Spline Interpolation in Piecewise Constant Tension

Masaru Kamada, Rentsen Enkhbat

Spline Interpolation in Piecewise Constant Tension

Masaru Kamada[(1)] and Rentsen Enkhbat[(2)]

(1) Ibaraki University, Hitachi, Ibaraki 316 8511, Japan.
(2) National University of Mongolia, P. O. Box 46/635, Ulaanbaatar, Mongolia.
kamada@mx.ibaraki.ac.jp, renkhbat46@ses.edu.mn

Abstract:

Locally supported splines in tension are constructed where the tension, which has ever been constant over the entire domain, is allowed to change at sampling points.

1. Introduction

A cubic spline gives the interpolation of data that minimizes the square integral of its second derivative [3, 5, 9] and is crowned as the smoothest interpolation in this sense. A linear spline gives the piecewise linear interpolation that is most straight but nonsmooth. The linear spline is characterized as minimizing the square integral of its first derivative [3, 9]. A *spline in tension* [1, 10] was devised as a generalization of those two splines. It minimizes the integral of a weighted sum of the squared second derivative and the squared first derivative. By increasing the weight called *tension*, we can make a spline in tension approach the most straight linear spline while retaining smoothness similar to that of the cubic spline.

The spline in tension has been known for more than 40 years. It has been extended even to the multidimensional cases [2, 7] and is now supported by a standard software library [8]. But the tension has ever been a single constant over the entire domain.

In this paper, we look at the splines as the output of a linear dynamical system with a series of delta functions input. That is the same way as how the exponential splines and their locally supported basis were successfully constructed in [12, 13]. In addition, attending to that the linear dynamical system theory [6] allows for time-varying dynamical parameters, we shall place different tension in each sampling interval. Then we will obtain locally supported splines in piecewise constant tension that can change the interpolation characteristics from a sampling interval to another.

2. Preliminaries

A spline f in tension interpolating the data $\{(t_k, f_k)\}_{k=-\infty}^{\infty}$ given at strictly increasing sampling points $(\cdots < t_{-2} < t_{-1} < t_0 < t_1 < t_2 < \cdots)$ on the real line is defined as the twice-differentiable function that minimizes the integral of a weighted sum

$$\int_{-\infty}^{\infty} (f^{(2)}(t))^2 + p(t)^2 (f^{(1)}(t))^2 \, dt \tag{1}$$

of its squared second derivative $f^{(2)}$ and squared first derivative $f^{(1)}$ subject to the constraints

$$f(t_k) = f_k, \quad k = 0, \pm 1, \pm 2, \cdots . \tag{2}$$

In the case $p = 0$, f is identical with the cubic spline [3, 5, 9]. By increasing p, f approaches the most straight linear spline as if the curve were pulled from both ends. That is why p is called *tension* [1, 10].

The tension p has originally been a single constant over the entire domain [10]. We shall now relax the tension to be a non-negative constant in each sampling interval, i.e.,

$$p(t) = p_k \geq 0, \ \text{ for } \ t \in [t_k, t_{k+1}), \tag{3}$$

which can change at the sampling points.

By the calculus of variation, the minimization problem is reduced to solution of the Euler-Lagrange differential equation

$$f^{(4)}(t) - 2p(t)p^{(1)}(t)f^{(1)}(t) - p(t)^2 f^{(2)}(t) = w(t), \tag{4}$$

where w is a series of the Dirac delta functions

$$w(t) = \sum_{n=-\infty}^{\infty} w_n \delta(t - t_n)$$

to be determined so that (2) holds good. We do not have, however, a practical means to decide the coefficients $\{w_n\}$ for given $\{(t_k, f_k)\}$.

In practice, it is convenient to have locally supported functions $\{y_n\}$ satisfying

$$y_n(t) = 0, \ \text{ for } \ t \notin [t_n, t_{n+4}] \tag{5}$$

of which linear combination

$$f(t) = \sum_{n=-\infty}^{\infty} c_n y_n(t) \tag{6}$$

represents any possible f. This y_n can be constructed by

$$y_n^{(4)}(t) - 2p(t)p^{(1)}(t)y_n^{(1)}(t) - p(t)^2 y_n^{(2)}(t) = u_n(t) \tag{7}$$

for some appropriately chosen

$$u_n(t) = \sum_{l=0}^{4} u_{l,n} \delta(t - t_{n+l}) \tag{8}$$

as long as the sampled data system (7) with the impulse input (8) is completely controllable [4]. Once we obtain $y_n(t)$, we have only to determine the coefficients $\{c_n\}$ by the linear equations

$$f_k = \sum_{n=-\infty}^{\infty} c_n y_n(t_k), \quad k = 0, \pm 1, \pm 2, \cdots$$

from $\{(t_k, f_k)\}$. Although infinitely many coefficients and data are involved in the equations, we can solve the linear equations for finitely many $\{c_n\}$ from finitely many $\{(t_k, f_k)\}$ in practice because $\{y_n\}$ are locally supported.

3. Construction of locally supported splines in piecewise constant tension

A state-space representation of the differential equation (7) is

$$\boldsymbol{x}_n^{(1)}(t) = F(t)x_n(t) + \boldsymbol{g}u_n(t), \;\; y_n(t) = \boldsymbol{h}\boldsymbol{x}_n(t), \quad (9)$$

where

$$F(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2p(t)p^{(1)}(t) & p(t)^2 & 0 \end{bmatrix},$$

$$\boldsymbol{x}_n(t) = \begin{bmatrix} y_n \\ y_n^{(1)} \\ y_n^{(2)} \\ y_n^{(3)} \end{bmatrix}, \; \boldsymbol{g} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \; \boldsymbol{h} = [1\,0\,0\,0]. \quad (10)$$

The state $\boldsymbol{x}_n$ can be expressed by

$$\boldsymbol{x}_n(t) = \Phi(t, v)\boldsymbol{x}_n(v) + \int_v^t \Phi(t, \tau)\boldsymbol{g}u_n(\tau)\, d\tau, \quad (11)$$

for any real numbers t and v, in terms of the state-transition matrix function Φ and the input u_n [11].

Since $u_n(t) = 0$ for $t \notin \{t_n, t_{n+1}, t_{n+2}, t_{n+3}, t_{n+4}\}$, it follows from (11) that

$$\boldsymbol{x}_n(t) = \begin{cases} \boldsymbol{0}, & t < t_n \\ \Phi(t, t_{n+l}\text{+}0)\boldsymbol{x}_n(t_{n+l}\text{+}0), & \\ \qquad t_{n+l} < t < t_{n+l+1}, (l = 0, 1, 2, 3) & \\ \Phi(t, t_{n+4}\text{+}0)\boldsymbol{x}_n(t_{n+4}\text{+}0), & t_{n+4} < t. \end{cases} \quad (12)$$

Because of the top and bottom lines of (12), $y_n = \boldsymbol{h}\boldsymbol{x}_n$ is locally supported as (5) if $\boldsymbol{x}_n(t_{n+4}\text{+}0) = \boldsymbol{0}$. In order to avoid the trivial case $u_{0,n} = u_{1,n} = u_{2,n} = u_{3,n} = u_{4,n} = 0$ that would result in $u_n \equiv y_n \equiv 0$, let us fix one of them as $u_{0,n} = 1$. Then the problem of constructing a locally supported y_n becomes a dead-beat control problem of finding $u_{1,n}$, $u_{2,n}$, $u_{3,n}$, and $u_{4,n}$ that make the terminal state dead as

$$\boldsymbol{x}_n(t_{n+4}\text{+}0) = \boldsymbol{0}. \quad (13)$$

Once the terminal state is controlled to $\boldsymbol{0}$, it will stay at $\boldsymbol{0}$ forever for $t > t_{n+4}$ without any beats.

We shall consider two types of state transitions: (i) Those within each sampling interval (t_{n+l}, t_{n+l+1}), and (ii) one across each sampling point t_{n+l}.

(i) In the open interval (t_{n+l}, t_{n+l+1}), (11) with $v = t_{n+l}\text{+}0$ is reduced to

$$\boldsymbol{x}_n(t) = \Phi(t, t_{n+l}\text{+}0)\boldsymbol{x}_n(t_{n+l}\text{+}0), \;\; l = 0, 1, 2, 3 \quad (14)$$

because $u_n(t) = 0$ for $t \in (t_{n+l}, t_{n+l+1})$. Besides, $F(t)$ in (10) is a constant matrix

$$F(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & p_{n+l}^2 & 0 \end{bmatrix} \quad (15)$$

because of (3) so that we can calculate the state-transition matrix by the matrix exponential function [11] as follows:

$$\begin{aligned} & \Phi(t, t_{n+l}\text{+}0) \\ = \;& e^{\int_{t_{n+l}+0}^{t} F(v)\, dv} \\ = \;& \begin{cases} \begin{bmatrix} 1 & t - t_{n+l} & \frac{(t-t_{n+l})^2}{2} & \frac{(t-t_{n+l})^3}{6} \\ 0 & 1 & t - t_{n+l} & \frac{(t-t_{n+l})^2}{2} \\ 0 & 0 & 1 & t - t_{n+l} \\ 0 & 0 & 0 & 1 \end{bmatrix} & \\ \qquad\qquad \text{if } p_{n+l} = 0 & \\ \begin{bmatrix} 1 & t - t_{n+l} & \frac{\cosh(p_{n+l}(t-t_{n+l}))-2}{p_{n+l}^2} & \frac{\sinh(p_{n+l}(t-t_{n+l}))}{p_{n+l}^3} - \frac{2p_{n+l}(t-t_{n+l})}{p_{n+l}^3} \\ 0 & 1 & \frac{\sinh(p_{n+l}(t-t_{n+l}))}{p_{n+l}} & \frac{\cosh(p_{n+l}(t-t_{n+l}))-2}{p_{n+l}^2} \\ 0 & 0 & \cosh(p_{n+l}(t-t_{n+l})) & \frac{\sinh(p_{n+l}(t-t_{n+l}))}{p_{n+l}} \\ 0 & 0 & p_{n+l}\sinh(p_{n+l}(t-t_{n+l})) & \cosh(p_{n+l}(t-t_{n+l})) \end{bmatrix} & \\ \qquad\qquad \text{if } p_{n+l} > 0. & \end{cases} \end{aligned} \quad (16)$$

In the special case that $t = t_{n+l+1}\text{-}0$, we have the state transition from $\boldsymbol{x}_n(t_{n+l}\text{+}0)$ to $\boldsymbol{x}_n(t_{n+l+1}\text{-}0)$ as follows:

$$\boldsymbol{x}_n(t_{n+l+1}\text{-}0) = \Phi(t_{n+l+1}\text{-}0, t_{n+l}\text{+}0)\boldsymbol{x}_n(t_{n+l}\text{+}0), \quad l = 0, 1, 2, 3. \quad (17)$$

The matrix $\Phi(t_{n+l+1}\text{-}0, t_{n+l}\text{+}0)$ can be evaluated by the right hand side of (16) with t replaced by t_{n+l+1}.

(ii) The state transition from $\boldsymbol{x}_n(t_{n+l}\text{-}0)$ to $\boldsymbol{x}_n(t_{n+l}\text{+}0)$ across the sampling point t_{n+l}, $(l = 0, 1, 2, 3, 4)$ finds a trouble that $F(t)$ in (10) contains a derivative of the function p being discontinuous at t_{n+l} as defined by (3). We had better consider this transition by way of the original differential equation (7). An equivalent form of (7) is

$$y_n^{(4)}(t) - \frac{d}{dt}\left(p(t)^2 y_n^{(1)}(t)\right) = u_n(t) \quad (18)$$

and its integration gives

$$y_n^{(3)}(t) = p(t)^2 y_n^{(1)}(t) + \int_{t_n - 0}^{t} u_n(\tau) d\tau + c, \quad (19)$$

where c is an integral constant. Substituting $t_{n+l}\text{+}0$ and $t_{n+l}\text{-}0$ for t of (19), we have

$$\begin{aligned} y_n^{(3)}(t_{n+l}\text{+}0) \;=\; & p(t_{n+l}\text{+}0)^2 y_n^{(1)}(t_{n+l}\text{+}0) \\ & + u_{0,n} + \cdots + u_{n+l,n} + c \end{aligned} \quad (20)$$

and

$$y_n^{(3)}(t_{n+l}\text{-}0) = p(t_{n+l}\text{-}0)^2 y_n^{(1)}(t_{n+l}\text{-}0) + u_{0,n} + \cdots + u_{n+l-1,n} + c, \quad (21)$$

respectively. Recall that the spline in tension is sought among the twice-differentiable functions and attend to the definition (3) of p. Then we can reduce (20) and (21) to

$$y_n^{(3)}(t_{n+l}\text{+}0) = p_{n+l}^2 y_n^{(1)}(t_{n+l}) + u_{0,n} + \cdots + u_{n+l,n} + c \quad (22)$$

and

$$y_n^{(3)}(t_{n+l}\text{-}0) = p_{n+l-1}^2 y_n^{(1)}(t_{n+l}) + u_{0,n} + \cdots + u_{n+l-1,n} + c, \quad (23)$$

respectively. Subtracting (23) from (22), we have

$$\begin{aligned} & y^{(3)}(t_{n+l}\text{+}0) - y^{(3)}(t_{n+l}\text{-}0) \\ = \; & (p_{n+l}^2 - p_{n+l-1}^2) y^{(1)}(t_{n+l}) + u_{l,n}, \end{aligned} \quad (24)$$

which tells how to update the state variable $y^{(3)}$ at t_{n+l} and implies that the other state variables $y^{(2)}$, $y^{(1)}$, and y are continuous at t_{n+l}. So we can write the state transition across the sampling point t_{n+l} as follows:

$$\boldsymbol{x}_n(t_{n+l}\text{+}0) = \Phi(t_{n+l}\text{+}0, t_{n+l}\text{-}0)\boldsymbol{x}_n(t_{n+l}\text{-}0) + \boldsymbol{g}u_{l,n}, \quad l = 0, 1, 2, 3, 4, \quad (25)$$

where

$$\Phi(t_{n+l}\text{+}0, t_{n+l}\text{-}0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & p_{n+l}^2 - p_{n+l-1}^2 & 0 & 1 \end{bmatrix}. \quad (26)$$

The two types of state transitions (17) and (25) can be combined into the recurrence formulae

$$\begin{aligned} \boldsymbol{x}_n(t_n\text{+}0) &= \boldsymbol{g}u_{0,n} = \boldsymbol{g}, \\ \boldsymbol{x}_n(t_{n+l}\text{+}0) &= \Psi_{n+l}\boldsymbol{x}_n(t_{n+l-1}\text{+}0) + \boldsymbol{g}u_{l,n}, \\ & \qquad l = 1, 2, 3, 4, \end{aligned} \quad (27)$$

where we have set

$$\Psi_{n+l} = \Phi(t_{n+l}\text{+}0, t_{n+l}\text{-}0)\Phi(t_{n+l}\text{-}0, t_{n+l-1}\text{+}0), \quad l = 1, 2, 3, 4, \quad (28)$$

and used our choice $u_{0,n} = 1$ and the initial state $\boldsymbol{x}(t_n\text{-}0) = \boldsymbol{0}$. By these recurrence formulae, we can write the terminal state as follows:

$$\begin{aligned} \boldsymbol{x}_n(t_{n+4}\text{+}0) = \; & \Psi_{n+4}\Psi_{n+3}\Psi_{n+2}\Psi_{n+1}\boldsymbol{g} \\ & + \Psi_{n+4}\Psi_{n+3}\Psi_{n+2}\boldsymbol{g}u_{1,n} \\ & + \Psi_{n+4}\Psi_{n+3}\boldsymbol{g}u_{2,n} \\ & + \Psi_{n+4}\boldsymbol{g}u_{3,n} \\ & + \boldsymbol{g}u_{4,n}. \end{aligned} \quad (29)$$

Then we can determine u_1, u_2, u_3, and u_4 that makes the terminal state $\boldsymbol{x}_n(t_{n+4}\text{+}0)$ be zero by

$$\begin{bmatrix} u_{1,n} \\ u_{2,n} \\ u_{3,n} \\ u_{4,n} \end{bmatrix} = -\left[\Psi_{n+4}\Psi_{n+3}\Psi_{n+2}\boldsymbol{g} \;\; \Psi_{n+4}\Psi_{n+3}\boldsymbol{g} \;\; \Psi_{n+4}\boldsymbol{g} \;\; \boldsymbol{g}\right]^{-1} \Psi_{n+4}\Psi_{n+3}\Psi_{n+2}\Psi_{n+1}\boldsymbol{g}. \quad (30)$$

Existence of the inverse matrix is equivalent to the complete controllability of the sampled-data system with the impulse control u_n input. We do not have the condition in a simpler form due to the complication caused by time-varying dynamics and non-uniform sampling. Even the uniform sampling case is yet to be investigated.

For the numerical evaluation of y_n, we first compute the states $\{\boldsymbol{x}_n(t_{n+l}\text{+}0)\}_{l=0}^3$ by (27) from $\{u_{l,n}\}_{l=0}^4$. Then we can evaluate y_n by

$$y_n(t) = \begin{cases} 0, & t \le t_n \\ \boldsymbol{h}\Phi(t, t_{n+l}\text{+}0)\boldsymbol{x}_n(t_{n+l}\text{+}0), & \\ \qquad t_{n+l} < t \le t_{n+l+1}, (l = 0, 1, 2, 3) & \\ 0, & t_{n+4} \le t \end{cases} \quad (31)$$

and

$$\begin{aligned} & \boldsymbol{h}\Phi(t, t_{n+l}\text{+}0) \\ = \; & \begin{cases} \begin{bmatrix} 1 & t - t_{n+l} & \frac{(t-t_{n+l})^2}{2} & \frac{(t-t_{n+l})^3}{6} \end{bmatrix} & \text{if } p_{n+l} = 0 \\ \begin{bmatrix} 1 & t - t_{n+l} & \frac{\cosh(p_{n+l}(t-t_{n+l}))-2}{p_{n+l}^2} & \\ & & \frac{\sinh(p_{n+l}(t-t_{n+l}))-2p_{n+l}(t-t_{n+l})}{p_{n+l}^3} \end{bmatrix} & \text{if } p_{n+l} > 0 \end{cases} \end{aligned} \quad (32)$$

which follow from (12), (16), and the continuity of y_n over the entire domain.

4. Numerical examples

Test data were prepared by concatenating a sampled smooth curve and a sampled polygonal line. Their interpolation was computed as a linear combination of the locally supported splines in tension.

The cubic spline interpolation (equivalent to the case $p(t) \equiv 0$) is shown in Fig. 1. The cursive part is reproduced in a good shape but the polygonal part suffers from inter-sample vibration. The linear spline interpolation (equivalent to the case $p(t) \to \infty$) in Fig. 2 behaves in the opposite way. Reproduction of the polygonal part is perfect but there is no smoothness. Interpolation by a spline in constant tension ($p(t) \equiv 10$) in Fig. 3 provides a good compromise between the cubic and linear spline interpolation. It is fairly smooth and has less vibration.

Some may say that the cursive part is not smooth enough and rather polygonal in Fig. 3. In this case, we can obtain a better interpolation by varying the tension in time. Figure 4 is an interpolation by a spline in piecewise constant tension. Higher tensions are imposed on the polygonal part to suppress the vibration. The interpolation is kept smooth elsewhere. The locally supported splines used to construct this curve are plotted in Fig. 5 where the plots are vertically scaled to have a common peak value.

5. Conclusions

Locally supported splines in tension were constructed where the tension is constant within each sampling interval and variable at the sampling points. They will hopefully contribute to the variety of curve drawing modules in the graphical design tools. Another application may be image enlargement tools which allow users to put higher tension manually at the portions where they want to suppress ringing effects.

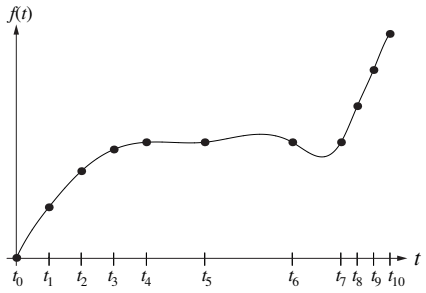

Figure 1: Interpolation by a cubic spline ($p(t) \equiv 0$).

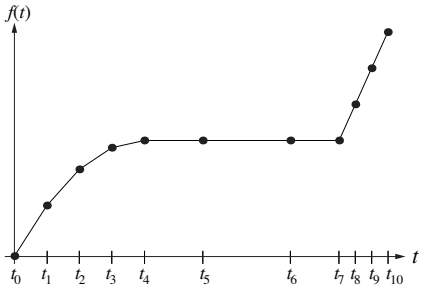

Figure 2: Interpolation by a linear spline ($p(t) \to \infty$).

References:

[1] J. H. Ahlberg, E. N. Nilson and J. L. Walsh. *The Theory of Splines and Their Applications*. Academic Press, London, 1967.

[2] M. N. Benbourhim and A. Bouhamidi. Approximation of vector fields by thin plate splines with tension. *J. Approx. Theory*, 136:198–229, 2005.

[3] C. de Boor. Best approximation properties of spline functions of odd degree. *J. Math. Mech.*, 12:747–750, 1963.

[4] Y. C. Ho, R. E. Kalman and K. S. Narendra. Controllability of linear dynamical systems. *Contrib. Diff. Eqs.*, 1:189–213, 1963.

[5] J. C. Holladay. Smoothest curve approximation. *Math. Tables and Aids to Comput.*, 11:223–243, 1957.

[6] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME*, 82(Series D):35–45, 1960.

[7] H. Mitasova and L. Mitas. Interpolation by regularized spline with tension: I. theory and implementation. *Mathematical Geology*, 25:641–655, 1993.

[8] A. Polyakov and V. Brusentsev. *Graphics Programming with GDI+ & DirectX*. A-List Publishing, Wayne, PA, 2005.

[9] I. J. Schoenberg. On interpolation by spline functions and its minimal properties. In P. L. Butzer and J. Korevaar, editors, *On Approximation Theory*, pages 109–129, June 1964.

[10] D. G. Schweikert. An interpolation curve using a spline in tension. *J. Math. Phys.*, 45:312–317, 1966.

[11] E. D. Sontag. *Mathematical Control Theory*. Springer, New York, 1990.

[12] M. Unser and T. Blu. Cardinal exponential splines: Part I—Theory and filtering algorithms. *IEEE Transactions on Signal Processing*, 53(4):1425–1438, April 2005.

[13] M. Unser. Cardinal exponential splines: Part II—Think analog, act digital. *IEEE Transactions on Signal Processing*, 53(4):1439–1449, April 2005.

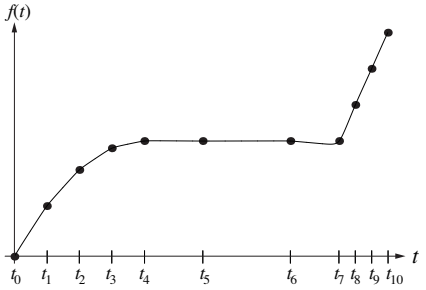

Figure 3: Interpolation by a spline in constant tension ($p(t) \equiv 10$).

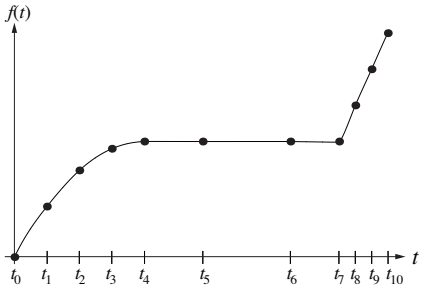

Figure 4: Interpolation by a spline in piecewise constant tension ($p(t) = 0$ for the cursive part ($t < t_4$), $p(t) = 10$ for the straight parts ($t_4 \le t < t_6$ and $t_7 \le t$), and $p(t) = 30$ for the breaking part ($t_6 \le t < t_7$)).

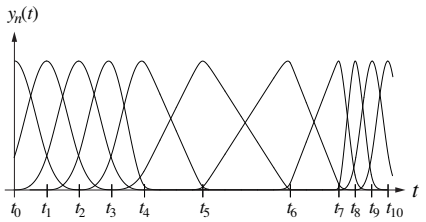

Figure 5: Locally supported splines used to construct the curve in Fig. 4.