optimal Smoothing Spline Surfaces with Constraints, The 18th IFAC World Congress, to be presented, Milan, Italy, Aug. 28 - Sept. 2, 2011.

Optimal Spline Surfaces and Extensions

➢ Introduction
➢ Optimal Smoothing Spline Surfaces
➢ Periodic Smoothing Spline Surfaces
Introduction

- Advantages of using B-spline functions

Computational feasibilities
• Using B-splines as the basis functions yields extremely simple algorithms for designing curves and surfaces.

Dimensional extendability
• The approach using B-splines enables us to extend the results for one-dimensional case (i.e. the case of curves) to two dimensional case (i.e. the case of surfaces) and to even higher dimensions.

- Our purpose here

• We develop the design method of optimal smoothing spline surfaces using normalized uniform B-splines.
• We extend the design and analysis method to the case of periodic splines, and the results are applied to the problem of dynamic contour modeling.

First we show how to construct spline surface, and describe the problem of optimal smoothing spline surfaces.
Optimal Smoothing Spline Surfaces

■ Spline Surface

We construct spline surfaces (bivariate splines) using product B-splines as the basis functions as

\[ x(s, t) = \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} \tau_{i,j} B_k(\alpha(s - s_i)) B_k(\beta(t - t_j)) \]

where,

- \( B_k(\cdot) \) : normalized uniform B-splines with degree \( k \) (\( k = 3 \)).
- \( \alpha, \beta \) : positive constants.
- \( s_i, t_j \) : equally-spaced knot points both in \( s \) and \( t \) directions:
  \[ s_i - s_{i-1} = \frac{1}{\alpha} \quad t_j - t_{j-1} = \frac{1}{\beta} \]
- \( \tau_{i,j} \) : weighting coefficient called control point.
- \( m_1, m_2 (> 2) \) : integers.

Choosing appropriate \( \tau_{i,j} \), \( x(s, t) \) can represent arbitrary spline surface on the rectangular domain \( S = [s_0, s_{m_1}] \times [t_0, t_{m_2}] \).
Suppose that a set of spatial data $D = \{(u_i, v_i, d_i) : (u_i, v_i) \in S, \; d_i \in \mathbb{R}, \; i = 1, 2, \ldots, N\}$ be given.

\[ x(s, t) \]

\[ u_i \]

\[ v_i \]

\[ d_i \]

\[ d_1 \]

\[ d_N \]

\[ s \]

\[ t \]

\[ \nabla^2 : \text{the Laplacian operator} \]

\[ \nabla^2 = \frac{\partial^2}{\partial s^2} + \frac{\partial^2}{\partial t^2} \]

\[ \tau = [\tau_{i,j}] \in \mathbb{R}^{M_1 \times M_2} \] is the control point matrix

\[ \tau = \begin{bmatrix}
\tau_{-k,-k} & \tau_{-k,-k+1} & \cdots & \tau_{-k,m_2-1} \\
\tau_{-k+1,-k} & \tau_{-k+1,-k+1} & \cdots & \tau_{-k+1,m_2-1} \\
\vdots & \vdots & \ddots & \vdots \\
\tau_{m_1-1,-k} & \tau_{m_1-1,-k+1} & \cdots & \tau_{m_1-1,m_2-1}
\end{bmatrix} \]

\[ M_1 = m_1 + k = m_1 + 3 \]

\[ M_2 = m_2 + k = m_2 + 3 \]

**Problem 1:** Find the control point matrix $\tau$ minimizing the cost function

\[ J(\tau) = \lambda \int_{I_1} \int_{I_2} (\nabla^2 x(s, t))^2 \, ds \, dt + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_{ij} (x(u_i, v_j) - d_{ij})^2 \]

where $I_1 = (s_0, s_{m_1})$, and $I_2 = (t_0, t_{m_2})$,

$\lambda (> 0)$ : smoothing parameter

$w_{ij}$ (0 < $w_{ij}$ ≤ 1): weights for approximation errors

Note: For simplicity, we assume cubic splines, i.e. $k=3$
Smoothing Spline Surface (continued)

$$J(\tau) = \lambda \int_{I_1} \int_{I_2} (\nabla^2 x(s, t))^2 dsdt + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_{ij} (x(u_i, v_j) - d_{ij})^2$$

**Remark:**
As in the case of curves, the first term is for the smoothness of the resulting curve, and the second term is for goodness of fit to the given data.

The smoothing parameter $\lambda$ adjusts the weight between the two terms.

The optimal solution is derived by introducing the concepts of vec-function and Kronecker product, which we present in the next few slides.

For solving this problem, we need the concept of vec-function and Kronecker product.
Vec-function and Kronecker product

For a matrix $A \in \mathbb{R}^{m \times n}$ with
\[ A = [a_1 \ a_2 \ldots \ a_n], \ a_i \in \mathbb{R}^m \]
\text{vec} \ A \ is \ a \ vector \ obtained \ by \ simply \ stacking \ each \ columns \ of \ A \ as
\[
\text{vec} \ A = \begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_n
\end{bmatrix} \quad (\mathbb{R}^{mn})
\]

For matrices $A = [a_{ij}] \in \mathbb{R}^{m_1 \times n_1}$ and $B = [b_{ij}] \in \mathbb{R}^{m_2 \times n_2}$, The Kronecker product of A and B, denoted by $A \otimes B$, is defined by
\[
A \otimes B = \begin{bmatrix}
a_{11}B & \cdots & a_{1n_1}B \\
\vdots & \ddots & \vdots \\
a_{m_11}B & \cdots & a_{m_1n_1}B
\end{bmatrix} \in \mathbb{R}^{m_1m_2 \times n_1n_2}
\]

Examples

\[
A = \begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{bmatrix} \quad B = \begin{bmatrix}
1 & 0 \\
0 & 2
\end{bmatrix}
\]

Then
\[
\text{vec} \ A = \begin{bmatrix}
1 \\
4 \\
2 \\
5 \\
3 \\
6
\end{bmatrix}
\]
\[
A \otimes B = \begin{bmatrix}
B & 2B & 3B \\
4B & 5B & 6B
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 2 & 0 & 3 & 0 \\
0 & 2 & 0 & 4 & 0 & 6 \\
4 & 0 & 5 & 0 & 6 & 0 \\
0 & 8 & 0 & 10 & 0 & 12
\end{bmatrix}
\]
Thus, $A \otimes B$ is a large matrix obtained by multiplying each element of $A$ matrix by $B$ matrix. Note that $A$ and $B$ can be matrices of any size.

### Useful properties

$\text{vec}(\alpha A + \beta B) = \alpha \cdot \text{vec}A + \beta \cdot \text{vec}B$

$(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha (A \otimes B)$

$(A + B) \otimes C = A \otimes C + B \otimes C$

$A \otimes (B + C) = A \otimes B + A \otimes C$

$A \otimes (B \otimes C) = (A \otimes B) \otimes C$

$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

$(A \otimes B)^T = A^T \otimes B^T$

$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$

$\text{vec}(AXB) = (B^T \otimes A)\text{vec}X$


### Useful properties (continued)

Let $A$ and $B$ be square matrices of any size $nxn$ and $mxm$, respectively, and their eigenvalues be

$A : \lambda_i, \ i = 1, 2, \ldots, n \quad B : \mu_i, \ i = 1, 2, \ldots, m$

Then the eigenvalues of their Kronecker product are mn numbers

$A \otimes B : \lambda_i \mu_j, \ i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$

**Lemma 1**

If $A$ and $B$ are positive-definite (resp., nonnegative-definite) matrices, then so is the Kronecker product $A \otimes B$

(Proof)

$(A \otimes B)^T = A^T \otimes B^T = A \otimes B \Rightarrow \text{symmetric}$

$\text{e.v.'s of } A \otimes B = \lambda_i \mu_j > 0 \ (\geq 0) \Rightarrow A > 0 \ (\text{resp. } A \geq 0)$
Solution of Sylvester (or Lyapunov) equation by Kronecker product

They can be used in many applications, e.g. to solve matrix equation. Consider the following linear matrix equation in $X \in \mathbb{R}^{n \times m}$

$$AX + XB = C$$

$A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, $C \in \mathbb{R}^{n \times m}$

This types of equation is called **Sylvester equation**, and solved as follows.

\[
\begin{align*}
AX + XB &= C \\
\Rightarrow \quad \text{vec}(AX + XB) &= \text{vec} C \\
\Rightarrow \quad \text{vec}(AX) + \text{vec}(XB) &= \text{vec} C \\
\Rightarrow \quad \text{vec}(AXI_m) + \text{vec}(I_nXB) &= \text{vec} C \\
\Rightarrow \quad (I_m \otimes A)\text{vec} X + (B^T \otimes I_n)\text{vec} X &= \text{vec} C \\
\text{since, in general,} \quad \text{vec}(AXB) &= (B^T \otimes A)\text{vec} X \\
\Rightarrow \quad (I_m \otimes A + B^T \otimes I_n)\text{vec} X &= \text{vec} C
\end{align*}
\]

This is a standard form of linear algebraic equation as

$$Ax = c$$

with $A = I_m \otimes A + B^T \otimes I_n$, $x = \text{vec} X$, $c = \text{vec} C$

Outline of Derivation of Optimal Solution

Expression of $x(s,t)$ by vec-function and Kronecker product

First note that $x(s,t)$ is rewritten as

\[
\begin{align*}
x(s,t) &= \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} \tau_{i,j} B_k(\alpha(s - s_i))B_k(\beta(t - t_j)) \\
&= b_1(s)^T \tau b_2(t)
\end{align*}
\]

where

\[
\begin{align*}
b_1(s) &= [B_k(\alpha(s - s_{-k})) \ B_k(\alpha(s - s_{-k+1})) \cdots \ B_k(\alpha(s - s_{m_1-1})]^T \\
b_2(t) &= [B_k(\beta(t - t_{-k})) \ B_k(\beta(t - t_{-k+1})) \cdots \ B_k(\beta(t - t_{m_2-1})]^T
\end{align*}
\]

Here we would like to express $x(s,t)$ in terms of the vectorized $\tau$, namely by $\text{vec} \ \tau$, as in the case of spline curves.
Then we get the expression

\[ x(s, t) = \text{vec} \ x(s, t) = \text{vec} \ (b_1(s)^T \tau b_2(t)) \]

\[ = (b_2^T(t) \otimes b_1^T(s)) \text{vec} \ \tau \]

\[ \text{vec}(AXB) = (B^T \otimes A) \text{vec} \ X \]

Using the property of Kronecker product \((A \otimes B)^T = A^T \otimes B^T\) we get the desired expression

\[ x(s, t) = (b_2(t) \otimes b_1(s))^T \hat{\tau} \]

where

\[ \hat{\tau} = \text{vec} \ \tau \]

---

**Expression of \(x(s,t)\):** The spline surface \(x(s,t)\) is expressed in terms of vec-function and Kronecker product as

\[ x(s, t) = (b_2(t) \otimes b_1(s))^T \hat{\tau} \quad \text{where} \quad \hat{\tau} = \text{vec} \ \tau \]

**Note:** In the case of curves, \(x(t)\) is expressed as \(x(t) = b^T(t)\hat{\tau}\)

**Expression of cost function \(J(\tau)\) in matrix \(\tau\)**

In the cost function

\[ J(\tau) = \lambda \int_{l_1} \int_{l_2} (\nabla^2 x(s, t))^2 \ ds dt + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \ w_{ij} (x(u_i, v_j) - d_{ij})^2 \]

the first term is obtained as

\[ \int_{l_1} \int_{l_2} (\nabla^2 x(s, t))^2 \ ds dt = \hat{\tau}^T Q \hat{\tau} \]

where

\[ Q = \int_{l_1} \int_{l_2} (\nabla^2 (b_2(t) \otimes b_1(s))) (\nabla^2 (b_2(t) \otimes b_1(s)))^T \ ds dt \]
Expression of cost function $J(\tau)$ in matrix $\tau$ (continued)

On the other hand, we can show that the second term becomes

$$\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_{ij} (x(u_i, v_j) - d_{ij})^2 = ((B_2 \otimes B_1)^T \hat{\tau} - d)^T W ((B_2 \otimes B_1)^T \hat{\tau} - d)$$

where

$$\hat{B}_1 = \begin{bmatrix} b_1(u_1) & b_1(u_2) & \cdots & b_1(u_{N_1}) \end{bmatrix}$$

$$\hat{B}_2 = \begin{bmatrix} b_2(v_1) & b_2(v_2) & \cdots & b_2(v_{N_2}) \end{bmatrix}$$

$$W = \text{diag} \{ w_{11}, w_{21}, \cdots, w_{N_1 1}, \cdots, w_{1 N_2}, w_{2 N_2}, \cdots, w_{N_1 N_2} \}$$

$$d = \begin{bmatrix} d_{11}, d_{21}, \cdots, d_{N_1 1}, \cdots, d_{1 N_2}, d_{2 N_2}, \cdots, d_{N_1 N_2} \end{bmatrix}^T$$

Thus the second term is expressed as

$$\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_{ij} (x(u_i, v_j) - d_{ij})^2 = (\Gamma^T \hat{\tau} - d)^T W (\Gamma^T \hat{\tau} - d)$$

where

$$\Gamma = \hat{B}_2 \otimes \hat{B}_1$$

Optimal Solution

We summarize the expression for the cost function $J(\tau)$

$\textbf{Expression of } J(\tau) :$ The cost function $J(\tau)$ in Problem 1 is given by

$$J(\hat{\tau}) = \lambda \hat{\tau}^T Q \hat{\tau} + (\Gamma^T \hat{\tau} - d)^T W (\Gamma^T \hat{\tau} - d)$$

where $\hat{\tau} = \text{vec} \ \tau$, $\Gamma = \hat{B}_2 \otimes \hat{B}_1$, and other symbols are defined previously.

Thus we obtain the optimal solution as follows.

$\textbf{Optimal solution to Problem 1:}$

Optimal $\tau$ is a solution of the following linear algebraic equation:

$$(\lambda Q + \Gamma W \Gamma^T) \hat{\tau} = \Gamma W d$$

Proof: $\nabla_{\hat{\tau}} J(\hat{\tau}) = 2\lambda Q \hat{\tau} + 2W (\Gamma^T \hat{\tau} - d) = 0$
Periodic Smoothing Spline Surfaces

In Problem 1, we impose a periodicity constraint.

**Problem 2**  \((x(s,t)\text{ periodic in } t)\)

\[
\min_{\tau \in \mathbb{R}^{M_1 \times M_2}} J(\tau),
\]

\[
J(\tau) = \lambda \int_{I_1} \int_{I_2} (\nabla^2 x(s,t))^2 \, ds \, dt + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_{ij} (x(u_i, v_j) - d_{ij})^2
\]

subject to

\[
\frac{\partial}{\partial t} x(s, t_0) = \frac{\partial}{\partial t} x(s, t_m) \quad \forall s \in [s_0, s_m], \quad l = 0, 1, \cdots, k - 1
\]

This periodicity is in \(t\)-direction.

\[
x(s, t_0) = x(s, t_m), \quad \frac{\partial}{\partial t} x(s, t_0) = \frac{\partial}{\partial t} x(s, t_m), \quad \cdots \quad \forall s \in [s_0, s_m]
\]

The periodicity constraints are satisfied as follows.

**Proposition 1**  The constraint is satisfied if and only if

\[
\tau_{ij} = \tau_{i, m_2+j} \quad \forall i = -k, -k + 1, \cdots, m_1 - 1
\]

holds for \(j = -k, -k + 1, \cdots, -1\)

This condition is expressed in terms of \(\tilde{\tau}\) as linear constraint:

\[
G \tilde{\tau} = 0
\]

where

\[
G = \begin{bmatrix}
I_{kM_1} & 0_{kM_1, M_1M_2-2kM_1} & -I_{kM_1}
\end{bmatrix}
\]

The condition is that the first \(k\) columns and the last \(k\) columns of \(\tau\) coincide.
Optimal spline surface $x(s,t)$ periodic in $t$

We form the following Lagrangian function.

$$L(\tilde{\varphi}, \mu) = \frac{1}{2} J(\tilde{\varphi}) + \mu^T G \tilde{\varphi}$$

$$= \frac{1}{2} \lambda \tilde{\varphi}^T Q \tilde{\varphi} + \frac{1}{2} (r^T \tilde{\varphi} - d)^T W (r^T \tilde{\varphi} - d) + \mu^T G \tilde{\varphi}$$

By the routine procedure, we get the following system of algebraic equation.

**Optimal solution to Problem 2:**

$$\begin{bmatrix} \lambda Q + \Gamma W \Gamma^T & G^T \\ G & 0_{kM_1,kM_1} \end{bmatrix} \begin{bmatrix} \tilde{\varphi} \\ \mu \end{bmatrix} = \begin{bmatrix} \Gamma W d \\ 0_{kM_1} \end{bmatrix}$$

**Existence of solution:**

If $\lambda Q + \Gamma W \Gamma^T > 0$, then the coefficient matrix is nonsingular since $G$ matrix is of row full rank, and the solution exists uniquely.

Properties of Smoothing Splines for Sampled Data

We analyze various properties of optimal smoothing surfaces.

**Assumption:**

We assume that the data is obtained by sampling some surface with and without noises.

**Question:**

Then how the optimal spline surface behave as the number of data increases, and what are the statistical properties?

We show that, under some natural condition, the optimal smoothing splines converge to some limiting surface as the number of sampled data increases.
This asymptotical and statistical analyses are important, since then
-- we can expect to get better results by increasing the number
of data, and
-- the data are often corrupted by noise and we can treat such data

Trivariate Optimal Smoothing Splines

Trivariate Splines

We construct trivariate splines (splines in three variables) using product
B-splines as the basis functions as

\[ x(r, s, t) = \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} \sum_{l=-k}^{m_3-1} \tau_{i,j,l} B_k(\alpha(r - r_i)) B_k(\beta(s - s_j)) B_k(\gamma(t - t_l)) \]

where,

- \( B_k(\cdot) \) : normalized uniform B-splines with degree \( k \)
- \( \alpha, \beta, \gamma \) : positive constants.
- \( r_i, s_j, t_l \) : equally-spaced knot points both in \( s \) and \( t \) directions:
  \[ r_i - r_{i-1} = \frac{1}{\alpha}, \quad s_i - s_{i-1} = \frac{1}{\beta}, \quad t_l - t_{l-1} = \frac{1}{\gamma} \]
- \( \tau_{i,j,l} \) : weighting coefficient called control point.
- \( m_1, m_2, m_3 \) : integers.
Trivariate smoothing splines

Suppose that a set of spatial data
\[ \mathcal{D} = \{ (p_i, u_i, v_i; d_i) : (p_i, u_i, v_i) \in \mathcal{S}, \ d_i \in \mathbb{R}, \ i = 1, 2, \cdots, N \} \]
be given, where \( \mathcal{S} = I_1 \times I_2 \times I_3 \) with
\[ I_1 = (r_0, r_{m_1}), \ I_2 = (s_0, s_{m_2}), \ I_3 = (t_0, t_{m_3}) \]
The standard optimal smoothing spline problem is stated as follows.

**Problem 1:** Find the control point \( \tau_{i,j,k} \) minimizing the cost function
\[
J(\tau) = \lambda \int_{I_1} \int_{I_2} (\nabla^2 x(r, s, t))^2 \, dr \, ds \, dt + \sum_{i=1}^{N} w_i \left( x(r_i, u_i, v_i) - d_i \right)^2
\]
where
\[
\lambda \ (> 0) \quad : \text{smoothing parameter} \\
w_i \ (0 < w_i \leq 1) \quad : \text{weights for approximation errors} \\
\nabla^2 \quad : \text{the Laplacian operator:} \quad \nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial s^2} + \frac{\partial^2}{\partial t^2}
\]

References

- **basics of spline surface**

- **periodic spline surface and modeling of wet material**

- **spline surface and extrema detection**

- **recursive construction of spline surface**
**Expression of \( x(s,t) \):** The spline surface \( x(r,s,t) \) is expressed in terms of vec-function and Kronecker product as

\[
x(r, s, t) = (b_3(t) \otimes b_2(s) \otimes b_1(r))^T \hat{\tau}
\]

where

\[ \hat{\tau} = \text{vec } \tau \]

**Note:**

(i) single variable case (univariate spline):

\[
x(t) = b^T(t) \hat{\tau}
\]

(ii) two variable case (bivariate spline):

\[
x(s, t) = (b_2(t) \otimes b_1(s))^T \hat{\tau}
\]

Thus, extensions of various theories to trivariate case are straightforward.

---

Thank you for your attention !!

おわり