


# Drawing Planar Graphs and 1-Planar Graphs Using Cubic Bézier Curves with Bounded Curvature

David Eppstein ✉ 🏠

University of California, Irvine, CA, USA

Michael T. Goodrich ✉ 🏠 

University of California, Irvine, CA, USA

Abraham M. Illickan ✉ 🏠 

University of California, Irvine, CA, USA

---

## Abstract

We study algorithms for drawing planar graphs and 1-planar graphs using cubic Bézier curves with bounded curvature. We show that any  $n$ -vertex 1-planar graph has a 1-planar RAC drawing using a single cubic Bézier curve per edge, and this drawing can be computed in  $O(n)$  time given a combinatorial 1-planar drawing. We also show that any  $n$ -vertex planar graph  $G$  can be drawn in  $O(n)$  time with a single cubic Bézier curve per edge, in an  $O(n) \times O(n)$  bounding box, such that the edges have  $\Theta(1/\text{degree}(v))$  angular resolution, for each  $v \in G$ , and  $O(\sqrt{n})$  curvature.

**2012 ACM Subject Classification** Theory of computation

**Keywords and phrases** graph drawing, planar graphs, Bézier curves, and RAC drawings

**Digital Object Identifier** 10.4230/LIPIcs.GD.2024.39

**Funding** This work was supported in part by NSF Grant 2212129.

## 1 Introduction

A Bézier curve is a parametric curve defined by a set of *control* points that determine a smooth, continuous curve in the plane [14, 30]. For example, one of the most common types, a *cubic Bézier* curve, is defined by four points,  $P_0, P_1, P_2, P_3$ , such that the curve starts at  $P_0$  tangent to the line segment  $\overline{P_0P_1}$  and ends at  $P_3$  tangent to the line segment  $\overline{P_2P_3}$ , with the lengths of  $\overline{P_0P_1}$  and  $\overline{P_2P_3}$  determining “how fast” the curve turns towards  $P_1$  before turning towards  $P_2$ . Formally, a cubic Bézier curve,  $f$ , has the following explicit form (see Figure 1):

$$f(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3, \quad \text{for } 0 \leq t \leq 1.$$

The speed in which a curve turns can be characterized by its *curvature*, which is a measure of the instantaneous rate of change of direction of a point that moves on the curve; hence, the larger the curvature, the larger this rate of change. For example, the curvature of a line is zero, the curvature of a polygonal chain with a bend is  $+\infty$ , and the curvature of a circle is the reciprocal of its radius.

Formally, the curvature of a twice-differentiable parameterized curve,  $\mathbf{c}(t) = (x(t), y(t))$ , can be defined as follows (e.g., see [33, p. 890]):

$$\kappa(t) = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}},$$

where  $x'$  and  $x''$  are the first and second derivatives of  $x$ , and  $y'$  and  $y''$  are the first and second derivatives of  $y$ , with respect to  $t$ . For this to be well-defined the curve must be smooth enough to have a second derivative, which is not true for polylines. In such cases the curvature can be thought of as infinite, as a limiting case of smooth perturbations of the given curve.



© David Eppstein, Michael T. Goodrich, and Abraham M. Illickan;  
licensed under Creative Commons License CC-BY 4.0

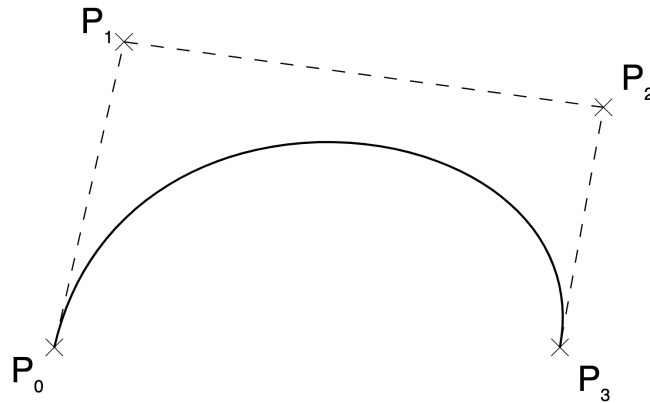
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 39; pp. 39:1–39:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** An example cubic Bézier curve. Public domain image by MarianSigler.

For a graph drawing, we desire the curvatures of the edges to be small, where we define the curvature of a drawing of a graph  $G$  to be the maximum curvature for a non-vertex point on an edge of  $G$ , taken over all edges in the drawing of  $G$ . For example, Xu, Rooney, Passmore, Ham, and Nguyen [36] empirically show that user performance on network tasks is better for low-curvature drawings than for high-curvature drawings.

Unfortunately, minimizing curvature may conflict with other goals for a drawing. For example, curvature can conflict with angular resolution for planar drawings of planar graphs.<sup>1</sup> For example, we can draw an  $n$ -vertex planar graph without crossings using straight edges (i.e., with curvature 0) but this can cause angular resolution to be  $O(1/n)$ , even for low-degree vertices, e.g., see [9, 28, 31], or even worse, e.g., see Tutte [35]. Indeed, Garg and Tamassia [22] show that, in general, the best angular resolution that any algorithm for drawing a degree- $d$  planar graph  $G$  using straight-line drawing can achieve is  $O(\sqrt{(\log d)/d^3})$ . Ideally, we would like the angular resolution for a drawing of a graph  $G$ , to be  $\Omega(1/\text{degree}(v))$ , for each  $v \in G$ , which Goodrich and Wagner show how to achieve [23], but their methods for achieving this bound either use polylines with bends (hence, with infinite curvature) or with Bézier curves that the authors admit have high curvature, and they pose as an open problem whether one can simultaneously achieve good angular resolution and relatively low curvature for planar graph drawings with edges represented with cubic Bézier curves.<sup>2</sup>

In terms of another trade-off for drawings with curves, Huang, Eades, Hong, and Duh [25] empirically show that users performing network tasks were quickest with drawings with curved crossing edges rather than mixed drawings with no crossings, and the authors conclude that for better human graph comprehension, it might be better to use curves to increase crossing angle, rather than to remove them completely. Similarly, Huang, Hong, and Eades [26] report on user studies showing that crossings with large angles are much less harmful to the readability of drawings than shallow crossings. Relatedly, there is considerable prior work on right angle crossing (RAC) drawings, where every pair of crossing edges must cross at right angles, but these drawings are typically achieved by using polygonal paths with bends, e.g.,

<sup>1</sup> Recall that a planar graph can be drawn in the plane without edge crossings and a 1-planar graph can be drawn in the plane so that each edge crosses at most one other edge. Also recall that the angular resolution for each vertex is the minimum angle between two edges incident on  $v$  in the drawing.

<sup>2</sup> For the sake of normalization of the curvature parameter, we assume in this paper that a drawing has an  $O(n) \times O(n)$  bounding box, as is common for drawings of planar and 1-planar graphs.

see [1, 2, 10, 11, 32, 34]; hence, these drawings can have unbounded curvature. Therefore, we are interested in methods for producing RAC drawings of 1-planar graphs using curves having bounded curvature, e.g., such as can be achieved with cubic Bézier curves.

Bézier curves are used extensively in computer graphics applications, where it is common to concatenate Bézier curves together to form a composite Bézier curve, e.g., see [30]. As long as each connection point is collinear with its two adjacent control points, then the resulting composite Bézier curve will be  $C^1$  continuous, but it will not necessarily have continuous curvature. In addition, such representations can be quite complex, depending on the number of pieces used, and the curvature at connection points might not be well-defined or, even if it exists, it might not be easy to bound. Thus, we are interested in this paper on studying drawings of planar graphs and 1-planar graphs using cubic Bézier curves where each edge is represented with a single cubic Bézier curve, so that each edge has bounded curvature. In the case of 1-planar graph drawings, we desire edge crossings to be at right angles, and in the case of planar graph drawings, we would like to simultaneously achieve good angular resolution and low curvature.

## 1.1 Related Prior Work

There is some notable previous work on using Bézier curves for graph drawing, which we review below, but we are not aware of previous work on using Bézier curves to draw planar graphs with low curvature per edge and optimal angular resolution or for RAC drawings of 1-planar graphs.<sup>3</sup>

In addition to the work cited above, there is some interesting prior work on using Bézier curves in graph drawing systems. For example, the Graphviz system of Gansner [21] can render edges using Bézier curves. Finkel and Tamassia [20] describe a force-directed graph drawing implementation that uses Bézier curves to render graph edges by integrating control points into the force equations. Brandes and Wagner [3] visualize railroad systems with some edges rendered using Bézier curves, and Fink, Haverkort, Nöllenburg, Roberts, Schuhmann, and Wolff [19] provide a similar type of system for drawing metro maps. The GDot system of Hong, Eades, and Torkel [24] uses Bézier curves to draw edges in graphs visualized as dot paintings. In addition, the CelticGraph system of Eades, Gröne, Klein, Eades, Schreiber, Hailer, and Schreiber [15] draws graphs using Celtic knots with edges represented as Bézier curves with limited curvature.

In terms of additional theoretical work, Eppstein, Goodrich, and Meng [17] show how to draw confluent layered drawings using Bézier curves that combine multiple edges, and Eppstein and Simons provide a similar result for Hasse diagrams [18]. In addition, there is considerable prior work on Lombardi drawings, where edges are drawn using circular arcs, e.g., see [5, 12, 13, 16, 18, 29], which we consider to be related work even though circular arcs are not Bézier curves. Cheng, Duncan, Goodrich, and Kobourov [4] show how to draw an  $n$ -vertex planar graph  $G$  with asymptotically optimal angular resolution,  $O(1/\text{degree}(v))$ , for each  $v \in G$ , using 1-bend polylines or circle-arc chains, both of which have unbounded curvature.

---

<sup>3</sup> At a workshop affiliated with GD 2023 to celebrate the 60th birthday of Beppe Liotta, Peter Eades advocated for more research on the topic of using Bézier curves to draw graphs, including results involving curvature guarantees.

## 1.2 Our Results

In this paper, we show how to draw 1-planar graphs as RAC drawings using a single cubic Bézier curve for each edge; hence, with bounded curvature. We also show how to draw planar graphs in an  $O(n) \times O(n)$  grid with good angular resolution by rendering each edge using a cubic Bézier curve with  $O(\sqrt{n})$  curvature. Our methods involve careful constructions and proof techniques for proving bounded curvature results, which may be applicable in other settings.

Our constructions are also based in part on refinements of the *convex hull property* of Bézier curves, which is that every point of a Bézier curve lies inside the convex hull of its defining control points, e.g., see [14]. In our results, however, the convex hull property is not sufficient, since the regions in which we desire Bézier curves to traverse are more restrictive than just the convex hulls of control points. Moreover, the convex hull property says nothing about right-angle crossing points for pairs of Bézier curves, which is an important component of our work, and one that requires considerable work, as we show.

## 2 Constrained Constructions for Pairs of Bézier Curves

We show in this paper that we can draw any 1-planar graph in the plane with right angle crossings, i.e., a RAC drawing, using Bézier curves for every pair of intersecting edges and straight line segments for the rest.

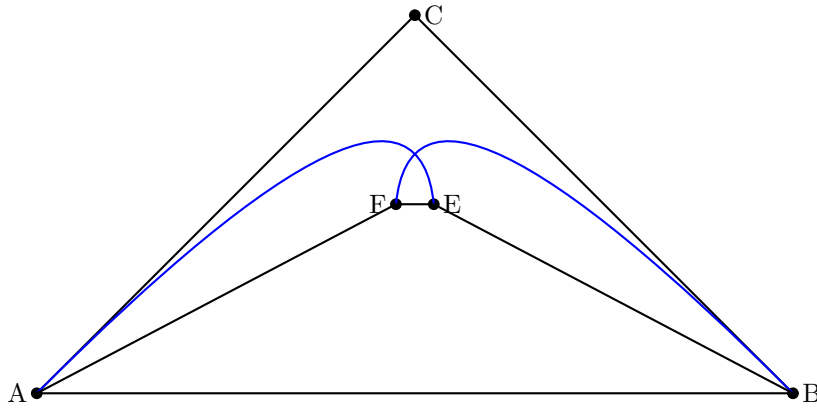
Bekos, Didimo, Liotta, Mehrabi, and Montecchiani [2] show that one can draw any 1-planar graph as a RAC drawing where each edge is represented by a polyline that has at most one bend. Their algorithm starts from a 1-planar (combinatorial) embedding of a 1-planar graph  $G$ , and proceeds via an induction proof involving augmentation and contraction steps to produce a RAC drawing of  $G$  with edges represented with polylines with at most one bend per edge. We show how to adapt their proof to use a single cubic Bézier curve per edge in place of a 1-bend polyline. To achieve this, we develop a number of constructions for pairs of Bézier curves that cross at right angles in specific ways while fitting in specified polygonal regions. As mentioned above, our constructions go well beyond the convex hull property for Bézier curves. We describe each of these constructions in this section and we show in the subsequent section how to use these constructions to prove our main result for RAC drawings of 1-planar graphs, which is the following.

► **Theorem 1.** *Any  $n$ -vertex 1-planar graph has a 1-planar RAC drawing using a single cubic Bézier curve per edge. Further, if a 1-planar embedding of the graph has been provided, a 1-planar RAC drawing using such cubic Bézier curves can be computed in  $O(n)$  time.*

Since Bézier curves have bounded curvature, we achieve a RAC drawing of any 1-planar graph using edges with bounded curvature. We give each of our constructions for constrained pairs of Bézier curves in the subsections that follow. Our constructions make use of another property of Bézier curves; namely, that applying an affine transformation (e.g., a rotation, reflection, translation, or scaling) to a Bézier curve is equal to the Bézier curve defined by applying that same transformation to the original control points, e.g., see [14].

### 2.1 Right-angle Crossing in a Triangle and Outside a Quadrilateral

Our first construction is for defining two cubic Bézier curves that have a right-angle crossing inside a triangle, each with two endpoints that form a quadrilateral with the base of the triangle, such that the curves lie outside of that quadrilateral. The curves that we describe are actually quadratic Bézier curves but any quadratic Bézier curve is also a cubic Bézier curve by de Casteljau's algorithm [7, 8]. See Figure 2.



■ **Figure 2** Triangle  $ABC$  with quadrilateral  $ABEF$  and pair of cubic Bézier curves that cross in a right angle outside the quadrilateral but inside  $ABC$ .

► **Theorem 2.** *For any triangle  $ABC$ , there is a quadrilateral  $ABEF$  contained in  $ABC$  such that there is a pair of Bézier curves with pairs of endpoints  $\{A, E\}$  and  $\{B, F\}$  that intersect each other at right angles, are contained within  $ABC$ , and lie outside of  $ABEF$ . The coordinates of  $E$  and  $F$  and the control points of the Bézier curves can be computed efficiently, given the coordinates of  $A$ ,  $B$  and  $C$ .*

**Proof.** It is enough to consider the case of an isosceles triangle with base  $\overline{AB}$ , since such a triangle can be found within any given triangle  $ABD$ . By the equivalence property for Bézier curves under angle-preserving affine transformations, let us assume  $A = (0, 0)$ ,  $B = (1, 0)$ ,  $C = (1/2, C_y)$ . Let  $E = \left(\frac{1}{4} \left(4C_y^2 + 6C_y + \sqrt{16C_y^4 + 48C_y^3 + 40C_y^2 + 12C_y + 1} + 3\right), C_y/2\right)$  with  $C_y > 0$  and  $F = (1 - E_x, C_y/2)$ . Let us define  $g_1(t) = At^2 + 2Ct(1 - t) + E(1 - t)^2$  and  $g_2(t) = Bt^2 + 2Ct(1 - t) + F(1 - t)^2$ . In the range  $(0, 1)$ , these curves only meet at one point,

$$\left(\frac{1}{2}, \frac{\left(-\sqrt{4C_y^2 + 8C_y + 1} + 6C_y + 1\right) \left(\sqrt{4C_y^2 + 8C_y + 1} + 2C_y - 1\right)}{24C_y}\right),$$

when  $t = \frac{4C_y + 1}{6C_y} - \frac{\sqrt{4C_y^2 + 8C_y + 1}}{6C_y}$  for both curves. At this point, the slope of one curve is  $-1$  and the other is  $+1$ , so they intersect at a right angle. At  $t = \frac{\sqrt{-4E_x^2 + 8E_x - 3} - 2E_x + 1}{2(1 - E_x)}$ ,  $g_2$  is at

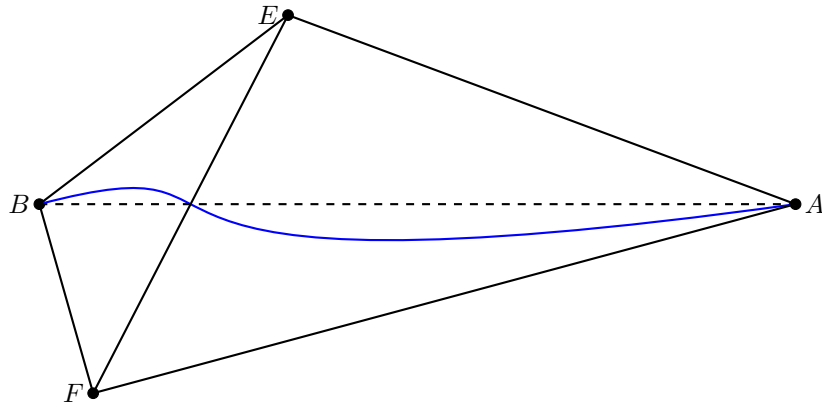
$$\left(E_x, -\frac{C_y \left(\sqrt{-4E_x^2 + 8E_x - 3} - 1\right) \left(3\sqrt{-4E_x^2 + 8E_x - 3} - 8E_x + 5\right)}{8(E_x - 1)^2}\right),$$

which is always higher than  $E_y = C_y/2$ . Thus,  $g_2$  cannot intersect  $\overline{EB}$  or  $\overline{EF}$  except at  $B$  and  $F$  respectively because it is a parabola and  $E$  is inside it. Similarly,  $g_1$  does not intersect  $\overline{AF}$  or  $\overline{EF}$  except at  $A$  and  $F$  respectively. ◀

## 2.2 Right-angle Crossing of a Diagonal of a Convex Quadrilateral

Our next construction is for a cubic Bézier curve  $f$  that replaces a diagonal of a convex quadrilateral so that  $f$  has a right-angle crossing with the other diagonal. See Figure 3.

We actually prove a slightly stronger result, for which, w.l.o.g., we assume the diagonal to be replaced is horizontal.



■ **Figure 3** Pair of intersecting diagonals,  $\overline{EF}$  and  $\overline{AB}$ , inside a convex quadrilateral,  $AEBF$ . We replace  $\overline{AB}$  with a Bézier curve that intersects  $\overline{EF}$  at a right angle.

► **Theorem 3.** For any convex quadrilateral with horizontal diagonal,  $\overline{AB}$ , a point  $X$  on  $\overline{AB}$ , and any real number,  $m$ , there is a simple cubic Bézier curve with endpoints  $A$  and  $B$  such that the Bézier curve intersects  $\overline{AB}$  at  $X$  and makes slope  $m$  at  $X$  and the curve is contained in the quadrilateral and also stays in a pair of opposite quadrants around  $X$ . For  $m \neq 0$ , the curve intersects  $\overline{AB}$  only at  $A$ ,  $X$ , and  $B$ . The control points of the Bézier curve can be computed efficiently, given the vertices of the quadrilateral.

We will work with the points  $A$  and  $B$  being at  $(1, 0)$  and  $(0, 0)$  respectively. For any other pair of points we can apply an angle-preserving affine transformation. W.l.o.g., we also work with the point of intersection being on the line segment from  $(0, 0)$  to  $(8/9, 0)$ , the portion of  $\overline{AB}$  closer to  $B$ , by symmetry. We will first show a curve that is perpendicular at  $X$  in Lemma 4. This first curve can be bounded by any arbitrary quadrilateral with  $\overline{AB}$  as diagonal and also leaves two opposite quadrants around  $X$  free for the other diagonal to be drawn as a straight line segment. Then, in Lemma 5, we show a curve that is a monotonic straight line that reaches  $X$  at the same value of the parameter,  $t$ , as defines the first curve. (See Figure 4.) Finally, in Lemma 6, we take a convex combination of these curves to get the required slope. This third curve is the one that we use.

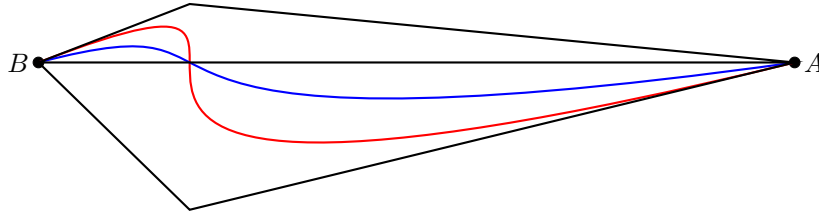
► **Lemma 4.** Let  $A = (1, 0)$ ,  $B = (0, 0)$ , and  $X = (x_0, 0)$  be a point on the line segment  $AB$ . For any convex quadrilateral with  $\overline{AB}$  as diagonal, there is a cubic Bézier curve with  $A$  and  $B$  as endpoints and  $C_1$  and  $D_1$  as control points that intersects the line segment  $\overline{AB}$  at  $X$  at a right angle at the parameter value  $t = t_0 = \frac{C_{1,x} - 2D_{1,x}}{-1 + 3C_{1,x} - 3D_{1,x}}$ , and is contained within this quadrilateral and also within any pair of opposite quadrants around the point of intersection.

**Proof.** Assume  $x_0 < 8/9$ . Otherwise mirror the plane. Let  $D_1 = (D_{1,x}, -r)$  and  $C_1 = (\frac{1}{2}(D_{1,x} - \sqrt{4D_{1,x} - 3D_{1,x}^2}), \frac{1 - 2C_{1,x} + D_{1,x}}{2D_{1,x} - C_{1,x}}r)$  where  $r \in \mathbb{R}$ . We will define  $D_{1,x}$  later. Let  $f_1(t) = At^3 + 3C_1t^2(1-t) + 3D_1t(1-t)^2 + B(1-t)^3$  be the Bézier curve we are constructing. Then the  $y$ -coordinate function for  $f_1$  is

$$f_{1,y}(t) = -3 \frac{1 - 2C_{1,x} + D_{1,x}}{2D_{1,x} - C_{1,x}} rt^2(1-t) + 3rt(1-t)^2.$$

At  $t = t_0$ ,  $f_{1,y}(t) = 0$ . Note that there are three roots for the cubic polynomial  $f_{1,y}$  and the other two are at  $t = 0$  and  $t = 1$ . The  $x$ -coordinate function for  $f_1$  is

$$f_{1,x}(t) = t^3 + 3 \frac{1}{2} (D_{1,x} - \sqrt{4D_{1,x} - 3D_{1,x}^2}) t^2(1-t) + 3D_{1,x} t(1-t)^2.$$



■ **Figure 4** Bézier curves intersecting the line segment,  $\overline{AB}$ , between the endpoints at different angles. The red curve is a curve which meets the line segment at that point at right angles. The blue curve is obtained by taking a convex combination with a curve that remains on the line segment and has the same parameter  $t$  as the red curve at the point of intersection.

At  $t = t_0$ ,  $f_{1,x}(t) = \frac{D_{1,x}(3D_{1,x} + \sqrt{-D_{1,x}(3D_{1,x} - 4)})}{3D_{1,x} + 3\sqrt{-D_{1,x}(3D_{1,x} - 4)} + 2}$  and we choose  $D_{1,x}$  to be  $x_0 + \sqrt[3]{x_0^2 - x_0^3}$  which is a root of  $f_{1,x}(t_0) = x_0$  when treated as an equation in  $D_{1,x}$ . This root is valid whenever  $C_{1,x}$  is real, which happens when  $4D_{1,x} - 3D_{1,x}^2 \geq 0$ , which is true when  $0 \leq D_{1,x} \leq 4/3$ . This happens when  $0 \leq x_0 \leq 8/9$ . Moreover,

$$\frac{df_{1,y}}{dt} = -\frac{3r(C_{1,x}(9t^2 - 8t + 1) + D_{1,x}(-9t^2 + 10t - 2) + (2 - 3t)t)}{C_{1,x} - 2D_{1,x}} \quad \text{and}$$

$$\frac{df_{1,x}}{dt} = 3(t(C_{1,x}(2 - 3t) + t) + D_{1,x}(3t^2 - 4t + 1)).$$

At  $t = t_0$ ,

$$\frac{df_{1,x}}{df_{1,y}} = \frac{\frac{df_{1,x}}{dt}}{\frac{df_{1,y}}{dt}} = 0,$$

which means that the angle at the point of intersection is  $\pi/2$ . Also, the value of  $C_{1,x}$  is such that  $\frac{df_{1,x}}{dt} = 3t^2(-3C_{1,x} + 3D_{1,x} + 1) + 3t(2C_{1,x} - 4D_{1,x}) + 3D_{1,x}$  is a quadratic polynomial in terms of  $t$  with discriminant 0. This means that  $t_0$  is a repeated root and, hence,  $f_{1,x}$  is monotonic. Thus, the curve remains in a pair of opposite quadrants, determined by the sign of  $r$ . Next, we show that this curve is bounded by the quadrilateral  $(0, 0), (x_0, \frac{-rx_0}{D_{1,x}}), (1, 0), (x_0, \frac{r(1-2C_{1,x}+D_{1,x})(x_0-1)}{(C_{1,x}-1)(2D_{1,x}-C_{1,x})})$ . This is obtained by the convex hull property of Bézier curves in conjunction with the fact that  $f_{1,x}$  is monotonic. For any value of  $x_0$ , these values are bounded. The value of  $r$  can be adjusted such that this quadrilateral is contained within any quadrilateral with the diagonal  $\overline{AB}$ . ◀

We next address the case for the point of intersection  $X$  having  $x$ -coordinate in  $(0, 8/9)$  and forming an angle of 0.

► **Lemma 5.** *Let  $A = (1, 0)$ ,  $B = (0, 0)$ , and  $X = (x_0, 0)$  a point on the line segment  $\overline{AB}$ . There is a cubic Bézier curve with  $A$  and  $B$  as endpoints that is at  $X$  when  $t = t_0$  as obtained from Lemma 4 for the same  $X$  and monotonically traces the straight line segment  $\overline{AB}$ .*



### 39:8 Drawing Planar Graphs and 1-Planar Graphs Using Cubic Bézier Curves

**Proof.** Assume  $x_0 < 8/9$ . Otherwise, mirror the plane. Let  $C_2 = D_2 = (\frac{x_0 - t_0^3}{3(1-t_0)t_0}, 0)$  and  $f_2(t) = At^3 + 3C_2t^2(1-t) + 3D_2t(1-t)^2 + B(1-t)^3$  be the Bézier curve we are constructing. It is easy to see that for  $f_2$ 's  $y$ -coordinate function,  $f_{2,y} = 0$  throughout. The  $x$ -coordinate function is

$$f_{2,x}(t) = t^3 + \frac{(1-t)t(x_0 - t_0^3)}{(1-t_0)t_0}.$$

At  $t = t_0$ ,  $f_{2,x}(t) = x_0$ . Also,  $C_2$  and  $D_2$ , the repeated control points are between  $A$  and  $B$ .

This means that the curve is always moving from  $B$  to  $A$  when  $t$  goes from 0 to 1, without turning back or overshooting  $A$ . ◀

Given the previous two lemmas, we now can use them in convex combination.

► **Lemma 6.** *Let  $A = (1, 0)$ ,  $B = (0, 0)$ ,  $X = (x_0, 0)$  a point on the line segment  $\overline{AB}$  and  $m$  a real number. For any convex quadrilateral with  $\overline{AB}$  as a horizontal diagonal, there is a simple cubic Bézier curve with  $A$  and  $B$  as endpoints that intersects the line segment  $\overline{AB}$  at  $X$  with slope  $m$  for the parameter value  $t = t_0$  as obtained from Lemma 4 for the same  $X$  and is contained within this quadrilateral and also within any pair of opposite quadrants around the point of intersection.*

**Proof.** Let  $C_3 = kC_1 + (1-k)C_2$  and  $D_3 = kD_1 + (1-k)D_2$  for  $k \in [0, 1]$  where  $C_1, D_1$  are control points of a curve  $f_1(t)$  obtained from the proof of Lemma 4 and  $C_2, D_2$  are control points of a curve  $f_2(t)$  obtained from the proof of Lemma 5. Let

$$f_3(t) = At^3 + 3C_3t^2(1-t) + 3D_3t(1-t)^2 + B(1-t)^3 = kf_1(t) + (1-k)f_2(t)$$

be the more-general Bézier curve we are now constructing as a convex combination of  $f_1$  and  $f_2$ . Clearly,  $f_3(t)$  is at  $X$  when  $t = t_0$ . In addition, we can write the slope of  $f_3$  at  $t_0$  as follows:

$$\frac{df_{3,y}}{df_{3,x}} = \frac{\frac{df_{3,y}}{dt}}{\frac{df_{3,x}}{dt}} = \frac{k \frac{df_{1,y}}{dt}}{k \frac{df_{1,x}}{dt} + (1-k) \frac{df_{2,x}}{dt}}.$$

To get a positive slope, we use a positive value of  $r$ . To get a negative slope, we use a negative value of  $r$ . Values of  $k$  in  $[0, 1]$  will span the full range of slopes (either positive or negative) for every value of  $r$ . To get a specific slope  $m$  at  $t = t_0$ , we set

$$k = \frac{m \frac{df_{2,x}}{dt}}{m \left( \frac{df_{1,x}}{dt} - \frac{df_{2,x}}{dt} \right) + \frac{df_{1,y}}{dt}}.$$

Now, we need to show that this curve does not self-intersect.  $f_{3,x}$  is monotonic as it is a convex combination of monotonic functions. This means that no  $x$ -coordinate is repeated and the curve does not self intersect. The curve,  $f_3$ , is also bounded by the quadrilateral,  $(0, 0), (x_0, \frac{-rx_0}{D_{1,x}}), (1, 0), (x_0, \frac{r(1-2C_{1,x}+D_{1,x})(x_0-1)}{(C_{1,x}-1)(2D_{1,x}-C_{1,x})})$ , as it is a convex combination of the curves from Lemma 4 and Lemma 5, which are bounded by the same. The value of  $r$  can be adjusted such that this quadrilateral is contained within any arbitrary quadrilateral. Again this curve also remains in a pair of opposite quadrants around  $X$  since  $f_{3,x}$  is monotonic. ◀



### 3 RAC Drawings of 1-Planar Graphs with Bézier Curves

We are now ready to prove Theorem 1, which is the following.

► **Theorem 1.** *Any  $n$ -vertex 1-planar graph has a 1-planar RAC drawing using a single cubic Bézier curve per edge. Further, if a 1-planar embedding of the graph has been provided, a 1-planar RAC drawing using such cubic Bézier curves can be computed in  $O(n)$  time.*

The proof goes through three stages, adapting a proof of Bekos, Didimo, Liotta, Mehrabi, and Montecchiani [2] for RAC drawings of 1-planar graphs with 1-bend polygonal edges.

#### 3.1 Augmentation

We start with a 1-plane combinatorial drawing  $G$  of the graph. We call every connected region of the plane bounded by edges and parts of edges a face. The number of such edges or parts is called the length of the face. The induction will be using triangulated 1-plane multigraphs, that is, 1-plane multigraphs in which every face is of length 3. For every pair of crossing edges  $ab, cd$ , add edges  $ac, cb, bd, ad$  such that the only edges contained within the cycle  $acbd$  are  $ab$  and  $cd$ . We call the subgraph consisting of these edges an empty kite. If there are 2-length faces in this drawing, remove one of the edges recursively until there are no more. Also remove any parallel edge that was crossed. Now all faces in this drawing are either of length three with 2 vertices and a crossing point or bounded only by vertices and no crossing points. In every face longer than 3 add a new vertex and connect it to all the vertices on the face. We call this 1-plane multigraph  $G^+$ .

#### 3.2 Contraction

A separation pair is a pair of vertices  $\{u, v\}$  whose removal disconnects the graph. Lemma 5 in [2] states that between any separation pair  $\{u, v\}$ , there exist two parallel edges  $e, e'$  such that  $\{u, v\}$  is not a separation pair for the graph obtained by removing everything inside the cycle  $\langle e, u, e', v \rangle$ . We call this removed subgraph along with the cycle  $G_{uv}$ . Replace  $G_{uv}$  with a thick edge and iterate until there are no separation pairs. We call this graph  $G^*$ .  $G^*$  is a simple triangulated 1-plane graph.

#### 3.3 Drawing

Obtain graph  $H^*$  by removing all crossing pairs in  $G^*$ . All the faces of  $H^*$  have either three or four vertices. Lemma 7 in [2] states that  $H^*$  is 3-connected. We can draw it with all faces convex and the outer face as a trapezoid or triangle using Tutte's method [35] or the method of Chiba *et al.* [6] to do it in linear time. Insert all the crossing edges in the interior four length faces by first drawing one of the edges as a straight line segment and then the other with the required slope using Theorem 3. If the outer face is of length four, use Theorem 2 with any large triangle.

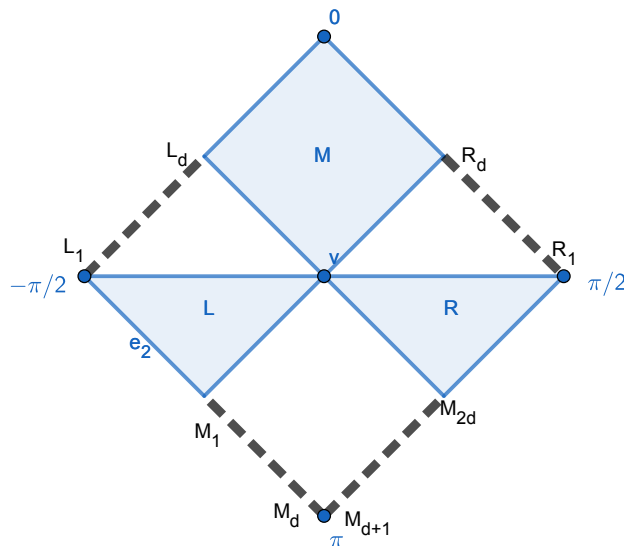
For any thick edge  $(u, v)$  with a triangle  $uvx$  that  $G_{uv}$  could be contained in (here  $x$  need not be a vertex, it could be any point), consider  $H_{uv}$  obtained by removing all crossing pairs. This is also 3-connected like  $H^*$ . If the outer face of  $H_{uv}$  is of length three and of the form  $uvw$ , recursively draw it inside  $uvx$ . If the outer face is of length four, use Theorem 2 to obtain the trapezoid and crossing pair and continue recursively.

## 4 Drawing Planar Graphs Using Bézier curves

Suppose we are given an  $n$ -vertex planar graph  $G$ . In this section, we describe a method of drawing  $G$  with asymptotically optimal angular resolution,  $\Theta(1/\text{degree}(v))$ , for each  $v \in G$ , using a single cubic Bézier curve of curvature  $O(\sqrt{n})$  for each edge. We start from an  $O(n) \times O(n)$  grid drawing  $D$  with asymptotically optimal angular resolution obtained by algorithm `OneBend` from Cheng, Duncan, Goodrich, and Kobourov [4], which uses one-bend edges. We describe some properties of this drawing in Section 4.1. We describe our new drawing using Bézier curves in Section 4.2. We show that the edges do not cross each other in Section 4.2.1. We show that the vertices in our drawing have asymptotically optimal angular resolution in Section 4.2.2. We show that the curvature of our drawing is  $O(\sqrt{n})$  in Section 4.2.3.

### 4.1 The Drawing Obtained by the `OneBend` Algorithm

In the `OneBend` algorithm,  $G$  is drawn in a drawing,  $D$ , in an  $O(n) \times O(n)$  grid such that every vertex,  $v$ , has a *joint box* – a square rotated  $\pi/4$  of width and height  $4\text{degree}(v) + 4$ , centered at  $v$ . Each joint box is divided into six regions. (See Figure 5.)



■ **Figure 5** A joint box for a degree- $d$  vertex,  $v$ , with left ports,  $L_1, \dots, L_d$ , right ports,  $R_1, \dots, R_d$ , and middle ports,  $M_1, \dots, M_{2d}$ , and free regions,  $L$ ,  $M$ , and  $R$ . Image from [4].

The regions of a joint box are of two types, free regions and port regions. The free regions are as follows –  $M$  is  $\pi/4$  on either side of the top corner and  $L$  (resp.,  $R$ ) is  $\pi/4$  below the left (resp., right) corner. The port regions are as follows –  $M$  is opposite the free  $M$  region,  $L$  (resp.,  $R$ ) is between the free  $M$  and  $L$  (resp.,  $R$ ) regions. The sides of the square on port regions have  $d$  evenly spaced ports. Every edge in the drawing is between a free  $M$  region and a port  $M$  region, a free  $L$  region and a port  $R$  region, or a free  $R$  region and a port  $L$  region. Every edge is drawn as two line segments starting at the endpoints of the edge and meeting at a distinct port on the port region [4].

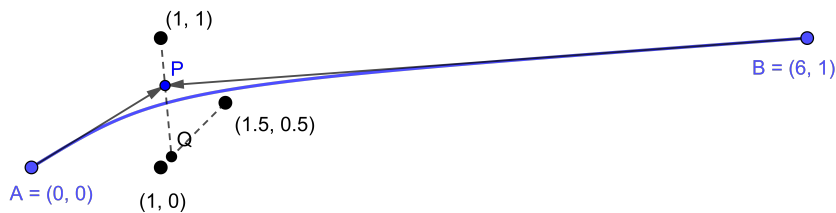
In our construction, we treat each of the  $M$  regions as two regions such that we have 8 regions which are congruent. However, only one of the two free  $M$  regions will have an edge.

## 4.2 Drawing a Planar Graph using Cubic Bézier Curves

We take the same embedding of the vertices as given by the `OneBend` algorithm. Since these curves can be rotated, translated and reflected without changing the curvature, for any edge  $(A, B)$  which is a 1-bend polyline with the port region at  $A$  and free region at  $B$ , without loss of generality, we may assume the following:  $A$  is at the origin,  $(0, 0)$ , and  $B = (b_1, b_2)$  is in the region bounded below by the  $x$ -axis and above by the line  $y = x - 1$ . After this transformation the edges through the same port region of  $A$  in the drawing obtained by the `OneBend` algorithm are ported through  $(d + 1, d + 1) + (i, -i)$  for distinct values of  $i$  which follow the same order along the vertex and  $1 \leq i \leq d$ . In our construction, we replace the polyline edges of the `OneBend` algorithm with cubic Bézier curves of the form

$$\gamma(t) = Bt^3 + 3Pt^2(1-t) + 3Qt(1-t)^2,$$

where  $P = (1 - k)(1, 1) + kQ$ ,  $Q = (1 - s)(1, 0) + s(3/2, 1/2)$ ,  $s = b_2/b_1$ , and  $k = i/(d + 1)$ . The repeated control point,  $P$ , is a convex combination of  $(1, 1)$  and  $Q$  depending on the parameter,  $k$ .  $Q$ , in turn, is a convex combination of  $(1, 0)$  and  $(3/2, 1/2)$ , depending on the angle  $b_2/b_1$  that  $B$  makes with the  $x$ -axis. See Figure 6.



■ **Figure 6** Bézier curve for an edge,  $(A, B)$ .

### 4.2.1 Planarity

In this subsection, we show that the cubic Bézier curves representing edges with a common endpoint do not cross each other. First we show that edges that meet at a common endpoint but through different ports do not intersect. We show that an edge through the ported  $R$  region does not intersect with an edge through the free  $R$  region or the free  $M$  region of the same vertex. A similar argument shows that an edge through the ported  $L$  region or either half of the ported  $M$  region does not intersect with an edge through an adjacent region.

► **Lemma 7.**  $\gamma(t)$  lies in the region  $x > y > 0$  for  $t \in (0, 1)$ .

**Proof.** The proof follows from the convex hull property of Bézier curves and the fact that this region itself is convex. The control points except  $A$  are within this region. Only  $A$  could be on the boundary of the region. Since all points on the curve except for the endpoints will have a positive coefficient for the two intermediate control points, these points will lie within the region. ◀

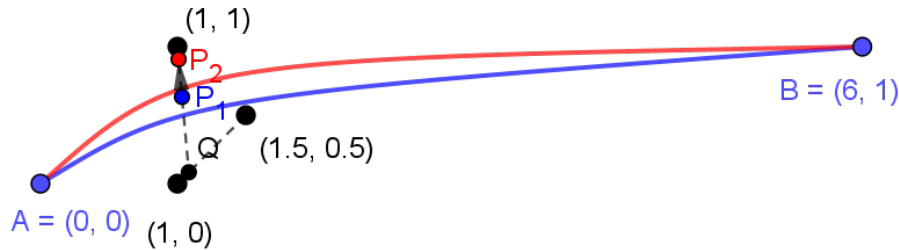
In other words, an edge ported through the  $R$  region will remain in the wedge of this  $R$  region extended to infinity. Let us next consider whether such an edge will intersect another one in the free  $M$  region. The control points of any edge through the free  $M$  region lie in the region  $y \geq x$  and therefore can't intersect with  $\gamma$ . The control points of any edge through the free  $R$  region lie in the region  $y \leq 0$  and therefore can't intersect with  $\gamma$ .

## 39:12 Drawing Planar Graphs and 1-Planar Graphs Using Cubic Bézier Curves

Next we show that two curves through the same ported region do not intersect. We will show this for the ported  $R$  region but the argument is similar for ported  $L$  region and both halves of the ported  $M$  region. We actually prove a stronger result, namely, that even if we allowed for parallel edges with the same two endpoints, they would not cross.

► **Lemma 8.** *Suppose  $\gamma_1$  and  $\gamma_2$  are two edge curves with the same endpoints using different values of  $k$ ,  $k_1 > k_2$ , respectively. The curves  $\gamma_1$  and  $\gamma_2$  do not intersect except at the endpoints, and, except at the endpoints,  $\gamma_2$  is above  $\gamma_1$ .*

**Proof.** For the common endpoint of  $\gamma_1$  and  $\gamma_2$ , which has the same  $s = b_2/b_1$ , the different values of  $k$  give different values of  $P$  along a line of slope  $1 - 2/s$ . For the values of  $B$  under consideration,  $0 \leq s < 1$  and hence the slope of this line is less than  $-1$ . (See Figure 7.) Consider a rotation such that this line is now perpendicular to the  $x$ -axis and the two corresponding values of  $P$  such that  $P_2$  is above  $P_1$ . All points of  $\gamma_2$  would lie directly above the corresponding points of  $\gamma_1$ . Both curves are monotonic in the horizontal direction since the repeated control point is between the two endpoints in the horizontal direction. The curves do not intersect except at the endpoints.  $\gamma_2$  is above  $\gamma_1$ . ◀



■ **Figure 7** Parallel edges would not cross. As  $k$  decreases, all point except the endpoints move up in a direction parallel to  $P_1P_2$ . The curve itself moves monotonically in the perpendicular direction with  $t$ .

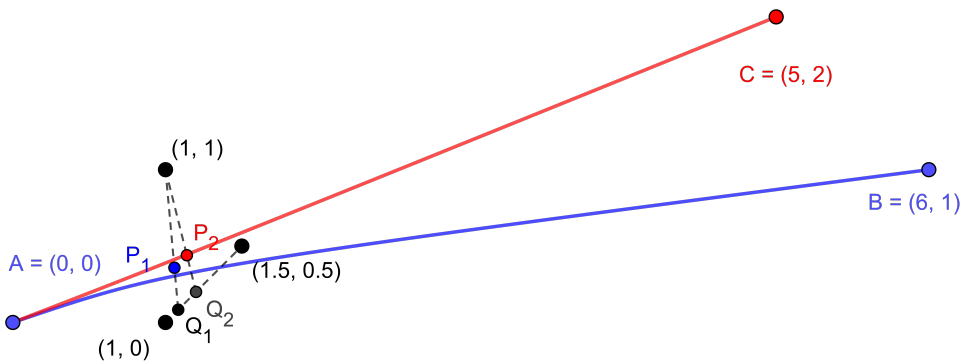
We next show that two curves with exactly one common endpoint and ported through the same  $R$  region do not cross.

► **Lemma 9.** *Let  $\gamma_1$  and  $\gamma_2$  be two curves with the same value of  $k$  but different vertices  $B$  and  $C$  which are endpoints of edges incident on  $A$  through the same region in the **OneBend** algorithm, with  $B$  coming counterclockwise first. The curves intersect only at  $A$  and  $\gamma_2$  lies above  $\gamma_1$  everywhere else.*

**Proof.** Let  $P_1$  and  $P_2$  be the corresponding values of  $P$ . Since the 1-bend edge  $BC$  has to pass through a pair of ports as described in Section 4.1, the straight line  $BC$  has to make an angle between  $\pi/4$  and  $3\pi/4$  with the  $x$ -axis. When moving from  $B$  to  $C$ ,  $s$  increases or remains the same. This means that  $P_2$  is the same as  $P_1$  or is further away along a line making angle  $\pi/4$  with the  $x$ -axis. (See Figure 8.) Consider the transformation  $\{x, y\} \mapsto \{x - y, x + y\}$ . Within this proof, unless noted otherwise, everything will be considered under this transformation. For any value of  $P$ , the  $x$ -coordinate is  $k$ , which is between 0 and 1 and the  $y$ -coordinate is  $2 - k(1 - s)$  which is between 1 and 2.  $A$  has both coordinates less than those of  $P_1$  and  $P_2$ .  $B$  and  $C$  have both coordinates greater than those of  $P_1$  and  $P_2$  respectively. This means that both curves are monotonic along both axes.  $C_x \leq B_x$  and  $C_y \geq B_y$  with at least one of the inequalities strict. Suppose there are  $t_1, t_2$  such that  $\gamma_1(t_1) = \gamma_2(t_2)$ .  $\gamma_2(t_1)_y \geq \gamma_1(t_1)_y = \gamma_2(t_2)_y$  with equality only when

$B_y = C_y$ . Monotonicity implies that  $t_1 \geq t_2$ . Similarly,  $\gamma_2(t_1)_x \leq \gamma_1(t_1)_x = \gamma_2(t_2)_x$  with equality only when  $B_x = C_x$ . Monotonicity implies that  $t_1 \leq t_2$ . The curves do not intersect unless  $B = C$  which is not the case.  $\gamma_2$  lies above  $\gamma_1$  because  $\gamma_2(t)_y \geq \gamma_1(t)_y$  and the curves do not intersect. ◀

For two curves through the same ported region, they will have different endpoints and different values of  $k$ . The curve with the same value of  $k$  as the upper edge and the endpoint the same as the lower edge, lies below the curve representing the upper edge and above the curve representing the lower edge. Therefore, the curves representing the upper and lower edges do not intersect.



■ **Figure 8** When the endpoint changes from  $B$  to  $C$  and  $k$  remains the same, all points except  $A$  move in the direction  $P_1P_2$  which is the same as  $Q_1Q_2$  or remain where they are. In the perpendicular direction, they move to the left, away from  $\gamma_1$ .

### 4.2.2 Angular Resolution

We next show that for each vertex,  $v \in G$ , in this drawing, we achieve angular resolution  $\Omega(1/\text{degree}(v))$ .

► **Lemma 10.** *Let  $v \in G$  be a vertex with degree  $\text{degree}(v)$ . The angular resolution of  $v$  in our drawing of  $G$  using cubic Bézier curves is  $\Omega(1/\text{degree}(v))$ .*

**Proof.** Without loss of generality, let us assume that the vertex  $v$  is at  $A = (0, 0)$  and has degree  $d$ . We will consider the angle between any pair of adjacent edges, at least one of which is ported through the  $R$  region. The other cases are similar. We first consider the case when both edges are ported through the  $R$  region. The tangent of each Bézier curve for an edge incident to  $v$ , at the point of contact is the same direction as the line segment between  $A$  and the control point  $P$ . For a pair of curves with the same pair of endpoints but different values of  $P$ , that is,  $P_1$  and  $P_2$  due to different values of  $k$  due to different values of  $i$ , consider the triangle  $P_1AP_2$ .  $P_1P_2$  has length at least  $1/(\sqrt{2}d)$ . From the law of sines,

$$\frac{1/(\sqrt{2}d)}{\sin \angle P_1AP_2} = \frac{P_1A}{\sin \angle P_1P_2A},$$

$$\frac{1}{\sqrt{2}d \sin \angle P_1AP_2} \leq \frac{\sqrt{10}/2}{1/\sqrt{2}}, \quad \text{and}$$

$$\sin \angle P_1AP_2 \geq \frac{1}{\sqrt{10}d}.$$

From, the Taylor expansion of  $\sin^{-1}$ ,

$$\angle P_1AP_2 \geq \frac{1}{\sqrt{10d}}.$$

Thus, we have  $\Omega(1/d)$  angle between this pair of parallel edges. From Lemma 9, we know that the second curve would be above these edges and would have an even larger angle. Note that these curves also leave at least the same angle with the boundaries of ported  $R$  region since we use  $k = i/(d+1)$  and  $i$  ranges from 1 to  $d$  and 0 and 1 correspond to the corners. This means that this minimum angle is maintained with any edge incident to  $v$  through any other region as well.  $\blacktriangleleft$

### 4.2.3 Curvature

We next show that the curvature of our construction is  $O(\sqrt{n})$ .

► **Lemma 11.** *The curvature of  $\gamma$  is  $O(\sqrt{b_1}) \subseteq O(\sqrt{n})$ .*

**Proof.** We denote the curvature by  $\kappa$ . By the definition of curvature,

$$\frac{\partial(\kappa^2/b_1)}{\partial b_1} = \frac{8(t-1)^2t^2(k(s^2-s+2)+2(s-1))^2 \left( \begin{array}{c} -2(5b_1^2(s^2+1)t^4 - 8b_1(s+1)t^3 + 4t^2(b_1s+b_1-2) + 8t-2) \\ +4k(s-1)(2t-1)(b_1st^2+2t-1) + k^2(s^2-2s+2)(1-2t)^2 \end{array} \right)}{9 \left( \begin{array}{c} 2b_1^2(s^2+1)t^4 - 2k(s-1)(2t-1)(b_1st^2-4t+2) \\ -8b_1(s+1)t^3 + 4t^2(b_1s+b_1+4) + k^2(s^2-2s+2)(1-2t)^2 - 16t+4 \end{array} \right)^4}.$$

In terms of  $b_1$ , the numerator of  $\frac{\partial(\kappa^2/b_1)}{\partial b_1}$  is a quadratic polynomial with the coefficient of the degree-2 term being negative. The denominator is a fourth power and hence always non-negative. For values of  $b_1$ ,  $s$ ,  $k$  and  $t$  that we care about,  $\frac{\partial(\kappa^2/b_1)}{\partial b_1}$  has at most one root,

$$\frac{-(2t-1) \left( \begin{array}{c} \sqrt{2k^2(7s^4-14s^3+17s^2-10s+10) + 8k(7s^3-5s^2+3s-5) + 8(7s^2+4s+7)} \\ -2ks^2 + 2(k-2)s - 4 \end{array} \right)}{10(s^2+1)t^2},$$

if  $t < 0.222$  and no roots otherwise. It is negative everywhere except possibly before this root. In addition,  $\kappa^2/b_1$  decreases as  $b_1$  increases after this point for fixed  $k$ ,  $s$  and  $t$ , because this a quadratic with a negative leading coefficient. At  $b_1 = 4$ , the curvature is

$$\frac{128(t-1)^2t^2(k(s^2-s+2)+2(s-1))^2}{9 \left( \begin{array}{c} k^2(s^2-2s+2)(1-2t)^2 \\ -4k(s-1)(2t-1)(2st^2-2t+1) \\ +4(8(s^2+1)t^4 - 8(s+1)t^3 + 4(s+2)t^2 - 4t+1) \end{array} \right)^3}.$$

This is less than 3 for all values of  $s$ ,  $k$  and  $t$  between 0 and 1. We evaluate  $\kappa^2/b_1$  at the possible root of  $\frac{\partial(\kappa^2/b_1)}{\partial b_1}$  and get the following:<sup>4</sup>

<sup>4</sup> We verified this expression, as well as the others in this proof, using Mathematica [27].

$$\frac{12500 \left( \frac{(s^2 + 1)^2 (k(s^2 - s + 2) + 2(s - 1))^2}{2ks^2 - 2(k - 2)s + 4} \right) (t - 1)^2}{243 \left( \begin{array}{l} \left( \frac{4k^2 (2s^4 - 4s^3 + 7s^2 - 5s + 5)}{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)} \right. \\ \left. + k(s - 1) \left( s \left( \sqrt{\frac{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)}{+8k (7s^3 - 5s^2 + 3s - 5) + 8 (7s^2 + 4s + 7)}} - 8 \right) \right) \right. \\ \left. + 2 \left( \begin{array}{l} \left( \frac{+32s^2 + 40}{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)} \right. \right. \\ \left. \left. + \sqrt{\frac{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)}{+8k (7s^3 - 5s^2 + 3s - 5) + 8 (7s^2 + 4s + 7)}} + 16s^2 + 16 \right) \right) \right) \end{array} \right)^3 (2t - 1)^5.$$

This is a product of two expressions, one independent of  $t$  and the other entirely in  $t$ . The magnitude of the expression in  $t$  is bounded by 12 for all  $0 \leq t \leq 0.222$ . The magnitude of the other expression is bounded by  $1/128$  for all values of  $s$  and  $k$  between 0 and 1. Putting these together we get that  $\kappa^2/b_1$  is bounded by  $12/128$ . The curvature of these curves is  $O(\sqrt{b_1})$  and hence  $O(\sqrt{n})$  since  $b_1$  is  $O(n)$ . ◀

To sum up, we have the following theorem.

► **Theorem 12.** *Given an  $n$ -vertex planar graph,  $G$ , we can draw  $G$  in an  $O(n) \times O(n)$  grid and  $\Omega(1/\text{degree}(v))$  angular resolution, for each vertex  $v \in G$ , using a single cubic Bézier curve with curvature  $O(\sqrt{n})$  per edge in  $O(n)$  time.*

## 5 Conclusion

In this paper, we have studied methods for drawing 1-planar and planar graphs using cubic Bézier curves with bounded curvature. Possible directions for future work and open problems include the following:

- Can the curvature for drawing an  $n$ -vertex planar graph using a single Bézier curve for each edge be improved from  $O(\sqrt{n})$  while still maintaining an angular resolution of  $\Omega(1/\text{degree}(v))$ , for each vertex  $v$  in the drawing?
- Our algorithm used to produce a RAC drawing of an  $n$ -vertex 1-planar graph  $G$ , given a combinatorial 1-planar drawing of  $G$ , is based on the recursive construction of Bekos, Didimo, Liotta, Mehrabi, and Montecchiani [2]. This allows us to achieve bounded curvature for each edge in the drawing, but it does not give us a bound on the curvature in terms of  $n$ . Is it possible to achieve such a bound?
- Our algorithm used to produce a RAC drawing of an  $n$ -vertex 1-planar graph uses “S”-shaped Bézier curves. Can the same result be achieved with “C”-shaped Bézier curves, e.g., quadratic Bézier curves, which are arguably more aesthetically pleasing?
- As mentioned above, Lombardi drawings are drawings where edges are drawn using circular arcs, e.g., see [5, 12, 13, 16, 18, 29], but circular arcs are not Bézier curves. Can every graph with a Lombardi drawing also be drawn with the same edge crossings using a single Bézier curve of bounded curvature for each edge?



## References

- 1 Patrizio Angelini, Michael A. Bekos, Henry Förster, and Michael Kaufmann. On RAC drawings of graphs with one bend per edge. *Theoretical Computer Science*, 828–829:42–54, 2020. doi:10.1016/j.tcs.2020.04.018.
- 2 Michael A. Bekos, Walter Didimo, Giuseppe Liotta, Saeed Mehrabi, and Fabrizio Montecchiani. On RAC drawings of 1-planar graphs. *Theoretical Computer Science*, 689:48–57, 2017. doi:10.1016/j.tcs.2017.05.039.
- 3 Ulrik Brandes and Dorothea Wagner. Using graph layout to visualize train interconnection data. *Journal of Graph Algorithms and Applications*, 4(3):135–155, 2000. doi:10.1007/3-540-37623-2\_4.
- 4 C. C. Cheng, Christian A. Duncan, Michael T. Goodrich, and Stephen G. Kobourov. Drawing planar graphs with circular arcs. *Discrete & Computational Geometry*, 25:405–418, 1999. doi:10.1007/s004540010080.
- 5 Roman Chernobelskiy, Kathryn I. Cunningham, Michael T. Goodrich, Stephen G. Kobourov, and Lowell Trott. Force-directed Lombardi-style graph drawing. In Marc van Kreveld and Bettina Speckmann, editors, *Graph Drawing (GD)*, pages 320–331. Springer, 2012. doi:10.1007/978-3-642-25878-7\_31.
- 6 Norishige Chiba. Linear algorithms for convex drawings of planar graphs. *Progress in graph theory*, pages 153–173, 1984.
- 7 Paul de Casteljau. Outillage methodes calcul. *Enveloppe Soleau 40.040*, 1959.
- 8 Paul de Casteljau. Courbes et surfaces à pôles. *Microfiche P 4147-1*, 1963.
- 9 Hubert De Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990. doi:10.1007/BF02122694.
- 10 Walter Didimo, Peter Eades, and Giuseppe Liotta. Drawing graphs with right angle crossings. *Theoretical Computer Science*, 412(39):5156–5166, 2011. doi:10.1016/j.tcs.2011.05.025.
- 11 Walter Didimo, Giuseppe Liotta, Saeed Mehrabi, and Fabrizio Montecchiani. 1-bend RAC drawings of 1-planar graphs. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing and Network Visualization (GD)*, pages 335–343. Springer, 2016. doi:10.1007/978-3-319-50106-2\_26.
- 12 Christian Duncan, David Eppstein, Michael Goodrich, Stephen Kobourov, and Martin Nöllenburg. Lombardi drawings of graphs. *Journal of Graph Algorithms and Applications*, 16(1):85–108, 2012. doi:10.7155/jgaa.00251.
- 13 Christian A. Duncan, David Eppstein, Michael T. Goodrich, Stephen G. Kobourov, Maarten Löffler, and Martin Nöllenburg. Planar and poly-arc Lombardi drawings. *Journal of Computational Geometry*, 9(1):328–355, 2018. doi:10.20382/jocg.v9i1a11.
- 14 Marsh Duncan. Bézier curves I. In *Applied Geometry for Computer Graphics and CAD*, pages 135–160. Springer, 2005. doi:10.1007/1-84628-109-1\_6.
- 15 Peter Eades, Niklas Gröne, Karsten Klein, Patrick Eades, Leo Schreiber, Ulf Hailer, and Falk Schreiber. CelticGraph: Drawing graphs as Celtic knots and links. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization (GD)*, pages 18–35. Springer, 2023. doi:10.1007/978-3-031-49272-3\_2.
- 16 David Eppstein. Planar Lombardi drawings for subcubic graphs. In Walter Didimo and Maurizio Patrignani, editors, *Proc. 20th Int. Symp. Graph Drawing*, volume 7704 of *Lecture Notes in Computer Science*, pages 126–137. Springer, 2012. doi:10.1007/978-3-642-36763-2\_12.
- 17 David Eppstein, Michael T Goodrich, and Jeremy Yu Meng. Confluent layered drawings. *Algorithmica*, 47:439–452, 2007. doi:10.1007/s00453-006-0159-8.
- 18 David Eppstein and Joseph Simons. Confluent Hasse diagrams. *Journal of Graph Algorithms and Applications*, 17(7):689–710, 2013. doi:10.7155/jgaa.00312.
- 19 Martin Fink, Herman Haverkort, Martin Nöllenburg, Maxwell Roberts, Julian Schuhmann, and Alexander Wolff. Drawing metro maps using Bézier curves. In Walter Didimo and Maurizio Patrignani, editors, *Graph Drawing (GD)*, pages 463–474. Springer, 2013. doi:10.1007/978-3-642-36763-2\_41.

- 20 Benjamin Finkel and Roberto Tamassia. Curvilinear graph drawing using the force-directed method. In János Pach, editor, *Graph Drawing (GD)*, pages 448–453. Springer, 2005. doi:10.1007/978-3-540-31843-9\_46.
- 21 Emden R. Gansner. Drawing graphs with Graphviz. Technical report, AT&T Bell Laboratories, 2009.
- 22 Ashim Garg and Roberto Tamassia. Planar drawings and angular resolution: Algorithms and bounds. In Jan van Leeuwen, editor, *Algorithms – ESA '94*, volume 855 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1994. doi:10.1007/BFB0049393.
- 23 Michael T. Goodrich and Christopher G. Wagner. A framework for drawing planar graphs with curves and polylines. *Journal of Algorithms*, 37(2):399–421, 2000. doi:10.1006/jagm.2000.1115.
- 24 Seok-Hee Hong, Peter Eades, and Marnijati Torkel. Gdot: Drawing graphs with dots and circles. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, pages 156–165, 2021. doi:10.1109/PacificVis52677.2021.00029.
- 25 Weidong Huang, Peter Eades, Seok-Hee Hong, and Henry Been-Lirn Duh. Effects of curves on graph perception. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 199–203, 2016. doi:10.1109/PACIFICVIS.2016.7465270.
- 26 Weidong Huang, Seok-Hee Hong, and Peter Eades. Effects of crossing angles. In *2008 IEEE Pacific Visualization Symposium*, pages 41–46, 2008. doi:10.1109/PACIFICVIS.2008.4475457.
- 27 Wolfram Research, Inc. Mathematica, Version 14.0. Champaign, IL, 2024. URL: <https://www.wolfram.com/mathematica>.
- 28 Goos Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996. doi:10.1007/BF02086606.
- 29 Philipp Kindermann, Stephen Kobourov, Maarten Löffler, Martin Nöllenburg, André Schulz, and Birgit Vogtenhuber. Lombardi drawings of knots and links. In *Graph Drawing and Network Visualization (GD)*, pages 113–126. Springer, 2017. doi:10.1007/978-3-319-73915-1\_10.
- 30 Michael E. Mortenson. *Mathematics for Computer Graphics Applications*. Industrial Press Inc., 1999.
- 31 Walter Schnyder. Embedding planar graphs on the grid. In *1st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 138–148, 1990. URL: <http://dl.acm.org/citation.cfm?id=320176.320191>.
- 32 Yusuke Suzuki. Re-embeddings of maximum 1-planar graphs. *SIAM Journal on Discrete Mathematics*, 24(4):1527–1540, 2010. doi:10.1137/090746835.
- 33 George B. Thomas and Ross L. Finney. *Calculus and Analytic Geometry*. Addison-Wesley, 9th ed. edition, 1996.
- 34 Csaba D. Tóth. On RAC drawings of graphs with two bends per edge. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization (GD)*, pages 69–77. Springer, 2023. doi:10.1007/978-3-031-49272-3\_5.
- 35 William T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, s3-13(1):743–767, 1963. doi:10.1112/plms/s3-13.1.743.
- 36 Kai Xu, Chris Rooney, Peter Passmore, Dong-Han Ham, and Phong H. Nguyen. A user study on curved edges in graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2449–2456, 2012. doi:10.1109/TVCG.2012.189.