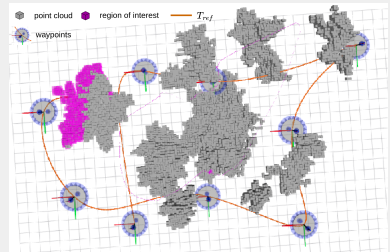


MOTION PLANNING FOR AUTONOMOUS VEHICLES

CURVE FITTING

GEESARA KULATHUNGA

MARCH 18, 2023



CURVE FITTING

CONTENTS

- n degree polynomial fitting
- Euler-Lagrange equation
- Minimum jerk trajectory (MJT) generation
- Quintic polynomial
- Lagrange polynomials
- Lagrange first-order, second-order, and nth-order interpolation
- Spline interpolation: Linear, Quadratic, and Cubic Spline
- Other types of curve fitting: Gradient descent, Double arc trajectory interpolation
- Nonlinear curve fitting
- Minimum-snap curve fitting
- Bezier curve fitting
- B-spline curve fitting

N DEGREE POLYNOMIAL FITTING

The least-squares method can be used to fit n th order fitting

- Consider a set $(x_i, y_i), i = 0, \dots, m - 1$

N DEGREE POLYNOMIAL FITTING

The least-squares method can be used to fit n th order fitting

- Consider a set (x_i, y_i) , $i = 0, \dots, m - 1$
- Define n th order fitting curve

$$a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

N DEGREE POLYNOMIAL FITTING

The least-squares method can be used to fit n th order fitting

- Consider a set (x_i, y_i) , $i = 0, \dots, m - 1$
- Define n th order fitting curve

$$a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

- Convert to mean square error

$$J(x, y) = \sum_{i=0}^{m-1} (a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n - y_i)^2$$

N DEGREE POLYNOMIAL FITTING

The least-squares method can be used to fit n th order fitting

- Consider a set $(x_i, y_i), i = 0, \dots, m - 1$
- Define n th order fitting curve

$$a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

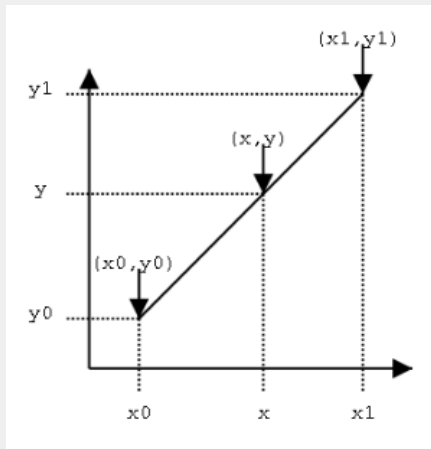
- Convert to mean square error

$$J(x, y) = \sum_{i=0}^{m-1} (a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n - y_i)^2$$

- Solve it

$$A^T A \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_n \end{bmatrix} = A^T \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}, A = \begin{bmatrix} 1 & x_1 & \dots & x_1^n \\ 1 & x_2 & \dots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \dots & x_m^n \end{bmatrix}$$

LINEAR INTERPOLATION



$$f(x) = f(x_0) + (x - x_0) \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (1)$$

EULER-LAGRANGE EQUATION

A **solution** of the **Euler-Lagrange** equation is called an **extremal** (minimum or maximum) of the **functional**. If **Lagrangian** $L(x, \dot{x})$ depends only on **first-order derivatives**, a **second-order equation of motion** can be found where only **two boundary conditions** are required, e.g., the **position** of the vehicle at an initial and final time. Such a **condition fixes the endpoint**. However, if **Lagrangian** $L(x, \dot{x}, \ddot{x})$ depends on **second-order derivatives**, a **fourth-order equation** of motion can be found. Hence, it requires **four boundary conditions** and **fixing the velocity** (as well as the **position**) at the **initial** and **final** time. Euler-Lagrange equations for a Lagrangian $L(x, \dot{x}, \ddot{x}, \dots)$ are given by

$$\frac{\partial L}{\partial x} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) + \dots + (-1)^n \frac{d^n}{dt^n} \left(\frac{\partial L}{\partial x^{(n)}} \right) = 0. \quad (2)$$

EULER-LAGRANGE EQUATION

Consider the AGV moves between the two positions within the time interval T .

$$J = \int_0^T L dt \quad (3)$$

where T is the motion duration and the performance index is L , the minimum jerk trajectory for unconstrained point-to-point movement, is

$$L = \left(\frac{\partial^3 x}{\partial t^3} \right)^2 + \left(\frac{\partial^3 y}{\partial t^3} \right)^2 \quad (4)$$

where, x and y indicates the position components. The objective is to deduce the local path minimizing the cost function J .

MINIMUM JERK TRAJECTORY (MJT) GENERATION

Jerk is **the time derivation** of **acceleration**. Jerk is the way to define **comfortness** mathematically (or suppressing vibration effects or sudden acceleration change). Additionally, the first and second derivatives are continuous, so **continuous velocity and curvature are satisfied**.

MINIMUM JERK TRAJECTORY (MJT) GENERATION

- To solve this, Euler–Lagrange equation can be utilized.

$$\begin{aligned}\frac{\partial L}{\partial x} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) + \cdots + (-1)^n \frac{d^n}{dt^n} \left(\frac{\partial L}{\partial x^{(n)}} \right) &= 0, \\ \frac{\partial L}{\partial y} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}} \right) + \cdots + (-1)^n \frac{d^n}{dt^n} \left(\frac{\partial L}{\partial y^{(n)}} \right) &= 0.\end{aligned}\tag{5}$$

MINIMUM JERK TRAJECTORY (MJT) GENERATION

- To solve this, Euler–Lagrange equation can be utilized.

$$\begin{aligned}\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) + \cdots + (-1)^n \frac{d^n}{dt^n}\left(\frac{\partial L}{\partial x^{(n)}}\right) &= 0, \\ \frac{\partial L}{\partial y} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) + \cdots + (-1)^n \frac{d^n}{dt^n}\left(\frac{\partial L}{\partial y^{(n)}}\right) &= 0.\end{aligned}\tag{5}$$

- Since jerk has to be minimized, $\frac{d}{dt}\left(\frac{\partial \ddot{x}^2}{\partial \ddot{x}}\right) = 0$ and $\frac{d}{dt}\left(\frac{\partial \ddot{y}^2}{\partial \ddot{y}}\right) = 0$ must be satisfied[1]. Hence,

$$\frac{d^6 x}{dt^6} = 0, \quad \frac{d^6 y}{dt^6} = 0\tag{6}$$

[1].<https://courses.shadmehrlab.org/Shortcourse/minimumjerk.pdf>

MINIMUM JERK TRAJECTORY (MJT) GENERATION

- To solve this, Euler–Lagrange equation can be utilized.

$$\begin{aligned}\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) + \cdots + (-1)^n \frac{d^n}{dt^n}\left(\frac{\partial L}{\partial x^{(n)}}\right) &= 0, \\ \frac{\partial L}{\partial y} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) + \cdots + (-1)^n \frac{d^n}{dt^n}\left(\frac{\partial L}{\partial y^{(n)}}\right) &= 0.\end{aligned}\tag{5}$$

- Since jerk has to be minimized, $\frac{d}{dt}\left(\frac{\partial \ddot{x}^2}{\partial \ddot{x}}\right) = 0$ and $\frac{d}{dt}\left(\frac{\partial \ddot{y}^2}{\partial \ddot{y}}\right) = 0$ must be satisfied[1]. Hence,

$$\frac{d^6 x}{dt^6} = 0, \quad \frac{d^6 y}{dt^6} = 0\tag{6}$$

- Therefore, $x(t)$ and $y(t)$ must having the 5th order polynomial as follows:

$$\begin{aligned}x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5\end{aligned}\tag{7}$$

[1].<https://courses.shadmehrlab.org/Shortcourse/minimumjerk.pdf>

QUINTIC POLYNOMIAL

A polynomial of degree five defines a quintic function.

$$\begin{aligned}x(t) &= a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \\y(t) &= b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4 + b_5t^5\end{aligned}\tag{8}$$

Consider the initial condition $x_0, y_0, \dot{x}_0, \dot{y}_0, \ddot{x}_0, \ddot{y}_0$ at $t = 0$ and final condition $x_f, y_f, \dot{x}_f, \dot{y}_f, \ddot{x}_f, \ddot{y}_f$ at $t = T$ are given.

Quintic polynomial curve fitting decouples the along **x** and **y** directions, however, **position, velocity acceleration and jerk** are solved by **coupling**.

QUINTIC POLYNOMIAL

Hence,

$$\begin{aligned} a_0 &= x_0, & a_1 &= \dot{x}_0, & a_2 &= \ddot{x}_0/2 \\ A &= \begin{bmatrix} t^3 & t^4 & t^5 \\ 3t^2 & 4t^3 & 5t^4 \\ 6t & 12t^2 & 20t^3 \end{bmatrix}, & b &= \begin{bmatrix} x_f - a_0 - a_1 - a_2 t^2 \\ \dot{x}_f - a_1 - 2a_2 t \\ \ddot{x}_f - 2a_2 \end{bmatrix} \\ & & & & & \begin{bmatrix} a_3 \\ a_4 \\ a_5 \end{bmatrix} = A^{-1}b \end{aligned} \quad (9)$$

similar way b_0, \dots, b_5 can be calculated. The higher-order derivatives can be estimated as follows:

$$\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ \dot{x}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ \ddot{x}(t) &= 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{aligned} \quad (10)$$

LAGRANGE POLYNOMIALS

- Given a set of points: $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$, to define a Lagrange polynomial, it is required to define a set of cardinal functions: $l_1, l_2, \dots, l_n \in \mathbb{P}^n$ such that

$$l_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (11)$$

for $\forall i \in [0, \dots, n]$. Term δ_{ij} is called Kronecker's delta.

LAGRANGE POLYNOMIALS

- Given a set of points: $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$, to define a Lagrange polynomial, it is required to define a set of cardinal functions: $l_1, l_2, \dots, l_n \in \mathbb{P}^n$ such that

$$l_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (11)$$

for $\forall i \in [0, \dots, n]$. Term δ_{ij} is called Kronecker's delta.

- Term \mathbb{P}^n , denoted polynomial of nth order.

$$\begin{aligned} l_i(x) &= \prod_{j=0, j \neq i}^n \left(\frac{x - x_j}{x_i - x_j} \right) \\ &= \frac{x - x_0}{x_i - x_0} \cdot \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_n}{x_i - x_n} \end{aligned} \quad (12)$$

Both conditions: $l_i(x_i) = 1$ and $l_i(x_k) = 0, i \neq k$ can be verified.

- Therefore, the Lagrange form of a polynomial interpolation can be defined as

$$P_n(x) = \sum_{i=0}^n l_i(x) \cdot y_i \quad (13)$$

- Therefore, the Lagrange form of a polynomial interpolation can be defined as

$$P_n(x) = \sum_{i=0}^n l_i(x) \cdot y_i \quad (13)$$

- With that interpolation property is expressed as

$$P_n(x_j) = \sum_{i=0}^n l_i(x_j) \cdot y_i = y_j \quad (14)$$

- Formulate a second-order Lagrange polynomial that goes through these points: $x, y = (0, 1), (2/3, 0.5), (1, 0)$.

- Formulate a second-order Lagrange polynomial that goes through these points: $x, y = (0, 1), (2/3, 0.5), (1, 0)$.
- Define cardinal functions $l_0(x), l_1(x), l_2(x)$, afterwards Lagrange polynomial can be determined as

$$p_2(x) = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2 \quad (15)$$

- Formulate a second-order Lagrange polynomial that goes through these points: $x, y = (0, 1), (2/3, 0.5), (1, 0)$.
- Define cardinal functions $l_0(x), l_1(x), l_2(x)$, afterwards Lagrange polynomial can be determined as

$$p_2(x) = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2 \quad (15)$$

- The main **disadvantage** of the **Lagrange polynomial** is that **adding or removing a new point, it has to recompute all the i'_i s**

LAGRANGE FIRST ORDER INTERPOLATION AND SECOND ORDER INTERPOLATION

■ Lagrange first-order interpolation

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0) \frac{f(x_0) - f(x_1)}{x_0 - x_1} \\ &= \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \end{aligned} \tag{16}$$

LAGRANGE FIRST ORDER INTERPOLATION AND SECOND ORDER INTERPOLATION

■ Lagrange first-order interpolation

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0) \frac{f(x_0) - f(x_1)}{x_0 - x_1} \\ &= \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \end{aligned} \quad (16)$$

■ Lagrange second-order interpolation

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0) \frac{f(x_0) - f(x_1)}{x_0 - x_1} + (x - x_0)(x - x_1) \frac{f(x_0, x_1) - f(x_1, x_2)}{x_0 - x_1} \\ &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \\ &\quad + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \end{aligned} \quad (17)$$

LAGRANGE NTH ORDER INTERPOLATION

$$f(x) = f(x_0)\delta_0(x) + f(x_1)\delta_1(x) + \dots + f(x_n)\delta_n(x) \quad (18)$$

where $\delta_i(x)$ can be determined as

$$\delta_i(x) = \prod_{j=0, j \neq i}^n \left(\frac{x - x_j}{x_i - x_j} \right) \quad (19)$$

Some example: <https://polympc.readthedocs.io/en/latest/ocp.html>

VARIATION OF CALCULUS TO SPLINE FITTING

Given

$$\begin{aligned} \min_{f:[0,1] \rightarrow \mathbb{R}} \quad & \int_0^1 \left[f^{(2)}(t) \right]^2 dt \\ \text{s.t.} \quad & f(0) = a, f^{(1)}(0) = c \\ & f(1) = b, f^{(1)}(1) = d \end{aligned} \tag{20}$$

VARIATION OF CALCULUS TO SPLINE FITTING

- The objective

$$J(f) = \int_0^1 \left[f^{(2)}(t) \right]^2 dt$$

VARIATION OF CALCULUS TO SPLINE FITTING

- The objective

$$J(f) = \int_0^1 \left[f^{(2)}(t) \right]^2 dt$$

- After considering the perturbation

$$J(f, p) = \int_0^1 \left[f^{(2)}(t) + hp^{(2)}(t) \right]^2 dt$$

VARIATION OF CALCULUS TO SPLINE FITTING

■ The objective

$$J(f) = \int_0^1 \left[f^{(2)}(t) \right]^2 dt$$

■ After considering the perturbation

$$J(f, p) = \int_0^1 \left[f^{(2)}(t) + hp^{(2)}(t) \right]^2 dt$$

■ Gateaux derivative

$$\begin{aligned} dJ(f, p) &= \frac{d}{dh} \left(\int_0^1 \left[f^{(2)}(t) + hp^{(2)}(t) \right]^2 dt \right)_{h=0} \\ &= \int_0^1 2 \left[f^{(2)}(t) + hp^{(2)}(t) \right] p^{(2)}(t) \big|_{h=0} dt \\ &= \int_0^1 2f^{(2)}(t)p^{(2)}(t) dt \end{aligned}$$

Taking integration by parts

$$\begin{aligned} dJ(f, p) &= [2f^{(2)}(t)p^{(1)}(t)]_0^1 - [2f^{(3)}(t)p(t)]_0^1 + \int_0^1 2f^{(4)}(t)p(t)dt \\ &= \int_0^1 2f^{(4)}(t)p(t)dt \end{aligned}$$

If $f(t)$ is optimal for the considered constraint problem, then $dJ(f, p) = 0$ as long as $f(t) + hp(t)$ is feasible for small $h \Rightarrow p(0) = p(1) = p^{(1)}(0) = p^{(1)}(1) = 0$. The function $p(t)$ can have infinitely many forms. Therefore, to obtain $dJ(f, p) = 0$, $f^{(4)}(t) = 0$. Hence, $f(t) = a^0 + a^1t + a^2t^2 + a^3t^3$.

SPLINE: PIECE-WISE INTERPOLATION

Only consider sub-interval without considering the **whole polynomial** as formulated in Lagrange nth order interpolation. Let $S(t)$ be interpolated function through a given set of points $(t_i, y_i)_{i=0}^n$. The ordered set $t_0 < t_1 < \dots < t_n$ is called knots vector. Hence, $S(t)$ contains a set of piece-wise polynomials

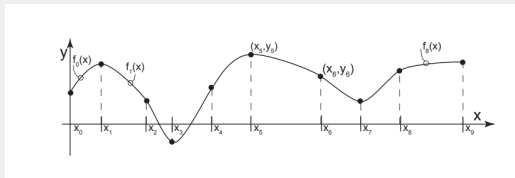
$$S(t) = \begin{cases} S_0(t), & t_0 \leq t \leq t_1 \\ S_1(t), & t_1 \leq t \leq t_2 \\ \vdots \\ S_{n-1}(t) & t_{n-1} \leq t \leq t_n \end{cases} \quad (21)$$

$S(t)$ is a polynomial of degree k , if and only if $S(t)$ is $k-1$ times continuous differentiable

$$S_{i-1}(t_i) = S_i(t_i), S'_{i-1}(t_i) = S'_i(t_i), \dots, S^{(k-1)}_{i-1}(t_i) = S^{(k-1)}_i(t_i), \quad (22)$$

When n equals **1 linear** Spline, equals **2 quadratic** Spline, and equals **3 cubic** spline

SPLINE: PIECE-WISE INTERPOLATION



In general, $f(x_i) = a_i + b_i x + c_i x^2 + d_i x^3$, is the function which depicts the curve in between i^{th} and $i + 1^{th}$ **control points**[1]. Hence, each curve represents by a cubic polynomial, with four coefficients for each. **How many parameters are to be solved?**

[1]. <https://people.cs.clemson.edu/~dhouse/courses/405/notes/splines.pdf>

SPLINE: PIECE-WISE INTERPOLATION

Each segment pass through its control points

$$f_i(x) = y_i, f_i(x_{i+1}) = y_{i+1}$$



Consecutive segments should have the **same slope** and **same curvature** where they **join together** $f'_i(x_{i+1}) = f'_{i+1}(x_{i+1})$,

$$f''_i(x_{i+1}) = f''_{i+1}(x_{i+1})$$



How many parameters are to be solved?

Piece-wise linear interpolation, i.e., straight-line. The constraints are

$$\begin{aligned} S_0(t_0) &= y_0 \\ S_{i-1}(t_i) &= S_i(t_i) = y_i, \quad i = 1, 2, \dots, n-1, \quad \Rightarrow S_i(t) = y_i + \frac{y_{i+1} - y_i}{t_{i+1} - t_i}(t - t_i) \\ S_{n-1}(t_n) &= y_n \end{aligned} \quad (23)$$

CUBIC SPLINE

Given ordered set $(t_i, y_i)_{i=0}^n$, cubic spline can be defined as

$$S(t) = S_i(t) \quad \text{for} \quad t_i \leq t \leq t_{i+1} \quad (24)$$

where $S_i(t) = d_i(t - t_i)^3 + c_i(t - t_i)^2 + b_i(t - t_i) + a_i, i = 0, 1, \dots, n - 1$.
Thus, the total number of unknown $4n$. However, the following constraints must be satisfied $S(t)$ is a polynomial of degree $k=3$, if and only if $S(t)$ is $k-1$ times continuous differentiable

$$\begin{aligned} S_i(t_i) = y_i, S_i(t_{i+1}) = y_{i+1}, i = 0, 1, \dots, n - 1 &\Rightarrow 2 \cdot n \text{ equations} \\ S'_i(t_{i+1}) = S'_{i+1}(t_{i+1}), i = 0, 1, \dots, n - 2 &\Rightarrow n - 1 \text{ equations} \\ S^{(2)}_i(t_{i+1}) = S^{(2)}_{i+1}(t_{i+1}), i = 0, 1, \dots, n - 2 &\Rightarrow n - 1 \text{ equations} \\ S^{(2)}_0(t_0) = 0, S^{(2)}_{n-1}(t_n) = 0 &\Rightarrow 2 \text{ equations} \end{aligned} \quad (25)$$

CUBIC SPLINE

- Consider $z_i = S^{(2)}(t_i)$, $i = 1, 2, \dots, n-1$, $z_0 = z_n = 0$. Since $S^{(2)}$ are linear functions, $S^{(2)}$ can be formulated in the Lagrange form

$$\begin{aligned} S_i^{(2)}(t) &= \frac{z_{i+1}}{t_{i+1} - t_i}(t - t_i) - \frac{z_i}{t_{i+1} - t_i}(t - t_{i+1}) \\ &= \frac{z_{i+1}}{h_i}(t - t_i) - \frac{z_i}{h_i}(t - t_{i+1}), \end{aligned} \tag{26}$$

where term $h_i = t_{i+1} - t_i$.

CUBIC SPLINE

- Consider $z_i = S^{(2)}(t_i)$, $i = 1, 2, \dots, n-1$, $z_0 = z_n = 0$. Since $S^{(2)}$ are linear functions, $S^{(2)}$ can be formulated in the Lagrange form

$$\begin{aligned} S_i^{(2)}(t) &= \frac{z_{i+1}}{t_{i+1} - t_i}(t - t_i) - \frac{z_i}{t_{i+1} - t_i}(t - t_{i+1}) \\ &= \frac{z_{i+1}}{h_i}(t - t_i) - \frac{z_i}{h_i}(t - t_{i+1}), \end{aligned} \quad (26)$$

where term $h_i = t_{i+1} - t_i$.

- After integration, terms $S_i'(t)$ and $S_i(t)$ can be derived as follows:

$$\begin{aligned} S_i'(t) &= \frac{z_{i+1}}{2h_i}(t - t_i)^2 - \frac{z_i}{2h_i}(t - t_{i+1})^2 + C_i - D_i \\ S_i(t) &= \frac{z_{i+1}}{6h_i}(t - t_i)^3 - \frac{z_i}{6h_i}(t - t_{i+1})^3 + C_i(t - t_i) - D_i(t - t_{i+1}) \end{aligned} \quad (27)$$