Research Article

Mathematical Methods in the Applied Sciences

Received 05 August 2013 Published online 17 March 2014 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/mma.3114
MOS subject classification: 15A15; 15A09; 15A23

G^1 fitting with clothoids

Enrico Bertolazzi and Marco Frego

Communicated by P. Colli

A new algorithm for the solution to the problem of Hermite G^1 interpolation with a clothoid curve is herein proposed, that is, a clothoid that interpolates two given points in a plane with assigned unit tangent vectors. The interpolation problem is formulated as a system of three nonlinear equations with multiple solutions, which is difficult to solve even numerically. In this work the solution of this system is reduced to the computation of the zeros of only one single nonlinear function in one variable. The location of the relevant zero is tackled analytically: it is provided the interval containing the zero where the solution is proved to exist and to be unique. A simple guess function allows to find that zero with very few iterations in all of the possible instances. Computing clothoid curves calls for evaluating Fresnel-related integrals, asymptotic expansions near critical values are herein conceived to avoid loss of precision. This is particularly important when the solution of the interpolation problem is close to a straight line or an arc of circle. The present algorithm is shown to be simple and compact. The comparison with literature algorithms proves that the present algorithm converges more quickly and accuracy is conserved in all of the possible instances, whereas other algorithms have a loss of accuracy near the transition zones. Copyright © 2014 John Wiley & Sons, Ltd.

Keywords: clothoid; Fresnel integrals; Hermite G^1 interpolation

1. Introduction

The fitting that allows a curve to interpolate two given points in a plane with assigned tangent directions is called G^1 Hermite interpolation (Figure 1). If the curvatures are also given at the two points, then this is called G^2 Hermite interpolation [1]. The G^2 interpolation provides a smoother curve with good properties, at the price of more constraints to be satisfied, and this implies heavier computational costs. In several applications (especially in those in real time), the G^1 Hermite interpolation is cost-effective in particular when the discontinuity of the curvature is acceptable. Clothoid curves are used in a variety of applications such as the trajectory planning of robots or autonomous vehicles [2–7], or in computer-aided design [8–10] or in other fields [11, 12]. It is well known that clothoids are extremely useful, and this is why they are being studied despite their transcendental form [13, 14].

The purpose of this paper is to describe a new method for the numerical computation of G^1 Hermite interpolation with a single clothoid segment. Nowadays, the best algorithms for solving the G^1 interpolation problem have been proposed by Kimia *et al.* [15] and Walton and Meek [16, 17]. An iterative method was proposed by Kimia *et al.* [15]; however, Walton and Meek [17] remarked that no existence and uniqueness theorem was provided, also because the convergence rate was linear instead of quadratic as in [17]. The algorithm proposed by Walton and Meek [17] performs generally better than in [15] in terms of accuracy and number of iterations. It requires to split the procedure in three mutually exclusive cases: straight lines, circles and clothoids, a geometrical fact that helps to understand the problem. For each of the mutually exclusive cases, the problem is reduced to find the root of a single nonlinear equation solved using *damped* Newton–Raphson algorithm. However, the root of the nonlinear equations are ill-conditioned near the transition region, for example, when the clothoid stretches to a straight line or a circle, as shown in the section of numerical tests.

The present algorithm does not need to separate straight lines, circles and clothoids. The G^1 Hermite interpolation is recast in term of computing a *well-conditioned* zero of a *unique* nonlinear equation, which is proven to exist and to be unique in a prescribed domain. The Newton–Raphson algorithm *without damping* is herein used to solve the nonlinear equation and the additional help of a good initial guess implies that few iterations (less than four) suffice.

The article is structured as follows: there are four logical parts, the first is analytic, constructive and leads to the solution of the problem; the second is strictly numeric and implements the algorithm described in the first part; the third discusses a good guess

Department of Industrial Engineering, University of Trento, I-38123 Trento, Italy
* Correspondence to: Enrico Bertolazzi, Department of Industrial Engineering, University of Trento, I-38123 Trento, Italy.
† E-mail: enrico.bertolazzi@unitn.it

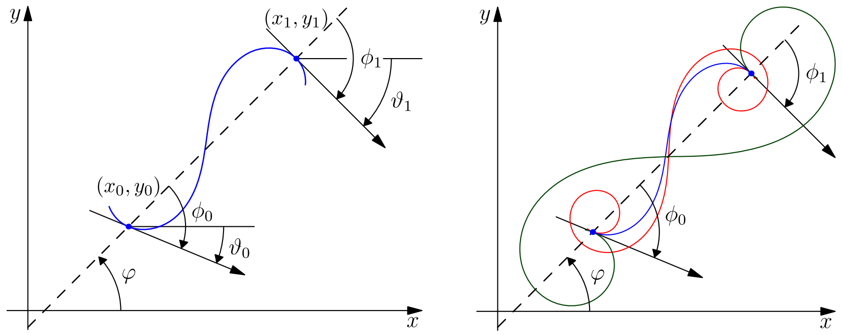


Figure 1. Left: G^1 Hermite interpolation schema and notation. Right: some possible solutions.

function in order to achieve a reduced number of iteration in all possible cases; and the last is a theoretical proof that covers existence and uniqueness of the solution under reasonable hypotheses.

Section 2 introduces the mathematical background and the notation used; there is a brief presentation of three possible definitions of the Fresnel integral functions and their momenta with some properties needed later. Section 3 defines the interpolation problem from the new analytical point of view. Section 4 describes the mathematical passages to reformulate it such that from the three equations in three unknowns it reduces to one nonlinear equation in one unknown. A summary of the algorithm and its issues are pointed out, such issues are solved in the following sections. It is given also a pseudo-code of the method. Section 5 considers an appropriate guess function to help the Newton–Raphson method to converge in few iterations, allowing the algorithm to be highly performant. Section 6 is devoted to answer to the numerical questions that arise when treating the Fresnel integral momenta such as stability and convergence. *Ad hoc* expressions for critical cases are discussed and provided. Section 7 covers the theoretical need of a proof of existence and uniqueness of the solution of the nonlinear equation used to solve the interpolation problem. It is explained how to select a valid solution among the infinite possibilities and a bounded range where this solution exists and is unique is exhibited. Although this proof is long, technical and not useful from an algorithmic point of view, the authors feel that it is necessary to complete the analysis of the algorithm. Section 8 is devoted to numerical tests and comparisons with other methods present in literature. Section 9 shows the application of the present algorithm in producing an interpolating clothoid spline that minimizes the jumps of the curvature at the joining points. In the Appendix, a pseudo code completes the presented algorithm for the accurate computation of the Fresnel-related integrals.

2. Some properties of Fresnel integrals

The fitting problem clearly involves the computation of Fresnel integrals. There are various possible definitions for Fresnel sine $\mathcal{S}(t)$ and cosine $\mathcal{C}(t)$ functions. Here, the choice is to follow reference [18].

Definition 1 (Fresnel integral functions)

$$\mathcal{C}(t) := \int_0^t \cos\left(\frac{\pi}{2}\tau^2\right) \mathrm{d}\tau, \qquad \mathcal{S}(t) := \int_0^t \sin\left(\frac{\pi}{2}\tau^2\right) \mathrm{d}\tau. \tag{1}$$

The literature reports different definitions of Fresnel integrals, such as

$$\widetilde{\mathcal{C}}(t) := \int_0^t \cos(\tau^2)\,\mathrm{d}\tau, \qquad \widetilde{\mathcal{S}}(t) := \int_0^t \sin(\tau^2)\,\mathrm{d}\tau, \qquad \widehat{\mathcal{C}}(\theta) := \frac{1}{\sqrt{2\pi}}\int_0^\theta \frac{\cos u}{\sqrt{u}}\,\mathrm{d}u, \qquad \widehat{\mathcal{S}}(\theta) := \frac{1}{\sqrt{2\pi}}\int_0^\theta \frac{\sin u}{\sqrt{u}}\,\mathrm{d}u. \tag{2}$$

The following identities allow to switch among these definitions:

$$\mathcal{C}(t) = \int_0^{\frac{\sqrt{2}}{\sqrt{\pi}}t} \cos(\tau^2)\,\mathrm{d}\tau = \frac{\operatorname{sign}(t)}{\sqrt{2\pi}}\int_0^{\frac{\pi}{2}t^2} \frac{\cos u}{\sqrt{u}}\,\mathrm{d}u, \qquad \mathcal{S}(t) = \int_0^{\frac{\sqrt{2}}{\sqrt{\pi}}t} \sin(\tau^2)\,\mathrm{d}\tau = \frac{\operatorname{sign}(t)}{\sqrt{2\pi}}\int_0^{\frac{\pi}{2}t^2} \frac{\sin u}{\sqrt{u}}\,\mathrm{d}u. \tag{3}$$

Also momenta of Fresnel integrals are used forward:

$$\mathcal{C}_k(t) := \int_0^t \tau^k \cos\left(\frac{\pi}{2}\tau^2\right) \mathrm{d}\tau, \qquad \mathcal{S}_k(t) := \int_0^t \tau^k \sin\left(\frac{\pi}{2}\tau^2\right) \mathrm{d}\tau. \tag{4}$$

Notice that $\mathcal{C}(t) := \mathcal{C}_0(t)$ and $\mathcal{S}(t) := \mathcal{S}_0(t)$ and that the first momenta are easily obtained:

$$\mathcal{C}_1(t) = \frac{1}{\pi}\sin\left(\frac{\pi}{2}t^2\right), \qquad \mathcal{S}_1(t) = \frac{1}{\pi}\left(1 - \cos\left(\frac{\pi}{2}t^2\right)\right). \tag{5}$$

It is possible to reduce the integrals (4) to a linear combination of standard Fresnel integrals (1) with some trigonometric functions. Closed forms via the exponential integral or the Gamma function are also possible; however, it is easy to express them as a recurrence. Integrating by parts, the following recurrence is obtained:

$$\mathcal{C}_{k+1}(t) = \frac{1}{\pi}\left(t^k \sin\left(\frac{\pi}{2}t^2\right) - k\,\mathcal{S}_{k-1}(t)\right), \qquad \mathcal{S}_{k+1}(t) = \frac{1}{\pi}\left(k\,\mathcal{C}_{k-1}(t) - t^k \cos\left(\frac{\pi}{2}t^2\right)\right). \tag{6}$$

Recurrence is started by computing standard Fresnel integrals (1) and first momenta (5). Notice that from recurrence (6), it follows that $\mathcal{C}_k(t)$ and $\mathcal{S}_k(t)$ with k odd do not contain Fresnel integrals (1) and are combination of elementary functions. It is convenient to introduce now the following functions whose properties are studied in Section 6:

$$X_k(a,b,c) := \int_0^1 \tau^k \cos\left(\frac{a}{2}\tau^2 + b\tau + c\right)\,\mathrm{d}\tau, \qquad Y_k(a,b,c) := \int_0^1 \tau^k \sin\left(\frac{a}{2}\tau^2 + b\tau + c\right)\,\mathrm{d}\tau. \tag{7}$$

Notice that, with a simple change of variable, one has the identities

$$\int_0^s \tau^k \cos\left(\frac{a}{2}\tau^2 + b\tau + c\right)\,\mathrm{d}\tau = s^{1+k} X_k(as^2, bs, c), \qquad \int_0^s \tau^k \sin\left(\frac{a}{2}\tau^2 + b\tau + c\right)\,\mathrm{d}\tau = s^{1+k} Y_k\left(as^2, bs, c\right),$$

which are used in the definition of the fitting problem.

3. The fitting problem

Consider the curve that satisfies the following system of ODEs:

$$\begin{array}{ll} x'(s) = \cos\vartheta(s), & x(0) = x_0, \\ y'(s) = \sin\vartheta(s), & y(0) = y_0, \\ \vartheta'(s) = \mathcal{K}(s), & \vartheta(0) = \vartheta_0, \end{array} \tag{8}$$

where s is the arc parameter of the curve, $\vartheta(s)$ is the direction of the tangent $(x'(s), y'(s))$, and $\mathcal{K}(s)$ is the curvature at the point $(x(s), y(s))$. When $\mathcal{K}(s) := \kappa' s + \kappa$, that is, when the curvature changes linearly, the curve is called clothoid. As a special case, when $\kappa' = 0$, the curve has constant curvature, that is, is a circle, and when both $\kappa = \kappa' = 0$, the curve is a straight line. The solution of ODEs (8) is given by the following:

$$\begin{aligned} x(s) &= x_0 + \int_0^s \cos\left(\frac{1}{2}\kappa'\tau^2 + \kappa\tau + \vartheta_0\right)\,\mathrm{d}\tau = x_0 + sX_0\left(\kappa' s^2, \kappa s, \vartheta_0\right), \\ y(s) &= y_0 + \int_0^s \sin\left(\frac{1}{2}\kappa'\tau^2 + \kappa\tau + \vartheta_0\right)\,\mathrm{d}\tau = y_0 + sY_0\left(\kappa' s^2, \kappa' s, \vartheta_0\right). \end{aligned} \tag{9}$$

Notice that $\frac{1}{2}\kappa' s^2 + \kappa s + \vartheta_0$ and $\kappa' s + \kappa$ are, respectively, the angle and the curvature at the abscissa s. Thus, the problem considered in this paper is stated next.

Problem 1 (G^1 Hermite interpolation)
Given two points (x_0, y_0) and (x_1, y_1), and two angles ϑ_0 and ϑ_1, find a curve segment of the form (9) that satisfies

$$\begin{array}{lll} x(0) = x_0, & y(0) = y_0, & (x'(0), y'(0)) = (\cos\vartheta_0, \sin\vartheta_0), \\ x(L) = x_1, & y(L) = y_1, & (x'(L), y'(L)) = (\cos\vartheta_1, \sin\vartheta_1), \end{array}$$

where $L > 0$ is the length of the curve segment.

The general scheme is shown in Figure 1 – left. Notice that Problem 1 admits an *infinite* number of solutions. In fact, the angle $\vartheta(s)$ of a clothoid, which solves Problem 1 satisfies $\vartheta(0) = \vartheta_0 + 2k\pi$ and $\vartheta(L) = \vartheta_1 + 2\ell\pi$ with $k, \ell \in \mathbb{Z}$: different values of k correspond to different interpolant curves that loop around the initial and the final point. Figure 1 – right shows possible solutions derived from the same Hermite data.

4. Recasting the interpolation problem

The solution of Problem 1 by (9) is a zero of the following nonlinear system involving the unknowns L, κ, κ':

$$\begin{cases} x_1 - x_0 - L\,X_0\left(\kappa' L^2, \kappa L, \vartheta_0\right) = 0 \\ y_1 - y_0 - L\,Y_0\left(\kappa' L^2, \kappa L, \vartheta_0\right) = 0 \\ \vartheta_1 - \left(\dfrac{1}{2}\kappa' L^2 + \kappa L + \vartheta_0\right) = 0. \end{cases} \tag{10}$$

Table I. The fitting algorithm.

```
Function buildClothoid($x_0$, $y_0$, $\vartheta_0$, $x_1$, $y_1$, $\vartheta_1$, $\epsilon$)
$\Delta x \leftarrow x_1 - x_0$;  $\Delta y \leftarrow y_1 - y_0$;  compute $r$, $\varphi$ from $r\cos\varphi = \Delta x$, $r\sin\varphi = \Delta y$;
$\phi_0 \leftarrow$ normalizeAngle($\vartheta_0 - \varphi$);  $\phi_1 \leftarrow$ normalizeAngle($\vartheta_1 - \varphi$);
Set $g$ as $g(A) = Y_0(2A, (\phi_1 - \phi_0) - A, \phi_0)$;
Set $A \leftarrow 3(\phi_1 + \phi_0)$;  // In alternative use (17a) or (17b)
while $|g(A)| > \epsilon$ do  $A \leftarrow A - g(A)/g'(A)$ ;
$L \leftarrow r/X_0(2A, \delta - A, \phi_0)$;  $\kappa \leftarrow (\delta - A)/L$;  $\kappa' \leftarrow (2\,A)/L^2$;
return $\kappa$, $\kappa'$, $L$
```

```
Function normalizeAngle($\phi$)
while $\phi > +\pi$ do
    $\phi \leftarrow \phi - 2\pi$
end while
while $\phi < -\pi$ do
    $\phi \leftarrow \phi + 2\pi$
end while
return $\phi$;
```

The third equation in equation (10) is linear so that solving it with respect to κ reduces the nonlinear system to

$$\begin{cases} x_1 - x_0 - L\,X_0\left(\kappa' L^2, \vartheta_1 - \vartheta_0 - \dfrac{1}{2}\kappa' L^2, \vartheta_0\right) = 0, \\ y_1 - y_0 - L\,Y_0\left(\kappa' L^2, \vartheta_1 - \vartheta_0 - \dfrac{1}{2}\kappa' L^2, \vartheta_0\right) = 0. \end{cases}$$

An approach based on the solution of a similar nonlinear system is proposed in reference [15], whereas references [16, 17] point out the criticity of this method by numerical examples. Introducing

$$A = \frac{1}{2}\kappa' L^2, \quad \Delta x = x_1 - x_0, \quad \Delta y = y_1 - y_0, \quad \delta = \vartheta_1 - \vartheta_0, \tag{11}$$

the nonlinear system is reduced to the solution of the nonlinear system of two equations in two unknowns, namely L and A:

$$\begin{cases} (*) := \Delta x - L\,X_0(2A, \delta - A, \vartheta_0) = 0, \\ (**) := \Delta y - L\,Y_0(2A, \delta - A, \vartheta_0) = 0. \end{cases} \tag{12}$$

Further simplification can be performed using polar coordinates for $(\Delta x, \Delta y)$, namely

$$\Delta x = r\cos\varphi, \qquad \Delta y = r\sin\varphi, \tag{13}$$

and the well-known trigonometric identities

$$\sin(\alpha - \beta) = \sin\alpha\cos\beta - \cos\alpha\sin\beta, \qquad \cos(\alpha - \beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta. \tag{14}$$

From equation (13) and $L > 0$, define two nonlinear functions $f(L, A)$ and $g(A)$, where $g(A)$ does not depend on L, as follows:

$$\begin{aligned} f(L, A) &:= (*)\cdot\cos\varphi + (**)\cdot\sin\varphi = \sqrt{\Delta x^2 + \Delta y^2} - L\,h(A), \\ g(A) &:= ((*)\cdot\sin\varphi - (**)\cdot\cos\varphi)\,/L = Y_0(2A, \delta - A, \phi_0). \end{aligned} \tag{15}$$

where $h(A) := X_0(2A, \delta - A, \phi_0), \phi_0 = \vartheta_0 - \varphi$ and ϕ_1, used later, is defined as $\phi_1 = \vartheta_1 - \varphi$.

Supposing to find A such that $g(A) = 0$, then from $f(L, A) = 0$ one computes L, κ and κ' using equations (15) and (11), respectively. Thus, the solutions of the nonlinear system (12) are known if the solutions of the single nonlinear function $g(A)$ of equation (15) are determined. The solution of Problem 1 is recapitulated in the following steps:

(1.) solve $g(A) = 0$;
(2.) compute $L = \sqrt{\Delta x^2 + \Delta y^2}/h(A)$; and
(3.) compute $\kappa = (\delta - A)/L$ and $\kappa' = 2A/L^2$.

This algorithm needs to compute the correct root of $g(A)$, which appropriately solves Problem 1 with the length L well defined and positive. These issues are discussed in Section 7.

The complete algorithm for the clothoid computation is written in the function `buildClothoid` of Table I. This function solves equation (15) and builds the coefficients of the interpolating clothoid.

5. Computing the initial guess for iterative solution

The zeros of function $g(A)$ are used to solve the interpolation problem and are approximated by the Newton–Raphson scheme. This algorithm needs 'a guess point' to converge to the appropriate solution. Notice that there is an infinite number of solutions of Problem 1, and criteria for the selection of a solution are needed. Uniqueness in appropriate range and existence of the root will be discussed in details in Section 7.

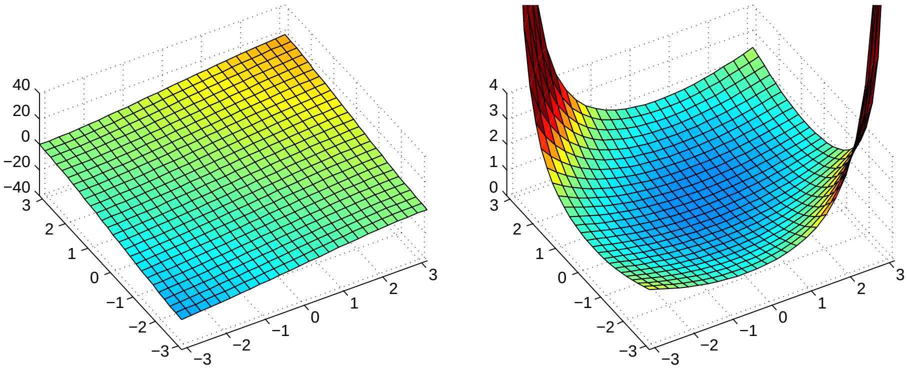


Figure 2. Left: the function $\mathcal{A}(\phi_0, \phi_1)$. Notice that $\mathcal{A}(\phi_0, \phi_1)$ is approximately a plane. Right: values of the length L of the computed clothoid as a function of ϕ_0 and ϕ_1. Notice that when angles satisfy $\phi_0 = \pi$, $\phi_1 = -\pi$ or $\phi_0 = -\pi$, $\phi_1 = \pi$ the length goes to infinity. The angles range in $[-\pi, \pi]$.

Table II. On the left, guess functions interpolation coefficients for guesses (17a) and (17b). On the right, iteration statistics for different guess functions.

	c	d	Iter.	Guess (16)		Guess (17a)		Guess (17b)	
1	3.070645	2.989696	1	1025	0.1%	1025	0.1%	1025	0.1%
2	0.947923	0.716220	2	6882	0.7%	10710	1.0%	34124	3.2%
3	-0.673029	-0.458969	3	238424	22.7%	702534	66.9%	1015074	96.6%
4		-0.502821	4	662268	63.0%	336356	32.0%	402	0.1%
5		0.261060	5	142026	13.5%				
6		-0.045854							

Denote with $\mathcal{A}(\phi_0, \phi_1)$, the selected zero of $g(A)$ as a function of ϕ_0 and ϕ_1. Figure 2 shows that $\mathcal{A}(\phi_0, \phi_1)$ is approximated by a plane. A simple approximation of $\mathcal{A}(\phi_0, \phi_1)$ is obtained by $\sin x \approx x$ in $Y_0(2A, \delta - A, \phi_0)$ and thus,

$$g(A) = Y_0(2A, \delta - A, \phi_0) \approx \int_0^1 A\tau^2 + (\delta - A)\tau + \phi_0 \, \mathrm{d}\tau = \frac{\phi_0 + \phi_1}{2} - \frac{A}{6},$$

and solving for A,

$$\mathcal{A}(\phi_0, \phi_1) \approx 3(\phi_0 + \phi_1). \tag{16}$$

This approximation is a fine initial point for Newton–Raphson; however, better approximation for $\mathcal{A}(\phi_0, \phi_1)$ are obtained by least squares. By invoking reflection and mirroring properties discussed in Section 7.1, the functional form of the approximation is simplified and results in the two following possible expressions for $\mathcal{A}(\phi_0, \phi_1)$:

$$\mathcal{A}(\phi_0, \phi_1) \approx (\phi_0 + \phi_1)\left(c_1 + c_2\overline{\phi}_0\overline{\phi}_1 + c_3\left(\overline{\phi}_0^2 + \overline{\phi}_1^2\right)\right), \tag{17a}$$

$$\mathcal{A}(\phi_0, \phi_1) \approx (\phi_0 + \phi_1)\left(d_1 + \overline{\phi}_0\overline{\phi}_1\left(d_2 + d_3\overline{\phi}_0\overline{\phi}_1\right) + \left(\overline{\phi}_0^2 + \overline{\phi}_1^2\right)\left(d_4 + d_5\overline{\phi}_0\overline{\phi}_1\right) + d_6\left(\overline{\phi}_0^4 + \overline{\phi}_1^4\right)\right), \tag{17b}$$

where $\overline{\phi}_0 = \phi_0/\pi, \overline{\phi}_1 = \phi_1/\pi$. The computed coefficients are reported in Table II on the left.

Using equations (16), (17a) or (17b) as the starting point for Newton–Raphson, the solution for Problem 1 is found in very few iterations. The three possible guess functions and their influence in the speed up process of the algorithm were checked in a battery of tests: computing the solution with Newton–Raphson starting with the proposed guesses in a 1024 × 1024 grid for ϕ_0 and ϕ_1 ranging in $[-0.9999\pi, 0.9999\pi]$ with a tolerance of 10^{-10}, results in the distribution of iterations resumed in Table II on the right.

Remark 2

For the Newton–Raphson method, the iteration is expressed as $A_{k+1} = A_k - g(A_k)/g'(A_k)$, near the root $A^\star$; there is the following well-known estimate for the error $e_k = A_k - A^\star$ when $|e_k| \le r$:

$$|e_{k+1}| \le C\,|e_k|^2, \qquad C = \frac{\max_{A\in\mathbb{R}} |g''(A)|}{2\min_{A\in[A^\star - r, A^\star + r]} |g'(A)|}.$$

The estimate for the second derivative of $g(A)$ is trivial:

$$\left|g''(A)\right| = \left|-\int_0^1 (\tau^2 - \tau)^2 \sin(A\tau^2 + (\delta - A)\tau + \phi_0)\,\mathrm{d}\tau\right| \le \int_0^1 (\tau^2 - \tau)^2\,\mathrm{d}\tau = \frac{1}{30}. \tag{18}$$

Using Taylor expansion yields the following estimate of $\min_{A\in[A^\star-r,A^\star+r]}|g'(A)|$:

$$g'(A)=g'(A^\star)+(A-A^\star)g''(\zeta),\qquad \left|g'(A)\right|\geq\left|g'(A^\star)\right|-\frac{|A-A^\star|}{30},\qquad \text{for}\quad \left|A-A^\star\right|\leq r.$$

Newton–Raphson is guaranteed to converge when $C\,|e_0|<1$. This more restrictive condition,

$$C\,|e_0|\leq\frac{|e_0|\max_{A\in\mathbb{R}}|g''(A)|}{2\min_{A\in[A^\star-r,A^\star+r]}|g'(A)|}\leq\frac{|e_0|}{2\,(30\,|g'(A^\star)|-|e_0|)}<1,$$

ensures that Newton–Raphson is convergent when $|e_0|<20\,|g'(A^\star)|$. Let $g_{\min}$ be the minimum value of the first derivative of $|g'(A^\star)|$ at the root for the angles ϕ_0,ϕ_1, then the computation on the previous 1024 × 1024 mesh yields

$$g_{\min}\approx 0.0505 \tag{19}$$

so that the estimate of the convergence radius becomes $r=20g_{\min}\approx 1.01$. On the same mesh, the maximum distance from the computed root with the guess (17b) results in a maximum distance of about 0.037, well below the estimated radius.

6. Accurate computation of Fresnel momenta

The computation of $g(A)$ defined in equation (15) ad $g'(A)$, employed in the Newton iteration, relies on the evaluation of integrals of kind (7), in fact, using integration by parts:

$$g'(A)=X_1(2A,\delta-A,\phi_0)-X_2(2A,\delta-A,\phi_0). \tag{20}$$

From the trigonometric identities (14), integrals (7) are rewritten as

$$X_k(a,b,c)=X_k(a,b,0)\cos c-Y_k(a,b,0)\sin c,\qquad Y_k(a,b,c)=X_k(a,b,0)\sin c+Y_k(a,b,0)\cos c.$$

Defining $X_k(a,b):=X_k(a,b,0)$ and $Y_k(a,b):=Y_k(a,b,0)$, the computation of equation (7) is reduced to the computation of $X_k(a,b)$ and $Y_k(a,b)$. From now on, it is assumed that the standard Fresnel integrals $\mathcal{C}_0$ and $\mathcal{S}_0$ can be computed with high accuracy. For this task, one can use algorithms described in references [19, 20] or simply use the available software [21]. It is convenient to introduce the following quantities:

$$\sigma:=\operatorname{sign}(a),\qquad z:=\sigma\frac{\sqrt{|a|}}{\sqrt{\pi}},\qquad \omega_+:=\frac{b+|a|}{\sqrt{\pi|a|}},\qquad \omega_-:=\frac{b}{\sqrt{\pi|a|}},\qquad \eta:=-\frac{b^2}{2a}, \tag{21}$$

so that it is possible to rewrite the argument of the trigonometric functions in $X_k(a,b)$ and $Y_k(a,b)$ as

$$\frac{a}{2}\tau^2+b\tau=\frac{\pi}{2}\sigma\left(\tau\frac{\sigma\sqrt{|a|}}{\sqrt{\pi}}+\frac{b}{\sqrt{\pi|a|}}\right)^2-\frac{b^2}{2a}=\frac{\pi}{2}\sigma(\tau z+\omega_-)^2+\eta.$$

By using the change of variable $\xi=\tau z+\omega_-$ with inverse $\tau=z^{-1}(\xi-\omega_-)$ for $X_k(a,b)$ and the identity (14), one has

$$\begin{aligned}X_k(a,b)&=z^{-1}\int_{\omega_-}^{\omega_+}z^{-k}(\xi-\omega_-)^k\cos\left(\sigma\frac{\pi}{2}\xi^2+\eta\right)\,\mathrm{d}\xi=z^{-k-1}\int_{\omega_-}^{\omega_+}\sum_{j=0}^{k}\binom{k}{j}\xi^j(-\omega_-)^{k-j}\cos\left(\frac{\pi}{2}\xi^2+\sigma\eta\right)\,\mathrm{d}\xi,\\&=z^{-k-1}\sum_{j=0}^{k}\binom{k}{j}(-\omega_-)^{k-j}[\cos\eta\Delta\mathcal{C}_j-\sigma\sin\eta\Delta\mathcal{S}_j],\\&=\frac{\cos\eta}{z^{k+1}}\left[\sum_{j=0}^{k}\binom{k}{j}(-\omega_-)^{k-j}\Delta\mathcal{C}_j\right]-\sigma\frac{\sin\eta}{z^{k+1}}\left[\sum_{j=0}^{k}\binom{k}{j}(-\omega_-)^{k-j}\Delta\mathcal{S}_j\right],\end{aligned} \tag{22}$$

where $\Delta\mathcal{C}_j=\mathcal{C}_j(\omega_+)-\mathcal{C}_j(\omega_-)$ and $\Delta\mathcal{S}_j=\mathcal{S}_j(\omega_+)-\mathcal{S}_j(\omega_-)$ are the evaluation of the momenta of the Fresnel integrals as defined in (4). Analogously for $Y_k(a,b)$, one has

$$Y_k(a,b)=\frac{\sin\eta}{z^{k+1}}\left[\sum_{j=0}^{k}\binom{k}{j}(-\omega_-)^{k-j}\Delta\mathcal{C}_j\right]+\sigma\frac{\cos\eta}{z^{k+1}}\left[\sum_{j=0}^{k}\binom{k}{j}(-\omega_-)^{k-j}\Delta\mathcal{S}_j\right]. \tag{23}$$

This computation is inaccurate when $|a|$ is small: in fact, z appears in the denominator of several fractions. For this reason, for small values of $|a|$, it is better to substitute equations (22) and (23) with asymptotic expansions. Notice that the recurrence (6) is unstable so that it produces inaccurate results for large k, but only the first two terms are needed, so this is not a problem for the computation of $g(A)$ and $g'(A)$.

6.1. *Accurate computation with small parameters*

When the parameter $|a|$ is small, identity (14) yields the series expansion:

$$
\begin{aligned}
X_k(a,b) = \int_0^1 \tau^k \cos\left(\frac{a}{2}\tau^2 + b\tau\right) \mathrm{d}\tau &= \int_0^1 \tau^k \left[\cos\left(\frac{a}{2}\tau^2\right)\cos(b\tau) - \sin\left(\frac{a}{2}\tau^2\right)\sin(b\tau)\right] \mathrm{d}\tau, \\
&= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!}\left(\frac{a}{2}\right)^{2n} X_{4n+k}(0,b) - \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}\left(\frac{a}{2}\right)^{2n+1} Y_{4n+2+k}(0,b), \\
&= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!}\left(\frac{a}{2}\right)^{2n}\left[X_{4n+k}(0,b) - \frac{a\,Y_{4n+2+k}(0,b)}{2(2n+1)}\right],
\end{aligned}
\tag{24}
$$

and, analogously, using again identity (14):

$$
Y_k(a,b) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!}\left(\frac{a}{2}\right)^{2n}\left[Y_{4n+k}(0,b) + \frac{a\,X_{4n+2+k}(0,b)}{2(2n+1)}\right]. \tag{25}
$$

From the inequalities,

$$
|X_k| \le \int_0^1 |\tau^k|\,\mathrm{d}\tau = \frac{1}{k+1}, \qquad |Y_k| \le \int_0^1 |\tau^k|\,\mathrm{d}\tau = \frac{1}{k+1},
$$

the remainder for the series of X_k becomes

$$
\begin{aligned}
R_{p,k} &= \left|\sum_{n=p}^{\infty} \frac{(-1)^n}{(2n)!}\left(\frac{a}{2}\right)^{2n}\left[X_{4n+k}(0,b) - \frac{a\,Y_{4n+2+k}(0,b)}{2(2n+1)}\right]\right| \le \sum_{n=p}^{\infty} \frac{1}{(2n)!}\left(\frac{a}{2}\right)^{2n}\left[\frac{1}{4n+1} + \frac{|a|}{2(2n+1)(4n+3)}\right] \\
&\le \left(\frac{a}{2}\right)^{2p} \sum_{n=p}^{\infty} \frac{(a/2)^{2(n-p)}}{(2(n-p))!} \le \left(\frac{a}{2}\right)^{2p} \sum_{n=0}^{\infty} \frac{1}{(2n)!}\left(\frac{a}{2}\right)^{2n} = \left(\frac{a}{2}\right)^{2p} \cosh(a).
\end{aligned}
$$

The same estimate is obtained for the series of Y_k. Both series (24) and (25) converge fast. For example, if $|a| < 10^{-4}$ and $p = 2$, the error is less than 6.26×10^{-18}; whereas if $p = 3$, the error is less than 1.6×10^{-26}.

By using a simple recurrence, it is possible to compute $X_k(0,b)$ and $Y_k(0,b)$ but it turns out to be unstable. A stable computation is obtained by using an explicit formula based on the Lommel function $s_{\mu,\nu}(z)$ (see reference [22]). The explicit formula is as follows:

$$
X_k(0,b) = \frac{k\,s_{k+\frac{1}{2},\frac{3}{2}}(b)\sin b + f(b)s_{k+\frac{3}{2},\frac{1}{2}}(b)}{(1+k)b^{k+\frac{1}{2}}} + \frac{\cos b}{1+k}, \qquad Y_k(0,b) = \frac{k\,s_{k+\frac{3}{2},\frac{3}{2}}(b)\sin b + f(b)(2+k)s_{k+\frac{1}{2},\frac{1}{2}}(b)}{(2+k)b^{k+\frac{1}{2}}} + \frac{\sin b}{2+k}, \tag{26}
$$

where $k = 1,2,3,\dots$ and $f(b) := b^{-1}\sin b - \cos b$. The Lommel function has the following expansion (see reference [23] or [24]):

$$
s_{\mu,\nu}(z) := z^{\mu+1}\sum_{n=0}^{\infty} \frac{(-z^2)^n}{\alpha_{n+1}(\mu,\nu)}, \qquad \alpha_n(\mu,\nu) := \prod_{m=1}^{n}\left((\mu+2m-1)^2 - \nu^2\right), \tag{27}
$$

and using this expansion in equation (26) results in the next explicit formula for $k = 1,2,3,\dots$:

$$
X_k(0,b) = A(b)w_{k+\frac{1}{2},\frac{3}{2}}(b) + B(b)w_{k+\frac{3}{2},\frac{1}{2}}(b) + \frac{\cos b}{1+k}, \qquad Y_k(0,b) = C(b)w_{k+\frac{3}{2},\frac{3}{2}}(b) + D(b)w_{k+\frac{1}{2},\frac{1}{2}}(b) + \frac{\sin b}{2+k}, \tag{28}
$$

where

$$
w_{\mu,\nu}(b) := \sum_{n=0}^{\infty} \frac{(-b^2)^n}{\alpha_{n+1}(\mu,\nu)}, \quad A(b) := \frac{kb\sin b}{1+k}, \quad B(b) := \frac{(\sin b - b\cos b)b}{1+k}, \quad C(b) := -\frac{b^2\sin b}{2+k}, \quad D(b) := \sin b - b\cos b.
$$

Notice that expression (28) is continuous in b at $b = 0$.

7. Theoretical development

In this section, existence and selection of the appropriate solution are discussed in detail. The computation of L requires only to verify that for $A^\star$ such that $g(A^\star) = 0$, then $h(A^\star) = X_0(2A^\star, \delta - A^\star, \phi_0) \neq 0$. This does not ensure that the computed L is positive; but positivity is obtained by an appropriate choice of $A^\star$.

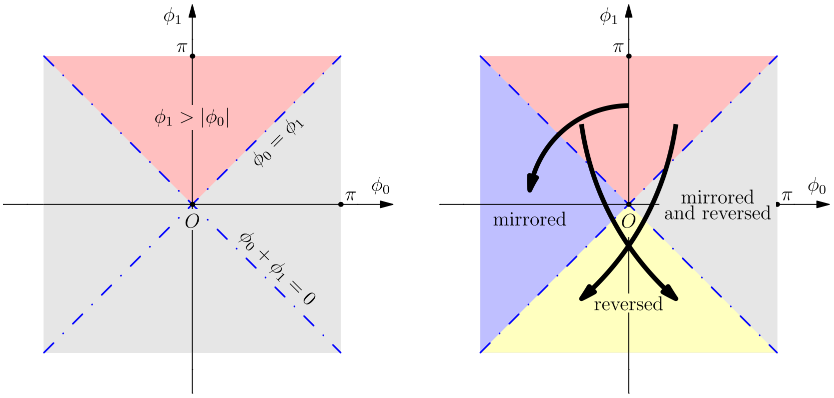


Figure 3. Left: the domain $\phi_1 > |\phi_0|$ with the special cases $\phi_1 = \phi_0$ and $\phi_1 + \phi_0 = 0$. Right: the domain mirrored and reversed.

7.1. *Symmetries of the roots of $g(A)$*

The general analysis of the zeros of $g(A)$ requires the angles ϕ_0 and ϕ_1 to be in the range $(-\pi, \pi)$. It is possible to restrict the domain of search stating the following auxiliary problems:

The reversed problem. The clothoid joining (x_1, y_1) to (x_0, y_0) with angles $\vartheta_0^R = -\vartheta_1$ and $\vartheta_1^R = -\vartheta_0$ is a curve with support, a clothoid that solves Problem 1 but running in the opposite direction (with the same length L). Let $\delta^R = \vartheta_1^R - \vartheta_0^R = -\vartheta_0 + \vartheta_1 = \delta$, it follows that $g^R(A) := Y_0(2A, \delta - A, -\phi_1)$ is the function whose zeros give the solution of the reversed interpolation problem.

The mirrored problem. The curve obtained connecting (x_0, y_0) to (x_1, y_1) with angle $\vartheta_0^M = \varphi - \phi_0$ and $\vartheta_1^M = \varphi - \phi_1$ is a curve with support, a curve solving the same problem but mirrored along the line connecting the points (x_0, y_0) and (x_1, y_1) (with the same length L). Let $\delta^M = \vartheta_1^M - \vartheta_0^M = -\phi_1 + \phi_0 = -\delta$, it follows that $g^M(A) := Y_0(2A, -\delta - A, -\phi_0)$ is the function whose zeros are the solution of the mirrored interpolation problem.

Lemma (3) shows that it is possible to reduce the search of the roots in the domain $|\phi_0| < \phi_1 \leq \pi$. The special cases $\phi_0 \pm \phi_1 = 0$ are considered separately.

Lemma 3
Let $g(A)$ and $h(A)$ defined in equation (15) with

$$\begin{aligned} g^R(A) &:= Y_0(2A, \delta - A, -\phi_1), & g^M(A) &:= Y_0(2A, -\delta - A, -\phi_0), \\ h^R(A) &:= X_0(2A, \delta - A, -\phi_1), & h^M(A) &:= X_0(2A, -\delta - A, -\phi_0), \end{aligned}$$

then $g(A) = -g^R(-A), g(A) = -g^M(-A), h(A) = h^R(-A) = h^M(-A)$.

Thus, $g(A)$ has the same roots of $g^R(A), g^M(A)$ with opposite sign.

Proof
(omitted) □

Figure 3 shows the domain $|\phi_0| < \phi_1 \leq \pi$ with the mirrored and reversed problem. Reflecting and mirroring allows to assume the constraints for the angles described in the following Assumption 4.

Assumption 4 (Angle domain)
The angles ϕ_0 and ϕ_1 satisfy the restriction: $|\phi_0| \leq \phi_1 \leq \pi$ with ambiguous cases $|\phi_0| = \phi_1 = \pi$ excluded (Figure 3).

This ordering ensures that when $|\phi_0| < \phi_1$, the curvature of the fitting curve is increasing, that is, $\kappa' > 0$. Notice that if A is the solution of nonlinear system (15), then $\kappa' = 2A/L^2$, that is, the sign of A is the sign of κ', and thus A must be positive. Finally, $\delta = \phi_1 - \phi_0 \geq 0$ with strict inequality when $|\phi_0| < \phi_1$. This assumption is not a limitation because any interpolation problem can be reformulated as a problem satisfying Assumption 4. The proof of existence and uniqueness of the fitting problem splits the angle domain in various subregions, whereas the special case $\phi_0 + \phi_1 = 0$ is performed apart.

7.2. *Localization of the roots of $g(A)$*

The problem $g(A) = 0$ has in general infinite solutions. The next Theorems show the existence of a *unique* solution in a prescribed range; they are in part new and in part taken in [17], here reported without proofs and notation slightly changed to better match our notation. By appropriate transformations, these Theorems permit to select the suitable solution and find the interval where the solution is unique. The transformation is contained in the following Lemma:

Lemma 5
The (continuous) functions $g(A)$ and $h(A)$ defined in equation (15) for $A > 0$, when ϕ_0 and ϕ_1 satisfy assumption 4, can be written as

$$g(A) = \frac{\sqrt{2\pi}}{\sqrt{A}} \begin{cases} p\left(\frac{(\delta-A)^2}{4A}\right) & 0 < A \le \delta, \\ q\left(\frac{(\delta-A)^2}{4A}\right) & A \ge \delta; \end{cases} \qquad h(A) = \frac{\sqrt{2\pi}}{\sqrt{A}} \begin{cases} \overline{p}\left(\frac{(\delta-A)^2}{4A}\right) & 0 < A \le \delta, \\ \overline{q}\left(\frac{(\delta-A)^2}{4A}\right) & A \ge \delta, \end{cases}$$

where

$$\begin{aligned} p(\theta) &= \int_\theta^{\theta+\delta} \frac{\sin(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u, \quad q(\theta) = p(\theta) + 2\int_0^\theta \frac{\sin(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u, \\ \overline{p}(\theta) &= \int_\theta^{\theta+\delta} \frac{\cos(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u, \quad \overline{q}(\theta) = \overline{p}(\theta) + 2\int_0^\theta \frac{\cos(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u. \end{aligned} \tag{29}$$

Proof
Standard trigonometric passages and assumption $A > 0$ yield the following expression for $g(A)$ and $h(A)$:

$$\begin{aligned} \sqrt{A}\,g(A) &= \sqrt{2\pi}\left[(\mathcal{C}(\omega_+) - \mathcal{C}(\omega_-))\sin\eta + (\mathcal{S}(\omega_+) - \mathcal{S}(\omega_-))\cos\eta\right], \\ \sqrt{A}\,h(A) &= \sqrt{2\pi}\left[(\mathcal{C}(\omega_+) - \mathcal{C}(\omega_-))\cos\eta - (\mathcal{S}(\omega_+) - \mathcal{S}(\omega_-))\sin\eta\right], \end{aligned}$$

where $\omega_\pm$ and η were previously defined in equation (21) and here, take the form

$$\omega_- = \frac{\delta - A}{\sqrt{2\pi A}}, \qquad \omega_+ = \frac{\delta + A}{\sqrt{2\pi A}}, \qquad \eta = \phi_0 - \theta, \qquad \theta = \frac{(\delta - A)^2}{4A}.$$

Combining equivalence (3) and the parity properties of $\sin x$ and $\cos x$, $g(A)$ and $h(A)$ take the following form:

$$\sqrt{A}\,g(A) = \Delta\widehat{\mathcal{C}}\sin(\phi_0-\theta) + \Delta\widehat{\mathcal{S}}\cos(\phi_0-\theta), \qquad \sqrt{A}\,h(A) = \Delta\widehat{\mathcal{C}}\cos(\phi_0-\theta) - \Delta\widehat{\mathcal{S}}\sin(\phi_0-\theta), \tag{30}$$

where $\Delta\widehat{\mathcal{C}} := \widehat{\mathcal{C}}(\theta+\delta) - \sigma_-\widehat{\mathcal{C}}(\theta)$, $\Delta\widehat{\mathcal{S}} := \widehat{\mathcal{S}}(\theta+\delta) - \sigma_-\widehat{\mathcal{S}}(\theta)$, $\sigma_- := \operatorname{sign}(\delta - A)$ and $\widehat{\mathcal{C}}(\theta)$ and $\widehat{\mathcal{S}}(\theta)$ are defined in equation (2). By using identities (14), equation (30) becomes

$$\begin{aligned} \hat{g}(\theta) &= \frac{\sqrt{A}}{\sqrt{2\pi}}g(A) = \int_0^{\theta+\delta} \frac{\sin(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u - \sigma_- \int_0^\theta \frac{\sin(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u, \\ \hat{h}(\theta) &= \frac{\sqrt{A}}{\sqrt{2\pi}}h(A) = \int_0^{\theta+\delta} \frac{\cos(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u - \sigma_- \int_0^\theta \frac{\cos(u+\phi_0-\theta)}{\sqrt{u}}\,\mathrm{d}u. \end{aligned}$$

It is recalled that A must be positive, so that when A ranges into $0 < A < \delta$ then $\sigma_- = 1$; otherwise, when $A > \delta$ then $\sigma_- = -1$. In case $A = \delta$, then $\theta = 0$ and the second integral are 0, and thus, $g(\delta) = p(0) = q(0)$ and $h(\delta) = \overline{p}(0) = \overline{q}(0)$. □

The next Theorems characterize the zeros of the functions (29) finding intervals where the solution exists and is unique.

Theorem 6 (see [17] th.2)
Let $0 < -\phi_0 < \phi_1 < \pi$. If $p(0) > 0$, then $p(\theta) = 0$ has no root for $\theta \ge 0$. If $p(0) \le 0$ then $p(\theta) = 0$ has exactly one root for $\theta \ge 0$. Moreover, the root occurs in the interval $[0, \theta^{\max}]$ where

$$\theta^{\max} = \frac{\lambda^2}{1-\lambda^2}(\phi_1 - \phi_0) > 0, \qquad 0 < \lambda = \frac{1-\cos\phi_0}{1-\cos\phi_1} < 1. \tag{31}$$

Theorem 7 (see [17] th.3)
Let $-\pi < -\phi_1 < \phi_0 < 0$ and $q(0) > 0$, then $q(\theta) = 0$ has exactly one root in the interval $[0, \pi/2 + \phi_0]$. If $q(0) < 0$, then $q(\theta) = 0$ has no roots in the interval $[0, \pi/2 + \phi_0]$.

Theorem 8 (see [17] th.4)
Let $\phi_0 \in [0, \pi)$ and $\phi_1 \in (0, \pi]$, then $q(\theta) = 0$ has exactly one root in $[0, \pi/2 + \phi_0]$; moreover, the root occurs in $[\phi_0, \pi/2 + \phi_0]$.

The following additional Lemmata are necessary to complete the list of properties of $p(\theta)$ and $q(\theta)$:

Lemma 9
Let $p(\theta)$ and $q(\theta)$ as defined in equation (29), then

(a) if $0 \le \phi_0 \le \phi_1 \le \pi$, then

- if $\phi_1 > \phi_0$, then $p(\theta) > 0$ for all $\theta \ge 0$; otherwise, $p(\theta) = 0$ for all $\theta \ge 0$;
- $q(\theta) = 0$ for $\theta \in [\phi_0, \pi/2 + \phi_0]$, and the root is unique in $[0, \pi/2 + \phi_0]$;

(b) if $-\pi \le -\phi_1 < \phi_0 < 0$

- if $p(0) = q(0) \le 0$, then
  - $p(\theta) = 0$ has a unique root θ in $[0, \theta_0]$ with θ_0 defined in equation (31).
  - $q(\theta) = 0$ has no roots in the interval $[0, \pi/2 + \phi_0]$;
- if $p(0) = q(0) > 0$, then
  - $p(\theta) > 0$ for all $\theta \ge 0$;
  - $q(\theta) = 0$ has a unique root in the interval $[0, \pi/2 + \phi_0]$;

(c) if $\phi_0 \le -\pi/2$, then $p(0) = q(0) < 0$.

Proof
A direct application of Theorems 6–8. For point (*c*), from equation (30), $p(0) = q(0) = \sqrt{\delta}\, g(\delta) = \Delta\widehat{\mathcal{C}} \sin\phi_0 + \Delta\widehat{\mathcal{S}} \cos\phi_0$, in addiction, since $-\pi \le \phi_0 \le -\pi/2$, both $\sin\phi_0 \le 0$ and $\cos\phi_0 \le 0$ resulting in $p(0) = q(0) < 0$. □

The combination of Lemma 3, together with reversed and mirrored problems, proves that any interpolation problem can be reduced to one which satisfies Assumption 4. Assumption 4 with Lemma 5 proves existence and uniqueness of $g(A) = 0$ in a specified range when $\phi_0 + \phi_1 \ne 0$. The case of $\phi_0 - \phi_1 = 0$ follows from the application of Theorem 8 for positive angles, because Assumption 4 forces $\phi_1 \ge 0$ and excludes the case of equal negative angles. The case $\phi_0 + \phi_1 = 0$ is considered in the following Lemma.

Lemma 10
Let $\phi_0 + \phi_1 = 0$ and $\phi_0 \in (-\pi, \pi)$, then $g(A) = 0$ has the unique solution $A = 0$ in the interval $(-2\pi, 2\pi)$.

Proof
For $\phi_0 + \phi_1 = 0$, one has $\delta = -2\phi_0$ and

$$\begin{aligned} g(A) &= Y_0(2A, -2\phi_0 - A, \phi_0) = \int_0^1 \sin(A\tau(\tau-1) + \phi_0(1-2\tau))\,\mathrm{d}\tau = \int_{-1}^1 \sin(A(z^2-1)/4 - z\phi_0)\frac{\mathrm{d}z}{2}, \quad [\tau = (z+1)/2] \\ &= \int_{-1}^1 \sin(A(z^2-1)/4)\cos(z\phi_0)\frac{\mathrm{d}z}{2} - \int_{-1}^1 \cos(A(z^2-1)/4)\sin(z\phi_0)\frac{\mathrm{d}z}{2}. \end{aligned}$$

By using properties of odd and even functions, the rightmost integral of the previous line vanishes yielding

$$g(A) = \int_0^1 \sin(A(z^2-1)/4)\cos(z\phi_0)\,\mathrm{d}z.$$

From this last equality, if $A = 0$, then $g(A) = 0$. If $0 < |A| < 4\pi$, the sign of the quantity $\sin(A(z^2-1)/4)$ is constant; if $|\phi_0| < \pi/2$, then $\cos(z\phi_0) > 0$, and thus, $g(A)$ has no roots. For the remaining values of ϕ_0, that is, $\pi/2 \le |\phi_0| < \pi$:

$$\int_0^{\pi/(2|\phi_0|)} \cos(z\phi_0)\,\mathrm{d}z = \frac{1}{|\phi_0|}, \qquad \int_{\pi/(2|\phi_0|)}^1 |\cos(z\phi_0)|\,\mathrm{d}z = \frac{1 - \sin|\phi_0|}{|\phi_0|} < \frac{1}{|\phi_0|}.$$

If in addition, $0 < |A| < 2\pi$, then $\left|\sin(A(z^2-1)/4)\right|$ is positive, and monotone decreasing so that

$$\left|\int_0^{\pi/(2|\phi_0|)} \sin\left(\frac{A}{4}(z^2-1)\right)\cos(z\phi_0)\,\mathrm{d}z\right| \ge \frac{C}{|\phi_0|}, \qquad \left|\int_{\pi/(2|\phi_0|)}^1 \sin\left(\frac{A}{4}(z^2-1)\right)\cos(z\phi_0)\,\mathrm{d}z\right| < \frac{C}{|\phi_0|},$$

where

$$C = \left|\sin\left(\frac{A}{16|\phi_0|^2}(\pi^2 - 4|\phi_0|^2\right))\right| > 0,$$

and thus, $g(A) \ne 0$ for $0 < |A| < 2\pi$ and $|\phi_0| < \pi$. □

It remains to proof that $h(A) > 0$ at the selected root of $g(A) = 0$. This is contained in the main Theorem of this study.

Theorem 11 (Existence and uniqueness of solution for system (15))
The function $g(A)$ defined in equation (15) when angles ϕ_0 and ϕ_1 satisfy Assumption 4, admits a unique solution for $A \in (0, A_{\max}]$, where

$$A_{\max} = \delta + 2\theta_{\max}\left(1 + \sqrt{1 + \delta/\theta_{\max}}\right), \qquad \theta_{\max} = \max\{0, \pi/2 + \phi_0\}.$$

Moreover, $h(A) > 0$ where $g(A) = 0$.

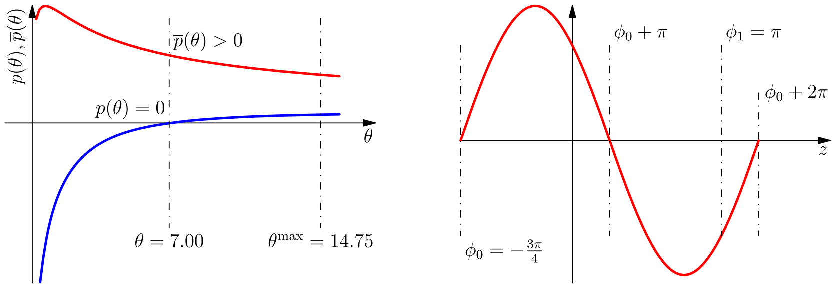


Figure 4. Left: functions $p(\theta)$ and $\overline{p}(\theta)$, when p vanishes, $\overline{p}$ is strictly positive. Right: the plot of $\cos z - w \sin z$. In both figures, $\phi_0 = -(3/4)\pi$ and $\phi_1 = \pi$.

Proof

The special cases $\phi_0 + \phi_1 = 0$ and $\phi_0 = \phi_1$ were previously considered and in Lemma 10. From Lemma 9, it follows that the two equations

$$p(\theta) = 0, \quad \text{for } \theta \geq 0, \qquad q(\theta) = 0, \quad \text{for } \theta \in [0, \theta_{\max}],$$

cannot be satisfied by the same θ in the specified range, so that they are mutually exclusive although one of the two is satisfied. Thus $g(A) = 0$ has a unique solution. To find the equivalent range of A, select the correct solution of $(\delta - A)^2 = 4A\theta_{\max}$. The two roots are the following:

$$A_1 = 2\theta_{\max} + \delta - 2\sqrt{\theta_{\max}^2 + \theta_{\max}\delta} = \frac{\delta^2}{2\theta_{\max} + \delta + 2\sqrt{\theta_{\max}^2 + \theta_{\max}\delta}} \leq \delta$$

$$A_2 = 2\theta_{\max} + \delta + 2\sqrt{\theta_{\max}^2 + \theta_{\max}\delta} \geq \delta,$$

and thus, A_2 is used to compute $A_{\max}$.

To check if $h(A) > 0$ when $g(A) = 0$, it suffices to consider the sign of $\overline{p}(\theta)$ and $\overline{q}(\theta)$. Suppose that $p(\theta) = 0$, there is to show that $\overline{p}(\theta) > 0$. For $|\phi_0| < \phi_1 \leq \frac{\pi}{2}$, the cosine in the numerator of $\overline{p}(\theta)$ is always positive, and so is the square root at the denominator, thus the integral $\overline{p}(\theta)$ is strictly positive. Now consider when $-\pi < \phi_0 < -\frac{\pi}{2}$. By using the change of variable $z + \theta - \phi_0$ in equation (29) for all $w \in \mathbb{R}$,

$$\overline{p}(\theta) = \overline{p}(\theta) + wp(\theta) = \int_{\phi_0}^{\phi_1} \frac{\cos z}{\sqrt{z + \theta - \phi_0}}\,\mathrm{d}z = \int_{\phi_0}^{\phi_1} \frac{\cos z - w \sin z}{\sqrt{z + \theta - \phi_0}}\,\mathrm{d}z. \tag{32}$$

In particular, it is true for $w = \frac{\cos\phi_0}{\sin\phi_0} > 0$ positive (which, incidentally is always positive because $-\pi < \phi_0 < -\frac{\pi}{2}$), so that the integrand function vanishes for the three values $z = \phi_0, \phi_0 + \pi, \phi_0 + 2\pi$. Moreover, $\cos z - w \sin z$ is strictly positive for $z \in (\phi_0, \phi_0 + \pi)$ and negative for $z \in (\phi_0 + \pi, \phi_0 + 2\pi)$ (Figure 4). Thus, integral (32) can be bound as

$$\overline{p}(\theta) > \int_{\phi_0}^{\phi_0 + 2\pi} \frac{\cos z - w \sin z}{\sqrt{z + \theta - \phi_0}}\,\mathrm{d}z \geq \frac{\int_{\phi_0}^{\phi_0 + 2\pi} \cos z - w \sin z\,\mathrm{d}z}{\sqrt{(\phi_0 + \pi) + \theta - \phi_0}} = 0.$$

If $q(\theta) = 0$, there is to show that $\overline{q}(\theta) > 0$. In this case, $A \geq \delta$ and from equation (30) $h(A)/\sqrt{A} = \Delta\widehat{\mathcal{C}} \cos(\phi_0 - \theta) - \Delta\widehat{\mathcal{S}} \sin(\phi_0 - \theta)$, with $\Delta\widehat{\mathcal{C}}, \Delta\widehat{\mathcal{S}} > 0$. If $\theta \in [0, \frac{\pi}{2} + \phi_0]$, then $-\frac{\pi}{2} \leq \phi_0 - \theta \leq 0$, thus the cosine is positive and the sine is negative; hence, the whole quantity is strictly positive. □

Corollary 12

All the solutions of the nonlinear system (10) are given by

$$L = \frac{\sqrt{\Delta x^2 + \Delta y^2}}{X_0(2A, \delta - A, \phi_0)}, \qquad \kappa = \frac{\delta - A}{L}, \qquad \kappa' = \frac{2A}{L^2},$$

where A is any root of $g(A) = Y_0(2A, \delta - A, \phi_0)$ provided that the corresponding $h(A) = X_0(2A, \delta - A, \phi_0) > 0$.

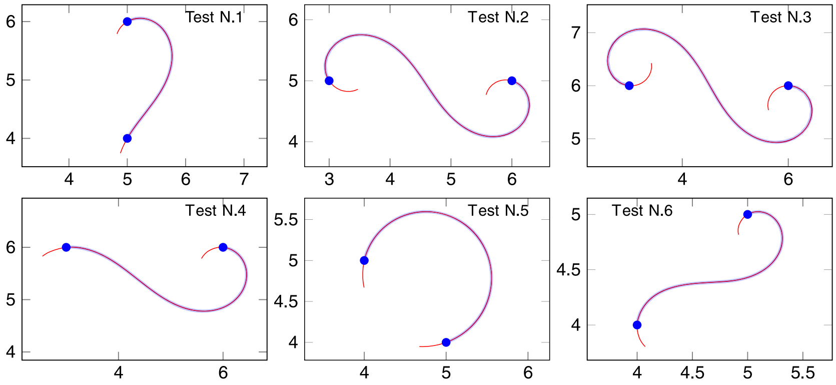
Corollary 13

If the angles ϕ_0 and ϕ_1 are in the range $[-\pi, \pi]$, with the exclusion of the points $\phi_0 = -\phi_1 = \pm\pi$, the solution exists and is unique for $-A_{\max} \leq A \leq A_{\max}$ where

$$A_{\max} = |\phi_1 - \phi_0| + 2\theta_{\max}\left(1 + \sqrt{1 + |\phi_1 - \phi_0|/\theta_{\max}}\right), \qquad \theta_{\max} = \max\{0, \pi/2 + \operatorname{sign}(\phi_1)\phi_0\}.$$

8. Numerical tests

The algorithm was implemented and tested in MATLAB (MATLAB Release 2013a, The MathWorks, Inc., Natick, Massachusetts, United States) and is available at Matlab Central [25]. For the Fresnel integrals computation, one can use the script by Telasula [26]. The first six tests are taken from reference [16], where the algorithm is presented; the algorithm is analyzed in reference [17]; moreover, a MATLAB implementation of the algorithm described in [16] is used for comparison.



	Tolerance 10^{-7}						Tolerance 10^{-14}					
#test	1	2	3	4	5	6	1	2	3	4	5	6
#iter. present method	3	3	2	2	2	3	4	4	3	3	3	4
#iter. method of ref. [22]	4	4	4	3	5	4	5	4	5	4	24	4

Figure 5. Results of test N.1 up to test N.6. The thick line is the result given by the present method, the thin line is the result obtained by reference [16]. The two results are indistinguishable, so the thin trajectory was prolonged beyond the end points to emphasize the overlapping.

Table III. Test N.7 and N.8 results.

	Test N.7				Test N.8			
	Present method		Meek & Walton		Present method		Meek & Walton	
k	iter	Error	iter	Error	iter	Error	iter	Error
1	2	2.6×10^{-16}	30	1.83×10^{-6}	3	3.18×10^{-14}	∞	3.76×10^{-9}
2	2	1.42×10^{-14}	29	1.85×10^{-6}	3	2.01×10^{-14}	∞	1.45×10^{-8}
3	3	0	28	1.38×10^{-6}	2	2.01×10^{-14}	∞	7.47×10^{-8}
4	2	4.33×10^{-17}	27	9.83×10^{-7}	2	2.84×10^{-14}	∞	3.47×10^{-8}
5	2	5.42×10^{-18}	26	6.96×10^{-7}	2	0	∞	1.07×10^{-9}
6	2	0	25	4.92×10^{-7}	2	1.42×10^{-14}	∞	5.53×10^{-9}
7	2	1.35×10^{-18}	24	3.48×10^{-7}	2	5.12×10^{-14}	∞	2.43×10^{-7}
8	2	0	23	2.46×10^{-7}	2	0	∞	3.09×10^{-6}
9	2	0	22	1.74×10^{-7}	2	0	∞	3.25×10^{-6}
10	2	0	21	1.23×10^{-7}	2	5.12×10^{-14}	∞	4.84×10^{-7}

Test 1 $\mathbf{p}_0 = (5,4), \mathbf{p}_1 = (5,6), \vartheta_0 = \pi/3, \vartheta_1 = 7\pi/6;$

Test 2 $\mathbf{p}_0 = (3,5), \mathbf{p}_1 = (6,5), \vartheta_0 = 2.14676, \vartheta_1 = 2.86234;$

Test 3 $\mathbf{p}_0 = (3,6), \mathbf{p}_1 = (6,6), \vartheta_0 = 3.05433, \vartheta_1 = 3.14159;$

Test 4 $\mathbf{p}_0 = (3,6), \mathbf{p}_1 = (6,6), \vartheta_0 = 0.08727, \vartheta_1 = 3.05433;$

Test 5 $\mathbf{p}_0 = (5,4), \mathbf{p}_1 = (4,5), \vartheta_0 = 0.34907, \vartheta_1 = 4.48550;$

Test 6 $\mathbf{p}_0 = (4,4), \mathbf{p}_1 = (5,5), \vartheta_0 = 0.52360, \vartheta_1 = 4.66003.$

The accuracy of fit (as in reference [16]) is determined by comparing the ending point as computed by both methods, with the given ending points. A tolerance of 10^{-7} and 10^{-14} is used in the stopping criterium for Newton iterations. For all the tests and for both methods, the position error of the solution does not exceed 10^{-14}. Also iterations are comparable, with a small advantage for the present method and are reported in the table of Figure 5, which also shows the computed curves.

The difference of the present method compared with the algorithm of reference [16] are in the transition zone (see e.g., test N.5) where the solution is close to be a circle arc or a segment. In fact, in this situation, the present method performs better without loosing accuracy or increasing in iterations. The following tests, which represent perturbed lines and circular arcs, highlight the differences as follows:

Test 7 $\mathbf{p}_0 = (0,0), \mathbf{p}_1 = (100,0), \vartheta_0 = 0.01 \cdot 2^{-k}, \vartheta_1 = -0.02 \cdot 2^{-k};$
Test 8 $\mathbf{p}_0 = (0,-100), \mathbf{p}_1 = (-100,0), \vartheta_0 = 0.00011 \cdot 2^{-k}, \vartheta_1 = \frac{3}{2}\pi - 0.0001 \cdot 2^{-k}.$

Table III collects the results for $k = 1,2,\ldots,10$. The error is computed as the maximum of the norm of the differences at the ending point as computed by the algorithm with the given ending point. The tolerance used for the Newton iterative solver for both the algorithms is 10^{-12}. Notice that the proposed algorithm computes the solution with constant accuracy and few iterations, whereas algorithm in [16] loses precision and uses more iterations. The ∞ symbol for iterations in Table III means that Newton method does not reach the required accuracy, and the solution is obtained using the last computed values. The maximum number of allowed iterations was 1000. In Table III, the algorithm of reference [16] has a large number of iteration in respect to the proposed method. To understand this behavior, notice that if $f(\theta)=0$ is the equation to be solved used in reference [17] for computing the clothoid curve and $g(A)=0$ is the equation to be solved in the present method, then the two functions f and g with the respective roots $\theta^\star$ and $A^\star$ are connected by the following relations:

$$f(\theta(A))\frac{\sqrt{2\pi}}{\sqrt{A}}=g(A),\qquad \theta(A)=\frac{(\delta-A)^2}{4A},\qquad \theta^\star=\theta(A^\star),\qquad f(\theta^\star)=g(A^\star)=0.$$

Both algorithms uses Newton–Raphson method to approximate the roots:

$$\theta_{k+1}=\theta_k-\frac{f(\theta_k)}{f'(\theta_k)},\qquad A_{k+1}=A_k-\frac{g(A_k)}{g'(A_k)}.$$

Denoting by $\epsilon_k=\theta_k-\theta^\star$ and $e_k=A_k-A^\star$, the error near the roots $\theta^\star$ and $A^\star$, at each iteration yields

$$\epsilon_{k+1}\approx C_f\epsilon_k^2,\qquad e_{k+1}\approx C_g e_k^2,\qquad C_f=-\frac{f''(\theta^\star)}{2f'(\theta^\star)},\qquad C_g=-\frac{g''(A^\star)}{2g'(A^\star)}.$$

Thus, the speed of convergence of the two methods is related to the constants C_f and C_g, respectively. Large values of the constants reflect a slow convergence. By using estimates (19) and (18) of Remark 2, the bound $|C_g|\lesssim 0.66$ is obtained. Joining this with the estimate (19) for the minimum of $|g'(A^\star)|$, it follows that the root is always well-conditioned and the Newton method converges quickly for the proposed algorithm. Thus, the proposed algorithm does not suffer of slow convergence as verified experimentally. To compare the constants C_f and C_g, notice that

$$\frac{g''(A^\star)}{g'(A^\star)}=\frac{\left((A^\star)^2-\delta^2\right)}{(2A^\star)^2}\frac{f''(\theta^\star)}{f'(\theta^\star)}-\frac{(A^\star)^2-3\delta^2}{A^\star\left((A^\star)^2-\delta^2\right)}\qquad\Rightarrow\qquad (2A^\star)^2C_g=((A^\star)^2-\delta^2)C_f+4A^\star\frac{(A^\star)^2-3\delta^2}{(A^\star)^2-\delta^2}. \tag{33}$$

Moreover,

- For $A^\star\ll\delta$, equation (33) is approximated with $(2A^\star)^2C_g\approx-\delta^2C_f-12A^\star$ and $C_f\approx-4A^\star(C_gA^\star+3)/\delta^2$. In this case, C_f is very low, and the algorithm of reference [16] converges faster than the proposed one.
- For $A^\star\gg\delta$, equation (33) is approximated with $(2A^\star)^2C_g\approx(A^\star)^2C_f+4A^\star$ and $C_f\approx4(C_g-1/A^\star)$. Thus, C_f is moderately low, and the algorithm of reference [16] converges more and less as the proposed one.
- For $A^\star=\delta+\epsilon$ with $\epsilon\approx0$, equation (33) is approximated with $4\delta^2C_g\approx2\delta\epsilon C_f-4\delta^2/\epsilon$ and $C_f\approx2\delta/\epsilon^2$. Thus, C_f may be huge for small ϵ, and the algorithm of reference [16] converges slowly or stagnates.

This behavior is verified in Table III. Notice that when $A^\star\ll\delta$ is true that the algorithm of reference [16] is faster but is also true that no more than 4 iterations are necessary for the present algorithm.

9. An application

The availability of a fast and reliable routine to compute Hermite G^1 interpolation as a black box opens the possibility of setting up more structured applications. When computing an interpolating spline, it is possible to take advantage of the linear varying curvature that clothoid curves offer with respect to other splines. In order to achieve the best results in terms of continuity of the curvature, a nonlinear least square problem is set up.

Problem 2 (Quasi G^2 fitting)
Let $\mathbf{p}_j=(x_j,y_j)$ for $j=1,\dots,N$ be assigned points and the free parameters θ_j be the angles associated to point $\mathbf{p}_j$. For each couple of the free parameters θ_j and θ_{j+1}, the G^1 Hermite interpolation problem is solved yielding the interpolating clothoid:

$$\kappa_j=\kappa_j(\theta_j,\theta_{j+1}),\qquad \kappa_j'=\kappa_j'(\theta_j,\theta_{j+1}),\qquad L_j=L_j(\theta_j,\theta_{j+1});$$

thus, the jump of curvature at point $\mathbf{p}_j$ for $j=2,3,\dots,N-1$ is $\Delta\kappa_j=(\kappa_{j-1}+L_{j-1}\kappa_{j-1}')-\kappa_j$. The objective function to be minimized is the sum of the squares of the jumps of the curvature at the extrema of each clothoid segments. The curvature at the first and the last point should minimize the squares of the jump with κ_{begin} and κ_{end},

$$F(\theta_1,\theta_2,\dots,\theta_N)=\frac{1}{\sqrt{N}}\left((\kappa_1-\kappa_{\text{begin}})^2+(\kappa_{N-1}+L_{N-1}\kappa_{N-1}'-\kappa_{\text{end}})^2+\sum_{j=2}^{N-1}\left((\kappa_{j-1}+L_{j-1}\kappa_{j-1}')-\kappa_j\right)^2\right)^{1/2}$$

The *quasi G^2 fitting problem* requires to find the angles $\theta_1,\theta_2,\dots,\theta_N$ that minimize $F(\theta_1,\theta_2,\dots,\theta_N)$.

Problem 2 involves several times the computation of the function `buildClothoid`. The nonlinear solver adopted in the numerical experiments was Levenberg–Marquardt implemented in `lsqnonlin` of the Optimization Toolbox of MATLAB, no information on the Jacobian was given; hence, derivatives were approximated by finite difference. This implies a heavier rely on the evaluation of the objective function itself. Four examples are herein proposed to compare the present algorithm with the algorithm of [16].

The tests have the following definition with MATLAB-like syntax; moreover, in all the test, $\kappa_{\text{begin}}=\kappa_{\text{end}}=0$ was chosen.

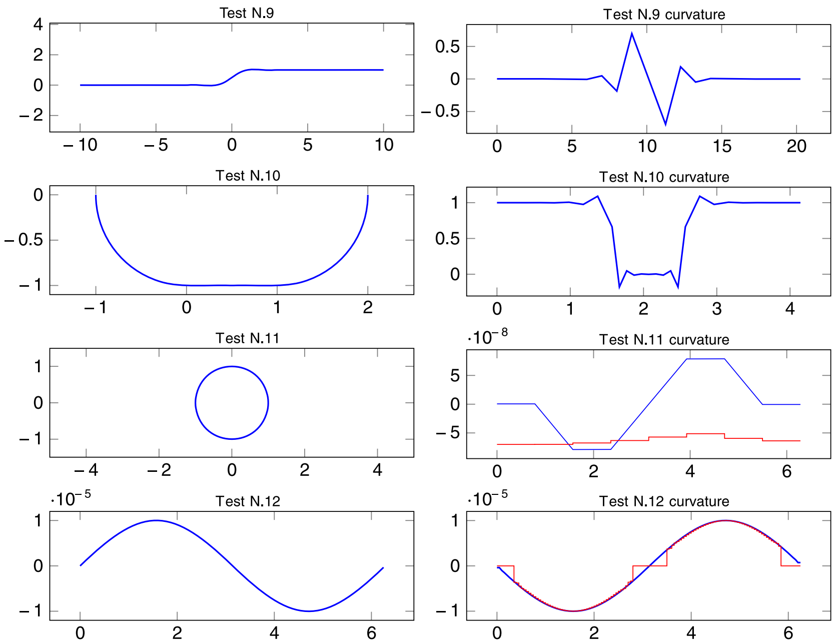


Figure 6. Results of the spline test N.9 up to test N.12. Left: interpolating spline. Right: arclength versus curvature. In blue, the curvature of the spline obtained with present method; in red, the curvature given by algorithm in [16]. Computed curvatures between the two methods for tests N.9 and N.10 are graphically indistinguishable. In the figure of the curvature of Test N.11, the plot represents the difference from 1 of the curvature, so that eventual jumps are magnified: the red line shows that reference [16] treats those clothoids as circles yielding a piecewise constant curvature. In test N.12, in red, the degenerate solution given by algorithm in [16].

Table IV. Results of the interpolating clothoid splines. it. is the number of iterations used by the MATLAB Levenberg–Marquardt algorithm, F ev. is the number of evaluations of the objective function, G^1 ev. is the number of evaluations of the routine that gives the G^1 interpolation, $F(\boldsymbol{\theta})$ is the value of the objective at the last computed point, and deg. is the number of times data was considered degenerate. The tolerance for `lsqnonlin` was 10^{-10}.

	Present method				Meek & Walton				
Test	it.	F ev.	G^1 ev.	$F(\boldsymbol{\theta})$	it.	F ev.	G^1 ev.	$F(\boldsymbol{\theta})$	deg.
9	4	70	840	$2.8{\cdot}10^{-15}$	29	441	5292	$1.3{\cdot}10^{-04}$	290
10	4	113	2938	$1.0{\cdot}10^{-12}$	8	257	6682	$7.6{\cdot}10^{-13}$	1195
11	4	50	400	$1.6{\cdot}10^{-15}$	19	214	1712	$4.5{\cdot}10^{-09}$	1497
12	2	381	47625	$6.2{\cdot}10^{-20}$	27	3581	447625	$7.2{\cdot}10^{-07}$	447506

Test 9 $x = [-10, -7, -4, -3, -2, -1, 0, 1, 2, 3, 4, 7, 10]$ and $y = [0, 0, 0, 0, 0, 0, 0.5, 1, 1, 1, 1, 1, 1]$;

Test 10 $x = [\cos(t), 0.1 : 0.1 : 0.9, 1 - \sin(t)]$, $y = [\sin(t), -\text{ones}(1, 9), \cos(t)]$, where $t = [\pi : \pi/16 : (3/2)\pi]$;

Test 11 $x = \cos([0 : \pi/4 : 2\pi]) + 10^{-7}\cos([0 : \pi/8 : \pi])$ and $y = \sin([0 : \pi/4 : 2\pi])$;

Test 12 $x = [0 : 0.05 : 2\pi]$ and $y = 10^{-5}\sin([0 : 0.05 : 2\pi])$.

Graphically, they represent two line segments, two arcs of circle joined by a segment, a perturbed circle and a perturbed line (Figure 6). The results are listed in Table IV, the residuals are very low for the present method, and the interpolating spline although computed as G^1 gives in practice a G^2 clothoid spline. The performance of the present algorithm yields a residual, which is several order lower than the residual of the algorithm present in literature. In computation of tests N.9 up to N.12, the G^1 fitting using algorithm in [16] is close to the transition zone where data are considered degenerate and therefore, approximated, respectively by a circle or a straight line. This results in a low precision fitting, which slows down the convergence of Levenberg–Marquardt algorithm. The last column of Table IV counts the number of times the G^1 fitting are considered degenerate. Although there is no direct correlation between the number of iterations and the number of degenerate cases, it is evident that degenerate cases corrupt both the accuracy of the final solution and the convergence speed. These test cases show that the present algorithm performs well also when used as an algorithmic kernel that is repeated several times.

10. Conclusions

An effective solution to the problem of Hermite G^1 interpolation with a clothoid curve is herein described with a full theoretical analysis. The present algorithm does not need the decomposition in mutually exclusive states as in previous geometric works. This introduces numerical instabilities and inaccuracies as it is shown in test N.7 and N.8 of Section 6 or test N.11 and N.12 of Section 9.

The interpolation problem was reduced to one single function in one variable, making the present algorithm compact, fast and robust. A guess function that allows to find that zero with very few iterations in all possible configurations was provided. Existence

Table A.I. Pseudocode for the computation of generalized Fresnel integrals (7) used for the computation of (15) and (20).

Function GeneralizedFresnelCS($a,\ b,\ c,\ k$)

```
$\varepsilon \leftarrow 0.01$;
if $|a| < \varepsilon$ then  $\hat{X}, \hat{Y} \leftarrow$ evalXYaSmall($a$,$b$,$k$,5) ;
else                        $\hat{X}, \hat{Y} \leftarrow$ evalXYaLarge($a$,$b$,$k$) ;
for $j = 0, 1, \ldots, k-1$ do
    $X_j \leftarrow \hat{X}_j \cos c - \hat{Y}_j \sin c$;   $Y_j \leftarrow \hat{X}_j \sin c + \hat{Y}_j \cos c$
end for
return $X,\ Y$
```

Function evalFresnelMomenta($t,\ k$)

```
$C_0 \leftarrow \mathcal{C}(t)$;   $S_0 \leftarrow \mathcal{S}(t)$;
$z \leftarrow \pi t^2/2$;   $c \leftarrow \cos z$;   $s \leftarrow \sin z$;
if $k > 1$ then  $C_1 \leftarrow s/\pi$;  $S_1 \leftarrow (1-c)/\pi$ ;
if $k > 2$ then  $C_2 \leftarrow (t\,s - S_0)/\pi$;  $S_2 \leftarrow (C_0 - t\,c)/\pi$ ;
return $C,\ S$
```

Function rLommel($\mu,\ \nu,\ b$)

```
$t \leftarrow (\mu+\nu+1)^{-1}(\mu-\nu+1)^{-1}$;
$r \leftarrow t$;   $n \leftarrow 1$;   $\varepsilon \leftarrow 10^{-50}$;
while $|t| > \varepsilon\,|r|$ do
    $t \leftarrow t\dfrac{(-b)}{2n+\mu-\nu+1}\dfrac{b}{2n+\mu+\nu+1}$;
    $r \leftarrow r + t$;   $n \leftarrow n + 1$
end while
return $r$
```

Function evalXYaLarge($a,\ b,\ k$)

```
$s \leftarrow a/|a|$;  $z \leftarrow \sqrt{|a|/\pi}$;  $\ell \leftarrow sb/(z\,\pi)$;
$\gamma \leftarrow -sb^2/(2\,|a|)$; $s_\gamma \leftarrow \sin\gamma$;  $c_\gamma \leftarrow \cos\gamma$;
$C^+, S^+ \leftarrow$evalFresnelMomenta($\ell + z, k$);
$C^-, S^- \leftarrow$evalFresnelMomenta($z, k$);
$\Delta\mathcal{C} \leftarrow C^+ - C^-$;   $\Delta\mathcal{S} \leftarrow S^+ - S^-$;
$X_0 \leftarrow z^{-1}\,(c_\gamma\,\Delta\mathcal{C}_0 - s\,s_\gamma\,\Delta\mathcal{S}_0)$;
$Y_0 \leftarrow z^{-1}\,(s_\gamma\,\Delta\mathcal{C}_0 + s\,c_\gamma\,\Delta\mathcal{S}_0)$;
if $k > 1$ then
    $d_c \leftarrow \Delta\mathcal{C}_1 - \ell\Delta\mathcal{C}_0$;
    $d_s \leftarrow \Delta\mathcal{S}_1 - \ell\Delta\mathcal{S}_0$;
    $X_1 \leftarrow (c_\gamma\,d_c - s\,s_\gamma\,d_s)\,/z^2$;
    $Y_1 \leftarrow (s_\gamma\,d_c + s\,c_\gamma\,d_s)\,/z^2$;
end if
if $k > 1$ then
    $d_c \leftarrow \Delta\mathcal{C}_2 + \ell(\ell\Delta\mathcal{C}_0 - 2\Delta\mathcal{C}_1)$;
    $d_s \leftarrow \Delta\mathcal{S}_2 + \ell(\ell\Delta\mathcal{S}_0 - 2\Delta\mathcal{S}_1)$;
    $X_2 \leftarrow (c_\gamma\,d_c - s\,s_\gamma\,d_s)\,/z^3$;
    $Y_2 \leftarrow (s_\gamma\,d_c + s\,c_\gamma\,d_s)\,/z^3$;
end if
return $X,\ Y$
```

Function evalXYaZero($b,\ k$)

```
if $|b| < \varepsilon$ then
    $X_0 \leftarrow 1 - \dfrac{b^2}{6}\left(1 - \dfrac{b^2}{20}\right)$;
    $Y_0 \leftarrow \dfrac{b^2}{2}\left(1 - \dfrac{b^2}{6}\left(1 - \dfrac{b^2}{30}\right)\right)$;
else
    $X_0 \leftarrow \dfrac{\sin b}{b}$;
    $Y_0 \leftarrow \dfrac{1-\cos b}{b}$;
end if
$A \leftarrow b\sin b$;
$D \leftarrow \sin b - b\cos b$;
$B \leftarrow bD$;
$C \leftarrow -b^2\sin b$;
for $k = 0, 1, \ldots, k-1$ do
    $t_1 \leftarrow \mathtt{rLommel}\left(k+\frac{1}{2}, \frac{3}{2}, b\right)$;
    $t_2 \leftarrow \mathtt{rLommel}\left(k+\frac{3}{2}, \frac{1}{2}, b\right)$;
    $t_3 \leftarrow \mathtt{rLommel}\left(k+\frac{3}{2}, \frac{3}{2}, b\right)$;
    $t_4 \leftarrow \mathtt{rLommel}\left(k+\frac{1}{2}, \frac{1}{2}, b\right)$;
    $X_{k+1} \leftarrow \dfrac{1}{1+k}(kA\,t_1 + B\,t_2 + \cos b)$;
    $Y_{k+1} \leftarrow \dfrac{1}{2+k}(C\,t_3 + \sin b) + D\,t_4$;
end for
return $X,\ Y$
```

Function evalXYaSmall($a,\ b,\ k,\ p$)

```
$\hat{X}, \hat{Y} \leftarrow$ evalXYaZero($b, k + 4p + 2$);
$t \leftarrow 1$;
for $j = 0, 1, \ldots, k-1$ do
    $X_j \leftarrow X_j^0 - \dfrac{a}{2}Y_{j+2}^0$;
    $Y_j \leftarrow Y_j^0 + \dfrac{a}{2}X_{j+2}^0$;
end for
for $n = 1, 2, \ldots, p$ do
    $t \leftarrow (-t\,a^2)/(16n(2n-1))$;
    $s \leftarrow a/(4n+2)$;
    for $j = 0, 1, \ldots, k-1$ do
        $X_j \leftarrow X_j + t(\hat{X}_{4n+j} - s\,\hat{Y}_{4n+j+2})$;
        $Y_j \leftarrow Y_j + t(\hat{Y}_{4n+j} + s\,\hat{X}_{4n+j+2})$;
    end for
end for
return $X,\ Y$
```

and uniqueness of the solution was discussed and proved in Section 7. Asymptotic expansions near critical values for Fresnel-related integrals are derived to keep the accuracy uniform. Implementation details of the present algorithm are given in Appendix using pseudo code and can be easily translated in any programming language.

The algorithm was successfully tested in any possible situation. The accurate computation of the clothoid needs an equally accurate computation of $g(A)$ and $g'(A)$ and thus, the accurate computation of Fresnel related functions $X_0(a,b,c)$ and $Y_0(a,b,c)$ with associated derivatives. These functions are a combination of Fresnel and Fresnel momenta integrals, which are precise for large $|a|$ and small momenta. For the computation, only the knowledge of the first two momenta are necessary so that the inaccuracy for higher momenta does not pose any problem. A different problem is the computation of these integrals for small values of $|a|$. In this case, demanding (but stable) expansion are used to compute the Fresnel momenta with high accuracy. Finally, a theoretical proof completes the exposition and guarantees the existence of the solution in all possible cases.

The solution of the interpolation problem is uniformly accurate even when close to a straight line or an arc of circle, and this was not the case of algorithms found in literature. In fact, even in domains where other algorithms solve the problem, the present method performs better in terms of accuracy and number of iterations. For example, in tests (1–6) proposed by Walton and Meek [16], the present method requires three iterations against 4 and 5; in critical tests (7 and 8), the present algorithm converges in all cases in two and three iterations (against 20–30 with loss of precision, or no convergence at all after 1000 iterations). It is to point out that critical situations such as those occur in practice every time the Hermite data are acquired with (even a low) corrupting noise and no longer represents straight lines or circles, as was described in the applications of Section 9.

Appendix A. Algorithms for the computation of Fresnel momenta

In Table A.I the algorithmic version of the analytical expression derived in the paper is herein presented. These algorithms are necessary for the computation of the main function `buildClothoid`, which takes the input data $(x_0, y_0, \vartheta_0, x_1, y_1, \vartheta_1)$ and returns the parameters (κ, κ', L) that solve the problem as expressed in equation (9). Function `GeneralizedFresnelCS` computes the generalized Fresnel integrals (7). It distinguishes the cases of a larger or smaller than a threshold ε. The role and the value of ε are discussed in Section 6. Formulas (22) and (23), used to compute $X_k(a,b)$ and $Y_k(a,b)$ at arbitrary precision when $|a| \geq \varepsilon$, are implemented in function `evalXYaLarge`. Formulas (24) and (25), used to compute $X_k(a,b)$ and $Y_k(a,b)$ at arbitrary precision when $|a| < \varepsilon$, are implemented in function `evalXYaSmall`. This function requires computation of equation (26) implemented in function `evalXYaZero`, which needs (reduced) Lommel function (27) implemented in function `rLommel`.

References

1. McCrae J, Singh K. Sketching piecewise clothoid curves. *Computers & Graphics* 2009; **33**:452–461.
2. Bertolazzi E, Biral F, Da Lio M. Symbolic-numeric efficient solution of optimal control problems for multibody systems. *Journal of Computational and Applied Mathematics* 2006; **185**:404–421.
3. Dai R, Cochran JE. Path planning for multiple unmanned aerial vehicles by parameterized cornu-spirals. *American Control Conference, 2009. ACC '09.*, St. Louis, Missouri, USA, 2009, 2391–2396.
4. De Cecco M, Bertolazzi E, Miori G, Oboe R, Baglivo L. PC-sliding for vehicles path planning and control - design and evaluation of robustness to parameters change and measurement uncertainty. *ICINCO-RA (2)'2007*, Angers, France, 2007, 11–18.
5. Labakhua L, Nunes U, Rodrigues R, Leite FS. Smooth trajectory planning for fully automated passengers vehicles: spline and clothoid based methods and its simulation. In *Informatics in Control Automation and Robotics*. Lecture Notes Electrical Engineering, Vol. 15. Springer: Berlin Heidelberg, 2008; 169–182.
6. Manz M, Von Hundelshausen F, Wuensche HJ. A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments. *2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010, 2410–2415.
7. Wilde DK. Computing clothoid segments for trajectory generation. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*, St. Louis, Missouri, USA, 2009, 2440–2445.
8. Baran I, Lehtinen J, Popović J. Sketching clothoid splines using shortest paths. *Computer Graphics Forum* 2010; **29**:655–664.
9. Bertails-Descoubes F. Super-clothoids. *Computer Graphics Forum* 2012; **31**:509–518.
10. Wang LZ, Miura KT, Nakamae E, Yamamoto T, Wang TJ. An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves. *Computer-Aided Design* 2001; **33**:1049–1058.
11. Arechavaleta G, Laumond J-P, Hicheur H, Berthoz A. An optimality principle governing human walking. *IEEE Transactions on Robotics* 2008; **24**:5–14.
12. Daily J, *Analysis of critical speed yaw scuffs using spiral curves*, Technical Report SAE Technical Paper, 2012-01-0606, 2012.
13. Kostov VP, Degtiariova-Kostova EV, *Some properties of clothoids*, Technical Report RR-2752, INRIA, 1995.
14. Stoer J. Curve fitting with clothoidal splines. *Journal of Research of the National Bureau of Standards* 1982; **87**:317–346.
15. Kimia BB, Frankel I, Popescu A-M. Euler spiral for shape completion. *International Journal of Computer Vision* 2003; **54**:157–180.
16. Walton DJ, Meek DS. An improved euler spiral algorithm for shape completion. In *Canadian Conference on Computer and Robot Vision, 2008. CRV '08.* IEEE: Windsor, Ontario, Canada, May 2008; 237–244.
17. Walton DJ, Meek DS. Interpolation with a single cornu spiral segment. *Journal of Computational and Applied Mathematics* 2009; **223**:86–96.
18. Abramowitz M, Stegun IA. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards Applied Mathematics Series, vol. 55. U.S. Government Printing Office: Washington, D.C., 1964.
19. Bulirsch Roland. Numerical calculation of the sine, cosine and fresnel integrals. *Numerische Mathematik* 1967; **9**(5):380–385.
20. Thompson WJ. *Atlas for Computing Mathematical Functions: An Illustrated Guide for Practitioners with Programs in C and Mathematica with Cdrom* 1st ed. John Wiley & Sons, Inc.: New York, NY, USA, 1997.
21. Press WH, Vetterling WT, Teukolsky SA, Flannery BP. *Numerical Recipes in C++: The Art of Scientific Computing* 2nd ed. Cambridge University Press: New York, NY, USA, 2002.

22. Shirley EL, Chang EK. Accurate efficient evaluation of lommel functions for arbitrarily large arguments. *Metrologia* 2003; **40**(S5):5–8.
23. Olver FWJ, Lozier DW, Boisvert RF, Clark CW (eds). *NIST Handbook of Mathematical Functions*. U.S. Department of Commerce National Institute of Standards and Technology: Washington, DC, 2010. With CD-ROM.
24. Watson GN. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press: Cambridge, England, 1944.
25. Bertolazzi E, Frego M. *G^1 fitting with clothoids*, 2013. Available from: http://www.mathworks.com/matlabcentral/fileexchange/42113-g1-fitting-with-clothoids.
26. Telasula V. *Fresnel cosine and sine integral function*, 2005. Available from: http://www.mathworks.it/matlabcentral.