Systematic Calculation of Loop Points for Parametric Curves

Skip Thompson
Department of Mathematics & Statistics
Radford University
Radford, VA 24142
thompson@radford.edu

Abstract *We consider the task of systematically locating loop points for parametric curves. We first discuss techniques that can be used to accomplish this task. We then give several examples to illustrate the methods discussed. We describe the manner in which these techniques can be used to locate loop points using the Maple Computer Algebra System (CAS). We describe Maple worksheets that can be used to perform the necessary calculations in an interactive fashion while at the same time using Maple to explore interesting properties of parametric curves.*

1 Introduction

This paper is devoted to the question of locating loop points for parametric curves. At first glance, calculating loop points appears to be a simple and straightforward task; but as we will see this is not necessarily the case. The procedure often used to locate specific loop points is to guess a value of either x_c or y_c, say y_c, near a loop point, solve the equation $y(t) = y_c$ for two different values of t, and use these values to approximate the x–coordinate at the loop point. In this paper we take a somewhat different approach. We describe methods for systematically locating all loop points. We describe how to use forward and backward parameterizations to locate loop points by finding where the forward and backward curves "intersect." To overcome an obvious problem with this approach we constrain the solution in a natural manner and formulate the problem as a least squares problem for which the Maple optimization solvers NLPSolve and LSSolve can be used to locate loop points. After identifying potential difficulties with this approach, we formulate the problem as a constrained system of nonlinear equations that can be solved using the Maple nonlinear equation solver fsolve. A troublesome issue arises since fsolve does not directly allow the type of constraint that needs to be used. However, we show that it is possible to force fsolve to incorporate the constraint indirectly. We describe graphical techniques for locating solution bracketing intervals and illustrate their use for several interesting parametric curves.

2 The Basic Problem

In order to calculate areas of regions enclosed by loops in a parametric curve or perform other tasks such as finding perimeters of loops, it is first necessary to find the loop points for the curve as well as the corresponding parameter values. Even for curves having only a few loop points, this is not necessarily easy to do. Suppose the curve in question is defined for $a \leq t \leq b$ by

$$(x, y) = (x(t), y(t)). \tag{1}$$

Some graphing calculators and CAS packages allow zooming of the graph of a curve to visually approximate loop points; but it is still necessary to find the values of t corresponding to these points. Usually this is done in a simplified fashion using whatever problem specific information is available to solve the necessary equations. This process can be illustrated using the following example from [2] (which we will refer to as Example 1 herein). The parametric curve of interest is defined by

$$x(t) = t^3 - 12t, \;\; y(t) = 3t^2 + 2t + 5, \;\; -4 \leq t \leq 3.5. \tag{2}$$

The curve has one loop point as depicted in Fig. 1.

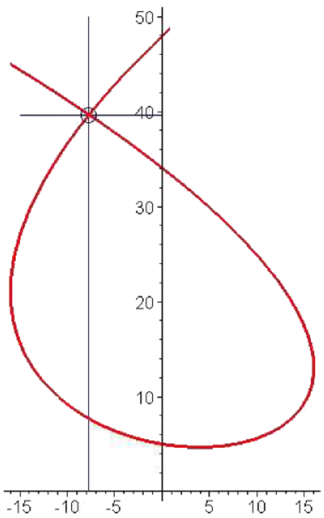

Figure 1: Parametric Curve and Loop Point for Example 1

What we wish to do is find two different values t_1 and t_2 for which

$$x(t_1) = x(t_2), \;\; y(t_1) = y(t_2). \tag{3}$$

By zooming the graph, the value of y at the loop point is estimated to be $y_c = 39.667$. Since $y(t) = y_c$ is quadratic, it is easy to find the two values of t corresponding to $y(t) = y_c$. The exact loop point for this curve is $(x, y) = (-208/27, 119/3)$ obtained for the values $t_1 = -(\sqrt{105} + 1)/3$ and $t_2 = (\sqrt{105} - 1)/3$. These values can be verified by solving the system

$$x_i(t) = x_i(s), \;\; y_i(t) = y_i(s)$$

after removing a common factor of $s - t$ (since $s = t$ is always a solution of this system).

More generally, it is necessary to solve the nonlinear equation $y(t) = y_c$ (or $x(t) = x_c$) and there may be any number of solutions that have to be considered and sorted out. This step can be frustrating for students (as well as the rest of us!). This issue will be addressed further in §3.

3 Solution Approaches

In this section we consider solution approaches, based on using the Maple CAS, for calculating loop points and the corresponding values of t for a given parametric curve. Throughout the discussion we assume that the parametric curves do not replicate any portions of themselves on any subinterval of $[a, b]$. If we were trying to locate a point of intersection of two different curves, $(x_1(t), y_1(t))$ and $(x_2(t), y_2(t))$, we could use any appropriate nonlinear equation solver such as fsolve in Maple to find the point of intersection by solving the system

$$x_1(t) = x_2(s), \;\; y_1(t) = y_2(s).$$

We could find a particular solution of interest by constraining s and t to belong to appropriate intervals. (For a specific example, refer to Item 10 in §7 which finds the points of intersection of the two curves in Example 7 of [4].) We will now explore ways of doing something similar to this despite the fact that we have only one parametric curve to work with.

3.1 Constrained Least Squares Approach

Suppose we parameterize the curve in the forward and backward directions using

$$x_f(t) = x(a+t), \;\; y_f(t) = y(a+t), \;\; 0 \le t \le b-a \tag{4}$$

and

$$x_b(s) = x(b-s), \;\; y_b(s) = y(b-s), \;\; 0 \le s \le b-a \tag{5}$$

and attempt to solve the nonlinear system

$$x_f(t) = x_b(s), \;\; y_f(t) = y_b(s)$$

subject to the constraints

$$0 \le s \le b-a, \; 0 \le t \le b-a. \tag{6}$$

The obvious problem with this approach is that there are too many solutions: every point on the curve is a solution! To be specific, any s and t for which $a + t = b - s$ yields a solution to the nonlinear system. One way to circumvent this difficulty is to enforce an additional constraint

$$0 \le s + t \le \alpha(b-a) \tag{7}$$

for some value of α where $0 < \alpha < 1$. We require $\alpha < 1$ because it is not possible to enforce a bound involving strict inequality such as $0 \le s + t < b - a$. We can then use either of the Maple minimization solvers NLPSolve and LSSolve, which allow such a constraint, to minimize $(x_f(t) - x_b(s))^2 + (y_f(t) - y_b(s))^2$. This approach often works well given a reasonably accurate initial guess (x_c, y_c). However, if care is not taken, it is possible for either solver to locate a local minimum other than a loop point; and there can be many such local minima even for simple parametric curves.

3.2 Constrained Nonlinear System Approach

NLPSolve and LSSolve are natural candidates since they allow the constraint $s + t \leq \alpha(b - a)$ to be enforced directly. We would also like to use fsolve to solve the nonlinear system

$$x_f(t) = x_b(s), \;\; y_f(t) = y_b(s) \tag{8}$$

or equivalently

$$F(t, s) = x_f(t) - x_b(s) = 0, \;\; G(t, s) = y_f(t) - y_b(s) = 0 \tag{9}$$

subject to constraint (6) and constraint (7) (if necessary).

fsolve directly accommodates the simple bounds defined by constraint (6) but does not directly accommodate the bound defined by constraint (7). However, we can force fsolve to enforce constraint (7) indirectly. To accomplish this, we add a third artificial equation to the system

$$z = \alpha(b - a) - (s + t) \tag{10}$$

and impose the constraint

$$0 \leq z \leq \alpha(b - a). \tag{11}$$

Note that system (9) is nonsingular at a loop point for which there are two tangent lines at the point. This is the case since the determinant of the Jacobian for system (9) is equal to

$$J\left(t_1, t_2\right) \;=\; -x'\left(t_1\right)\, y'\left(t_2\right) \;+\; x'\left(t_2\right)\, y'\left(t_1\right). \tag{12}$$

If this determinant is zero and both $x'\left(t_1\right)$ and $x'\left(t_2\right)$ are nonzero, we obtain

$$\frac{y'\left(t_1\right)}{x'\left(t_1\right)} = \frac{y'\left(t_2\right)}{x'\left(t_2\right)}$$

so that

$$\frac{dy}{dx}\left(t_1\right) = \frac{dy}{dx}\left(t_2\right).$$

We note that the determinant of the system containing the artificial variable (10) is also equal to $J\left(t_1, t_2\right)$.

4 Computing Loop Points

It is necessary to control the solution appropriately to solve system (9) since care must be taken to achieve convergence. In some cases one need only supply a reasonably accurate guess (x_c, y_c) but this generally is inadequate due to the nature of the zeroes of system (9) as well as the behavior of $J\left(t_1, t_2\right)$. For many problems if an isolated loop containing the loop point is found it is possible to find the loop point by experimenting with different values of α. This approach is illustrated in Item 3 in §7.

A more reliable approach is to isolate the loop point and supply what we will call an `X` at which two portions of the curve cross only at the loop point as illustrated for Example 1 in Fig. 2. (Note that when this is done it is not necessary to use constraint (7)). Doing this requires finding bracketing intervals for the forward and backward curves that intersect only at the loop point. Refer to §7 for Maple worksheets for accomplishing this. The worksheets include several problem dependent sections that allow the system to be solved in an interactive fashion. In the first section, students define the problem and various solution options. Plots are then generated that can be used to help determine approximate loop points. The next section of the worksheet can be run as many times as desired. By experimenting with subintervals of $[a, b]$ and inspecting plots, students are able to locate an X for the loop point. The loop point can then be found using either of fsolve, LSSolve, or NLPSolve.

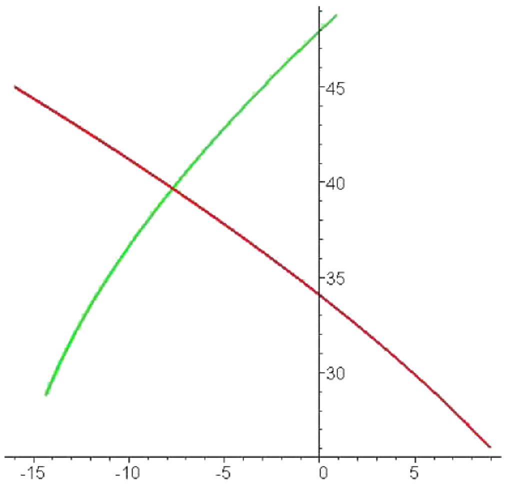

Figure 2: Isolated X for Example 1

The worksheets in §7 contain several graphical aids that we find useful to minimize (and in most cases eliminate) the potential frustration of finding an appropriate X. The purpose of these graphical aids is to find bracketing intervals (t_1, t_2) and (s_1, s_2) with $t_1 < t_2 < s_1 < s_2$ that produce an X and which can be used in lieu of a and b in constraint (7). By inspecting the graphs for Example 1, we find that $t_1 = -4.0$, $t_2 = -3.0$, $s_1 = 2.5$, and $s_2 = 3.5$ provide suitable bracketing intervals. Fig. 3 depicts $x(t)$, $y(t)$, and horizontal lines corresponding to x_c and y_c. In the plot $x(t)$ is shifted vertically by the amount $y_c - x_c$. Loop points correspond to points at which the shifted $x(t) + y_c - x_c$ and $y(t)$ curves intersect near the line $y = y_c$. Because $x(t) + y_c - x_c$ and $y(t)$ are plotted as functions of t the different intersections near the line $y = y_c$ correspond to the same loop point.

Fig. 4 illustrates a second graphical aid that is useful for determining the X bracketing intervals. It depicts the space curve $(x(t), y(t), t)$. Of interest are the points at which this curve intersects the vertical line (x_c, y_c, t) as the curve winds around the cylinder determined by the curve. These intersections correspond to the loop point; and the corresponding values of t yield the loop point.

Using this approach, Item 1 in §7 produces the following results for this example.

```
Computed results:
      NLPSolve     LSSolve     fsolve   Exact solution:
T1 = -3.7489836  -3.7489836  -3.7489836   -3.7489836
T2 =  3.0823169   3.0823169   3.0823169    3.0823169
```

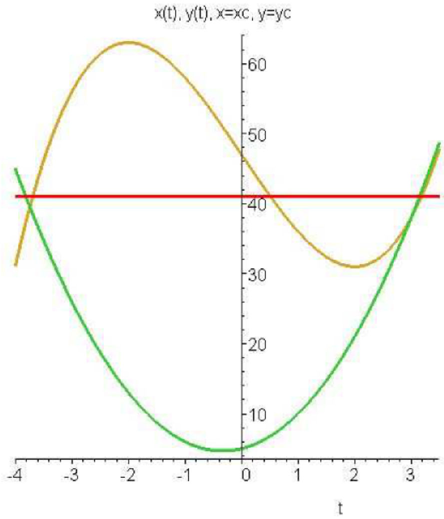

Figure 3: Temporal Curves for Example 1

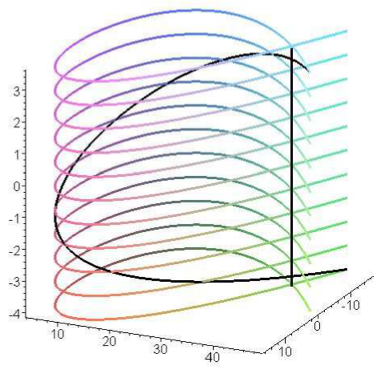

Figure 4: Space Curve for Example 1

```
x  = -7.7037037  -7.7037037  -7.7037037   -7.7037037
y  = 39.6666667  39.6666667  39.6666667   39.6666667
```

To explain the behavior of fsolve, it is instructive to consider where a loop point falls relative to the curves for which the Jacobian for system (9) is singular. The worksheets in §7 generate the necessary plots. Fig. 5 depicts (using original coordinates) the values of t and s for which the Jacobian is singular along with the t values that correspond to the loop point. The diagonal $s = t$ corresponds to the line $s + t = b - a$ using the forward and backward parameterization coordinates. Note the presence of the additional curved segment along which the Jacobian is singular.

4.1 Example 2

A bit more complicated curve further illustrates the need for a systematic solution. Consider the following curve.

$$x(t) = (t^2 - t)\cos(t) + 2,\ y(t) = (t^3 - 3)\sin(t) + 3,\ -0.5 \le t \le 1.6. \tag{13}$$

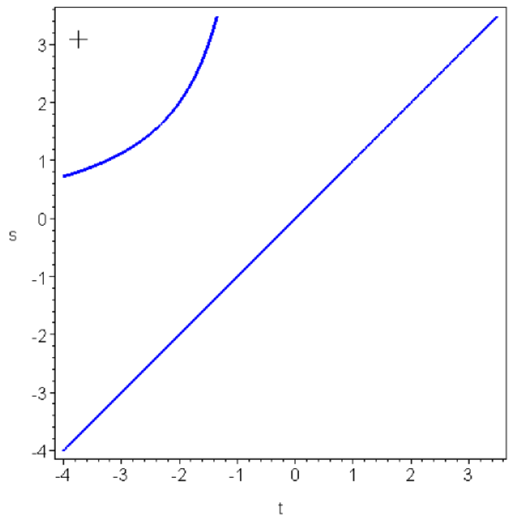

Figure 5: Singular Jacobian Curves and Loop Point for Example 1

Although the curve has only one loop point, it is not really feasible to use a simplified approach such as that used in [2] for Example 1 to find the point due to the nonlinear equations that must be solved. However, a quick inspection of the graphs in the worksheet for this example shows that a reasonable approximation for the loop point is $(2.1, 3.1)$ and suitable bracketing intervals correspond to $t_1 = -0.5$, $t_2 = 0.25$, $s_1 = 1.3$, and $s_2 = 1.6$. Using these values, each of the three solvers correctly locates the loop point at $(2.068, 3.193)$. These remarks can be verified using Item 4 of the Maple worksheets in §7.

Fig. 6 depicts the curves on which the Jacobian is singular along with the (t, s)–values that correspond to the loop point. Note that the additional curves along which the Jacobian is singular are a bit more complex than were the curves for Example 1.

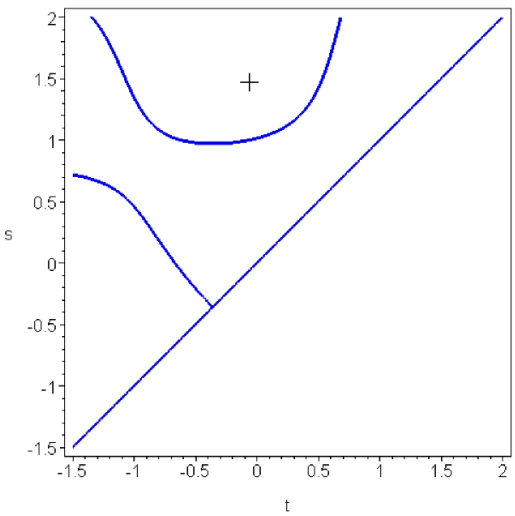

Figure 6: Singular Jacobian Curves and Loop Point for Example 2

Comments are in order regarding the situation in which only a single loop has been isolated rather than an X. Often fsolve will find the loop point for such a curve. This is illustrated in Items 3 and 5 in §7 for Examples 1 and 2. The worksheets do not attempt to find an X yet they are able to successfully locate the loop point using constraint (6) (with or without using constraint (7)). However, the success of fsolve in this situation can be problematic. For Example 2, if $a = -1.5$ and $b = 2$, fsolve is not successful if an X is not located. It is successful however when applied to the smaller loop obtained for $a = -0.5$ and $b = 1.6$. These comments can be verified using Item 4 in §7.

4.2 Example 3

The previously described methods are also applicable to more complicated parametric curves. For example, consider the following curve

$$x(t) = t + 2\sin(2t),\ \ y(t) = t + \cos(5t),\ \ 0 \le t \le 2\pi. \tag{14}$$

The graph of this curve depicted in Fig. 7 indicates some of the difficulties that must be considered.

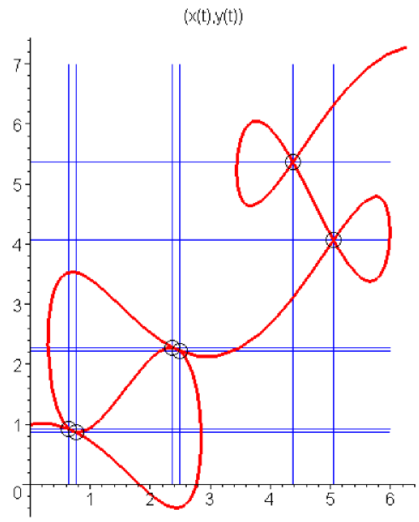

Figure 7: Parametric Curve and Loop Points for Example 3

There are six loop points for this curve. There are two pairs of two very close loop points. (Depending on the plot resolution used each pair appears as a single loop point.) The corresponding values of t and s are considerably different. For each pair, the second loop point occurs after the curve passes through the first point, moves away from this point, and then comes back to the second nearby loop point. For each pair, it is difficult to solve system (9) without locating an X for each of the two points in the pair. However, Item 6 in §7 can be used interactively to locate an X for each point. Without doing this it is somewhat unpredictable which point (if either) each of the three solvers will locate. In fact, convergence is problematic for one of the fixed points due to the fact that the curve is almost vertical just before the second point in the pair occurs.

Fig. 8 depicts the values of t and s for which the Jacobian is singular along with the t–values for the six loop points. Note that the two pairs of closely spaced loop points are very near portions of the curve on which the Jacobian is singular. This further demonstrates the need to carefully control the fsolve solution. Additionally, the structure of the singular Jacobian curve is more complicated than for the previous examples. This too demonstrates the need for constraining the solution in order to obtain convergence of the Newton iteration used in fsolve. As in Example 2, it is necessary to reduce the interval $[a, b]$ considerably in order for fsolve to be successful.

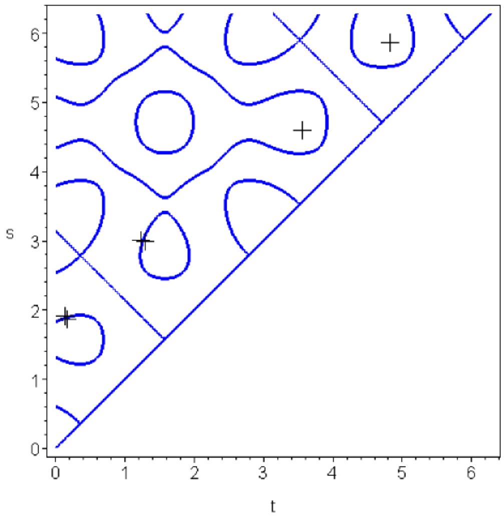

Figure 8: Singular Jacobian Curves and Loop Points for Example 3

4.3 Example 4

As the complexity of the graph of the parametric curve grows or the number of loop points increases, the work needed to find appropriate bracketing intervals using the repetitive calculations in the worksheets can become cumbersome. In such cases, by first studying the analytic properties of the curve, one can often determine the intervals more efficiently. The following example from [2] is such a curve.

$$x(t) = r(t)\ \cos(t), \ \ y(t) = r(t)\ \sin(t)$$

where

$$r(t) \ = \ \sin(t) + \sin^3(2.5t).$$

This curve has a total of 12 loop points as depicted in Fig. 9. Finding the loop points "on the fly" without first studying carefully how the curve behaves analytically can be quite an undertaking. However, by noting that the curve is traced out exactly once for $a = 0 \leq t \leq b = 4\pi$ and that it is symmetric with respect to the y–axis, we can simply plot the curve using an increment of $\frac{\pi}{4}$ in order to easily determine an appropriate `X` for each loop point. The adventure is just beginning though.

The four loop points located at $(\pm 0.29, 0.90)$ and $(\pm 0.48, 0.35)$ are particularly troublesome for fsolve. Although the forward and backward curves cross at each point, system (9) is poorly conditioned at these loop points. The two curves are virtually coincidental near the loop point. In addition, each curve has an inflection point near the loop point. Depending on the precision used, fsolve is not able to converge to the loop point (even if very small bracketing intervals are used), This is the case if the equivalent of double precision is used as it was for the other examples in this paper. However, with default precision fsolve is able to find each point. Both NLPSolve and LSSolve locate the four

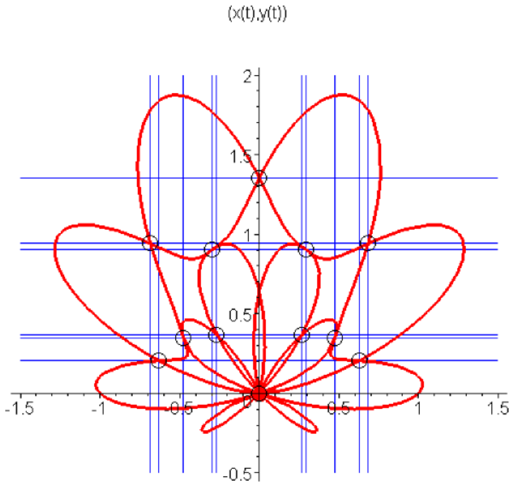

Figure 9: Parametric Curve and Loop Points for Example 4

points for each precision (although with a slight loss of accuracy in some cases). At default precision, each of the three solvers locate each of the 12 loop points. Item 7 in §7 can be used to verify these observations.

Although we forego a detailed discussion, we note that for polar curves such as $r(t)$ in this example and similar examples, the performance of fsolve near troublesome loop points sometimes can be improved by solving equivalent polar systems; but this too can lead to badly conditioned systems.

With regard to the convergence of the Newton iteration for fsolve, as depicted in Fig. 10, the structure of the singular Jacobian curve is considerably more complicated for this example than for the previous three examples.

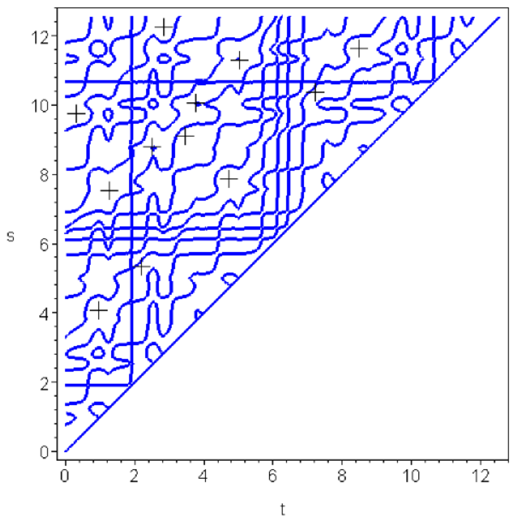

Figure 10: Singular Jacobian Curves and Loop Points for Example 4

4.4 Example 5

We offer this example as a curve students might wish to explore both analytically and numerically. It is a parametric curve from [2] defined as follows. For $-6\pi \le t \le 6\pi$,

$$x(t) = r(t)\,\cos(t),\;\; y(t) = r(t)\,\sin(t)$$

where

$$r(t) \;=\; 2 - 5\sin(t/6).$$

As depicted in Fig. 11, the curve is symmetric with respect to the x–axis and has 10 loop points with 6 of the points on the x–axis. It is a simple matter to locate the loop points on the x–axis. By solving $y(t) = 0$ it is seen that the t–values for these points are $\{k\pi,\ k = -6, \ldots, 6\}$, $6\sin^{-1}(2/5)$, and $6\,\pi - 6\sin^{-1}(2/5)$. These 15 values yield the 6 loop points (and two non–loop points). It is straightforward to see which loop points correspond to each of the 15 values of t. Interestingly enough, it is possible to determine the (rather complicated) analytical values of the other first loop points. Alternatively, this can be accomplished using fsolve to find the solutions of either of the equations

$$r(t)\,\cos(t) = r(t + k\pi)\,\cos(t + k\pi)$$

or

$$r(t)\,\sin(t) = r(t + k\pi)\,\sin(t + k\pi)$$

for selected values of k. In particular, $k = 1$ and $k = 3$ yield two of the off–axis loop points and the other two points can be determined using the symmetry of the curve.

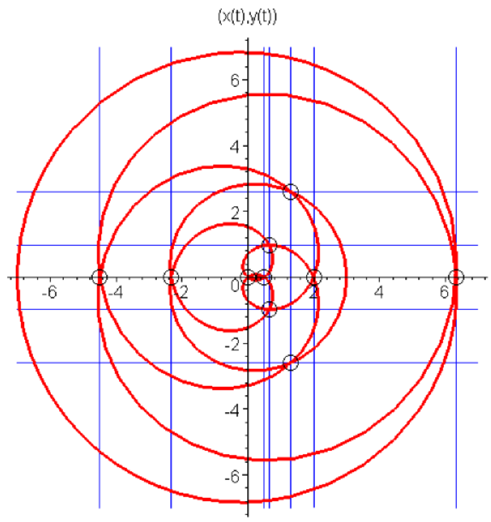

Figure 11: Parametric Curve and Loop Points for Example 5

After exploring this curve analytically, students can also determine any of the 10 loop points using the methods discussed in this paper. By plotting the curve in increments of $\frac{\pi}{6}$, it is a simple matter to determine appropriate bracketing intervals for each loop point. This is done in Item 8 in §7. Finally, an adventurous student may wish to perturb the coefficients in the parametric equations and determine the manner in which the loop points change.

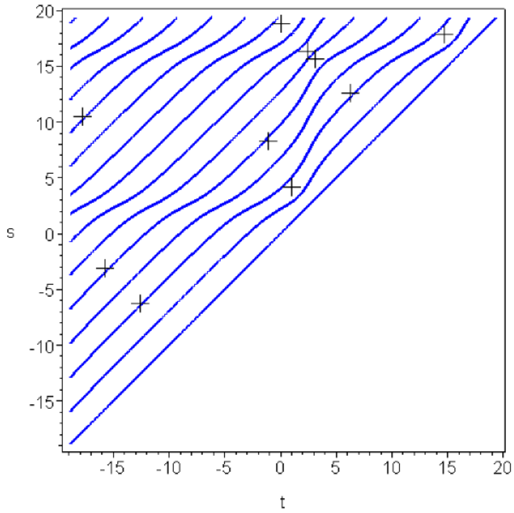

Figure 12: Singular Jacobian Curves and Loop Points for Example 5

5 Suggested Starter Problem

Before attempting to find loop points for general curves, students may find this problem interesting and instructive. Consider the two somewhat simple parametric curves

$$(x_1(t), y_1(t)) = \left(t^2 - t + 1, t^3 - 3t\right)$$

and

$$(x_2(t), y_2(t)) = \left(t^3 - 3t, t^2 + t + 1\right)$$

Each of the curves has exactly one loop point. It is straightforward to find the loop points and corresponding values of t either analytically or by using the methods described in this paper. We add a wrinkle and consider the family of parametric curves consisting of linear combinations of these two curves

$$(x_\lambda(t), y_\lambda(t)) = (\lambda\, x_1(t) + (1-\lambda)x_2(t)\, , \lambda\, y_1(t) + (1-\lambda)y_2(t))$$

where λ is a real constant. Students will find it instructive to determine the loop points for a few selected values of λ. Exploring these curves using different values of λ provides an opportunity to become familiar with the task of locating bracketing intervals and calculating the loop points using each of NLPSolve, LSSolve, and fsolve.

In addition to serving as practice in determining loop points, this starter problem easily can be turned into a challenge problem. We note the following facts about this family of curves. (The interested reader is referred to Item 11 in §7 for the complete details of the solution.) The Maple solve and simplify commands can be used to show that for a given value of λ, if we denote by z_λ either root of the equation

$$p_\lambda(z) = \left(-2 + 6\lambda + 2\lambda^2 - 12\lambda^3 + 4\lambda^4\right) - \left(1 - 2\lambda + 2\lambda^2\right)\, z + z^2 \ = 0,$$

the parameter values yielding the loop point are given by

$$s_\lambda = \frac{2\lambda^2 - 2\lambda - z_\lambda + 1)}{2\lambda - 1}, \;\; t_\lambda = \frac{z_\lambda}{2\lambda - 1}. \tag{15}$$

This is the case since removing a factor of $s - t$ from the equations $x(s) - x(t) = 0$ and $y(s) - y(t) = 0$ yields two implicit conic equations. The discriminants of the two corresponding conics are $-3(1-\lambda)^2$ and $-3\lambda^2$, respectively. The graphs are thus ellipses (unless λ is 0 or 1 in which case the graph of one of the equations is a line). Whether $(x_\lambda(t), y_\lambda(t))$ has a loop point depends on the intersection of these ellipses. When s_λ and t_λ are real, the corresponding loop point has coordinates

$$x_\lambda = \frac{2 - 15\lambda + 40\lambda^2 - 32\lambda^3 - 44\lambda^4 + 92\lambda^5 - 48\lambda^6 + 8\lambda^7}{(-1+2\lambda)^3} \tag{16}$$

and

$$y_\lambda = \frac{-3 + 21\lambda - 48\lambda^2 + 28\lambda^3 + 40\lambda^4 - 68\lambda^5 + 40\lambda^6 - 8\lambda^7}{(-1+2\lambda)^3} \tag{17}$$

s_λ and t_λ are real and unequal only if the discriminant of $p_\lambda(z)$ is positive. This is the case only if λ belongs to the union of the intervals (c_1, c_2) and (c_3, c_4) where

$$\begin{aligned}
c_1 &= \frac{5 - \sqrt{31} - 2\sqrt{(11 - \sqrt{31}}}{6} \approx -0.87 \\
c_2 &= \frac{5 + \sqrt{31} - 2\sqrt{(11 + \sqrt{31}}}{6} \approx 0.40 \\
c_3 &= \frac{5 - \sqrt{31} - 2\sqrt{(11 - \sqrt{31}}}{6} \approx 0.68 \\
c_4 &= \frac{5 + \sqrt{31} - 2\sqrt{(11 + \sqrt{31}}}{6} \approx 3.12.
\end{aligned}$$

The c_i are the zeroes of the equation

$$12c^4 - 40c^3 + 28c - 9 = 0.$$

Item 12 in §7 contains an animation of the parametric curves belonging to this family. The curves with loop points resemble the original curves. All but four of the other curves resemble s–shaped curves and do not have loop points. The other four curves, those corresponding to the ends of the above two intervals (at which $s_\lambda = t_\lambda$), stop at the corresponding singular point with $x'_\lambda = y'_\lambda = 0$. They then change direction abruptly in a cusp–like fashion. A careful study of the looped curves for values of λ near c_2^- and c_3^+ shows that the loop points are contained in small loops that approach the reversal points for c_2 and c_3. For complex s_λ and t_λ the curves resemble s–shaped curves without such cusps.

As an amusing and interesting tidbit, we can now find the loop points for the loop point locus itself. Fig. 13 depicts the locus of loop points for this family of curves. Different colors are used for the two intervals yielding loop points. The blue curve for the interval (c_1, c_2) starts just below the point $(1.4, 0)$ for $\lambda = c_1$, and the maroon curve for the interval (c_3, c_4) starts just below the point $(0, -0.4)$ for $\lambda = c_3$. The three large circles correspond to the loop points for these two segments of the loop point locus. The loop points of the loop point locus can be obtained by treating the blue and maroon curves as separate parametric curves and using fsolve to locate the loop points indicated by the three large circles. Specifically, for each of the two curves, fsolve can be used to solve the system

$$x_{\lambda_1}(s_{\lambda_1}) = x_{\lambda_1}(t_{\lambda_2}), \quad y_{\lambda_1}(s_{\lambda_1}) = y_{\lambda_1}(t_{\lambda_2})$$

for λ_1 and λ_2. We can then use either $(x_{\lambda_1}, y_{\lambda_1})$ or $(x_{\lambda_2}, y_{\lambda_2})$ to determine the loop points. The approximate loop points obtained in this fashion or by using the methods discussed in §3 are $(0.537, 0.655)$, $(-1.571, 2.454)$, and $(2.102, 1.571)$.

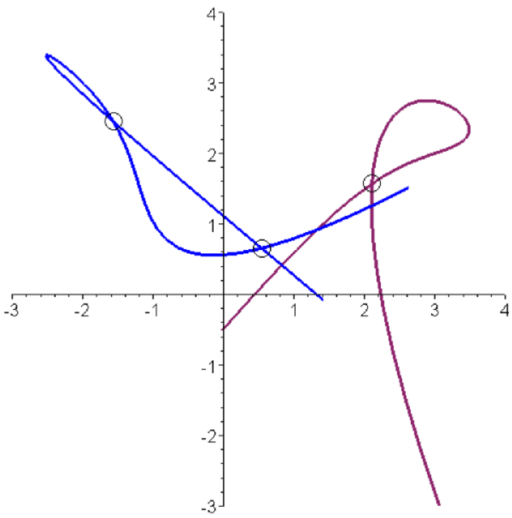

Figure 13: Starter Problem Loop Point Locus and Its Loop Points

To fully understand the behavior of the fixed point locus, especially for $c_2 < \lambda < c_3$, it is useful to consider a complex extension of the locus. If s_λ and t_λ in Eq. (15) are not real, the point determined by Eq. (16) and Eq. (17) is not a loop point but is nevertheless a unique real point on the graph of the corresponding parametric curve. Fig. 14 depicts the locus of loop points along with the extended locus of such points and indications of the orientation of the curve. The upper right green extension corresponds to the locus for $c_2 < \lambda < \frac{1}{2}$ while the lower green extension corresponds to the locus for $\frac{1}{2} < \lambda < c_3$. The upper segment extends from c_2 to $+\infty$ while the lower segment extends from $-\infty$ to c_3. The figure also includes segments of the extensions (in black) from $-\infty$ to c_1 and from c_4 to $+\infty$.

6 Summary

In this paper we considered the seemingly easy task of locating loop points for parametric curves. We showed that the problem can be posed and solved as either a constrained least squares problem or as a constrained system of nonlinear equations. We described ways to locate all loop points in a straightforward and systematic manner and the manner in which this can be done using Maple. We illustrated the solution for several representative parametric curves.

Item 13 in §7 includes Maple worksheets for several other curves of various degrees of complexity. Instructors may wish to have students explore these or similar parametric curves graphically before

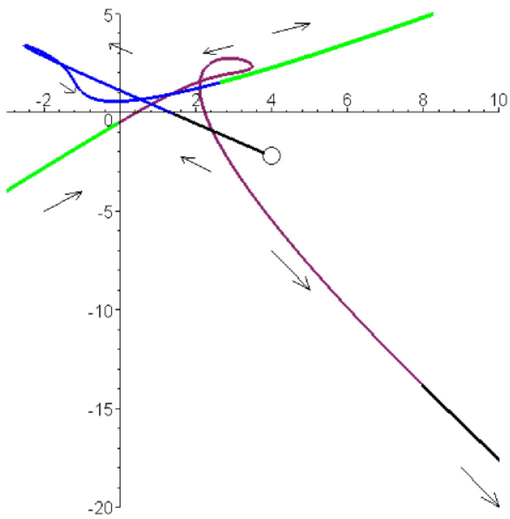

Figure 14: Complex Extension of the Starter Problem Loop Point Locus

using the worksheets. The worksheets each contain several other features of interest including plots of the sum of squares surfaces, checks for common tangent lines at loop points, and the ability to experiment with the various methods discussed in this paper. They demonstrate that the loop points and corresponding values of t for some very interesting curves can determined in a straightforward and systematic manner.

7 Supplemental Electronic Materials

Reference [3] contains links to the following supplemental electronic files.

1. Results.pdf, text file containing the computed loop points and values of t for Examples 1–5
2. Example1J.mws, Maple worksheet for Example 1
3. Example1s.mws, simplified Maple worksheet for Example 1
4. Example2J.mws, Maple worksheet for Example 2
5. Example2s.mws, simplified Maple worksheet for Example 2
6. Example3J.mws, Maple worksheet for Example 3
7. Example4J.mws, Maple worksheet for Example 4
8. Example5J.mws, Maple worksheet for Example 5
9. Example6J...14J.mws, Maple worksheets for several other curves
10. TwoCurvesIntersection.mws, Maple worksheet for the two curves in Example 7 from [4]
11. Suggested.mws, Maple worksheet for the suggested starter problem

12. SuggestedAnimation.pdf, Animation of the curves for the starter problem

13. OtherCurves.pdf, Description of parametric curves 6–14

References

[1] Maplesoft, Waterloo Maple Inc., 615 Kumpf Avenue, Waterloo, Ontario, Canada, 2010.

[2] Stewart, J., *Calculus Concepts & Contexts*, Thomson Brooks/Cole, 2005.

[3] Thompson, S., Maple Worksheets for Calculating Loop Points, 2011, `http://www.radford.edu/~thompson/LoopPoints/index.html`

[4] Yang, Wei–Chi and Lo, Min–Lin, Finding Signed Areas and Volumes Inspired by Technology, *The Electronic Journal of Mathematics and Technology*, Volume 2, Number 2, pp. 133–149, 2008.