SECOND ORDER SPIRAL SPLINES
LYLE NOAKES

ABSTRACT. Second order spiral splines are unit-speed planar curves that can be used to interpolate a list \( Y \) of \( n+1 \) points in \( R^2 \) at times specified in some list \( T \), where \( n \geq 2 \). We show that, for well behaved data \( Y \) and \( T \), there exist exactly \( 2^n \) natural second order spiral splines \( y_0 \) interpolating \( Y \) at \( T \). An algorithm is given for computing all the \( y_0 \) and is illustrated by examples.

1. Introduction

Given a finite sequence \( T \) of real numbers \( T_0 < T_1 < \ldots < T_n \) and a finite sequence \( Y \) of points \( Y_0, Y_1, \ldots, Y_n \in R^m \) where \( m \geq 1 \), it is a standard task to find a \( C^2 \) curve \( y : [T_0, T_n] \to R^m \) that interpolates \( Y \) at \( T \), namely \( y(T_j) = Y_j \) for \( 0 \leq j \leq n \).

A standard method for selecting such a curve \( y \) from all possible interpolants is to minimise

\[
J(y) := \int_{T_0}^{T_n} \|y''(t)\|^2 \, dt
\]

where \( \| \cdot \| \) denotes the Euclidean norm, and \( y^{(i)} \) is the \( i \)th derivative of \( y \). The infimum of \( J \) is achieved uniquely, by the so-called natural cubic spline, namely the \( C^2 \) piecewise cubic polynomial \( y \) with knots at \( T \), satisfying the auxiliary end conditions \( y''(T_0) = 0 = y''(T_n) \). The natural cubic spline always exists, is unique, and is easily calculated from \( T \) and \( Y \) by solving a tridiagonal system of linear equations [10]. None of these things hold when the \( C^2 \) interpolants \( y \) are required to be unit-speed namely \( \|y'(t)\| = 1 \) for all \( t \in [T_0, T_n] \).

Unit-speed curves are significant for high-speed and railway design [19, 20], motion planning for robots [13], and path planning for unmanned aerial vehicles and military aircraft [9, 15]. Whereas natural cubic splines always exist, an evident necessary condition for existence of a \( C^2 \) unit-speed interpolant is

\[
\|Y_j - Y_{j-1}\| \leq L_j := T_j - T_{j-1} \quad \text{for } 1 \leq j \leq n.
\]

By a natural elastic spline we mean\(^{\dag}\) a critical point of the restriction of \( J \) to the space of unit-speed interpolants. A natural elastic spline is known to be precisely a \( C^2 \) track-sum \( y \) of elastica (elastic curves) \( y_j : [T_{j-1}, T_j] \to R^m \) satisfying the auxiliary end conditions \( y''(T_{j-1}) = 0 = y''(T_j) \). By an elastica we mean\(^{\ddagger}\) a critical point of \( J \) restricted to unit-speed curves \( y_j \), where \( y(T_{j-1}), y''(T_{j-1}), y_j(T_1), y''(T_1) \) are all prescribed in advance. Elastic splines are extensively studied, with well-developed algorithms for interpolation, as discussed in \([\ddagger]\). These algorithms require significant effort, compared with interpolation by (non-unit-speed) polynomial splines.

Besides elastica there are simpler classes of unit-speed track-summands called polynomial spirals \([\ddagger]\). The well-studied class of clothoidal spirals, whose curvatures are \( C^0 \) and piecewise-affine, is insufficiently rich for interpolation when \( T \) is given in advance. We work with the larger class of second order spiral splines, namely \( C^2 \) track-sums \( y_0 \) of second order generalised Cornu spirals, whose curvatures are \( C^0 \) and piecewise-quadratic in the parameter \( t \). We impose the end conditions \( y''(T_{j-1}) = 0 = y''(T_j) \) and suppose that the inequalities (1) hold strictly.

Modulo conditions on how \( Y \) is generated from \( T \), Theorem \([\ddagger]\) says \( Y \) is interpolated at \( T \) by \( 2^n \) second order spiral spline interpolants, as illustrated in Examples \([\ddagger\ddagger]\)[\ddagger\ddagger][\ddagger\ddagger][\ddagger\ddagger]. So computation of second order spiral splines cannot be as easy as computations of cubic splines, where there is a unique solution, found from solving a tridiagonal system of linear equations.

Instead we identify \( 2^n \) discrete design parameters \( \sigma \) corresponding to all possible natural second order spiral splines \( y_0 \) interpolating \( Y \) at \( T \). The proof of Theorem \([\ddagger\ddagger]\) yields for each \( \sigma \) a construction of the corresponding \( y_0 \), echoing the standard algorithm for cubic splines, and with some similar advantages, namely

- \( \sigma, T \) and \( Y \) define a pair of tridiagonal linear systems: for \( n \geq 3 \) the solution of the first system is needed to define the right hand side of the second.
- Solution of the second system defines a curve \( y_0 \) that \emph{approximately} interpolates \( Y \) at \( T \) with good accuracy.
- Taking \( y_0 \) as an initial guess, standard software for systems of nonlinear equations is used to find the interpolating second order spiral spline \( y_0 \).

This is more costly than the standard algorithm for cubic splines, but not incomparably so, with an extra tridiagonal system for \( n \geq 3 \), and an application of Mathematica’s \texttt{FindRoot}. For all \( n \) to be considered, the low cost is multiplied by \( 2^n \). This seems manageable for moderate values of \( n \); in Example \([\ddagger\ddagger]\) we have \( n = 5 \) and the 32 possible second order spiral splines are found in a total of 0.23 seconds.

\textit{Date:} January 16, 2018.

\dag The term elastic spline is sometimes used differently, to mean an interpolant where \( T \) is not specified in advance \([\ddagger]\).

\ddagger The length of \( y_j \) is then \( T_j - T_{j-1} \). Sometimes the term elastica has a different meaning, where the length of \( y_j \) is not prescribed in advance \([\ddagger]\).

\ddagger It seems that the plural of elastica is elastica.
Following a little more discussion of unit-speed interpolants in [8] we proceed as follows. In [3] we specify how $Y$ and $T$ should be related for our results to hold. Relative to some small parameter $\epsilon > 0$, the gaps between subsequent $T_j$ should be neither very large nor very small. Then $Y$ is obtained by sampling from an unknown well-behaved curve $x$, with discrete curvatures $k_j$ bounded away from 0. Naturally, second order spiral splines $y_0$ are defined in terms of real-valued $C^3$ cubic splines $\theta$. Then we state Theorem 1, which says there are $2n$ interpolants $y_0$ (one of these approximates $x$ to order $O(\epsilon^2)$).

The proof of Theorem 1 begins in [3] where asymptotic equations are found, relating the coefficients of $\theta$ to $Y$ and $T$. Equations (9), (10) are linear in coefficients of $\theta$, and equation (11) is quadratic. The quadratic equations are replaced by linear equations (13) depending on $\sigma_j = \pm 1$, with right hand sides depending on higher order terms in the unknown coefficients. As illustrated in [3] this is enough for approximate recovery of $\theta$ from $Y$ and $T$ by solving systems of linear equations, one for each value of $\sigma$. In Example 2 we find the approximations $\theta$ to be fairly accurate, as indicated by the asymptotic analysis.

In [3] we deal with the case $n \geq 3$, where there is an additional complication due to a residual nonlinearity in equations (22), whose right hand sides depend partly on the unknown coefficients. In [7] a tridiagonal system of linear equations is used to find $O(\epsilon^2)$ estimates for some of the coefficients. The estimates need to be improved, but are sufficient to estimate the right hand sides of (26). Equations (26) are then solved approximately in [6.2] as part of a second tridiagonal system. The resulting natural second order spiral splines are very nearly interpolating, as illustrated in Examples 2 and 3.

For $y_0$ to interpolate $Y$ at $T$ we would need to have $\theta = \hat{\theta} + O(\epsilon^2)$ where $\theta$ is one of our $2n$ estimates. To guarantee existence of such a $C^3$ spline $\theta$, a little more work is needed in [3], using the implicit function theorem. In [3] we take this further, formulating the interpolation conditions as a system of nonlinear equations, using the $\theta$ to provide $2n$ initial guesses. The interpolation conditions are then met exactly, using standard software to solve the nonlinear systems. The complete algorithm is summarised in 17 steps and applied in Example 3. It is good to be able to do this, but sometimes the estimates $\hat{\theta}$ may already be sufficiently accurate, as seen when comparing Examples 2 and 3.

So far, there is no optimisation, except that the discrete design parameter $\sigma$ may be chosen to meet a performance criterion. However, additional parameters for $\theta$ permit continuous optimisation of a functional $J$ of $y_0$, as described in [8]. This is carried out in Example 4 with $n = 5$ additional parameters chosen to optimise elastic energy, with $Y$ and $T$ from Examples 2 and 3. The reductions in elastic energy turn out to be small, suggesting that the interpolating natural second order spiral splines of Example 3 are very nearly natural elastic splines.

2. More Background on Unit-Speed Interpolants

A natural class of unit-speed interpolants is the unit-speed reparameterised polynomial splines, studied in [21, 11, 22] where algorithms are for efficient computation. Whereas a polynomial spline can be found to interpolate $Y$ at $T$, the unit-speed reparameterisation interpolates $Y$ at a different set of parameter values. This does not work well when $T$ is given in advance, as in our situation. Similar difficulties arise with clothoidal splines [23, 13, 1, 10, 12] and their $C^4$ generalisations [15]. The algorithms in these papers are of significant interest, but do not apply when $T$ is already prescribed.

Interpolation by unit-speed curves can also be performed using elastic splines [12, 22], namely critical points of the restriction of the functional $J$ to the space of unit-speed $C^2$ interpolants. An elastic spline turns out to be a $C^2$ track-sum $y$ of elastica $y_j : [T_{j-1}, T_j] \to \mathbb{R}^d$, satisfying the auxiliary end conditions $y''(T_0) = 0 = y''(T_n)$. Elastica (elastic curves) have a long and interesting history [12, 13, 1], and their study reduced to the case where $m = 3$. The elastica are completely known in terms of elliptic functions [22], with simplifications for $m = 2$, which is the case of interest for the present paper. Although elastic splines (sometimes called nonlinear splines or true splines) are highly regarded as interpolants, they are less widely used than cubic polynomial splines. This is because the interpolation conditions for elastic splines require the solution of a system of nonlinear equations that is even more complicated than for clothoidal splines. However, the relative sophistication of elastica compared with cubic polynomials is not the main difficulty in computing elastic splines.

The more challenging problem is that there are usually many elastic splines interpolating $Y$ at $T$, and a good initial guess is required for each, in order to start the numerical solution of a sparse system of complicated nonlinear equations. The appearance of elliptic functions is of no special concern for numerical solvers, and sparseness is good: the critical task is to generate appropriate initial guesses for a rich variety of cases.

The algorithms in [10, 12] and [13] are for sliding elastic splines, where $T$ is not given in advance. Another condition that is sometimes imposed is that the total time $T_n - T_0$ should be fixed or, equivalently, the length of the spline is prescribed

---

This is not needed for the rest of the paper.

Just as for natural cubic splines, the precise details of our analysis would change if the auxiliary end conditions $y''(T_0) = 0 = y''(T_n)$ were replaced by other conditions.

A clothoidal spline is a unit-speed $C^2$ planar curve $t \mapsto y(t)$ whose curvature is $C^3$ and piecewise-affine in $t$.

Elastic splines and elastica mean different things depending on the context. In the present paper, elastic splines are pinned and unpinned, and elastica are fixed-length. In [10] $T$ is not given in advance and the elastic splines are sliding.

On the other hand, the study of elastic splines does not reduce in this way.

The objection raised at the end of [10] does not really apply when $T$ is given.

As noted on p. 184 of [12], clothoidal splines are sometimes used to construct initial guesses for the computation of elastic splines.

In our setting there are usually $2n$ elastic spline interpolants. This contrasts with the situation for polynomial splines, which are unique, depending only on $Y$ and $T$. 

in advance. In the present paper the entire sequence \(T\) is assumed to be given in advance, by analogy with standard interpolation by natural cubic polynomial splines.

What is different from natural cubic splines is that in our case is that the interpolants are required to be unit-speed. Because of this additional requirement, once \(T_0\) is given, fixing the rest of the sequence \(T\) is equivalent to prescribing the lengths \(L_j\) of the interpolant between all consecutive data points. In these circumstances, the main contribution to interpolation by elastic splines is Theorem 4.1 of \[17\], where a nonlinear system of equations is given for determining elastic splines in great generality.

The key to solving such a nonlinear system is a suitable initial guess, and \[17\] says very little about how this might be constructed. The present paper contributes something in this respect, although strictly speaking our focus is primarily on spiral splines, whereas \[17\] is concerned with elastic splines. Our contribution includes a list of initial guesses for spiral optimisation, whose outcome depends critically on an unspecified initial guess in a high dimensional space.

We also mention the discrete elastic splines of \[3\], where a discrete analogue of \(J\) is optimised with respect to a variable finite sequence of points approximating \(y\). In effect optimisation of \(J\) with respect to \(y\) is replaced by a large finite-dimensional optimisation, whose outcome depends critically on an unspecified initial guess in a high dimensional space.

In summary, reparameterised polynomial splines and clothoidal splines are unsuitable for interpolating \(Y\) by \(T\). Elastic splines come with non-negligible computational issues, especially the comprehensive generation of suitable initial guesses.

### 3. Admissible Samplings

Let \(0 < A_m < A_M\) be given, together with \(B_i > 0\) for \(2 \leq i \leq 5\), \(C > 0\), and an integer \(n \geq 2\). Given \(T_0 < b\) let \(x : [T_0, b] \to \mathbb{R}^2\) be an unknown \(C^0\) unit-speed curve satisfying \(\|x^{(i)}\| \leq B_i\) for \(2 \leq i \leq 5\), where \(x^{(i)}\) denotes the \(i\)th derivative of \(x\), and \(\|\|\) is the uniform norm.

For \(\epsilon > 0\) let \(T_{\epsilon}\) be the set of finite sequences \(T\) of reals \(T_0 < T_1 < T_2 < \ldots < T_n \leq b\) such that \(A_m \epsilon < L_j < A_M \epsilon\) for all \(1 \leq j \leq n\). Given \(T \in T_{\epsilon}\), define \(Y(T) = (Y_0, Y_1, \ldots, Y_n) \in (\mathbb{R}^3)^n\) by \(Y_j = x(T_j)\) for \(0 \leq j \leq n\). Our task is to find a \(C^2\) unit-speed curve \(y : [T_0, T_n] \to \mathbb{R}^2\) with \(y^{(2)}(T_0) = 0 = y^{(2)}(T_n)\) interpolating \(Y\) at \(T\).

Because \(x\) is unit-speed \(q_j := \langle Y_j - Y_{j-1} \rangle / L_j\) has length no greater than 1. Indeed \(\|q_j\| < 1\) except possibly when \(x\) has linear segments. It also follows from our hypotheses that the \(Y_j\) are separated by no more than \(A_M \epsilon\), and so they all lie inside the disc of radius \(nA_M \epsilon\) centred on \(Y_0\). Looking forward, these conditions are needed to guarantee the conclusions of Theorem 1, but may be troublesome to check in applications. In such cases (including the examples of the present paper) Theorem 1 may be taken to assert that when its conclusions are false, the sampling of \(x\) is too irregular or too sparse. The cue, which seems to be rarely needed, is to sample more regularly and more often from \(x\), to generate better data \(Y\).

For \(\epsilon\) sufficiently small \(q_j \neq 0\), and then we define

\[
 k_j := \frac{\sqrt{12(1 - \|q_j\|^2)^\ell}}{L_j\|q_j\|} \quad \text{for} \quad 1 \leq j \leq n.
\]

Because \(\|x^{(i)}\| \leq B_i\) for \(i = 2, 3, 4\), a calculation using Taylor polynomials shows that the \(k_j\) are also bounded above, independently of \(x\) and \(T\), provided \(\epsilon\) is sufficiently small. We say \(T\) is admissible when \(k_j \geq C\) for \(1 \leq j \leq n\).

Let \(\theta : [T_0, T_n] \to \mathbb{R}\) be a \(C^1\) cubic spline with knots \(T_j\) such that \(\theta^{(1)}(T_0) = 0 = \theta^{(1)}(T_n)\). For \(1 \leq j \leq n\) and all \(s \in [0, L_j]\) write \(\theta_j(s) := \theta(T_{j-1} + s) = a_j + b_j s + c_j s^2 + d_j s^3\).

Besides the two auxiliary end conditions,

\[
 b_1 = 0 = b_n + 2c_n L_n + 3d_n L_n^2,
\]

there are \(2n - 2\) conditions for \(\theta\) to be \(C^1\), namely

\[
 a_{j+1} = a_j + b_j L_j + c_j L_j^2 + d_j L_j^3
\]

\[
 b_{j+1} = b_j + 2c_j L_j + 3d_j L_j^2,
\]

where \(1 \leq j \leq n - 1\). So we have \(2n\) affine inequality constraints on \(4n\) coefficients \(a_j\), \(b_j\), \(c_j\), \(d_j\).

The \(C^1\) spline \(\theta\) determines a natural second order spiral spline, namely a \(C^2\) unit-speed curve \(y_\theta : [T_0, T_n] \to \mathbb{R}^2\) given by

\[
y_\theta(t) := Y_0 + \int_T^t (\cos \theta(s), \sin \theta(s)) \, ds.
\]

Then \(y_\theta^{(2)}(T_0) = 0 = y_\theta^{(2)}(T_n)\), and we are interested in finding \(\theta\) such that \(y_\theta\) interpolates \(Y\) at \(T\), namely for \(1 \leq j \leq n\),

\[
 \int_{T_j}^{T_{j+1}} (\cos \theta_j(s), \sin \theta_j(s)) \, ds = Y_j - Y_{j-1}.
\]

Condition \(\theta\) amounts to another \(2n\) conditions on the \(4n\) coefficients, making \(4n\) conditions in total. There may be multiple solutions for \(\theta\) because equations \(0\) are non-affine.
Theorem 1. For \( \epsilon \) sufficiently small, there are \( 2^n \) choices of \( \theta \) for which \( y_\theta \) interpolates \( Y \) at \( T \). For one such \( \theta \) and all \( t \neq T_j \) where \( 1 \leq j \leq n - 1 \),

\[
y^{(i)}_\theta(t) = \hat{x}^{(i)}(t) + O(\epsilon^{i-1}) \quad \text{for } 0 \leq i \leq 4.
\]

The proof begins in \( 3 \) where \( T \in \mathcal{T} \) is admissible and \( y_\theta \) interpolates \( Y \) at \( T \) where \( \theta \) satisfying \( 2 \) is not yet determined. The main effort is to find the \( 2^n \) estimates \( \theta = \theta + O(\epsilon) \). The \( \theta \) are found by supplementing the \( 2n \) affine equations \( 2, 3, 4, 5 \) for the \( a_j, b_j, c_j, d_j \) by systems of \( 2n \) asymptotic equations \( 6, 7, 8, 9 \), with one such system for each of \( 2^n \) parameters \( \sigma = (\pm 1, \pm 1, \pm 1, \ldots, \pm 1) \). Here the changes in sign correspond roughly to changes in concavity of \( y_\theta \). Algorithms for \( \theta \) are given in \( 3 \) for \( n = 2 \) and in \( 6 \) for \( n \geq 3 \). The proof is completed in \( 7 \) using the \( \theta \) with the implicit function theorem.

In \( 8 \) we turn the proof of Theorem 1 into an effective algorithm for finding \( \theta \). First, interpolation by \( y_\theta \) of \( Y \) at \( T \) is rewritten as a system of \( 2n \) equations for \( (u, v) \in \mathbb{R}^{2n} \). Using standard nonlinear solvers, such as Mathematica’s FindRoot, the critical ingredient for finding \( (u, v) \) is a good initial guess. It follows from Theorem 1 that, for some choice of \( \sigma \), \( (u, v) \in \mathbb{R}^{2n} \) corresponding to \( \theta \) is suitable.

Except for the significant condition that interpolants should be unit-speed, our problem has a lot in common with interpolation by natural cubic polynomial splines, and there are some traces of the classical method in our algorithm. Whereas calculating polynomial splines requires solution of a single tridiagonal linear system, our algorithm requires solution of two tridiagonal linear systems for each initial guess. So, for any given \( \sigma \), very little effort is needed for us to calculate \( (u, v) \in \mathbb{R}^{2n} \). Because there are \( 2^n \) choices of \( \sigma \), the total computational effort is small for moderate values of \( n \) (certainly for \( n \leq 5 \)), and grows exponentially with \( n \).

Even for \( n \leq 5 \), \( 2^n \) choices of \( \sigma \) may be too many design parameters. One option is to limit attention to \( \hat{\sigma} \), such as the elastic energy, namely the restriction of \( J \) defined in \( 4 \) to the space of unit-speed interpolants.

4. ASYMPTOTIC EQUATIONS

Let \( U \) be an open neighbourhood of \( c := (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n) \in \mathbb{R}^n \) where \( m = 4n \), and the \( a_j, b_j, c_j, d_j \) are the coefficients for a \( C^4 \) cubic spline \( \theta \) where \( y_\theta \) interpolates \( Y \) at \( T \). We also require the auxiliary end conditions \( 4 \). Here \( c \) is unknown, and determines the \( q_j = (Y_j - Y_{j-1})/L_j \) for \( 1 \leq j \leq n \), because of the interpolation condition \( 1 \). Write \( r_j(\cos(\omega_j), \sin(\omega_j)) := q_j \), where \( r_j > 0 \) and \( \omega_j \in \mathbb{R} \).

Lemma 1. For some \( 0 < \delta_0 < \delta_1 < 1 \) depending only on \( A_{m, A_M}, B_2, B_1, B_1, C \) and \( c \), we have \( r_j \in [\delta_0, \delta_1] \) for all \( 1 \leq j \leq n \).

Proof: Because \( T \) is admissible, and because the \( k_j \) are uniformly bounded above, \( 0 < \delta_0 \leq k_j \leq D_1 \), where \( D_0, D_1 \) depend only on \( c \) and the given constants. So

\[
D_0^2 L_j^2 r_j^2 \leq 12(1 - r_j^2) \leq D_1^2 L_j^2 \Rightarrow (1 + \frac{D_0^2 L_j^2}{12}) r_j^2 \leq 1 \leq (1 + \frac{D_1^2 L_j^2}{12}) r_j^2 \Rightarrow \frac{12}{12 + D_0^2 A_{m, A_M} c^2} \leq r_j^2 \leq \frac{12}{12 + D_1^2 A_{m, A_M} c^2}.
\]

Considering the \( r_j \) and likewise (modulo \( 2 \pi \)) the \( \omega_j \) as functions of \( c \), equation \( 4 \) and polynomial expansion in \( L_j \) to degree \( 4 \) yield

\[
\begin{bmatrix} a_j & -b_j & \cos(\omega_j) \\ -\beta_j & a_j & \sin(\omega_j) \end{bmatrix} := \frac{r_j}{1 - \frac{b_j c_j L_j^4}{6}} + \frac{b_j c_j L_j^4}{4} - \frac{(b_j^2 - 12 b_j^2 - 24c_j d_j L_j^4)}{120} \quad \text{and} \quad \beta_j := \frac{b_j c_j L_j^4}{2} + \frac{c_j L_j^4}{3} - \frac{(b_j^2 - 6d_j L_j^4)}{24} - \frac{b_j c_j L_j^4}{10}.
\]

From \( 5 \) we find that

\[
\cos(\omega_j - \omega_j) \sin(\omega_j - \omega_j) = \frac{(a_j, \beta_j)}{(\alpha_j, \beta_j)} + O(\epsilon^5).
\]

Further calculation shows that, for some integer \( m_j \),

\[
a_j = 2m_j \pi + \omega_j - \frac{b_j c_j L_j^4}{2} - \frac{c_j L_j^4}{3} - \frac{d_j L_j^4}{4} + \frac{b_j^2 c_j L_j^4}{360} + O(\epsilon^5).
\]

Substituting from \( 7 \) for \( a_j, \beta_j \) in \( 5 \), \( \omega_{j+1} - \omega_j - (m_{j+1} - m_j) \pi = O(\epsilon) \). Because \( T \in \mathcal{T} \), we can choose \( \omega_{j+1} - \omega_j = O(\epsilon) \). Then, for small \( \epsilon \), the \( m_j \) are all equal, and there is no loss in taking all \( m_j = 0 \). By \( 6 \),

\[
a_j = \omega_j - \frac{b_j c_j L_j^4}{2} - \frac{c_j L_j^4}{3} - \frac{d_j L_j^4}{4} + O(\epsilon^5).
\]
Substituting from (9) for \( a_j, a_{j+1} \) in (3) for \( 1 \leq j \leq n - 1 \), and using \( b_{j+1} = b_j + O(\epsilon) \) from (4),

\[
\begin{align*}
\frac{(b_{j+1}L_{j+1} + b_jL_j)}{2} + \frac{(c_jL_j^3 + 2c_jL_j^2)}{3} + \frac{(d_{j+1}L_{j+1}^3 + 3d_jL_j^3)}{4} &= \omega_{j+1} - \omega_j + O(\epsilon^2).
\end{align*}
\]

Eliminating \( a_j \) from (7),

\[
(b_j + c_jL_j + \frac{9}{10} d_jL_j^2)^2 + \frac{c_j^2L_j^2}{15} + \frac{b_j^2L_j^2}{20} = k_j^2 + O(\epsilon^2) \Rightarrow b_j + c_jL_j = \sigma_j k_j + O(\epsilon^2)
\]

where \( \sigma_j^2 = 1 \). Write \( \sigma := (\sigma_1, \sigma_2, \ldots, \sigma_n) \). Because \( b_j + c_jL_j = \sigma_j k_j + O(\epsilon^2) \),

\[
k_j^2 = \frac{1}{\sigma_j} b_j + c_jL_j
\]

In particular, from (2),

\[
\begin{align*}
\frac{(c_jL_j + \frac{9}{10} d_jL_j^2)^2}{3} &= k_j^2(1 - \frac{k_j^2L_j^2}{20}) + O(\epsilon^2),
\end{align*}
\]

\[
\begin{align*}
\frac{(c_nL_n + \frac{21}{10} d_nL_n^2)^2}{3} &= k_n^2(1 - \frac{k_n^2L_n^2}{20}) + O(\epsilon^2). \quad \Rightarrow
\end{align*}
\]

\[
\begin{align*}
c_jL_j + \frac{9}{10} d_jL_j^2 &= \rho_j + O(\epsilon^2) \quad \text{where} \quad \rho_j := \sigma_j k_j \sqrt{1 - \frac{k_j^2L_j^2}{20}},
\end{align*}
\]

\[
\begin{align*}
-c_nL_n - \frac{21}{10} d_nL_n^2 &= \rho_n + O(\epsilon^2) \quad \text{where} \quad \rho_n := \sigma_n k_n \sqrt{1 - \frac{k_n^2L_n^2}{20}},
\end{align*}
\]

and \( \epsilon \) is small enough for the terms inside the square root signs to be bounded away from 0. For \( 2 \leq j \leq n - 1 \), from (11) and (12),

\[
(b_j + c_jL_j + \frac{9}{10} d_jL_j^2)^2 = k_j^2 \left(1 - \frac{k_j^2L_j^2}{20}\right) - \frac{(b_{j+1} - b_j)^2}{60} + O(\epsilon^2) \Rightarrow
\]

\[
(b_j + c_jL_j + \frac{9}{10} d_jL_j^2)^2 = \sigma_j \sqrt{k_j^2 \left(1 - \frac{k_j^2L_j^2}{20}\right) - \frac{(b_{j+1} - b_j)^2}{60}} + O(\epsilon^2),
\]

Again, \( \epsilon \) is so small that the terms inside the square root signs to be bounded away from 0. Then \( \theta(t) = \theta(t) + O(\epsilon^4) \), where \( \theta \) is obtained by substituting the estimates for \( a_j, b_j, c_j, d_j \) in \( \theta \), and the unit-speed curve \( y_j : [T_0,T_n] \to \mathbb{R}^2 \) satisfies \( y_j(T_j) = y_j + O(\epsilon^2) \) for \( 1 \leq j \leq n \). As for \( y_0 \), the curvature of \( y_0 \) vanishes at \( T_0 \) and \( T_n \).

Notice, by the way, that \( \theta_j^{(1)}(t) = b_j + O(\epsilon) = \sigma_j k_j + O(\epsilon) \). So, for sufficiently small \( \epsilon \), a change in sign of \( \sigma_j \) corresponds to a change in concavity of \( y_j(T_j) \).

Although \( y_j \) does not exactly interpolate \( Y \) at \( T_j \), it is true that \( \tilde{y}_j(T_j) = y_j \) for all \( j \), where

\[
\tilde{y}_j(t) := y_{j-1} + \int_{T_{j-1}}^{T_j} (\cos \theta_j(s), \sin \theta_j(s)) \, ds
\]

for \( t \in [T_{j-1}, T_j] \). For this reason, although it is usually not exactly \( C^0 \) at \( T_1, T_2, \ldots, T_n \), the interpolating curve \( \tilde{y}_j \) may sometimes be preferable to \( y_j \) (of course \( y_0 = y_0 \)).

5. \( n = 2 \)

Consider first the case where \( n = 2 \). From the auxiliary conditions and equations (3), (12), (13), (10), (9), we find that

\[
(a_1, a_2) = (\tilde{a}_1, \tilde{a}_2) + O(\epsilon^4), \quad (b_1, b_2) = (\tilde{b}_1, \tilde{b}_2) + O(\epsilon^3), \quad (c_1, c_2) = (\tilde{c}_1, \tilde{c}_2) + O(\epsilon^2), \quad (d_1, d_2) = (\tilde{d}_1, \tilde{d}_2) + O(\epsilon),
\]

where...
\begin{align*}
\hat{a}_1 &= \frac{12(2L_2 \omega_1 + 3L_2 \omega_2 + L_2 \omega_2) - 5J_1(4J_1 L_1 + 3J_1 L_2 + 3J_2 L_2)}{60(L_1 + L_2)}, \\
\hat{a}_2 &= \frac{12(L_1 \omega_2 + L_2 \omega_1) + 5L_1 L_2 (p_1 - p_2)}{12(L_1 + L_2)}, \\
\hat{b}_1 &= 0, \\
\hat{b}_2 &= \frac{24(\omega_2 - \omega_1) - 10(p_2 L_2 + p_1 L_1)}{3(L_1 + L_2)}, \\
\hat{c}_1 &= \frac{12(\omega_1 - \omega_2) + 5(2p_1 L_1 + p_1 L_2 + p_2 L_2)}{2L_1(L_1 + L_2)}, \\
\hat{c}_2 &= \frac{84(\omega_1 - \omega_2) + 5(7p_1 L_1 + 3p_2 L_2)}{6L_2(L_1 + L_2)}, \\
\hat{d}_1 &= \frac{60(\omega_2 - \omega_1) - 5(8p_1 L_1 + 3p_1 L_2 + 5p_2 L_2)}{9L_1^2(L_1 + L_2)}, \\
\hat{d}_2 &= \frac{60(\omega_2 - \omega_1) - 5(16p_1 L_1 + 3p_2 L_1 + 8p_2 L_2)}{9L_1^2(L_1 + L_2)}.
\end{align*}

Equations (15), (16) still hold when the \(a_j, b_j, c_j, d_j\) are replaced by the estimates \(\hat{a}_j, \hat{b}_j, \hat{c}_j, \hat{d}_j\), and so \(\theta\) is \(C^1\). The approximately-interpolating curve \(\hat{y}_\theta : [T_0, T_2] \to \mathbb{R}^2\) is therefore \(C^2\).

**Example 1.** Choose \(y_0 = (0, 0), y_1 = (0.48, 0.12), y_2 = (1, 0)\), and \(T_0 = 0, T_1 = 0.5, T_2 = 1\). The four approximately \(C^0\) interpolating curves \(\hat{y}_\theta\) depending on \(\sigma = (\pm 1, \pm 1)\) are shown, together with the \(y_i\) and the elastic energies \(J(\sigma)\), in Figures 1, 2, 3, and 4. We have \(J_1(1, -1) < J(-1, -1) < J(1, -1) < J(1, 1)\). As might be expected from our hypotheses of moderate curvature and moderate derivatives of curvature, \(\hat{y}_\theta\) is very nearly \(C^2\) for the lower values of \(J\) in Figures 1, 3, and 4. Figure 2, where \(J\) is largest, shows obvious discontinuities at \(T_0\) and \(T_2\). Figure 4 shows an obvious discontinuity at \(T_2\). The four \(\theta\) were computed in a total of 0.000054 seconds, with Mathematica on a mid-2015 MacBook Pro (2.5GHz Intel Core i7).

**Figure 1.** \(n = 2\), and \(\hat{y}_\theta\) with \(\sigma = (1, 1)\) in Example 1. \(J = 17.60\)

**Figure 2.** \(n = 2\), and \(\hat{y}_\theta\) with \(\sigma = (-1, 1)\) in Example 1. \(J = 10.59\)
For $n \geq 3$, the auxiliary constraints $b_1 = 0 = b_n + 2c_nL_n + 3d_nL_n^2$ together with equations (4), (10), (13), (12) and (13), give

\begin{align}
(23) \quad b_{j+1} - b_j &= 2c_jL_j + 3d_jL_j^2 + O(\epsilon^3), \quad \text{for } 1 \leq j \leq n - 1, \\
(24) \quad \frac{b_{j+1} + b_j}{2} + \frac{c_{j+1}L_{j+1}^2 + 2c_jL_j^2 + c_jL_j}{3} + \frac{d_{j+1}L_{j+1}^3 + 3d_jL_j^3}{4} = \omega_{j+1} - \omega_j + O(\epsilon^3), \quad \text{for } 1 \leq j \leq n - 1, \\
(25) \quad b_j + c_jL_j + \frac{9}{10}d_jL_j^2 &= \rho_j(b_j, b_{j+1}) + O(\epsilon^3), \quad \text{for } 2 \leq j \leq n - 1, \\
(26) \quad c_jL_j + \frac{9}{10}d_jL_j^2 &= \rho_j + O(\epsilon^3), \\
(27) \quad -c_nL_n - \frac{21}{10}d_nL_n^2 &= \rho_n + O(\epsilon^3),
\end{align}

which is a system of 3n equations for 3n unknowns $b_j, c_j, d_j$ where $1 \leq j \leq n$. Here, unlike $\rho_1, \rho_n$, $\rho_j$ depend not only on $\sigma$, but also on the $b_j, b_{j+1}$ for $2 \leq j \leq n - 1$. As usual, $\epsilon$ is chosen small enough for the term inside the square root sign to be bounded away from 0.

6.1. $O(\epsilon^2)$ Estimates of the $b_j$.

From the auxiliary conditions, $b_1 = 0$ and $b_n = -2c_nL_n + O(\epsilon^2) = 2\sigma_nL_n + O(\epsilon^2)$ using (27). From (11), $2c_jL_j = b_{j+1} - b_j + O(\epsilon^2)$. Substituting in equation (24), $T^{(2)}b = R^{(2)} + O(\epsilon^3)$ where $b$ is the n-dimensional column vector whose $j$th coordinate is $b_j$, $T^{(2)}$ is the $n \times n$ tridiagonal matrix

\[
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
L_1 & 2(L_1 + L_2) & L_2 & \cdots & 0 \\
0 & L_2 & 2(L_2 + L_3) & \cdots & 0 \\
0 & 0 & L_3 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & L_{n-2} \\
0 & 0 & 0 & 0 & L_{n-1} \\
L_{n-1} & L_{n-2} & 2(L_{n-2} + L_{n-1}) & \cdots & L_{n-1} \\
2(L_{n-1} + L_n) & L_{n-1} & 2(L_{n-2} + L_{n-1}) & \cdots & \cdots \\
\end{bmatrix}
\]

and

\[
R^{(2)} := 6 \begin{bmatrix}
0 \\
\omega_2 - \omega_1 \\
\omega_3 - \omega_2 \\
\omega_4 - \omega_3 \\
\cdots \\
\omega_{n-1} - \omega_{n-2} \\
\omega_n - \omega_{n-1}
\end{bmatrix}
\]

Solving $T^{(2)}y^{(2)} = R^{(2)}$ we have $b = y^{(2)} + O(\epsilon^2)$.
6.2. $O(\epsilon^3)$ Estimates of the $b_j$.

Equations (23), (25) are used to solve for $c_jL_j$ and $d_jL_j^2$ for $2 \leq j \leq n - 1$ in terms of $b_2, \ldots, b_n$, namely

\begin{align}
    c_jL_jF L_j &= -\frac{3b_{j+1} - 7b_j}{4} + \frac{5\rho_j}{2} + O(\epsilon^3), \\
    d_jL_j^2 &= \frac{5(b_{j+1} + b_j)}{6} - \frac{5\rho_j}{3} + O(\epsilon^3),
\end{align}

supplemented as follows, using (26), (27), and the auxiliary conditions:

\begin{align}
    c_1L_1 &= \frac{7b_2 + 10\rho_2}{4} + O(\epsilon^3), \\
    c_nL_n &= -\frac{7b_n + 10\rho_n}{4} + O(\epsilon^3), \\
    d_1L_1^2 &= \frac{5(b_2 - 2\rho_2)}{6} + O(\epsilon^3), \\
    d_nL_n^2 &= \frac{5(b_n - 2\rho_n)}{6} + O(\epsilon^3).
\end{align}

Substituting for the $c_jL_j, d_jL_j^2$ in terms of $b_2, \ldots, b_n$ in (24), we find that $T^{(3)}b = R^{(3)}(b) + O(\epsilon^3)$, where $T^{(3)}$ is the $n \times n$ tridiagonal matrix

\[
    \begin{bmatrix}
        1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
        -L_1 & 3(L_1 + L_2) & -L_2 & 0 & 0 & 0 & \cdots & 0 \\
        0 & -L_2 & 3(L_2 + L_3) & -L_3 & 0 & 0 & \cdots & 0 \\
        \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vvdots & \vdots \\
        0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
    \end{bmatrix}
\]

and

\[
    R^{(3)}(b) := \begin{bmatrix}
        24(\omega_2 - \omega_1) - 10(L_1\rho_1 + L_2\rho_2) \\
        24(\omega_2 - \omega_1) - 10(L_2\rho_2 + L_3\rho_3) \\
        24(\omega_4 - \omega_3) - 10(L_3\rho_3 + L_4\rho_4) \\
        \vdots \\
        24(\omega_{n-1} - \omega_n) - 10(L_{n-2}\rho_{n-2} + L_{n-1}\rho_{n-1}) \\
        24(\omega_n - \omega_{n-1}) - 10(L_{n-1}\rho_{n-1} + L_n\rho_n)
    \end{bmatrix}
\]

Solving $T^{(3)}b = R^{(3)}(b)$, we have $b = b^{(3)} + O(\epsilon^3)$.

Substituting for $b$ in equations (26), (29), (31), (32), (33), the $c_jL_j$ and $d_jL_j^2$ are also found to $O(\epsilon^3)$. The $a_j$ are then found to $O(\epsilon^4)$ by substitution in (19).

As before, $\theta(t) = \theta(t) + O(\epsilon^3)$, and the unit-speed curve $y_b : [T_0, T_n] \rightarrow \mathbb{R}^2$ satisfies $y_b(T_i) = y_i + O(\epsilon^3)$.

**Example 2.** For $n = 5$ take $y_0 = (0, 0)$, $y_1 = (0.5, 0.15)$, $y_2 = (1, 0)$, $y_3 = (1.5, -0.1)$, $y_4 = (2, -0.2)$, $y_5 = (2.5, -0.5)$, with $T_0 = 0, T_1 = 0.55, T_2 = 1.1, T_3 = 1.7, T_4 = 2.4, T_5 = 3.0$. The 32 possibilities for $a$ yield curves $y_\xi$, with elastic energies $J$ ranging from 20.71 to 100.60. The time to compute all 32 functions $\theta$ totalled 0.0114 seconds. The three approximately C4 interpolating curves $\hat{y}_\xi$ with lowest $J$ appear nearly C4, as shown in Figure 8. There are only some small indications of discontinuity at $T_i$ in Figure 8. Some discontinuities of $\hat{y}_\xi$ with larger $J$ are more obvious in Figures 9, 10, especially at $T_4, T_5$. A small discontinuity can also be seen at $T_1$ in Figure 8.

**Example 3.** For $0 \leq j \leq n = 7$ take $T_j = j\pi/10$ and $y_j = (\cos(T_j), \sin(T_j))$. The circular arc from $(1, 0)$ to $(\cos(7\pi/10), \sin(7\pi/10))$ interpolates $Y$ at $T$ with elastic energy $\pi/10 \approx 2.12$. The 128 estimates $\hat{\theta}$ are found in a total of 0.051 seconds, and it is extremely difficult to tell by eye that the $\hat{y}_\xi$ are not C2. Their elastic energies $J$ range from 2.12 for $a = (1, 1, 1, 1, 1, 1, 1)$ to 37.21 for $a = (-1, -1, -1, -1, -1, -1, -1)$. For $a = (1, 1, 1, 1, 1, 1, 1)$ it is extremely difficult to distinguish by eye between $\hat{y}_\xi$ and the circular arc where $J = 7\pi/10 \approx 2.12$. Figure 8 shows $\hat{y}_\xi$ for $a = (-1, -1, -1, -1, -1, -1, -1)$, where $J$ is largest.

**7. Proof of Theorem 3**

For sufficiently small $\epsilon > 0$, we have $\gamma = \hat{\gamma} + O(\epsilon)$ where

\[
    \hat{\gamma} := (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_n, \hat{b}_1, \hat{b}_2, \ldots, \hat{b}_n, \hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n, \hat{d}_1, \hat{d}_2, \ldots, \hat{d}_n) \in U \subseteq \mathbb{R}^n.
Figure 5. $n = 5$, and $\hat{y}_j$ with $\sigma = (-1, -1, 1, 1)$ in Example 2. $J = 20.71$

Figure 6. $n = 5$, and $\hat{y}_j$ with $\sigma = (-1, 1, -1, -1)$ in Example 2. $J = 21.51$

Figure 7. $n = 5$, and $\hat{y}_j$ with $\sigma = (1, -1, 1, -1)$ in Example 2. $J = 23.31$

Figure 8. $n = 5$, and $\hat{y}_j$ with $\sigma = (1, 1, 1, 1)$ in Example 2. $J = 100.60$

is expressed in terms of $\sigma = (\pm 1, \pm 1, \ldots, \pm 1)$ and the $q_i \in \mathbb{R}^2$, according to the formulae of §5 for $n = 2$, and the equations of §6 for $n \geq 3$. Tracing through the construction of $\hat{\theta}$ we see that

\begin{equation}
\gamma - \hat{\gamma} = \epsilon g(\gamma, \epsilon)
\end{equation}

where the derivatives of $g$ are uniformly bounded in norm, independently of $\epsilon$, by quantities determined by the constants $A_m, A_M, B_2, B_3, B_4, B_5, C$. Once $\sigma$ is given we may consider $\hat{\gamma}$ as a function of $\gamma \in U$, by using (10) to express the $q_i$ in terms of $c$.

By (34) and the implicit function theorem, for $\epsilon$ sufficiently small, the assignment $\gamma \mapsto \hat{\gamma}$ is a local diffeomorphism. So in some small open neighbourhood $V$ contained in $U$, $\gamma$ is determined uniquely by $\hat{\gamma}$. As previously noted, $\hat{\gamma}$ is determined by
Figure 9. \(n = 5\), and \(\hat{y}_\theta\) with \(\sigma = (-1, 1, 1, 1, 1)\) in Example \(2\) \(J = 83.22\)

Figure 10. \(n = 5\), and \(\hat{y}_\theta\) with \(\sigma = (-1, -1, 1, 1, 1)\) in Example \(2\) \(J = 66.56\)

Figure 11. \(n = 7\), and \(\hat{y}_\theta\) with \(\sigma = (-1, -1, -1, -1, -1, -1, 1)\) in Example \(3\) \(J = 37.21\)

\(\sigma\) and the \(q_j\). So \(\gamma \in V\) is determined uniquely by \(\sigma\) and \(T\). Therefore, given \(T \in T\) for sufficiently small \(\epsilon\), the possibilities for interpolating curves \(y_\theta\) correspond to the \(2^n\) choices for \(\sigma\). \(\square\)

8. Closing Gaps

Although the \(\hat{y}_\theta\) are nearly \(C^2\), the \(C^1\) spline \(\hat{\theta}\) is only an estimate of the unknown \(\theta : [T_0, T_n] \rightarrow \mathbb{R}\) where \(y_\theta\) interpolates \(Y\) at \(T\). So \(\hat{y}_\theta\) is almost always discontinuous at \(T_1, T_2, \ldots, T_n\) even though, as in Examples \(1\), \(2\), \(3\) the discontinuities are often too small to be visible. Sometimes \(\hat{y}_\theta\) may be all that is needed.

In any event \(\theta\) can be found from \(\hat{\theta}\) with only a small additional effort. Standard packages for solving nonlinear systems of equations are effective when given a suitable initial guess. As shown in \(4\), \(5\), \(6\) and as seen from the previous examples, \(\hat{\theta}\) is a useful guess for \(\theta\); in the sense that \(\hat{y}_\theta\) has the kinds of properties that might be required for \(y_\theta\), namely moderate curvature and moderate derivatives of curvature. So we formulate the conditions on \(\theta\) as a nonlinear system of equations, then solve numerically with \(\theta\) as an initial guess. This second step is easier than finding \(\hat{\theta}\) and simpler to code.\(^{13}\)

\(^{13}\)The runtime is longer though, at least in our implementation.
Define \( z : \mathbb{R}^n \times \mathbb{R}^n \to (\mathbb{R}^2)^n \) by \( z_j(u, v) := \int_0^L e^{\theta(t)} \, dt \) for \( 1 \leq j \leq n \). Here \( u, v \in \mathbb{R}^n \), \( \theta \in C \subseteq \mathbb{R}^2 \) corresponds to (0,1) with \( p^2 = -1 \), and

\[
\begin{align*}
a_1 &= v_1, \\
a_j &= u_{j-1}, & &\text{for } 2 \leq j \leq n, \\
b_1 &= 0, \\
b_j &= \frac{v_j}{L_j}, & &\text{for } 2 \leq j \leq n, \\
c_1 &= -L_1 v_1^2 - 3L_1_2 (u_1 - v_1), \\
c_j &= -L_j v_{j+1}^2 + 2L_j v_j v_{j+1} - 3L_{j+1} (u_j - v_{j+1}), & &\text{for } 2 \leq j \leq n - 1, \\
c_n &= -3u_n + v_n, \\
d_1 &= \frac{L_1 v_1^2 - 2L_2 (u_1 - v_1)}{L_1^2 L_2}, \\
d_j &= \frac{L_j v_{j+1}^2 + L_{j+1} v_j v_{j+1} - 2L_{j+1} (u_j - v_{j+1})}{L_j^2 L_{j+1}}, & &\text{for } 2 \leq j \leq n - 1, \\
d_n &= \frac{u_n}{L_n^2}.
\end{align*}
\]

As may easily be verified,

**Lemma 2.** With \( a_1, b_1, c_1, d_1 \) defined as above, equations \( 1, 2 \), hold, and so do the auxiliary conditions. \( \square \)

For \( y_y \) to interpolate \( Y \) at \( T \), namely to satisfy equation \( 1 \), we must find \( u, v \in \mathbb{R}^n \) so that \( z_j(u, v) = Y - Y_{j-1} \) for \( 1 \leq j \leq n \). Our initial guess is \((u, v) = (\hat{u}, \hat{v}) + O(\epsilon)\) where \((\hat{u}, \hat{v})\) is obtained from the coefficients of \( \theta \) in \( \hat{u} \) or \( \hat{v} \) according as \( n = 2 \) or \( n \geq 3 \). In practice we replace the \( z_j \) by numerical integrals, using the composite Simpson’s Rule.

For \( n \geq 2, \epsilon \) sufficiently small, and \( 1 \leq p \leq 2^n \), the algorithm for finding the \( p \)-th natural spiral spline \( y_y \) interpolating \( Y \) at \( T \) is summarised as follows (the first 5 steps are independent of \( p \) and the last 6 may sometimes be unnecessary).

1. Writing \( L_j := T_j - T_{j-1} \) for \( 1 \leq j \leq n \), let the \( n \times n \) tridiagonal matrix \( T(2) \) in \( [5] \).
2. For \( 1 \leq j \leq n \) set \( q_j := (Y_j - Y_{j-1})/L_j \) and check that its length \( r_j \) lies in \((0,1)\). Set \( k_j := \sqrt{12(1 - r_j^2)/(L_j r_j)} \).
3. Write \( q_1 := \hat{r}_1 \cos \omega_1, \sin \omega_1 \). For \( 2 \leq j \leq n \) let \( \omega_j \) be nearest \( \omega_1 \) while satisfying \( q_j = k_j (\cos \omega_j, \sin \omega_j) \).
4. If \( n \geq 3 \) form the \( n \times n \) tridiagonal matrix \( T(3) \) in \( [6] \), the \( n \)-dimensional column vector \( R(2) \) in \([6,1]\) and solve the linear system \( T(3) R(2) = R(2) \) for the \( n \)-dimensional column vector \( R(2) \).
5. Form the \( n \times n \) tridiagonal matrix \( T(3) \) in \([6,2]\).
6. For \( 1 \leq j \leq n \) set \( \sigma_j := (-1)^j \) where \( p_1 p_2 p_3 \ldots p_n \) is the binary representation of \( p \).
7. Set \( p_1 = \sigma_1 k_1 / \sqrt{1 - k_1^2 L_j^2} / 20 \), \( p_m = \sigma_m k_1 / \sqrt{1 - k_1^2 L_j^2} / 20 \) and \( p_j = \sigma_j \sqrt{k_j^2 (1 - k_j^2 L_j^2) / 20} - (k_{j+1}^2 - k_j^2)^2 / 60 \) where \( 2 \leq j \leq n - 1 \).
8. Form the \( n \)-dimensional column vector \( R(3) \) in \([7,2]\) and solve the linear system \( T(3) R(3) = R(3) \) for the \( n \)-dimensional column vector \( R(3) \).
9. Take \( \hat{c}_j, \hat{d}_j \) to be the \( n \)-dimensional column vectors \( c, d \) found by substituting \( \hat{b}_j \) for \( b_j \) in the right hand sides of equations \( [21], [22], [23], [24], [25], [26], [27], [28] \).
10. For \( 1 \leq j \leq n \) define \( \beta_j : [0, L_j] \to \mathbb{R}^n \) by \( \beta_j(s) := \hat{a}_j + \hat{b}_j s + \epsilon_j s^2 + \epsilon_j s^3 \) where \( s \in [0, L_j] \). Let \( \theta : [T_0, T_n] \to \mathbb{R} \) be the track-sum of the \( \beta_j \), namely \( \theta(t) = \beta_j(t - T_{j-1}) \) for \( t \in [T_{j-1}, T_j] \).
11. If the natural second-order spiral spline \( y_y \) interpolates \( Y \) at \( T \) to sufficient accuracy, stop.
12. Using \( \hat{a}_j, \hat{b}_j, \hat{c}_j, \hat{d}_j \) in place of \( a, b, c, d \), take \((\hat{u}, \hat{v}) \in \mathbb{R}^n \times \mathbb{R}^n \) to be \((u, v)\) as determined from the formulae preceding Lemma 2.
13. For \( 1 \leq j \leq n \) define \( z_j : \mathbb{R}^n \times \mathbb{R}^n \to (\mathbb{R}^2)^n \) by the formula in \([6]\) replacing the integral on the right hand side by the approximation using the composite Simpson’s Rule, starting with (say) \( 4 \) subintervals.
14. Using \((\hat{u}, \hat{v})\) as an initial estimate, use Mathematica’s FIndRoot to solve the nonlinear equations \( z_j(u, v) = Y_j - Y_{j-1} \) for \((u, v) \in \mathbb{R}^n \times \mathbb{R}^n \) where \( 1 \leq j \leq n \).
15. Calculate \( a, b, c, d \in \mathbb{R}^n \) from \((u, v)\), using the formulae preceding Lemma 2.
16. For \( 1 \leq j \leq n \) define \( \beta_j : [0, L_j] \to \mathbb{R} \) by \( \beta_j(s) := a_j + b_j s + c_j s^2 + d_j s^3 \) where \( s \in [0, L_j] \). Take \( \theta : [T_0, T_n] \to \mathbb{R} \) to be the track-sum of the \( \beta_j \).

\[1\] If this or any subsequent checks fail, the problem is that \( \epsilon \) is not sufficiently small. In such cases we have nothing to say.
(17) If the natural second-order spiral spline $y_0$ interpolates $Y$ at $T$ to sufficient accuracy, stop. Otherwise return to step (13) and increase the number of subintervals for composite Simpson’s.

The steps are elementary, with the exception of step (14) where a numerical package is used to solve a nonlinear system. Because $(u, v) = (u, v) + O(\epsilon^4)$, we start with an excellent initial guess, at least when $\epsilon$ is not too large.

**Example 4.** Taking the $y_0$ from Example 3 we use the composite Simpson’s Rule with 4 subintervals to evaluate the $z_j$. Using Mathematica’s FindRoot, it takes a total of 0.23 seconds to calculate the 32 possible $C^1$ splines $\theta$, namely 0.007 seconds each. Except for closing the gaps, the $y_0$ resembles the $\tilde{y}_0$ in Example 3 and appear $C^3$. The elastic energies for the 6 cases considered in Example 3 are listed in Table 1:

**Table 1.** Energies for Estimates and Interpolants in Example 3

| $\sigma$ | $J(\tilde{y}_0)$ | $J(y_0)$ |
|----------|-----------------|-----------|
| (-1, -1, 1, 1, 1) | 39.71 | 39.98 |
| (-1, 1, 1, -1, 1) | 21.51 | 21.8 |
| (-1, -1, 1, -1, 1) | 23.31 | 22.85 |
| (1, 1, 1, 1, 1) | 100.60 | 61.60 |
| (1, 1, 1, -1, 1, 1) | 83.2 | 54.59 |
| (1, -1, 1, 1, 1) | 66.56 | 45.89 |

The first three cases are where approximation of $\theta$ by $\tilde{\theta}$ is best, and the last three are where it is worst. Accordingly, very little change is needed to close the gaps for the first 3 cases with much less change in elastic energy. □

9. Optimisation

Just as it was straightforward to find the $\theta$ by closing gaps, availability of the $\tilde{\theta}_j$ is also helpful for optimisation within more general classes of unit-speed curves. We first extend the definition of the $\theta_j : [0, L_j] \rightarrow \mathbb{R}$ to include curves

$$\theta_j(t) = a_j + b_j t + c_j t^2 + d_j t^3 + F(p_j, t) t^4$$

where $q \geq 1$, and $p_j \in \mathbb{R}^q$ is a $q$-dimensional parameter for $1 \leq j \leq n$. Here $F : \mathbb{R}^q \times \mathbb{R} \rightarrow \mathbb{R}$ is a $C^3$ function prescribed in advance, permitting considerable generality in the class of curves $\theta_j$. For instance $F(p, t)$ could be polynomial of degree $q - 1$ in $t$, with $q$ the list of coefficients. Up to now $F$ has been identically 0.

In general, for $u, v \in \mathbb{R}^n$ and for $p \in (\mathbb{R}^3)^n$, write

- $a_1 = v_1$,
- $a_j = u_{j-1}$, for $2 \leq j \leq n$,
- $b_1 = 0$,
- $b_j = \frac{v_j}{L_j}$, for $2 \leq j \leq n$,
- $c_1 = \frac{L_1 v_1^2 - 3 L_2 (u_1 - v_1)}{L_2^2} + F(p_1, L_1) L_1^2 - F(p_1, L_1) L_1^3$,
- $c_j = \frac{L_j v_{j+1}^2 + 2 L_{j+1} (u_j - v_{j+1})}{L_{j+1}^2} + F(p_j, L_j) L_j^2 - F(p_j, L_j) L_j^3$ for $2 \leq j \leq n - 1$,
- $c_n = \frac{3 u_n + v_n}{2 L_n} - 2 F(p_n, L_n) L_n^2 - F(p_n, L_n) L_n^3$,
- $d_1 = \frac{L_1 v_1^2 - 3 L_2 (u_1 - v_1)}{L_2^2} - 2 F(p_1, L_1) L_1 + F(p_1, L_1) L_1^2$,
- $d_j = \frac{L_j v_{j+1}^2 + 2 L_{j+1} (u_j - v_{j+1})}{L_{j+1}^2} - 2 F(p_j, L_j) L_j - F(p_j, L_j) L_j^2$ for $2 \leq j \leq n - 1$,
- $d_n = \frac{u_n}{L_n}$.

Then the track-sum $\theta : [T_0, T_n] \rightarrow \mathbb{R}$ of the $\theta_1, \theta_2, \ldots, \theta_n$ is $C^4$ and $\theta^{(1)}(T_0) = 0 = \theta^{(1)}(T_n)$. The additional $q n$ real parameters $p = (p_1, p_2, \ldots, p_n)$ allow us to describe a larger class of well-behaved curves, within which some functional $J(y_0)$ can be optimised, subject to the interpolation condition 6, where $y_0$ is defined by formula 3 with respect to the more general definition of $\theta$. We again find 6 is equivalent to

$$z_j(u, v, p) := \int_0^{L_j} e^{\theta_j(t)} \, dt \quad \text{for} \ 1 \leq j \leq n,$$

\[\text{Footnote 15:}\] The calculation might have been made faster by feeding approximate Jacobians to Mathematica, but we were satisfied with this additional complication.

\[\text{Footnote 16:}\] It does not seem worthwhile to plot the $y_0$. 

12 LYLE NOAKES
where \( z : \mathbb{R}^n \times \mathbb{R}^n \times (\mathbb{R}^q)^n \to (\mathbb{R}^2)^n \) is given by

\[
z_j(u, v, p) := \int_0^{L_j} e^{i \theta_j(t)} \, dt.
\]

Standard packages for local optimisation are effective when provided with a good initial guess. For curves of moderate curvature and moderate elastic energy, it makes sense to use \((u, v, p) = (\hat{u}, \hat{v}, 0) + O(\epsilon^4)\), where \((\hat{u}, \hat{v})\) is obtained from \(\hat{\theta}\) in the same way as before, and numerical integration is used to estimate the \(z_j\), this time for constrained numerical local optimisation. In the following example \(q = 1\), \(F(p_j, t) = p_j\), and \(J\) is taken as the energy integral \(J\).

**Example 5.** Starting with the data of Example 2, the \(y_\theta\) are very similar to those in Example 4, with decreases in elastic energy ranging between 0.0122 and 0.3058, with mean 0.1019 and standard deviation 0.0775. By comparison, the elastic energies for \(y_p\) with the additional parameters \(p \in \mathbb{R}^5\) range between 20.970 and 61.576. The minimal energy curve in Example 4 is therefore a good initial guess for the minimal energy curve \(y_p\) with the additional parameters \(p\). For practical purposes it is probably preferable to choose the smallest elastic energy in Example 4 without introducing \(p\). Even better, very little is lost in choosing the smallest elastic energy in Example 2 with no nonlinear equations at all. \(\Box\)
References

[1] Di Antonio L, The fabric of the universe is most perfect: Euler’s research on elastic curves, in Euler at, 300: an appreciation, 239–260, Mathematical Association of America (2007).
[2] Andoly, B., Ponno, Yves, Elasticity and Geometry: From Hair Curls to the Non-linear Response of Shells, Oxford University Press (2010).
[3] Baran, Ilya, Lehtinen, Jaako and Popovic, Jovan, Sketching Clothoid Splines Using Shortest Paths, Eurographics 0 (1981).
[4] Florence Bertails-Deschamps, Super-Clothoids, Eurographics 31 (2012)
[5] G. Birkhoff, H. Burchard and D. Thomas, Nonlinear Interpolation by Splines, Pseudosplines and Elasticia, General Motors Research Laboratories Report 466, (1965), 1-13.
[6] Bruckstein, Alfred M., Helia Robert, J., and Netravala, Arun, N., Discrete elastica, Applicable Analysis 78 (2001) 453–485.
[7] Brunnett, Guido and Wurfl, Joz, A Univariate Method for Plane Elastic Curves, Computer Aided Geometric Design 14 (1997) 273–292.
[8] Coupé, Ian, D., Curve Fitting With Nonlinear Spiral Splines, Department of Mathematics, University of Canterbury NZ, No 63 (1991), 1-14.
[9] Dai, Han and Cochran, John E. Jr. (2010) Path Planning and State Estimation for Unmanned Aerial Vehicles in Hostile Environments, Journal of Guidance, Control and Dynamics 33 (2), 595–601.
[10] de Boor, Carl, A Practical Guide to Splines, Applied mathematical Sciences 27, Springer (1978).
[11] Elber, David, Moving Along a Curve with Specified Speed, http://www.geomitrictools.com/ (2007) 1-15.
[12] Edwards, J.A., Exact equations of the nonlinear spline, ACM Transactions on Mathematical Software 18 (1992), 174–192.
[13] Golomb, Michael and Jerome, Joseph, Equilibria of the Curvature Functional and Manifolds of Interpolating Spline Curves, SIAM J. on Mathematical Analysis 13 (3) (1982), 412–458.
[14] Alonzo Kelly and Brian Nagy, Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control, International Journal of Robotics Research 22 (2003), 583-601.
[15] Levien, Raph, The Elastica: a Mathematical History, Technical Report No. UCB/EECS-2008-103 http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-103.html, August 23, 2008.
[16] E.H. Lee and G.E. Forsythe, Variational Study of Nonlinear Spline Curves, SIAM Review 15 (1973), 120–133.
[17] Linness, Anders, Unified Representations of Nonlinear Splines, Journal of Approximation Theory 84 (1996), 315–350.
[18] Looker, Jason R., Constant Speed Interpolating Paths, DSTO Defence Science and Technology Organisation, AR 014-939, DSTO-TN-0989, March 2011.
[19] D.S. Meek and R.S.D. Thomas, A Guided Clothoid Spline, Computer Aided Geometric Design 8 (1991) 163-174.
[20] D.S. Meek and D.J. Walton, An Arc Spline Approximation to a Clothoid, Journal of Computational and Applied Mathematics 170 (2004), 59–77.
[21] Peterson, John W., Arc-Length Parameterization of Spline Curves http://www.saccade.com/writing/graphics/RE-PARAM.PDF
[22] Singer, David A., Lectures on Elastic Curves and Rods, in AIP Conference Proceedings 1002, Curvature and Variational Modelling in Physics and Heliophysics, Santiago de Compostela, Spain, 17–18 September 2007 (eds Oscar J. Garay, Eduardo Garcia-Hu, Ramón Vázquez-Lorenzo), 3–32.
[23] Stoor, Joao, Curve Fitting With Clothoidal Splines, J. Research of the National Bureau of Standards 87 No. 4 (1982) 318–346.
[24] Arc-Length Parameterized Spline Curves for Real-Time Simulation, Wang, Hongling, Keaney, Joseph, Atkinson, Kendall, Curve and Surface Design: Saint Malo (2002) Lych, T., Mazure, M.-L., Schumacher, L. (eds) 387–396.

LYLE.NOAKES@UWA.EDU.AU (SCHOOL OF MATHEMATICS & STATISTICS, THE UNIVERSITY OF WESTERN AUSTRALIA, 35 STIRLING HIGHWAY, CRAWLEY, WA 6009, AUSTRALIA)