Purdue University
Purdue e-Pubs

Department of Computer Science Technical Reports
Department of Computer Science

1994

Constraint-Based Parametric Conics for CAD

Ioannis Fudos

Christoph M. Hoffmann
Purdue University, cmh@cs.purdue.edu

Report Number:
94-065

CONSTRAINT-BASED PARAMETRIC CONICS FOR CAD

Ioannis Fudos
Christoph M. Hoffmann

CSD-TR-94-065
September 1994

Constraint-Based Parametric Conics for CAD*

Ioannis Fudos[†] Christoph M. Hoffmann

Department of Computer Science
Purdue University
West Lafayette, IN 47907-1398

Report CSD 94-065[‡]
September 1994

Abstract

Current commercial CAD systems allow constraint-based profile sketching. Algebraic constructive solvers are among the fastest algorithms for solving the resulting systems of geometric constraints. Such solvers decompose the system of nonlinear equations describing the constraints into small subsystems that correspond to simple construction steps.

The repertoire of geometric elements supported by algebraic solvers is usually restricted to points, lines and circles. Consequently, when blending two line segments, only circular arcs can be used. This fact imposes serious restrictions on the relative position of the two segments. To overcome this limitation, general conic arcs should be allowed because they can blend any two line segments. Moreover, such arcs have an additional shape parameter that can be used to satisfy one more geometric constraint.

We explain how to construct conic arcs from constraints, using a unified rational parametric representation that combines the separate cases of blending parallel and nonparallel edges. Our representation can be converted easily into a two-piece rational B-spline with positive weights, and is therefore compatible with internal representations used by most solid modeling systems. We show how to determine arcs that must have a given distance from a line, a point, or a circle, or else intersect a circle or a line at a prescribed angle. Finally, we discuss an implementation that integrates our techniques with our algebraic constraint solver.

*Work supported in part by ONR contract N00014-90-J-1599, by NSF Grant CDA 92-23502, and by NSF Grant ECD 88-03017.

†Supported by a Purdue Research Foundation Fellowship.

‡This and related reports are available via anonymous ftp to arthur.cs.purdue.edu under pub/cmh/Reports and subsidiaries.

1 Introduction

Blending two line segments in a sketch is an important operation in CAD/CAM systems and their user interfaces. The majority of CAD/CAM systems provide circular arcs for this purpose. When connecting two given line segments at their end points, four degrees of freedom are needed. Circular arcs, however, have only three degrees of freedom, and therefore can be used only when the two line segments are in special position — when the end points to be connected by the arc are equidistant from the intersection of the extended line segments. General conic arcs, on the other hand, offer five degrees of freedom. Consequently, they can be used to blend any two line segments with one additional degree of freedom available as shape parameter.

In CAD/CAM systems, the segments to be blended are often parallel or nearly so. Therefore, it is highly desirable to use representations and constructions that are capable of handling both parallel and intersecting segments uniformly. Such representations, moreover, can be expected to increase the robustness of the system.

These considerations motivate the following technical contributions our paper makes:

- We develop a uniform representation, based on the rational quadratic Bézier form, that describes a conic arc that is tangent to two segments and passes through a third point. Our representation applies whether the segments to be blended are parallel or not.

- We present a geometric construction for finding a blending arc that has been specified to be tangent to a line or at a certain distance from a line. The construction is valid for parallel and nonparallel segments.

- We give an algebraic procedure that determines a conic blending arc that is tangent to a circle, has a specified distance from a point, or intersects a given line or circle at a specified angle. Again, the computations are valid for both parallel and nonparallel segments.

- This work has been be integrated into a constructive constraint solver that supports points, lines and circles [2, 5, 6], thus increasing the design vocabulary available to CAD users. Our representation is converted easily into a two-piece rational quadratic B-spline with positive weights, and is therefore compatible with internal representations used by most solid modeling systems.

In prior work, an explicit parametric form for a conic arc that blends two segments and passes through a third point has been studied by Liming [9] and Faux [4], but the cases of parallel and nonparallel segments must be handled separately. A rational quadratic Bézier formula for the nonparallel case is presented in [3]. In [11], Piegl proposes "infinite" control points which he uses to to handle parallel end tangents [12]. In [3], Farin derives a solution for finding a conic arc that blends two given segments and is tangent to a third line. He assumes nonparallel segments. A unified representation for circular arcs using B-splines is presented in [13, 14]. In [15], a sufficient condition for the weights of a rational cubic Bézier curve is derived so that it represents a conic arc. In the same work they prove that one convex control polygon can define only one such conic arc. Finally, in [1], regular rational Bézier curves are used for representing conic arcs and other free form edges. In this work, the degree of the rational Bézier curve and the appropriate parametrization are specified on a case by case basis.

Dimensional geometric constraint solvers usually restrict the shape vocabulary to line segments and circular arcs. There seems to be little published work that addresses the incorporation of more general geometric shape primitives. Malraison [10] develops a technique for constraining a control net of a rational quadratic Bézier curve to be always an elliptical arc. No constraints may be imposed on the elliptical arc itself except at the end points. In [1], a number of constraints are allowed between an edge, which is described by a classical rational Bézier curve of arbitrary degree, and other geometries. The constraints are then translated into equations by making use of the implicit equation for the Bézier curve, and the final system of equations is solved using an iterative method. The authors mention that the solution derived by this method is sensitive to the initial positioning of geometric objects, making the problem of root selection hard to solve (see [6] for a discussion of the problem of root selection in geometric constraint solving). [7] considers the problem of conics that have C^2 contact with a plane curve, that is, tangency and curvature of the two curves agree at the contact point. A representation for conics of contact is derived using rational quadratic Bézier curves. Finally in [8], Hoffmann and Peters discuss how to construct a class of cubic Bézier curves from geometric constraints.

The remainder of this paper is structured as follows. In Section 2, we specify our representation for a conic arc that is tangent to two given segments and passes through a given point. In Section 2.4, we describe how our basic representation can be converted to a two-piece rational quadratic B-spline curve with positive weights. In Section 3, we give algebraic algorithms for constructing a blending arc which is constrained to a given line, point or circle by a distance, tangency or angle constraint. Finally, in Section 4, we discuss how our method has been integrated with our graph-constructive, variational constraint solver [2].

2 A Uniform Representation for Conics

We develop a uniform rational Bézier representation for a conic arc that blends two segments at the end points and interpolates a third point. We first review the nonparallel [3] and the parallel [11] cases separately. Then we construct a unified representation that we use in both cases.

Our representation may have negative weights, but most commercial solid modeling software restricts to positive weights. The negative weights occur in cases where the conic arc subtends an angle of more than 180°. In the implementation, therefore, we translate our representation, used internally, to a B-spline representation when interfacing to the solid modeler. The B-spline breaks such arcs into two adjacent pieces, each subtending an angle smaller than 180°. By doing so, we also benefit from a large repertoire of algorithms for handing rational B-splines with positive weights; e.g., [13]. We sketch the method at the end of this section.

It is well-known that a rational quadratic Bézier curve is a conic arc. However, when the denominator of the coordinate functions vanishes in the interval $[0, 1]$, the arc will contain points at infinity. Below, we will exclude those arcs because they are unsuitable for applications.

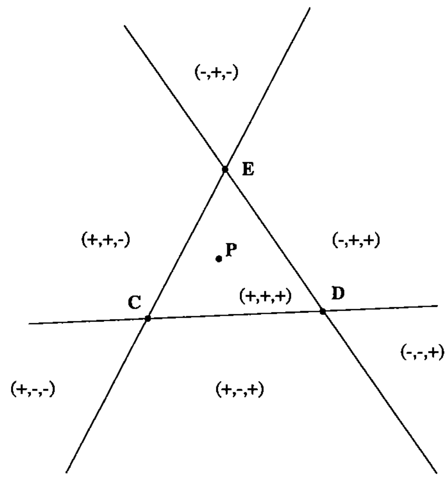
2.1 Nonparallel Tangents

A rational quadratic Bézier curve with nonparallel end tangents has the form:

$$\mathbf{c}(t) = \frac{w_0(1-t)^2\mathbf{C} + 2w_1t(1-t)\mathbf{E} + w_2t^2\mathbf{D}}{w_0(1-t)^2 + 2w_1t(1-t) + w_2t^2}, \qquad t \in [0,1]$$

Where $\mathbf{C}$, and $\mathbf{D}$ are the end points of the arc and $\mathbf{E}$ is the intersection of the end tangents. Let $\mathbf{P} = (P_x, P_y)$ be the point we wish to interpolate, and let (τ_0, τ_1, τ_2) be the barycentric coordinates of $\mathbf{P}$ with respect to the triangle $\triangle\, \mathbf{C}, \mathbf{E}, \mathbf{D}$. The lines of the triangle partition the plane into several regions. Figure 1 (left) shows the signs of the barycentric coordinates when $\mathbf{P}$ lies in each region. We write $\mathbf{P} = \mathbf{c}(t_P) = \tau_0\mathbf{C} + \tau_1\mathbf{E} + \tau_2\mathbf{D}$. By comparing coefficients (see [3]), we derive the implicit formula $\tau_1^2 w_0 w_2 = 4w_1^2\tau_0\tau_2$. When τ_0 and τ_2 are positive and $\tau_1/(2\sqrt{\tau_0\tau_2}) > -1$, an acceptable solution is obtained. In all other cases, the conic arc will pass through infinity or degenerate into a pair of line segments. The unique solution is given by

$$\begin{aligned} w_1 &= \frac{\tau_1}{2\sqrt{\tau_0\tau_2}} \\ \mathbf{c}(t) &= \frac{(1-t)^2\mathbf{C} + 2w_1t(1-t)\mathbf{E} + t^2\mathbf{D}}{(1-t)^2 + 2w_1t(1-t) + t^2}, \qquad t \in [0,1] \end{aligned} \tag{1}$$



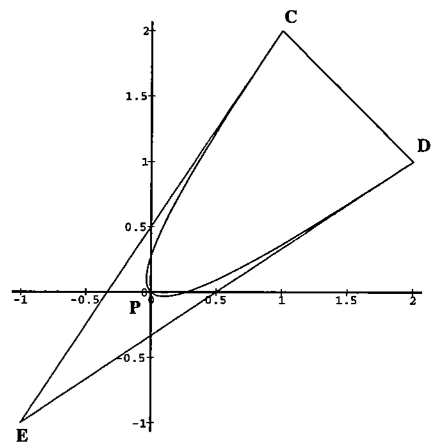


Figure 1: Left: the sign of the barycentric coordinates. Right: A conic arc blending two segments and interpolating the origin.

2.2 Parallel Tangents

Let $\mathbf{C} = (C_x, C_y)$, $\mathbf{D} = (D_x, D_y)$ be two endpoints and let $\vec{\mathbf{v}} = (v_x, v_y)$ be a tangent vector of the two parallel segments. Then the rational Bézier form of the blending conic can be written

$$\mathbf{c}(t) = \frac{w_0(1-t)^2\mathbf{C} + 2w_1 t(1-t)\vec{\mathbf{v}} + w_2 t^2 \mathbf{D}}{w_0(1-t)^2 + w_2 t^2}, \qquad t \in [0,1] \tag{2}$$

In [12], $\vec{\mathbf{v}}$ has been called a control point at infinity.

Let $d(\mathbf{S}, \vec{\mathbf{r}}, \mathbf{R})$ be the unnormalized distance of the point $\mathbf{S}$ from the line through $\mathbf{R}$ with direction $\vec{\mathbf{r}}$; that is, $d(\mathbf{S}, \vec{\mathbf{r}}, \mathbf{R}) = (S_y - R_y)r_x - (S_x - R_x)r_y$. By $Area(\mathbf{C}, \mathbf{P}, \mathbf{D})$ we denote the signed area of the triangle $\triangle\, \mathbf{C}, \mathbf{P}, \mathbf{D}$.

When the two intersecting tangent lines become parallel, their intersection $\mathbf{E}$ moves to infinity; i.e., $\mathbf{E} = \lim_{r\to\infty} r\vec{\mathbf{v}}$. By taking the barycentric coordinates in the limit, we get

$$\mathbf{P} = \mathbf{c}(t_P) = T_0\mathbf{C} + T_1\vec{\mathbf{v}} + T_2\mathbf{D}$$

where

$$T_0 = \frac{d(\mathbf{P}, \vec{\mathbf{v}}, \mathbf{D})}{d(\mathbf{C}, \vec{\mathbf{v}}, \mathbf{D})} \qquad T_2 = \frac{d(\mathbf{P}, \vec{\mathbf{v}}, \mathbf{C})}{d(\mathbf{D}, \vec{\mathbf{v}}, \mathbf{C})} \qquad T_1 = \frac{2\, Area(\mathbf{C}, \mathbf{P}, \mathbf{D})}{d(\mathbf{C}, \vec{\mathbf{v}}, \mathbf{D})} \tag{3}$$

By comparing the coefficients we obtain the implicit equation:

$$T_1^2 w_0 w_2 = 4w_1^2 T_0 T_2$$

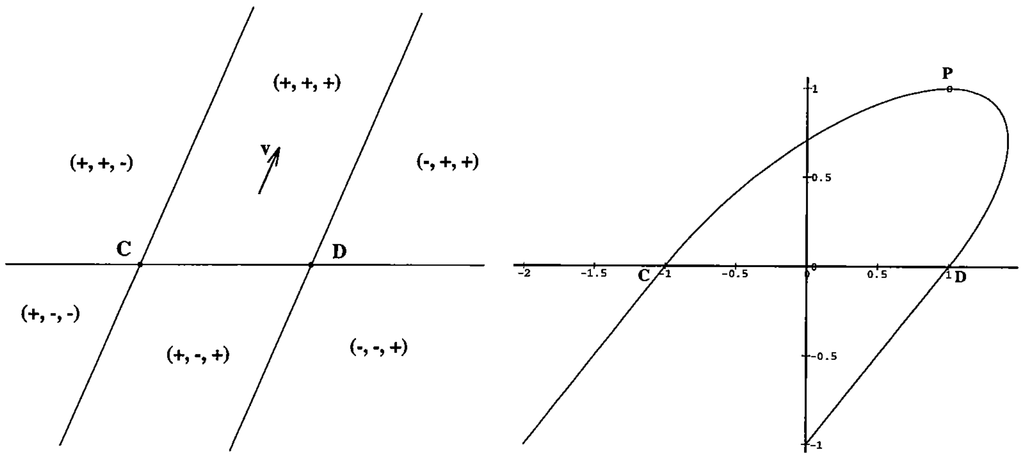


Figure 2: Left: the sign of the coefficients for the case of parallel tangents. Right: A conic arc blending two parallel segments and passing through $(1, 1)$.

The signs of the coordinates (T_0, T_1, T_2) of $\mathbf{P}$ are shown in Figure 2 (left). An acceptable solution will require that $T_0 > 0$ and $T_2 > 0$. Here, $\mathbf{P}$ is in the strip defined by the two parallel lines. The unique solution is the elliptic arc

$$\begin{aligned} w_1 &= \frac{T_1}{2\sqrt{T_0 T_2}} \\ \mathbf{c}(t) &= \frac{(1-t)^2\mathbf{C} + w_1 t(1-t)\vec{\mathbf{v}} + t^2\mathbf{D}}{t^2 + (1-t)^2}, \qquad t \,\epsilon\, [0,1] \end{aligned} \tag{4}$$

Note that the solution is not affected by changing the sign of $\vec{\mathbf{v}}$ or the order of the points $\mathbf{C}$ and $\mathbf{D}$. An example is shown in Figure 2 (right) for $\mathbf{C} = (-1, 0)$, $\mathbf{D} = (1, 0)$, $\vec{\mathbf{v}} = (1, 1)$, and $\mathbf{P} = (1, 1)$.

2.3 A Unified Representation

Let $\vec{\mathbf{v}}$, $\vec{\mathbf{u}}$ be the two tangent vectors of the conic arc at the end points. Let $\mathbf{C}$ and $\mathbf{D}$ be the two endpoints of the conic arc, and let $\mathbf{P}$ be the third point we wish to interpolate. Furthermore, let $\mathbf{U} = (v_x M_2 - u_x M_1, v_y M_2 - u_y M_1)$, where $M_1 = C_x v_y - C_y v_x$ and $M_2 = D_x u_y - D_y u_x$. We will prove that the solution $\mathbf{c}(t)$, if it exists, is given by:

$$\mathbf{c}(t) = \frac{(1-t)^2\mathbf{C} + 2t(1-t)\mathbf{W} + t^2\mathbf{D}}{(1-t)^2 + 2wt(1-t) + t^2}, \qquad t \in [0,1] \tag{5}$$

where

$$\begin{aligned}
\mathbf{W} &= \frac{Area(\mathbf{C},\mathbf{P},\mathbf{D})\,\mathbf{U}}{\sqrt{d(\mathbf{D},\vec{\mathbf{v}},\mathbf{C})\,d(\mathbf{P},\vec{\mathbf{v}},\mathbf{C})\,d(\mathbf{C},\vec{\mathbf{u}},\mathbf{D})\,d(\mathbf{P},\vec{\mathbf{u}},\mathbf{D})}} \\
w &= \frac{Area(\mathbf{C},\mathbf{P},\mathbf{D})(\vec{\mathbf{v}}\times\vec{\mathbf{u}})}{\sqrt{d(\mathbf{D},\vec{\mathbf{v}},\mathbf{C})\,d(\mathbf{P},\vec{\mathbf{v}},\mathbf{C})\,d(\mathbf{C},\vec{\mathbf{u}},\mathbf{D})\,d(\mathbf{P},\vec{\mathbf{u}},\mathbf{D})}}
\end{aligned} \tag{6}$$

and that the formula is valid whether the vectors $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$ are linearly dependent or not. We assume that $\mathbf{P}$ is not on any of the three lines, through $\mathbf{C}$ with direction $\vec{\mathbf{u}}$, through $\mathbf{D}$ with direction $\vec{\mathbf{v}}$, and through $\mathbf{C}$ and $\mathbf{D}$. Moreover, the three lines are assumed distinct, so that the denominators of $\mathbf{W}$ and w do not vanish.

Scale and Sign Invariance

We observe the following:

1. $d(\cdot,\lambda\vec{\mathbf{v}},\cdot) = \lambda d(\cdot,\vec{\mathbf{v}},\cdot)$.
2. Let $\mathbf{U} = U(\vec{\mathbf{u}},\vec{\mathbf{v}})$, where U is defined by the expression for $\mathbf{U}$ above. Then $U(\lambda\vec{\mathbf{u}},\vec{\mathbf{v}}) = \lambda U(\vec{\mathbf{u}},\vec{\mathbf{v}})$. and $U(\vec{\mathbf{u}},\lambda\vec{\mathbf{v}}) = \lambda U(\vec{\mathbf{u}},\vec{\mathbf{v}})$.
3. $(\lambda\vec{\mathbf{v}}\times\vec{\mathbf{u}}) = \lambda(\vec{\mathbf{v}}\times\vec{\mathbf{u}})$ and $(\vec{\mathbf{v}}\times\lambda\vec{\mathbf{u}}) = \lambda(\vec{\mathbf{v}}\times\vec{\mathbf{u}})$.

Consequently, if we replace $\vec{\mathbf{u}}$ with $\lambda\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$ with $\mu\vec{\mathbf{v}}$ in (5), the same arc is obtained. When the sign of the square root in the denominator is chosen equal to the sign of $d(\mathbf{D},\vec{\mathbf{v}},\mathbf{C})\,d(\mathbf{C},\vec{\mathbf{u}},\mathbf{D})$, the representation becomes independent of sign and magnitude of the end tangent vectors.

Correctness for Nonparallel Tangents

Using the well-known interpretation of barycentric coordinates as area ratios in the subdivision of $\triangle\,\mathbf{C},\mathbf{D},\mathbf{E}$ by $\mathbf{P}$, we can rewrite w_1 from (1) as

$$w_1 = \frac{Area(\mathbf{C},\mathbf{P},\mathbf{D})}{2\sqrt{Area(\mathbf{P},\mathbf{E},\mathbf{D})Area(\mathbf{C},\mathbf{E},\mathbf{P})}}$$

But

$$\begin{aligned}
Area(\mathbf{P},\mathbf{E},\mathbf{D}) &= \frac{1}{2}\|\mathbf{ED}\|\,d(\mathbf{P},\overline{(\mathbf{E},\mathbf{D})},\mathbf{E}) = \frac{d(\mathbf{P},\vec{\mathbf{u}},\mathbf{D})d(\mathbf{D},\vec{\mathbf{v}},\mathbf{C})}{2(\vec{\mathbf{v}}\times\vec{\mathbf{u}})} \\
Area(\mathbf{C},\mathbf{E},\mathbf{P}) &= \frac{1}{2}\|\mathbf{EC}\|\,d(\mathbf{P},\overline{(\mathbf{E},\mathbf{C})},\mathbf{C}) = \frac{d(\mathbf{P},\vec{\mathbf{v}},\mathbf{C})d(\mathbf{C},\vec{\mathbf{u}},\mathbf{D})}{2(\vec{\mathbf{v}}\times\vec{\mathbf{u}})}
\end{aligned} \tag{7}$$

so that w_1 in (1) is equal to w in (6). Now the intersection $\mathbf{E}$ must satisfy the system:

$$\begin{aligned}
(E_y - D_y)u_x - (E_x - D_x)u_y &= 0 \\
(E_y - C_y)v_x - (E_x - C_x)v_y &= 0
\end{aligned} \tag{8}$$

By algebra, $\mathbf{E} = \dfrac{\mathbf{U}}{(\vec{\mathbf{v}} \times \vec{\mathbf{u}})}$. Therefore, (5) is equivalent to (1) for the case of nonparallel tangents.

Correctness for Parallel Tangents

We set $\vec{\mathbf{u}} = \vec{\mathbf{v}}$ because of scale and sign invariance. Then $w = 0$ and $\mathbf{U} = (M_2 - M_1)\vec{\mathbf{v}} = d(\mathbf{C}, \vec{\mathbf{v}}, \mathbf{D})\vec{\mathbf{v}}$. From (3), and (6) we obtain

$$\mathbf{W} = \frac{Area(\mathbf{C}, \mathbf{P}, \mathbf{D})}{\sqrt{d(\mathbf{D}, \vec{\mathbf{v}}, \mathbf{C})d(\mathbf{P}, \vec{\mathbf{v}}, \mathbf{C})d(\mathbf{C}, \vec{\mathbf{v}}, \mathbf{D})d(\mathbf{D}, \vec{\mathbf{v}}, \mathbf{C})}} d(\mathbf{C}, \vec{\mathbf{v}}, \mathbf{D})\vec{\mathbf{v}} = \frac{T_1}{2\sqrt{T_0 T_2}}\vec{\mathbf{v}}$$

from which we derive that $\mathbf{W} = w_1\vec{\mathbf{v}}$. Hence the formula is valid for parallel tangents.

Acceptable Solutions

Observing the formal correspondences of (1) and (2) with (5), an acceptable solution exists if, and only if,

$$d(\mathbf{D}, \vec{\mathbf{v}}, \mathbf{C})d(\mathbf{P}, \vec{\mathbf{v}}, \mathbf{C}) > 0, \qquad d(\mathbf{C}, \vec{\mathbf{u}}, \mathbf{D})d(\mathbf{P}, \vec{\mathbf{u}}, \mathbf{D}) > 0, \qquad w > -1$$

For instance, we obtain the arc of Figure 1 (right) if we set $\mathbf{C} = (1, 2)$, $\mathbf{D} = (2, 1)$, $\vec{\mathbf{v}} = (-2, -3)$, $\vec{\mathbf{u}} = (-3, -2)$ and $\mathbf{P} = (0, 0)$, in (5). Similarly we obtain the arc of Figure 2 (right) if we set $\mathbf{C} = (-1, 0)$, $\mathbf{D} = (1, 0)$, $\vec{\mathbf{v}} = (1, 1)$, $\vec{\mathbf{u}} = (1, 1)$ and $\mathbf{P} = (1, 1)$.

2.4 Converting to a Rational B-spline

No class of rational Bézier curves of any fixed degree is capable of representing all acceptable solutions defined by (5) with positive weights alone. Since, CAD systems often disallow negative weights, we use a two-piece rational quadratic B-spline with positive weights. By choosing the segmentation appropriately, we can represent all acceptable solutions with positive weights.

We subdivide our conic arc defined by (5) at $t = 1/2$ such that the two resulting conic arcs are the two pieces of a C^1-continuous rational B-spline. C^2-continuity of the quadratic B-spline is then implied. Since the tangent at $t = 1/2$ is parallel to $\mathbf{CD}$, it must intersect both end tangents. Let $\mathbf{A}$ and $\mathbf{B}$ be the intersection points (see Figure 3).

It is well-known that a rational parametric curve can be considered to be the projection of a parametric space curve to a plane; e.g., [3]. Specifically, we take the Bézier space curve defined by $[\mathbf{C}, 1]$, $[\mathbf{W}, w]$ and $[\mathbf{D}, 1]$, where w and $\mathbf{W}$ are as in (5). Its projection to the plane $w = 1$ is the original rational curve. Using this formulation a routine computation shows that the following rational

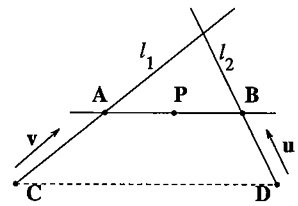


Figure 3: Subdividing a conic arc at $t = 1/2$. The tangent at that point is parallel to **CD**.

B-spline represents the original conic arc exactly:
rational quadratic B-spline (see e.g. chapter 7 of [3]) with:

Control Points:	$[\mathbf{C}, \frac{\mathbf{C}+\mathbf{W}}{1+w}, \frac{\mathbf{D}+\mathbf{W}}{1+w}, \mathbf{D}]$
Knot Vector:	$[0, 0, \frac{1}{2}, 1, 1]$
Weights:	$[1, \frac{1+w}{2}, \frac{1+w}{2}, 1]$

3 Constructions

We present geometric and algebraic methods for constructing a blending arc that satisfies an additional geometric constraint with another geometric object. All computations are valid independently of the relative position of the geometric objects involved.

3.1 Tangency to or Distance from a Line

In this section we discuss a geometric construction for constructing a conic arc that blends two segments and is tangent to a line. The case of requiring that the arc have a given distance from a line directly reduces to this case. The construction is in terms of the uniform representation explained before.

Let Λ be the line which is to be tangent to the conic arc. We will determine the point **P** where the conic touches the line Λ, thereby reducing the problem to the interpolation problem solved in the previous section. First, we will describe a geometric construction that derives **P** and then we shall prove its soundness.

The construction for intersecting tangents is from [3], and is shown in Figure 4 (left). For parallel tangents, the construction is illustrated in Figure 4 (right). Let l_1 be the line that passes through **C** and is parallel to $\vec{\mathbf{v}}$, and l_2 be the line that passes through **D** and is parallel to $\vec{\mathbf{u}}$. We assume that Λ intersects both l_1 and l_2 at two points **A**, **B** other than **C**, **D**, and the intersection, if any, of l_1 and

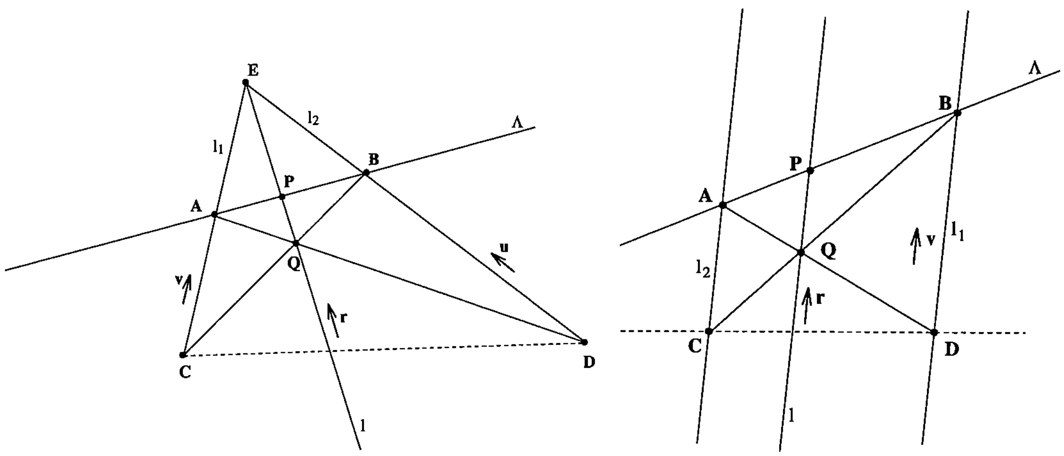


Figure 4: Left: finding the point of tangency **P** between the conic arc and the line in the case of nonparallel tangents. Right: finding the point of tangency **P** between the conic arc and the line in the case of parallel tangents.

l_2. We then find the intersection point **Q** of **AD** and **BC**. Then the intersection **P** of line Λ with the line l through **Q** with direction $\vec{\mathbf{r}} = \mathbf{U} - (\vec{\mathbf{v}} \times \vec{\mathbf{u}})\mathbf{Q}$ will be shown to be the point of tangency. Here **U** is as before in (6).

(i) *Nonparallel tangents:* The correctness of the construction uses Pascal's theorem; [3, 9]. Since $\vec{\mathbf{QE}} = \dfrac{1}{(\vec{\mathbf{v}} \times \vec{\mathbf{u}})}\vec{\mathbf{r}}$, the geometric construction is consistent with the definition of $\vec{\mathbf{r}}$.

(ii) *Parallel tangents:* The correctness of the geometric construction is clear from the projective interpretation of Pascal's theorem; $(\vec{\mathbf{v}} \times \vec{\mathbf{u}}) = 0$, so the geometric construction is consistent with the definition of $\vec{\mathbf{r}}$.

The construction of **Q** assumes that Λ intersects both l_1 and l_2. However, it is possible that Λ is parallel to l_1 or l_2, yet we are still able to find a blending arc that is tangent to Λ. We include this case by determining **Q** from a computation similar to the one used to compute **P**: We intersect the line through **C** with direction $\vec{\mathbf{r}}_1 = \mathbf{U}_1 - (\vec{\mathbf{n}} \cdot \vec{\mathbf{v}})\,\mathbf{C}$, and the line though **D** with direction $\vec{\mathbf{r}}_2 = \mathbf{U}_2 - (\vec{\mathbf{n}} \cdot \vec{\mathbf{u}})\,\mathbf{D}$. Here $\vec{\mathbf{n}}$ is the normal vector of Λ and $\mathbf{U}_1$ and $\mathbf{U}_2$ are defined as in Section 3.2.

For this problem, there is only one solution. Figure 5 shows two examples, one with intersecting, the other with parallel tangents.

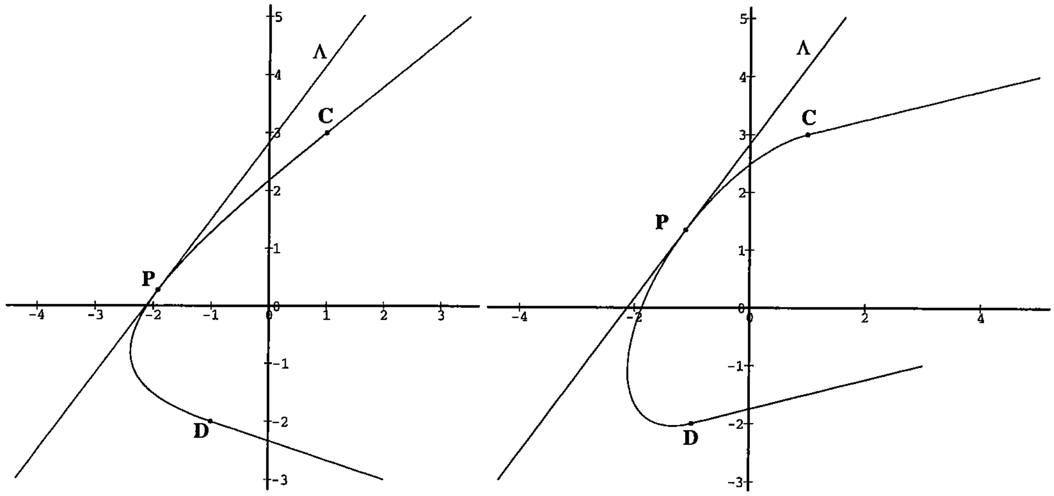


Figure 5: Blending two segments by a conic arc that is tangent to a given line

3.2 Tangency to a Circle or Distance from a Point

We seek blending arcs that are tangent to a given circle. The case is equivalent to requiring that the arc have nonzero distance from a given point. To solve this problem, we will determine the point **P** at which the arc must touch the circle.

Without loss of generality, we assume that the circle **R** is centered at the origin and has radius $d > 0$. Let **P** be the point of tangency with the conic arc $\mathbf{c}(t)$, and let Λ be the common tangent through **P**. Let $\vec{\mathbf{n}} = (n_x, n_y)$ be the unit normal of Λ. If we determine $\vec{\mathbf{n}}$, then we have reduced the problem to interpolating the point $\mathbf{P} = (d\,n_x, d\,n_y)$. Our strategy is to use an approach similar to that of Section 3.1.

Let $\mathbf{A} = (A_x, A_y)$ be the intersection point of Λ and l_1, $\mathbf{B} = (B_x, B_y)$ be the intersection of Λ and l_2, and $\mathbf{Q} = (Q_x, Q_y)$ be the intersection point of **AD** and **BC**. If Λ is parallel to l_1, then **A** is at infinity. This means that **Q** can be found by intersecting **BC** and the line that passes through **D** and is parallel to Λ. Similarly, if Λ is parallel to l_2, we intersect **AD** and the line that passes through **D** and is parallel to Λ. All cases can be expressed uniformly by the following system of equations:

$$\begin{aligned} \vec{\mathbf{r}}_1 \times \vec{\mathbf{CQ}} &= 0 \\ \vec{\mathbf{r}}_2 \times \vec{\mathbf{DQ}} &= 0 \end{aligned} \tag{9}$$

where $\vec{\mathbf{r}}_1 = \mathbf{U}_1 - (\vec{\mathbf{n}} \cdot \vec{\mathbf{v}})\,\mathbf{C}$ and $\vec{\mathbf{r}}_2 = \mathbf{U}_2 - (\vec{\mathbf{n}} \cdot \vec{\mathbf{u}})\,\mathbf{D}$. Finally $\mathbf{U}_1$ and $\mathbf{U}_2$ are defined as follows, $U_1 = (n_y L_1 - u_x d, -n_x L_1 - u_y d)$, $U_2 = (n_y L_2 - u_x d, -n_x L_2 - u_y d)$, where $L_1 = v_y C_x - v_x C_y$, $L_2 = u_y D_x - u_x D_y$.

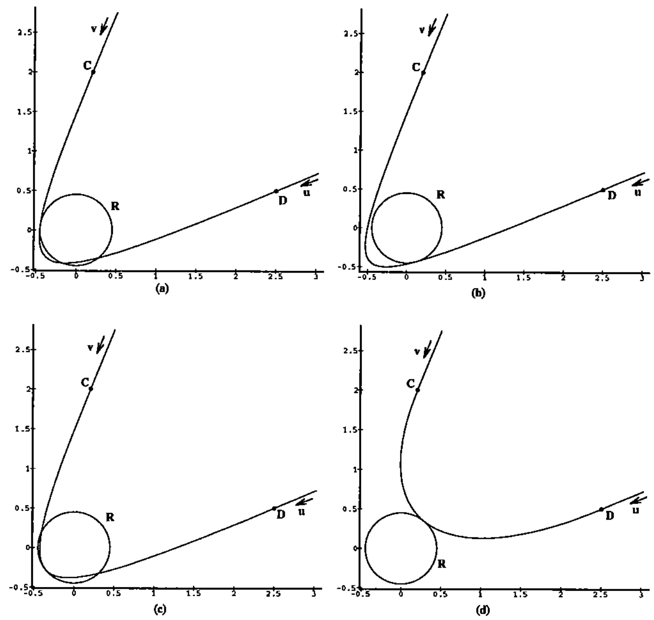


Figure 6: Four solutions for a conic arc that blends two segments and is tangent to a circle

Since Λ is tangent to the curve $\mathbf{c}$ at $\mathbf{P}$, $\mathbf{P}$ is on the line l that passes through $\mathbf{Q}$ in the direction $\vec{\mathbf{r}} = (r_x, r_y) = \mathbf{U} - (\vec{\mathbf{v}} \times \vec{\mathbf{u}})\mathbf{Q}$. Therefore

$$-r_y(d\,n_x - Q_x) + r_x(d\,n_y - Q_y) \quad = \quad 0 \tag{10}$$

By solving (9) we determine Q_x, Q_y in terms of n_x, n_y. Substitution of Q_x, Q_y in (10) yields

$$(\mathbf{U}\vec{\mathbf{n}} - (\vec{\mathbf{v}} \times \vec{\mathbf{u}})d)(a_0 + a_1 n_x + a_2 n_y + a_3 n_x^2 + a_4 n_y^2 + a_5 n_x n_y) = 0 \tag{11}$$

where the a_i are constant expressions.

The first term of (11) corresponds to the case where the line Λ passes through the intersection point E of l_1 and l_2 (this point is at infinity for parallel tangents). This case does not yield a solution, so it suffices to solve

$$\begin{aligned} a_0 + a_1 n_x + a_2 n_y + a_3 n_x^2 + a_4 n_y^2 + a_5 n_x n_y &= 0 \\ n_x^2 + n_y^2 &= 1 \end{aligned} \tag{12}$$

A routine Gröbner basis computation produces an equivalent system in which n_y is determined from a univariate polynomial and n_x from a linear one:

$$\begin{aligned} c_4 n_y^4 + c_3 n_y^3 + c_2 n_y^2 + c_1 n_y + c_0 &= 0 \\ e_0 n_x + e_1 &= 0 \end{aligned} \tag{13}$$

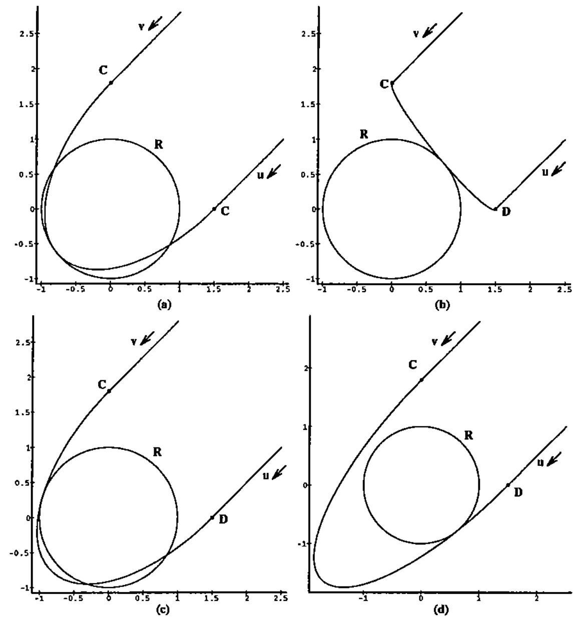


Figure 7: Four solutions for a conic arc that blends two parallel segments and is tangent to a circle

Next, we determine $\mathbf{P} = (d\,n_x, d\,n_y)$ and use (5) to derive the arc $\mathbf{c}(t)$.

As indicated by the system (13), up to four distinct solutions are possible. Figure 6 shows an example with $\mathbf{C} = (1/5, 2)$, $\mathbf{D} = (5/2, 1/2)$, $\vec{\mathbf{v}} = (6/5, 3)$, $\vec{\mathbf{u}} = (7/2, 3/2)$, and $d = 0.45$. Solutions (a), (b) and (c) are hyperbolic arcs, while solution (d) is an elliptic arc. The tangents are not parallel. Figure 7 shows an example with $\mathbf{C} = (0, 1.8)$, $\mathbf{D} = (3/2, 0)$, $\vec{\mathbf{v}} = (-1, -1)$, $\vec{\mathbf{v}} = (-1, -1)$, and $d = 1$. In this case the segments we blend are parallel, so all solutions are elliptical arcs.

3.3 Angle with a Line

Let Λ_0 be a line and α be the specified signed angle which we impose between Λ_0 and the conic arc $\mathbf{c}$. Let $\mathbf{P}$ be the intersection of the conic and the line, and let Λ be the tangent to the conic arc at $\mathbf{P} = (P_x, P_y)$. From the unit normal $\vec{\mathbf{n}}_0 = (n_{0x}, n_{0y})$ of Λ_0 and α we determine the normal $\vec{\mathbf{n}}$ of Λ:

$$\vec{\mathbf{n}} = (n_{0x}\cos\alpha - n_{0y}\sin\alpha, n_{0y}\cos\alpha + n_{0x}\sin\alpha)$$

See [5, 6] for the definition of a signed angle between oriented lines. The sign of Λ depends only on Λ_0 and α, and is not related to the sign of the derivative of $\mathbf{c}$. To simplify the calculations, we do a rigid motion so that $\mathbf{C} = (0, 0)$ and $\vec{\mathbf{n}} = (0, 1)$. That is, the tangent through $\mathbf{P}$ is parallel to the x-axis.

Let r be the signed distance of the origin from Λ_0, and let d be the signed distance of the origin from Λ. We will compute d from a necessary condition, and then apply the inverse motion to find the actual position of $\mathbf{P}$.

We express $\mathbf{Q}$ as a function of d, as in Section 3.2. Then we compute the coordinates of $\mathbf{P}$ as a function of d, by intersecting Λ with the line l that is through $\mathbf{Q}$ in the direction $\vec{\mathbf{r}} = (r_x, r_y) = \mathbf{U} - (\vec{\mathbf{v}} \times \vec{\mathbf{u}})\mathbf{Q}$

$$\begin{aligned} -r_y(P_x - Q_x) + r_x(P_y - Q_y) &= 0 \\ P_y &= d \end{aligned} \tag{14}$$

Since $\mathbf{P}$ is on Λ_0 we obtain:

$$P_x n_{0x} + P_y n_{0y} = r \tag{15}$$

Substituting P_x and P_y from (14) into (15), we obtain for d a quadratic equation. The equation can have up to two real solutions, so we can have two distinct conic arcs that form the specified angle with the given line. In Figure 8 (a),(b) we see two solutions for a conic arc that blends two intersecting segments and forms a 45° angle with a line Λ_0, and in Figure 8 (c),(d) we see the two solutions for a conic arc that blends two parallel segments and forms a 45° angle with a line Λ_0.

3.4 Angle with a Circle

In this section, we extend the method of Section 3.2 to computing a conic arc that blends two segments and forms a specified angle with a given circle $\mathbf{R}$ which we consider (without loss of generality) to be centered at the origin and have radius d.

Let $\mathbf{P}$ be an intersection point of the circle and the conic arc, such that the tangent Λ' to the circle at $\mathbf{P}$ and the tangent Λ to the conic at the same point form a signed angle α. Also let $\vec{\mathbf{n}} = (n_x, n_y)$ be the unit normal vector of Λ, and $\vec{\mathbf{n}}' = (n'_x, n'_y)$ be the unit normal vector of Λ'. In this setting $\mathbf{P} = (d\,n'_x, d\,n'_y)$.

As in Section 3.2 we derive an expression for $\mathbf{Q}$ involving n_x, n_y. Then we substitute n_x and n_y from:

$$\vec{\mathbf{n}} = (n'_x \cos\alpha - n'_y \sin\alpha, n'_y \cos\alpha + n'_x \sin\alpha)$$

Since $\mathbf{P} = (d\,n'_x, d\,n'_y)$, (10) becomes:

$$-r_y(d\,n'_x - Q_x) + r_x(d\,n'_y - Q_y) = 0 \tag{16}$$

Substitution of Q_x and Q_y in (16) and elimination of the factors that do not yield a solution gives:

$$\begin{aligned} b_0 + b_1 n'_x + b_2 n'_y + b_3 {n'_x}^2 + b_4 {n'_y}^2 + b_5 n'_x n'_y &= 0 \\ {n'_x}^2 + {n'_y}^2 &= 1 \end{aligned} \tag{17}$$

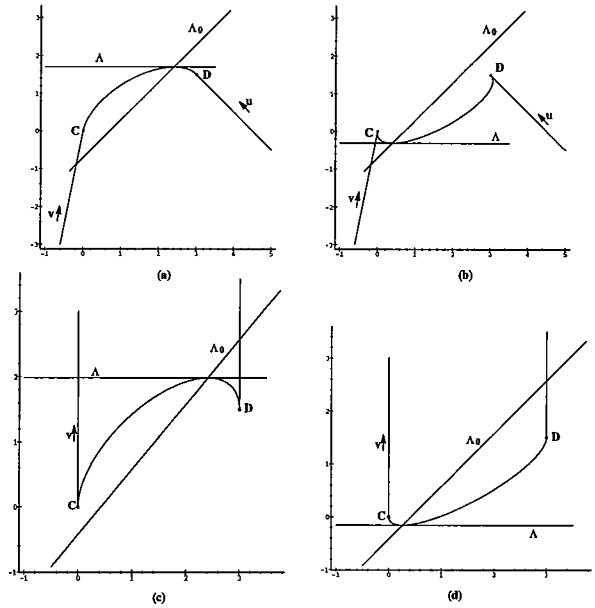


Figure 8: (a),(b): Two solutions for a conic arc that blends two segments and forms a 45° angle with a line. (c),(d): Two solutions for a conic arc that blends two parallel segments and forms a 45° angle with a line.

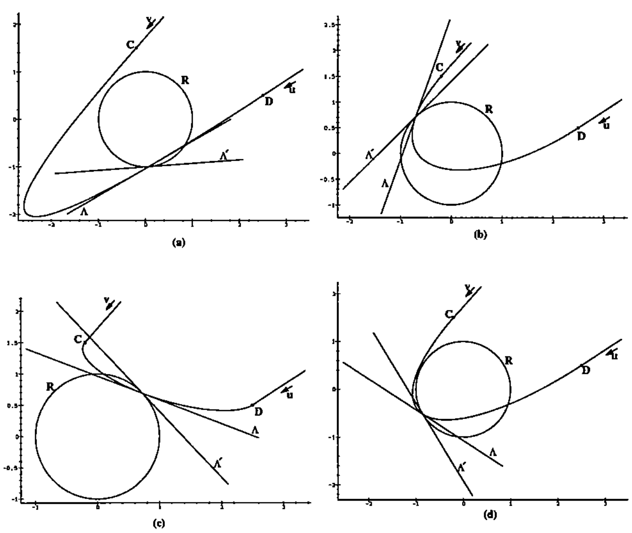


Figure 9: Four solutions for a conic arc that blends two segments and forms a $\pi/7$ angle with a circle.

where b_i are constants. We then proceed by solving (17) as in Section 3.2.

In Figure 9 we see the four solutions for a conic arc that blends two segments and intersects a circle centered at the origin with radius $d = 0.45$ under an angle $\alpha = \pi/7$. Solution (a) is a hyperbolic arcs, while solutions (b)-(d) are elliptical arcs.

4 Integration into a Geometric Constraint Solver

We have incorporated conic arcs into our geometric constraint solver [2]. The constraint solving algorithm algorithm works in two phases:

(i) A constraint graph is analyzed by a reduction process that produces a sequence in which geometric elements must be constructed.

(ii) The actual construction of the geometric elements is carried out, in the order determined by Phase 1, by solving certain standard sets of algebraic equations.

In general, a conic arc can be determined from five constraints, but we require that four of them make the construction a Hermite problem. That is, there must be two end points and two end tangents to the conic arc.

Phase 2 of the solver has been extended to provide for the constructions described in Section 3. After the arc has been determined, geometric constraints involving the conic may be used to construct other points and lines. For example, we can constrain a line to go through a fixed point and be tangent to the conic.

The combination of these requirements entails special rules for analyzing the constraint problem in Phase 1 of our constraint solving algorithm, and we now explain them. Note that constraints between two conic arcs are not permitted. The constraint graph initially has vertices corresponding to all geometric elements, including the conic arc itself. Phase 1 proceeds as in [2] except that conic arc nodes and the constraints on them are ignored. Whenever a cluster (i.e. a rigid set of geometries which is formally defined in [5]) is formed that contains geometric elements with five constraints to a conic arc $\mathbf{c}(t)$, the construction of $\mathbf{c}(t)$ is attempted. For Phase 1 of our solver this means simply adding the node corresponding to $\mathbf{c}(t)$ to the cluster, and considering in the subsequent processing all constraint edges incident to $\mathbf{c}(t)$.

Adding the node for $\mathbf{c}(t)$ is restricted to correspond to a Hermite problem. We require that the user designate in the sketch input which points are to be end points and end tangents. Four of the five constraints must then be the incidence and tangency conditions thereby implied.

It is possible to avoid having to designate the end conditions explicitly, but this is not necessarily desirable. For an example, consider Figure 10 where the user designates tangency conditions at A, B, D and to the line HF. The

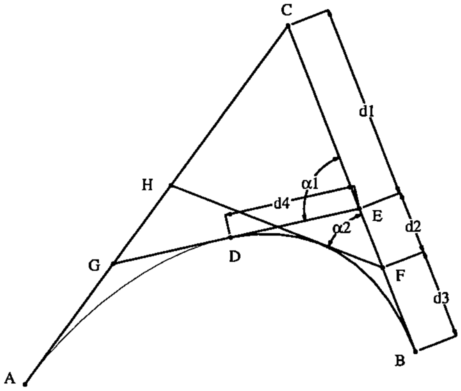


Figure 10: Arc extension implied

solver would construct first the line CB and the incident points E and F. The lines FH and EG can be constructed next, although the positions of H and G cannot yet be determined. Then, point D can be constructed, whereupon the conic arc between D and B can be found using the computations explained before. Now the line CA can be constructed, as tangent to the conic from C, thereby determining G and H, by intersection, and A by tangency. This requires extending the conic arc to A as shown. However, if the arc obtained before is hyperbolic, it is entirely possible that the tangent to A lies on a different branch of the hyperbola. Since the conic arc $\mathbf{c}(t)$ from D to B had to be extended, testing for this undesirable situation is more complicated than verifying that the parameter value corresponding to A is in the interval $[0, 1]$.

References

[1] Paula L. Beaty, Patrick A. Fitzhorn, and Gary J. Herron. Extensions in variational geometry that generate and modify object edges composed of rational Bézier curves. *Computer Aided Design*, 26(2):98–107, 1994.

[2] W. Bouma, Ioannis Fudos, Christoph Hoffmann, Jiazhen Cai, and Robert Paige. A Geometric Constraint Solver. *Computer Aided Design*, To appear, 1994. Also available as CSD-TR-93-054, Purdue University, 1993.

[3] Gerald E. Farin. *Curves and surfaces for computer aided geometric design : a practical guide.* Academic Press, 1993. ISBN: 0122490525.

[4] I. D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis Horwood, Chichester, 1979.

[5] I. Fudos and C. M. Hoffmann. Correctness Proof of a Geometric Constraint solver. *International Journal of Computational Geometry & Applications*, May 1995. To appear.

[6] Ioannis Fudos. Editable Representations for 2D Geometric Design. Master's thesis, Dept of Computer Sciences, Purdue University, December 1993. Available through Mosaic, http://www.cs.purdue.edu/people/fudos.

[7] G. Geise and Th. Nestler. Bézier representation of conics of contact in the projective plane. *Computer Aided Geometric Design*, 11:233–245, 1994.

[8] C. M. Hoffmann and J. Peters. Geometric Constraints for CAGD. Technical Report CSD-TR-94-068, Purdue University, CS Dept, Purdue University, West Lafayette, IN 47907, 1994.

[9] Roy A. Liming. *Mathematics for computer graphics.* Fallbrook, CA : Aero Publishers, 1979. ISBN: 0816867518.

[10] Pierre J. Malraison. Constraining a Quadratic Rational Bezier to be an Elliptical Arc. Presented in 3rd SIAM Conf. on Geom. Design., 1994.

[11] L. Piegl. On the use of infinite control points in CAGD. *Computer Aided Geometric Design*, 4:155–166, 1987.

[12] L. Piegl. Interactive Data Interpolation by Rational Bezier Curves. *IEEE Computer Graphics and Applications*, pages 45–58, April 1988.

[13] W. Tiler. Rational B-splines for Curve and Surface Representation. *IEEE Computer Graphics & Applications*, 3:61–69, 1983.

[14] W. Tiler and E. G. Hanson. Offsets of Two-Dimensional Profiles. *IEEE Computer Graphics & Applications*, 4:36–46, 1984.

[15] Guo-Jin Wang and Guo-Zhao Wang. The rational cubic Bézier representation of conics. *Computers in Industry*, 16(3):283–288, 1991.