Closing Differentiable Piecewise Circular Curves

dino

Abstract

We study differentiable piecewise circular curves in the plane (which we will call *circle curves*) and determine the conditions which allow these curves to be *closed* with one last circular arc while keeping the differentiability of the curve.

Index

1 Introduction

The name *differentiable piecewise circular curve* is a complicated mouthful for a very simple concept. Constructing these curves is an exercise in high-school technical drawing classes about tangency. I remember doing these fun exercises where we'd find the circles that make up a curve to complete a picture in a sort of over-complicated connect-the-dots game.

You would have this set of points in the plane in a sequence and you would have to find the center of the circular arc that joins a point with the next in such a way that this arc continues the previous one smoothly, without making a corner. You would go on until all the points are connected by a "smooth" curve made of arcs.

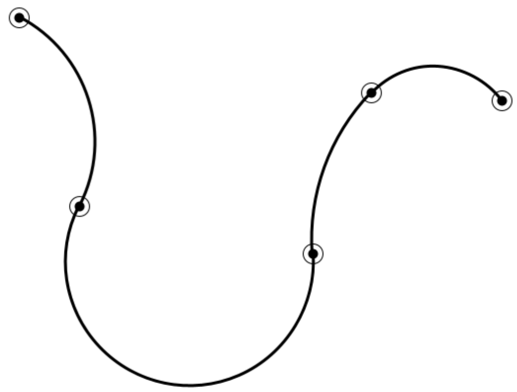

Figure 1: Five points "smoothly joined" by four circular arcs

However, maybe you would be tempted to close this curve, finding one more arc that joins the last point with the first one, only to find that this extra arc doesn't "smoothly" join with the first one.

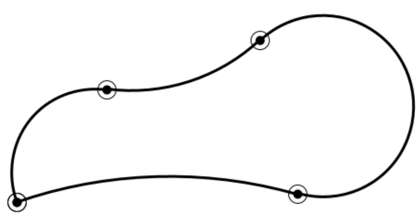

Figure 2: A curve that doesn't close "nicely"

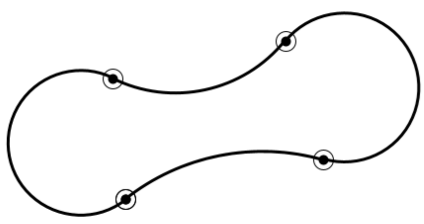

Figure 3: A curve that closes "nicely"

But you think that this *could* be possible. Maybe not for these points, but there definitely are sets of points for which it is possible. The obvious one is any number of points on a circle, the circle itself being the curve that joins them. So there are sets that can be "smoothly closed" and others that can't? What difference is there between these sets?

2 Basic concepts

To start, let's define what these curves are, though these are intuitively very simple concepts.

Definition 1: Let $t_1, \dots, t_n \in [0,1]$ be such that $t_1 = 0 < t_2 < \dots < t_n = 1$ ($t_1, \dots, t_n$ are a partition of $[0,1]$). We say that the plane curve $\gamma\colon [0,1] \to \mathbb{R}^2$ is a *piecewise circular* curve passing through the n points $p_i := \gamma(t_i) \in \mathbb{R}^2$ if the restrictions $\gamma|_{[t_i,t_{i+1}]}$ are circular arcs.

Because circular arcs are differentiable, we just need to check differentiability at the points t_i to determine if γ is differentiable. In the introduction, we used the word "smooth" in quotations to refer to differentiability in these points, though *smooth* often refers to infinitely-differentiable curves. However, for a piecewise circular curve to be $\mathcal{C}^\infty$ (actually just $\mathcal{C}^2$), all the circular arcs would need to be arcs of the same circle, so only circles are *smooth* piecewise circular curves.

Let $\gamma_i := \gamma|_{[t_i,t_{i+1}]}, i = 1, \dots, n-1$ be the circular arcs, which are differentiable. As we said, for γ to be differentiable we just need

$$\lim_{t \to t_{i+1}^-} \gamma'(t) = \gamma_i'(t_{i+1}) = \gamma_{i+1}'(t_{i+1}) = \lim_{t \to t_{i+1}^+} \gamma'(t) \tag{1}$$

This is just fancy talk to say that each circular arc is tangent to the next one (and the previous one). Figure 1 is an example of a differentiable piecewise simple curve.

Proposition 1: Once we've fixed the points $p_1, \dots, p_n$, the differentiable piecewise circular curve passing through them is uniquely characterized (save reparametrization) by the direction[1] of the vector $\gamma'(t_i)$ for any $i = 1, \dots, n$.

Proof: Consider γ_i (unless if $i = n$, in which case consider this proof for the opposite curve $\sim\gamma$ that passes through the point in the reverse order). Since this is a circular arc, it is uniquely determined by $\gamma_i(t_i) = p_i$, $\gamma_i(t_{i+1}) = p_{i+1}$ and the tangent direction $\gamma_i'(t_i) = \gamma'(t_i)$. This circular arc determines a tangent direction at the next (and previous) arc, so all arcs are uniquely determined. □

In particular, γ is uniquely determined by the direction of $\gamma'(0)$, so we will write $\gamma := C(\boldsymbol{t}; p_1, \dots, p_n)$ for the differentiable piecewise circular curve that passes through $p_1, \dots, p_n$ and has $\gamma'(0)$ in the same direction as $\boldsymbol{t}$. Differentiable piecewise circular curve is a very long name, so we will call these curves *circle curves*.

We could have proved Proposition 1 by construction. This would be the same process as those technical drawing exercises. We would find the centers of the circular arcs considering that γ_i passes through p_i and p_{i+1}. Then the center of γ_i sits at the bisection of the segment $\overline{p_i p_{i+1}}$. The center would also sit at the line passing through p_i and perpendicular to $\gamma_i'(t_i)$. Then the center sits at the intersection of these two lines and γ_i is uniquely determined. We would continue to find the rest of the centers using the same method, which requires only ruler and compass.

[1]We won't consider opposite vectors to have the same direction. We'll say the vectors a and b have

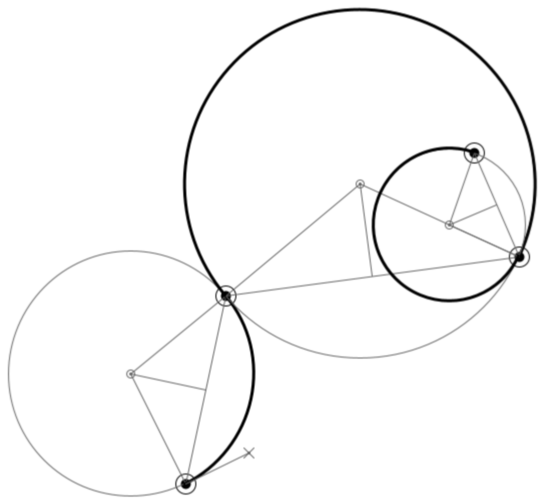

Figure 4: Circle curve showing the complete circles of each circular arc, including the bisecting and perpendicular lines. The vector t is shown with a cross.

However, this construction spells a little problem with our definition: the case where $\gamma_i'(t_i)$ has the same or opposite direction as $p_{i+1} - p_i$. In the construction, this means that the two lines are parallel and there is no intersection for this circle's center to be placed at. The solution to this is to consider straight lines to be circular arcs. These circular arcs would have "infinite radius" and their center would be "at infinity".

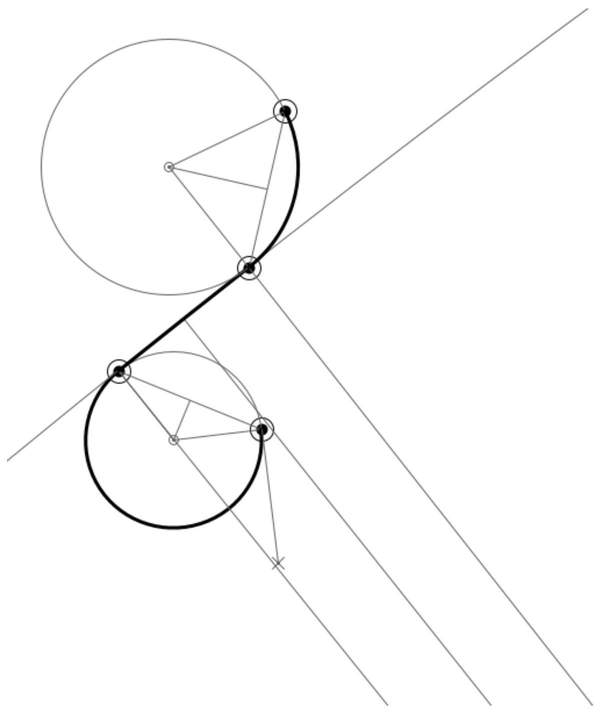

Figure 5: Circle curve with a straight segment. Notice how the bisecting and perpendicular lines are parallel, resulting in a straight line instead of a proper circular arc.

---

the same direction if $a = kb$ for some scalar $k \geq 0$.

3 Closing the circle curve

We now want to consider closed circle curves: Circle curves that have $\gamma(0) = \gamma(1)$, or equivalently, $p_1 = p_n$. As we mentioned in the introduction, we are interested in the cases where this curve closes "nicely", which is when $\gamma'(0)$ and $\gamma'(1)$ have the same direction:

Definition 2: The closed circle curve $\overline{C}(\boldsymbol{t}; p_1, \ldots, p_n) := C(\boldsymbol{t}; p_1, \ldots, p_n, p_1)$ *closes nicely* if $\gamma'(0)$ has the same direction as $\gamma'(1)$.

Playing around with different points $p_1, \ldots, p_n$ and different vectors $\boldsymbol{t}$, we can notice an interesting distinction:

- If n is odd, we seem to always be capable of finding a direction $\boldsymbol{t}$ for which γ closes nicely. Furthermore, there are only two of these directions: $\boldsymbol{t}$ and $-\boldsymbol{t}$. (Figure 6)
- If n is even, it seems like either the curve doesn't close nicely for any $\boldsymbol{t} \in S^1$, or it does close nicely for every $\boldsymbol{t} \in S^1$, which depends on the chosen points. (Figures 7 and 8)

Furthermore, it seems that for an even value of n, the angle that $\gamma'(0)$ and $\gamma'(1)$ make is kept constant as $\boldsymbol{t}$ changes. For odd values of n, it seems like $\gamma'(0)$ and $\gamma'(1)$ are reflections of each other across a specific direction. We are gonna try to prove that this is true.

Definition 3: For a closed circle curve $\overline{C}(\boldsymbol{t}; p_0, \ldots, p_{n-1}) = C(\boldsymbol{t}; p_0, \ldots, p_{n-1}, p_0)$ (notice the change in the indices), we're gonna write $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_n \in S^1$ as:

$$\boldsymbol{t}_i := \frac{\gamma_i'(t_i)}{\|\gamma_i'(t_i)\|}, \quad i = 0, \ldots, n \tag{2}$$

With this notation, the curve closing nicely is equivalent to $\boldsymbol{t}_0 = \boldsymbol{t}_n$. The behaviour of these unit vectors is key to our proof.

Proposition 2: The vectors $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_n$ of a closed circle curve $\overline{C}(\boldsymbol{t}; p_0, \ldots, p_{n-1})$ satisfy that

$$\boldsymbol{t}_{i+1} = F_i \boldsymbol{t}_i, \quad i = 0, \ldots, n-1 \tag{3}$$

where F_i denotes the matrix for a reflection (a flip) across the direction $p_{i+1} - p_i$.

Proof: We can focus solely on the arc γ_i, which has $\gamma_i'(t_i)$ in the direction $\boldsymbol{t}_i$ and $\gamma_i'(t_{i+1})$ in the direction $\boldsymbol{t}_{i+1}$. Consider, without loss of generality, that this circular arc has radius 1; that the center of this circular arc is at the origin; and that the line $\overline{p_i p_{i+1}}$ is parallel to the horizontal axis in such a way that we can write

$$p_i = (-\sin\theta, \cos\theta), \quad p_{i+1} = (\sin\theta, \cos\theta)$$

where $\theta \in (0, \pi)$. Through a strictly increasing reparametrization, the arc γ_i can be thought of as equivalent to the curve

$$\begin{aligned} \delta_i \colon [-\theta, \theta] &\longrightarrow \mathbb{R}^2 \\ \varphi &\longmapsto (\sin\varphi, \cos\varphi) \end{aligned}$$

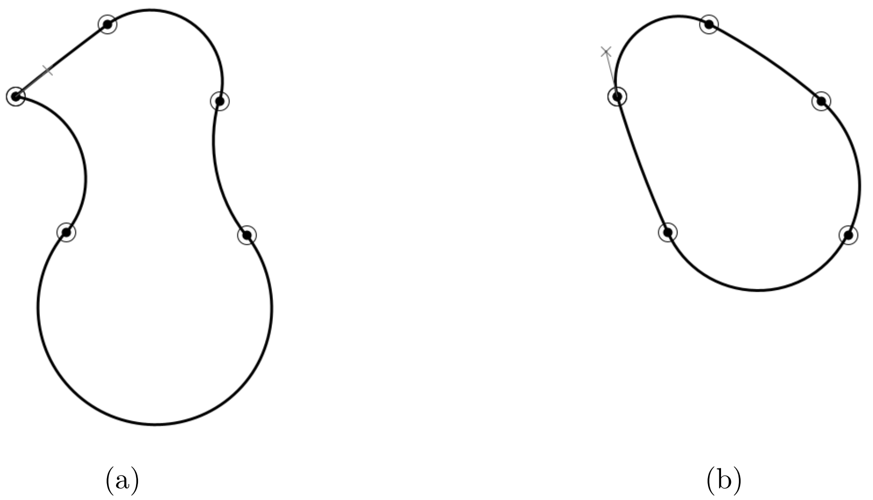

Figure 6: Closed curves of the same set of odd points. (b) closes nicely while (a) doesn't.

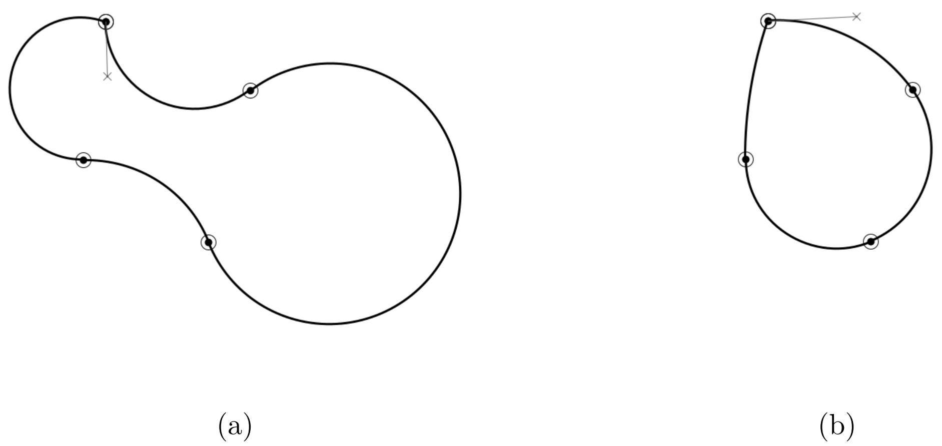

Figure 7: Closed curves of the same set of even points. Neither (a) or (b) close nicely.

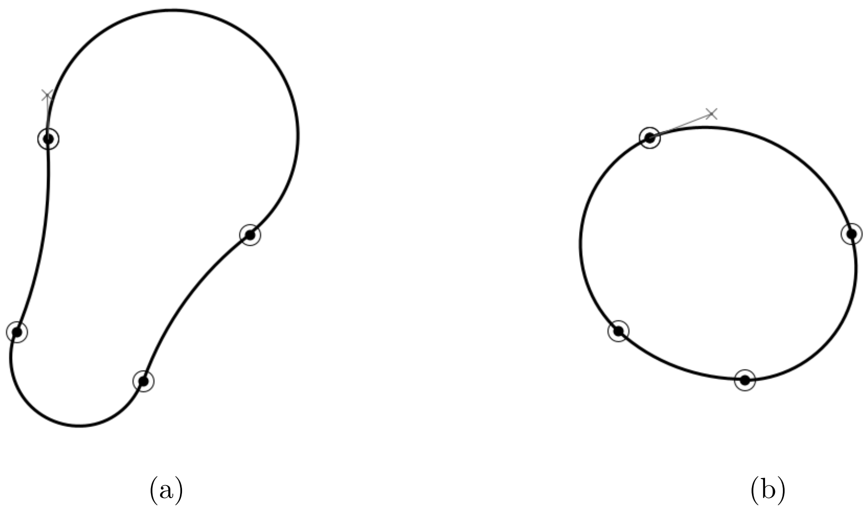

Figure 8: Closed curves of the same set of even points. Both (a) and (b) close nicely.

Then, $\gamma_i'(t_i)$ is in the same direction as $\delta_i'(-\theta) = \boldsymbol{t}_i$; and $\gamma_i'(t_{i+1})$ is in the same direction as $\delta_i'(\theta) = \boldsymbol{t}_{i+1}$. We have that

$$\delta_i'(\varphi) = (\cos\varphi, -\sin\varphi)$$

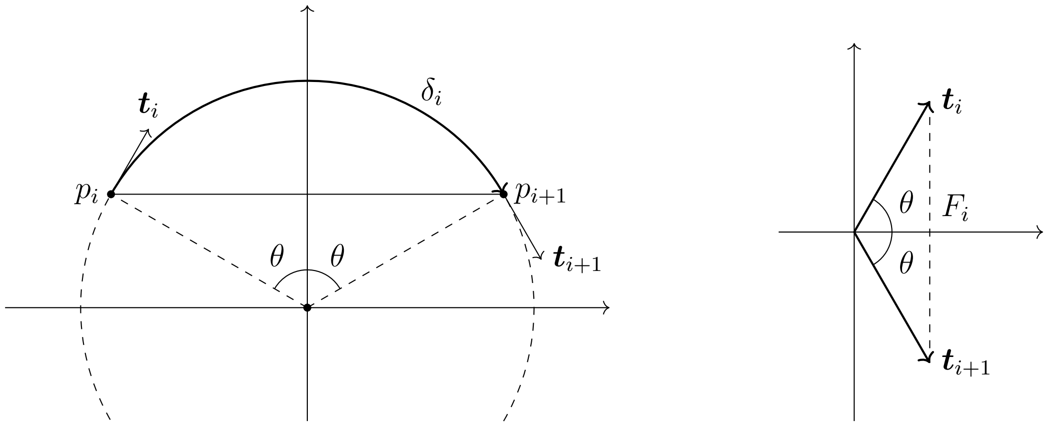

Figure 9: Diagrams showing the reflection from $\boldsymbol{t}_i$ to $\boldsymbol{t}_{i+1}$.

Therefore,

$$\left.\begin{aligned} \boldsymbol{t}_i &= \delta_i'(-\theta) = (\cos\theta, \sin\theta) \\ \boldsymbol{t}_{i+1} &= \delta_i'(\theta) = (\cos\theta, -\sin\theta) \end{aligned}\right\} \Longrightarrow \boldsymbol{t}_i = F_i \boldsymbol{t}_{i+1}$$

where F_i is the reflection across the horizontal axis. Undoing this change of coordinates we've made, we obtain that $\boldsymbol{t}_{i+1}$ is the reflection of $\boldsymbol{t}_i$ across the direction of the line $\overline{p_i p_{i+1}}$. □

Then, we can write

$$\boldsymbol{t}_n = F_{n-1}F_{n-2}\cdots F_1 F_0 \boldsymbol{t}_0 \tag{4}$$

$\boldsymbol{t}_n$ is the result of reflecting $\boldsymbol{t}_0$ across the directions of all the sides of the polygon made up by $p_1, \ldots, p_n$. Note that this transformation is independent of the choice of $\boldsymbol{t}$ and only depends on $p_1, \ldots, p_n$.

In order to continue, we need some basic properties of reflections and orthogonal matrices. A reflection is an orthogonal linear transformation. That is to say that its matrix A satisfies that $AA^T = A^TA = I$ (its columns are orthogonal to each other and have norm 1, and the same thing happens with its rows). From this we can deduce that $\det A = \pm 1$. We can also prove that the product of orthogonal matrices is an orthogonal matrix:

$$A_1A_2\cdots A_n(A_1A_2\cdots A_n)^T = A_1A_2\cdots A_nA_n^T\cdots A_2^TA_1^T = I \tag{5}$$

In the plane, there are only two kinds of orthogonal transformations: Reflections and rotations. This means that an orthogonal transformation in the plane is either a reflection or a rotation (the identity transformation can be thought as a 0° rotation) and we can tell which one a transformation is only through its determinant. An orthogonal transformation in the plane with $\det A = 1$ is a rotation, while if $\det A = -1$, the transformation is a reflection.

Therefore, the matrix $T := F_{n-1}F_{n-2}\cdots F_0$ from equation (5) is an orthogonal matrix too, with determinant

$$\det T = \det F_{n-1}\det F_{n-2}\cdots\det F_0 = \overbrace{(-1)(-1)\cdots(-1)}^{n} = (-1)^n \tag{6}$$

This is where the distinction between n even and n odd comes from. If n is even, $\det T = 1$, so T is a rotation. This confirms the behaviour we saw before, where the angle between $\boldsymbol{t}_0$ and $\boldsymbol{t}_n$ was always the same. Therefore, a circle curve with an even amount of points will only close nicely if $T = I$, since that's the only case where a rotation has 1 as an eigenvalue.

If n is odd, then $\det T = -1$ and T is a reflection. Again, this is what we saw earlier. Any reflection has 1 as an eigenvalue and its corresponding eigenvector is the direction of the reflection, so if $\boldsymbol{t}$ is in this direction (or $-\boldsymbol{t}$), then $T\boldsymbol{t} = \boldsymbol{t}$ and the curve closes nicely.

This explains and proves the behaviour we described. Now we'll try to find the condition that makes the curve close nicely if n is even; and the vector $\boldsymbol{t}$ that closes the curve nicely if n is odd.

From now on, we will study angles. These angles will be oriented. "A rotation with angle $-\theta$" will rotate vectors with an angle of the same amplitude as θ, but in the opposite direction.

Proposition 3: F_iF_{i-1} is a rotation and the angle of this rotation is twice the opposite of the angle $\alpha_i := \angle p_{i-1}p_ip_{i+1}$. The direction of this rotation is the opposite of the turn made by this corner.

Proof: As we've seen, we know that F_iF_{i-1} is a rotation. Then, we just need to find the image of one (non-zero) vector to know its angle. Consider $v := p_i - p_{i-1}$, which is in the direction of the reflection F_{i-1}. This way, $F_{i-1}v = v$ and

$$F_iF_{i-1}v = F_iv$$

As we see in Figure 10, F_iv is the same as rotating v $-2\alpha_i$ (following $p_{i-1}p_ip_{i+1}$ in Figure 10, this corner is a clockwise turning, so the rotation is counter-clockwise). Because F_iF_{i-1} is a rotation and $F_iF_{i-1}v = F_iv$, we know F_iF_{i-1} is a rotation of $-2\alpha_i$. □

We'll then write $F_iF_{i-1} =: R_i$. We have then that:

$$\boldsymbol{t}_n = \begin{cases} R_{n-1}\cdots R_3R_1\boldsymbol{t}_0 & \text{if } n \text{ is even} \\ F_{n-1}R_{n-2}\cdots R_3R_1\boldsymbol{t}_0 & \text{if } n \text{ is odd} \end{cases} \tag{7}$$

Now, the product of rotations is a rotation too, and its angle if the sum of all the rotations' angles. Therefore, in the even case, we can write the following condition for nice closure:

$$2\alpha_1 + 2\alpha_3 + \cdots + 2\alpha_{n-1} = 2k\pi, \quad k \in \mathbb{Z}$$

Equivalently,

$$\alpha_1 + \alpha_3 + \cdots + \alpha_{n-1} = k\pi, \quad k \in \mathbb{Z} \tag{8}$$

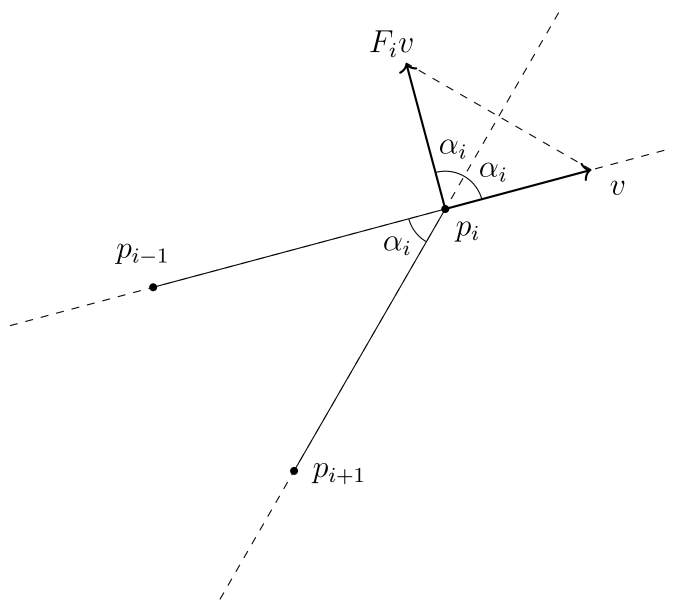

Figure 10: Effect of F_i on v. This reflection on this vector is equivalent to a counter-clockwise rotation of $2\alpha_i$.

When this happens, the tower of rotations from Equation (7) is a rotation of $-2k\pi$, so it is the identity, $\boldsymbol{t}_n = \boldsymbol{t}_0$ and the curve closes nicely.

If n is odd, the tower of rotations is a rotation R of angle $\beta := -2\alpha_1 - 2\alpha_3 - \cdots - 2\alpha_{n-2}$ and $\boldsymbol{t}_n = F_{n-1}R\boldsymbol{t}_0$. Now, because $\det(F_{n-1}R) = -1$, this is a reflection across some direction. If we find a vector v with $F_{n-1}Rv = v$, this will be that direction.

Proposition 4: $F_{n-1}R$ is a reflection across the direction of $p_0 - p_{n-1}$ rotated by $-\frac{\beta}{2}$.

Proof: We already reasoned that this transformation is a reflection. No we just have to see that the vector from the statement v is the direction of the reflection. This can be easily seen in Figure 11.

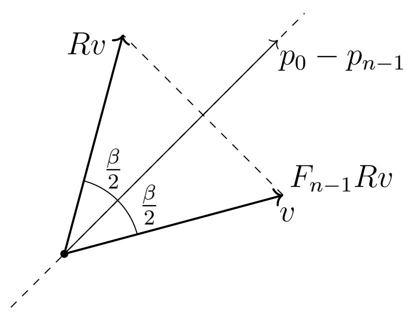

Figure 11: Effect of $F_{n-1}R$ on v.

Because $F_{n-1}Rv = v$, v is the direction of this reflection. □

We have that

$$-\frac{\beta}{2} = \alpha_1 + \alpha_3 + \cdots + \alpha_{n-2}$$

So the direction $\boldsymbol{t}$ (or $-\boldsymbol{t}$) that makes the curve close nicely for n odd is $p_0 - p_{n-1}$ rotated by $\alpha_1 + \alpha_3 + \cdots + \alpha_{n-2}$.

Results: The closed circle curve $\overline{C}(\boldsymbol{t}; p_0, \ldots, p_{n-1})$ closes nicely if:

- n is even and the sum $\alpha_1 + \alpha_3 + \cdots + \alpha_{n-1}$ is a multiple of π, in which case it will close nicely for any $\boldsymbol{t} \in S^1$.
- n is odd and either $\boldsymbol{t}$ or $-\boldsymbol{t}$ are at an angle $\alpha_1 + \alpha_3 + \cdots + \alpha_{n-2}$ with the line $\overline{p_{n-1}p_0}$.

Examples: Consider Figure 12, which represents the curve $\overline{C}(\boldsymbol{t}; p_0, p_1, p_2, p_3, p_4)$.

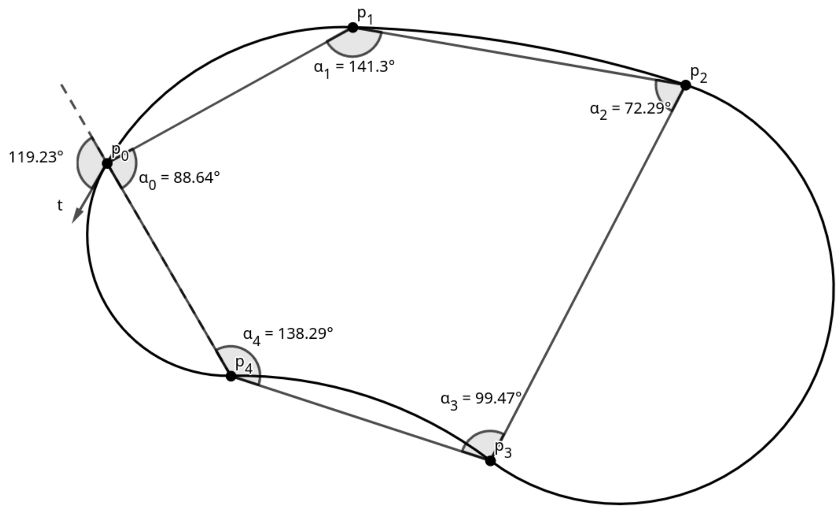

Figure 12

The vector $\boldsymbol{t}$ in the figure is actually $-\boldsymbol{t}$, but the result should be the same. If we check α_1 and α_3, these angles add up to $141.3° + 99.47° = 240.77°$. If we follow the points p_i, these angles are clockwise, and so the angle measured in a clockwise fashion between $\overline{p_4p_0}$ and $\boldsymbol{t}$ should be 244.77°. In Figure 12, this angle is 119.23°, but counter-clockwise, which is the same thing. So our result tells us that the curve closes nicely.

Now, consider Figure 13 for a curve with six points. This curve has $\alpha_1 + \alpha_3 + \alpha_5 = 116.28° + 84.18° + 96.66° = 299.34°$, which is not a multiple of π, so it doesn't close nicely. Not only does this particular curve not close nicely, but neither will any curve with these same points.

Consider instead the curve in Figure 14. It has $\alpha_1 + \alpha_3 + \alpha_5 = 96.1° + 119.24° + 144.66° = 360°$, so it does close nicely and it will still close nicely for any $\boldsymbol{t} \in S^1$.

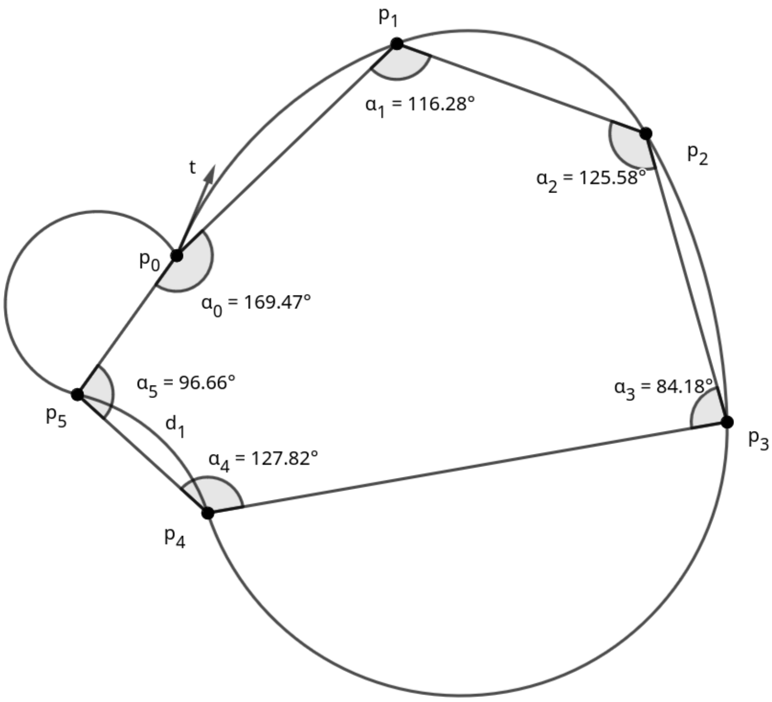

Figure 13

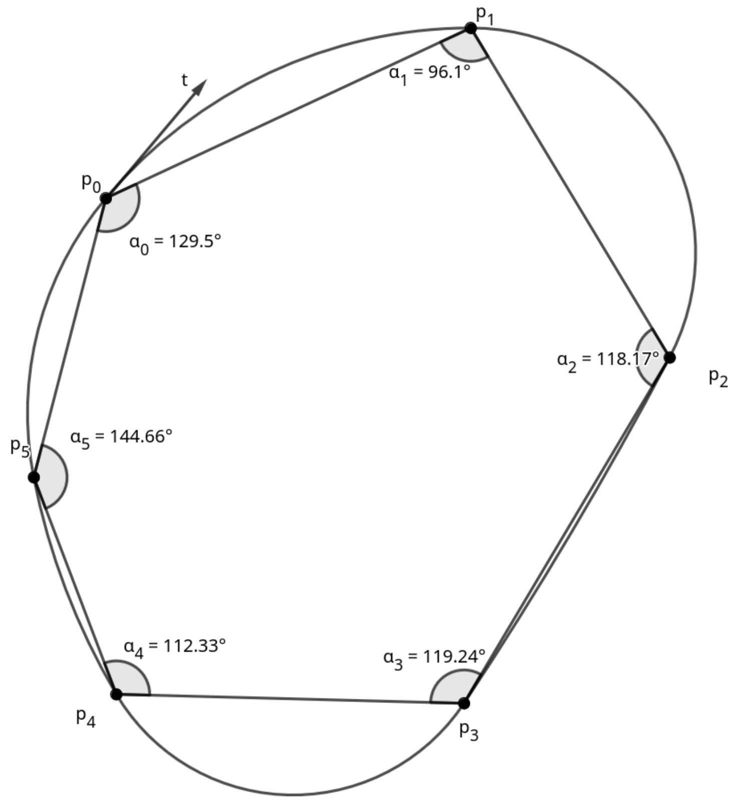

Figure 14

4 Appendix

Figures 12, 13 and 14 were made using Geogebra. Figures 1 to 8 were made using the following Processing code:

```
PVector[] points;
PVector[] centers;
float[] radii;
boolean dirs[];
PVector start;

int N = 5;

boolean close = true;

int target = -2;

boolean showGuidelines = true;

void setup() {
  size(1000, 1000);
  ellipseMode(RADIUS);

  if (close) N++;

  points = new PVector[N];
  centers = new PVector[N-1];
  radii = new float[N-1];
  dirs = new boolean[N-1];

  for (int i = 0; i < N - (close?1:0); i++) points[i] = new PVector(width/2
    + (i-N/2)*100, height/2 + 0);
  for (int i = 0; i < N-1; i++) {
    centers[i] = new PVector();
    radii[i] = 0;
    dirs[i] = false;
  }

  if (close) points[N-1] = points[0];

  start = points[0].copy().add(20,20);
}

void draw() {
  background(255);

  calcCenters();

  noFill();
  for (int i = 0; i < N; i++) {
    PVector p = points[i];

    if (i == target) stroke(255, 0, 0);
    else stroke(0);

    strokeWeight(3);
    point(p.x, p.y);
    strokeWeight(1);
```

```
    circle(p.x, p.y, 5);

    if (i > 0) {
      PVector pp = points[i-1];

      PVector c = centers[i-1];
      if (showGuidelines) {
        stroke(255,0,0);
        line(p.x, p.y, pp.x, pp.y);
        strokeWeight(3);
        point(c.x, c.y);
        strokeWeight(1);
        circle(c.x, c.y, 4);

        line(pp.x,pp.y,c.x,c.y);
        line(p.x,p.y,c.x,c.y);
      }

      //circle(c.x,c.y,radii[i-1]);
      stroke(0);
      strokeWeight(2);
      if (radii[i-1] >= 0) myArc(c.x,c.y,radii[i-1],PVector.sub(p,c).
   heading(),PVector.sub(pp,c).heading(),dirs[i-1]);
      else line(p.x,p.y,pp.x,pp.y);
    }
  }

  stroke(0, 100, 100);
  strokeWeight(3);
  point(start.x, start.y);
  strokeWeight(1);
  circle(start.x, start.y, 5);

  line(points[0].x, points[0].y, start.x, start.y);

  if (target > -1) {
    points[target].x = mouseX;
    points[target].y = mouseY;
  } else if (target == -1) {
    start.x = mouseX;
    start.y = mouseY;
  }

  if (close) points[N-1] = points[0];
}

void mousePressed() {
  for (int i = 0; i < N; i++) {
    PVector p = points[i];
    if (dist(p.x,p.y,mouseX,mouseY) <= 10) {
      target = i;
      return;
    }
  }
  if (dist(start.x,start.y,mouseX,mouseY) <= 10) {
    target = -1;
    return;
  }
```

```
  target = -2;
}

void mouseReleased() {
  target = -2;
}

void keyPressed() {
  if (keyCode == 'D') showGuidelines = !showGuidelines;
}

void calcCenters() {
  PVector v = PVector.sub(start,points[0]).normalize();
  for (int i = 0; i < N-1; i++) {
    PVector dif = PVector.sub(points[i+1],points[i]);
    float d = dif.mag()*dif.mag()/2;
    float det = dif.x*v.y - dif.y*v.x;

    if (det != 0) {
      centers[i].x = d*v.y/det;
      centers[i].y = -d*v.x/det;
      dirs[i] = det > 0;
      radii[i] = centers[i].mag();

      centers[i].add(points[i]);
      v = PVector.sub(points[i+1],centers[i]).normalize().rotate(HALF_PI*(
    dirs[i] ? -1 : 1));
    } else {
      centers[i] = dif.copy().mult(0.5).add(points[i]);
      radii[i] = -1;
    }
  }
}

void myArc(float cx, float cy, float r, float a, float b, boolean dir) {
  float minAngle = dir ? a : b;
  float maxAngle = dir ? b : a;

  if (minAngle > maxAngle) maxAngle += TWO_PI;

  arc(cx, cy, r, r, minAngle, maxAngle);
}

int sign(float val) {
  if (val == 0) return 0;
  if (val > 0) return 1;
  return -1;
}
```