POINT X-RAYS OF A CONVEX BODY FROM AN INTERIOR AND AN EXTERIOR SOURCE

CHRIS PRYBY

Advisor: Dr. Don Solmon
Oregon State University

Abstract. The focus of this study is to determine what information can be obtained about a convex body in the plane given two point X-ray functions for the body; in particular, whether a convex body is uniquely determined by two such X-rays. We shall specifically investigate the case in which one X-ray is taken from a source exterior to the body and the other is taken from an interior source. We shall derive formulae for the tangent lines and curvature of the body's boundary at its x-intercepts, and we shall use the Stable Manifold Theorem to design an algorithm that attempts to construct bodies with identical X-ray functions at both sources. Finally, we shall consider the possibility of forming an analytic argument proving the existence of exactly two bodies whose X-rays at each of the two points are indeed equal.

1. Introduction

Geometric tomography is the branch of mathematics concerned with the reconstruction of geometric objects from the knowledge of their sections or projections. Following the development of medical radiology, it became natural to look at mathematical objects modeling X-rays to study their properties, particularly whether they allow for accurate reconstructions of geometric objects like those that might be found in the human body.

We utilize a mathematical abstraction of X-rays in this study; given a point P and a convex body K in $\mathbb{R}^2$, we consider the lengths of the intersections of K with every line passing through P. This differs from the model of so-called directed X-rays in that we do not know on which "side" of P the object lies (see Figure 1).

Gardner [5] lists two cases of point X-rays of a convex body that are open problems of mathematical interest. There has already been research conducted on the first: X-rays from two exterior sources whose connecting line intersects the body [8]. The second—X-rays from one exterior source and one interior source—will be the topic of this paper. We will be adapting many of the methods used by researchers of other problems of X-ray tomography for our use in this particular case.

Date: August 17, 2007.

This work was done during the Summer 2007 REU program in mathematics at Oregon State University. I extend much-needed gratitude for the insight and assistance of Dr. Don Solmon and Dr. David Finch of Oregon State University, my colleagues Kirsten Aagesen and David Steinberg in the REU program, and Anthony Coulter of the Georgia Institute of Technology.

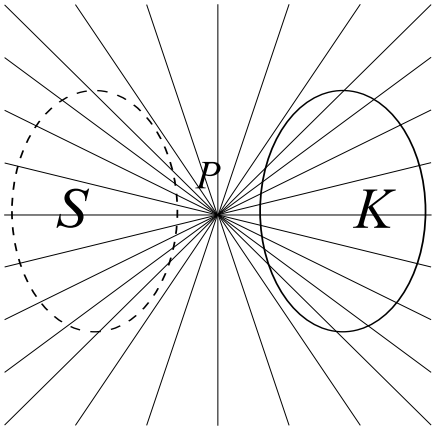

FIGURE 1. The point X-ray functions from source P for the bodies K and S are identical, but the directed X-ray function from P for S is offset by π from that of K.

In particular, we will attempt to determine if there is more than one body which generates the same point X-ray data from two sources. That is, given a convex body, K, and its X-rays, X_P and X_Q, from the points P and Q, we seek to find (or prove the nonexistence of) a second convex body, S, such that the X-rays of S from P and Q are precisely X_P and X_Q. We will attack this problem by first finding the x-intercepts, or "basepoints," of S (should it exist), approximating S by its tangent lines, and then using the Stable Manifold Theorem to write an algorithm that will construct an approximation of S. It is our hope that, through numerical computations, we will gather evidence that the existence of S is plausible as well as find directions in which to pursue an analytic proof of its existence.

1.1. **Definitions.** We shall now introduce some definitions that will be important for later discussion.

Definition 1.1. *A* ***convex body****, K, is a compact, convex subset of $\mathbb{R}^2$ with nonempty interior; we denote its boundary ∂K.*

Definition 1.2. *Given a point, $P \in \mathbb{R}^2$, and a convex body, K, the* ***point X-ray function****, or simply* ***X-ray****, of K from P is a function, $X_P(\varphi)$, $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2})$, such that $X_P(\varphi)$ is the length of $\ell \cap K$, where ℓ is the line passing through P at angle φ.*

Unless otherwise noted, all angles mentioned in this paper will be oriented counterclockwise from the positive x-axis.

Definition 1.3. *For a point, P, exterior to a convex body, K, the* ***nearside point*** *at angle φ is the point, $r_{P,\varphi} \in \ell \cap K$, closest to P, where ℓ is again the line passing through P at angle φ. Similarly, the* ***farside point*** *at angle φ is the point, $R_{P,\varphi} \in \ell \cap K$, farthest from P.*

We also define functions of φ called the ***nearside*** *and* ***farside functions*** *from P. The nearside function, $r_P(\varphi)$, is the distance from P to $r_{P,\varphi}$, and the farside function, $R_P(\varphi)$, is the distance from P to $R_{P,\varphi}$. Thus, the X-ray function from P satisfies the identity $X_P(\varphi) = R_P(\varphi) - r_P(\varphi)$.*

Definition 1.4. *For a point Q in the interior of K, we dispense with the notions of "near-side" and "farside" and instead define $R_Q(\varphi)$, with $\varphi \in [-\pi, \pi)$, as the distance from Q to the point farthest from Q in $\rho \cap K$, where ρ is the ray originating from Q at angle of inclination φ. In this case, we see that the point X-ray function from Q satisfies the identity $X_Q(\varphi) = R_Q(\varphi) + R_Q(\varphi + \pi)$, with φ in this case within $[-\frac{\pi}{2}, \frac{\pi}{2})$.*

A short remark on our choice of notation: despite abandoning the terminology "farside," we use a capitalized 'R' for the function R_Q. We do this to maintain consistency with R_P; notice that R_P and R_Q are always concave towards their sources, while r_P is concave away from its source.

Definition 1.5. *The **basepoints** of a convex body, K, given the points $P \notin K$ and $Q \in \text{int } K$, are the points of intersection in $\ell \cap \partial K$, where ℓ is the line passing through P and Q.*

The distances from P to the basepoints are $r_P(0)$ and $R_P(0)$, and the distances to the basepoints from Q are $R_Q(0)$ and $R_Q(\pi)$. From a result of Falconer [4], we are able to establish the locations of the basepoints given P, Q, and K; thus, we are able to determine the values of these four distances.

Definition 1.6. *Given a $\mathcal{C}^2$ function f in polar coordinates, the **signed curvature** of f, denoted κf, is given by the equation*

$$\kappa f = \frac{f^2 + 2(f')^2 - ff''}{(f^2 + (f')^2)^{\frac{3}{2}}}.$$

κf will be positive at θ when the graph of f is concave towards the origin at $(\theta, f(\theta))$, and it will be likewise negative when the graph of f is concave away from the origin.

Note that curvature is independent of a curve's parametrization; thus, changing the location of the origin in measuring a curve by polar coordinates will not change the signed curvature at any point (except perhaps by a factor of -1 to account for changes in concavity). We will use this property when measuring the curvature of a convex body from our two different point sources.

Further, to simplify computations of curvature, we introduce the following:

Definition 1.7. *The **curvature operator** of f, denoted $\mathcal{K}f$, is equal to $\kappa f \cdot (f^2 + (f')^2)^{\frac{3}{2}}$.*

Incidentally, $\mathcal{K}(-f) = \mathcal{K}f$. This fact comes in handy for curvature calculations appearing later in this paper.

Definition 1.8. *If K is a convex body and P a point, consider the function $X_P(\varphi)$. We call the angles $\alpha = \sup\{\varphi \mid X_P(\varphi) > 0\}$ and $\beta = \inf\{\varphi \mid X_P(\varphi) > 0\}$ the **support angles** of X_P. Moreover, if ℓ is a line passing through P at angle of inclination α or β, we call ℓ a **support line** of K through P.*

Finally, we define a term that will be important in our investigation of a convex body's uniqueness with respect to its X-rays.

Definition 1.9. *Given a convex body, K, and points $P, Q \in \mathbb{R}^2$, a **shadow body**, S, of K is a convex body such that its point X-ray functions from P and Q match the point X-ray functions of K, respectively, from P and Q.*

Other literature in the field may call any compact planar region with X-rays equal to those of K a shadow body; however, we shall restrict our usage of the term only to *convex* objects exhibiting this property.

1.2. **An Example Convex Body.** Throughout this paper we shall work with a particular body chosen for its possession of certain "nice" properties. However, when possible, we shall try to prove properties of convex bodies in general. The example convex body is defined as the intersection of the two closed disks $(x-8)^2+y^2 \leq 9$ and $(x-12)^2+y^2 \leq 9$, with the point P located at the origin and the point Q located at $(10,0)$, the centroid of the convex body. Notice that the line passing through the two vertices of our body also contains Q. Our body's X-ray at angle 0 from both P and Q will be 2, and we can also calculate the support angles of our body with respect to P: they are $\alpha = \pm \arccos\left(2\sqrt{\frac{5}{21}}\right)$.

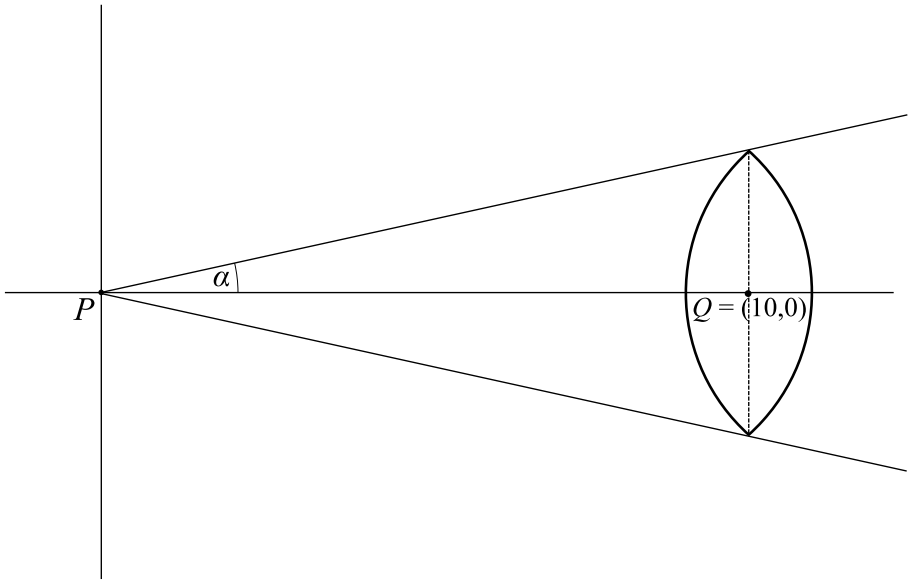

FIGURE 2. Our example convex body.

2. Falconer's Lemma and Computing the Basepoints of a Shadow Body

The ultimate goal of our study is to determine whether or not two convex bodies can exist which each have the same two X-ray functions from point sources. In other words, we may ask whether, given a convex body and its X-rays from the points P and Q, there exists a second convex body whose X-rays from P and Q are precisely the same as those of K. We begin by determining where, if it does exist, such a shadow body must be located relative to the original body. Indeed, it is possible to find the basepoints of the potential shadow body using Lemmas 3 and 4 from Falconer [4]. By the uniqueness theorems of Falconer [4] and Gardner [6], we know that there can be at most one convex body passing through these particular points on the x-axis which also satisfies the X-ray functions X_P and X_Q. Thus, finding these points will aid us greatly in our attempt to discover a shadow body.

Let K be a convex body in the plane, and let P and Q be points in the plane such that the line passing through P and Q intersects the interior of K. Orient the plane so that P lies on the origin and so that Q lies to the right of P on the x-axis. Denote by N and F the points on ∂K that intersect the x-axis, with N nearer to P than F.

Denote by p_1, q_1, p_2, and q_2 the signed distances from P to F, from P to N, from Q to F, and from Q to N, respectively (see Figure 3). Let A denote the distance from P to Q (so A is effectively $p_1 - p_2$). Also let $f(t) = t \log|t| - (t-m)\log|t-m|$, where m is the distance from N to F (note that $m = p_1 - q_1 = p_2 - q_2$). Falconer's Lemma tells us that

$$\lim_{\varepsilon \to 0} \frac{1}{2}\left[\int_{\pi-\varepsilon}^{\varepsilon} \frac{X_Q(\varphi)}{\sin(\varphi)} d\varphi - \int_{\pi-\varepsilon}^{\varepsilon} \frac{X_P(\varphi)}{\sin(\varphi)} d\varphi\right] = p_1 \log|p_1| - q_1 \log|q_1| - (p_2 \log|p_2| - q_2 \log|q_2|),$$

which happens to equal $f(p_1) - f(p_2)$. We label this quantity B.

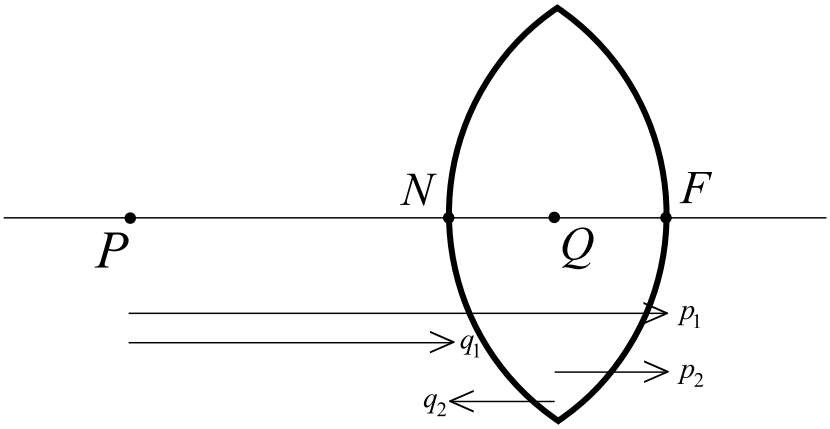

FIGURE 3. An example of a convex body with labeled distances between points. Note that the distance from Q to N will have the opposite sign of the other distances.

Now, we wish to know what other basepoints (that is, other possible locations for N and F) will give us the same value of B for fixed sources. Thus, following our example, we fix $A = 10$. We may also fix $m = 2$ since we know the length of the X-ray along the x-axis must be the same for any potential shadow body. Our question therefore reduces to finding solutions to the equation $f(p_1) - f(p_1 - 10) = B$, where $B \approx 6.60183$ in our example.

Using the `FindRoot` command of Mathematica 6.0, we are able to find that the solutions of this equation are $p_1 = 11$ (as expected) and $p_1 \approx 11.1974$ (see Figures 4 and 5). Thus, the farside basepoint of our possible shadow body is approximately $(11.1974, 0)$, and the corresponding nearside basepoint is $(9.1974, 0)$.

It is also worth noting that, if the original body is $\mathcal{C}^1$ except at two points (we shall call them "vertices") and the line connecting those vertices contains the point Q, the shadow body will have vertices at exactly the same points as those of the original body. This is true because at each vertex the X-ray functions will have discontinuous derivatives. Since the X-rays of the shadow body are the same as those of the original body, each vertex of the shadow body must therefore lie on the line connecting the two vertices of the original body

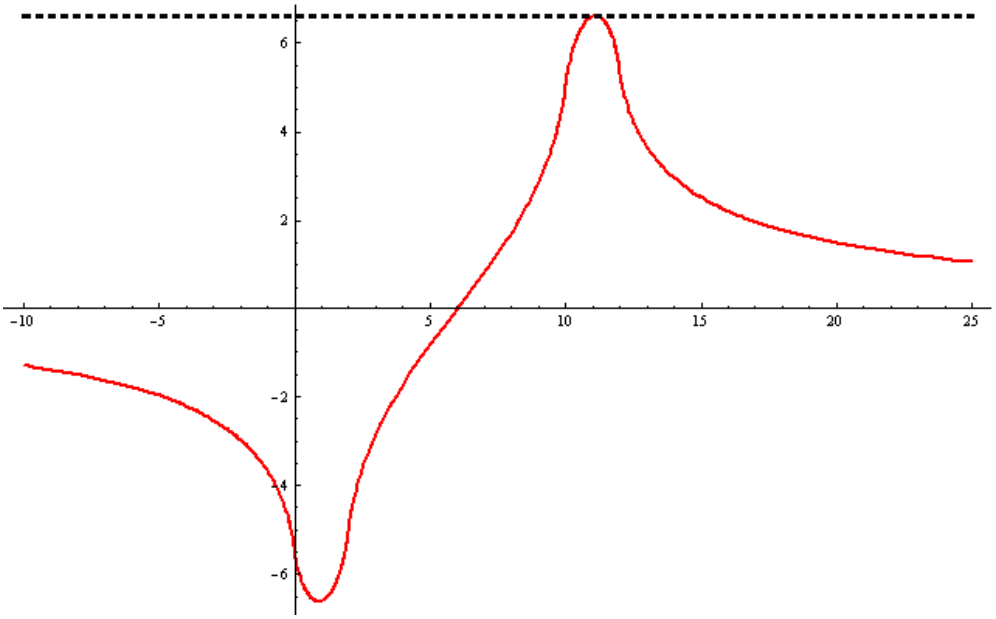

FIGURE 4. A plot of the Falconer function, $g(t) = f(t) - f(t - A)$, against $y = B$.

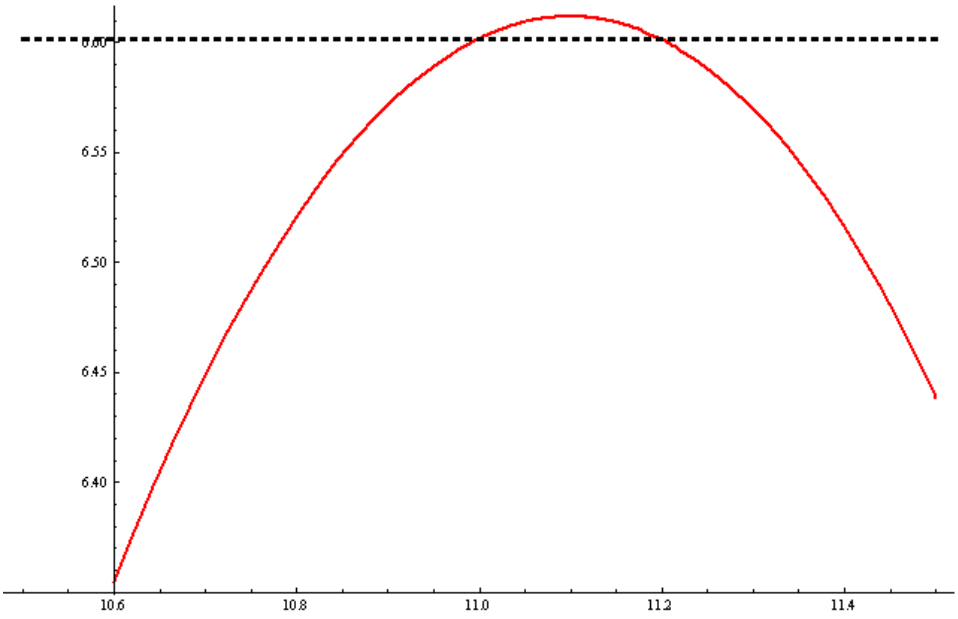

FIGURE 5. A second plot of the above, scaled to clearly show the intersection of $g(t)$ with B.

as well as a line passing through P and a vertex of the original body. The intersections of these lines are, in fact, the vertices of the original body.

3. Computing Tangent Lines at Basepoints

We will now compute the tangent lines at the basepoints of our convex body, K, using two X-rays: one from P in the exterior of K and one from Q in the interior. This work draws heavily from the processes used in §2 of [9] and §4 of [8], albeit with some variations to account for our use of an interior source instead of two exterior sources.

Though in the particular problem we will be considering it is unnecessary to explicitly recalculate the tangent lines and curvature of our convex body (as this body is given to us), we can use this same process to calculate the tangent lines of a shadow body (presuming one exists) at its basepoints. In fact, this is our motivation for calculating the tangent line formulae—it will give us an initial approximation of the shadow body which we will use to construct it.

Theorem. *Let K be a convex body ($\mathcal{C}^1$ near its basepoints), and let $P \notin K$, $Q \in \operatorname{int} K$ such that P and Q lie on the x-axis with P to the left of Q. Then the angles of inclination of the tangent lines at K's basepoints are computable. Specifically,*

$$\cot(\eta_0) = \frac{R_P(0)[R_Q(\pi)X'_P(0) - r_P(0)X'_Q(0)]}{r_P(0)[R_Q(\pi)R_P(0) + r_P(0)R_Q(0)]} - \frac{X'_P(0)}{r_P(0)} \text{ and}$$
$$\cot(\omega_0) = \frac{R_Q(\pi)X'_P(0) - r_P(0)X'_Q(0)}{R_Q(\pi)R_P(0) + r_P(0)R_Q(0)},$$

where η_0 is the angle of inclination of the tangent line at the basepoint nearer to P and ω_0 is that of the farther basepoint's tangent.

Proof. Our first goal in this proof is to find relationships among the functions r_P, R_P, and R_Q and their derivatives. Using the identities of the X-ray functions,

$$X_P(\varphi) = R_P(\varphi) - r_P(\varphi), \tag{1}$$
$$X_Q(\varphi) = R_Q(\varphi) + R_Q(\varphi + \pi), \tag{2}$$

we will write these relationships as a linear system of equations and then solve this system.

Let us begin by placing our point P at the origin and letting Q lie along the x-axis to the right of P; thus, ℓ becomes the x-axis. Per our convention, we shall use φ to denote the measure of the angle between the positive x-axis and a ray from P, and we shall denote by ψ the measure of the angle between the positive x-axis and a ray from P (thus, $\pi - \psi$ is the measure of the angle from this ray to the negative x-axis). Finally, we let $|PQ|$ represent the distance from P to Q along the x-axis.

Consider a point γ on the nearside of K from P (see Figure 6). If we consider the triangle $\triangle\gamma PQ$, the Law of Sines gives us the following relation:

$$\frac{R_Q(\pi - \psi)}{\sin(\varphi)} = \frac{r_P(\varphi)}{\sin(\pi - \psi)} \left(= \frac{r_P(\varphi)}{\sin(\psi)}\right). \tag{3}$$

After cross-multiplication and differentiation with respect to ψ, we find that

$$\varphi'(\psi)[r_P(\varphi)\cos(\varphi) + r'_P(\varphi)\sin(\varphi)] = R_Q(\pi - \psi)\cos(\psi) - R'_Q(\pi - \psi)\sin(\psi),$$

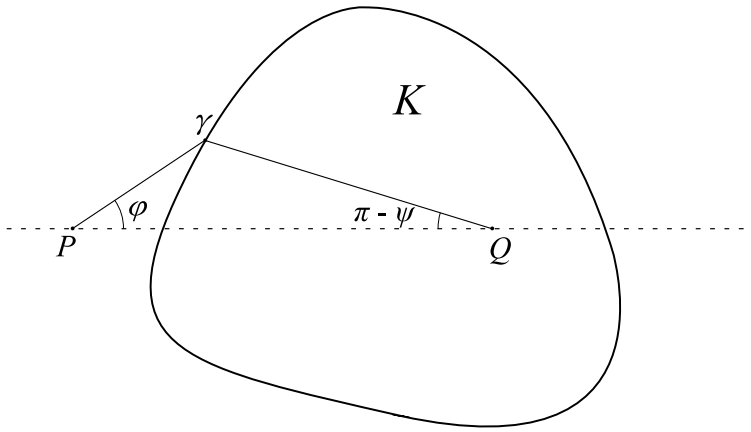

FIGURE 6. γ is on the nearside of ∂K with respect to P.

or, after solving for $\varphi'(\psi)$,

$$\varphi'(\psi) = \frac{R_Q(\pi-\psi)\cos(\psi) - R'_Q(\pi-\psi)\sin(\psi)}{r_P(\varphi)\cos(\varphi) + r'_P(\varphi)\sin(\varphi)}.$$

Now, since these functions are continuous, we may evaluate them at $\psi = \pi$. Noticing from Figure 6 that $\varphi = 0$ when $\psi = \pi$, we have

$$\varphi'(\pi) = -\frac{R_Q(0)}{r_P(0)}. \tag{4}$$

Looking again at Figure 6, we may also use the Law of Cosines to obtain

$$R_Q(\pi-\psi)^2 = |PQ|^2 + r_P(\varphi)^2 - 2|PQ|r_P(\varphi)\cos(\varphi),$$

which, after differentiation with respect to ψ, becomes

$$-2R_Q(\pi-\psi)R'_Q(\pi-\psi) = 2r_P(\varphi)r'_P(\varphi)\varphi'(\psi) - 2|PQ|[r'_P(\varphi)\cos(\varphi)\varphi'(\psi) - r_P(\varphi)\sin(\varphi)\varphi'(\psi)],$$

or

$$R'_Q(\pi-\psi) = -\varphi'(\psi)\frac{r_P(\varphi)r'_P(\varphi) - |PQ|[r'_P(\varphi)\cos(\varphi) - r_P(\varphi)\sin(\varphi)]}{R_Q(\pi-\psi)}.$$

Evaluating at $\psi = \pi$ and substituting from (4), we find

$$R'_Q(0) = [r_P(0) - |PQ|]\frac{r'_P(0)}{r_P(0)},$$

or, noting that $r_P(0) + R_Q(\pi) = |PQ|$,

$$R'_Q(0) = -R_Q(\pi)\frac{r'_P(0)}{r_P(0)}. \tag{5}$$

Now, suppose γ is on the farside of K from P (see Figure 7). Using the Law of Sines and a similar calculation to that used to reach (4), remembering that ψ is now 0 when $\varphi = 0$,

we find that

$$\psi'(0) = \frac{R_P(0)}{R_Q(\pi)}. \tag{6}$$

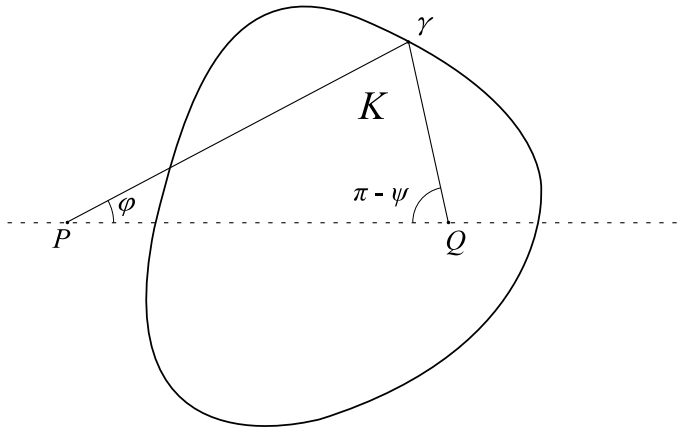

FIGURE 7. γ is on the farside of ∂K with respect to P.

Using the Law of Cosines (with $R_Q(\pi - \psi)$ expressed in terms of φ), differentiating with respect to φ, evaluating at $\varphi = 0$, and substituting $R_Q(0)$ for $R_P(0) - |PQ|$, we get

$$R'_Q(\pi) = -R_Q(0)\frac{R'_P(0)}{R_P(0)}. \tag{7}$$

We can rewrite (5) and (7) as:

$$r_P(0)R'_Q(0) = -R_Q(\pi)r'_P(0) \text{ and} \tag{8}$$

$$R_P(0)R'_Q(\pi) = -R_Q(0)R'_P(0), \tag{9}$$

from which we can form the following linear system of equations by adding $R_Q(\pi)R'_P(0)$ to each side of (8) and $R_P(0)R'_Q(0)$ to each side of (9) and then using identities (1) and (2):

$$R_Q(\pi)R'_P(0) + r_P(0)R'_Q(0) = R_Q(\pi)[R'_P(0) - r'_P(0)] = R_Q(\pi)X'_P(0)$$
$$-R_Q(0)R'_P(0) + R_P(0)R'_Q(0) = R_P(0)[R'_Q(0) + R'_Q(\pi)] = R_P(0)X'_Q(0).$$

We translate this system into a matrix equation:

$$\begin{bmatrix} R_Q(\pi) & r_P(0) \\ -R_Q(0) & R_P(0) \end{bmatrix} \begin{bmatrix} R'_P(0) \\ R'_Q(0) \end{bmatrix} = \begin{bmatrix} R_Q(\pi)X'_P(0) \\ R_P(0)X'_Q(0) \end{bmatrix}.$$

Notice that $\det \begin{pmatrix} R_Q(\pi) & r_P(0) \\ -R_Q(0) & R_P(0) \end{pmatrix} = R_Q(\pi)R_P(0) + r_P(0)R_Q(0) > 0$ since $r_P(0)$, $R_P(0)$, $R_Q(0)$, and $R_Q(\pi)$ are positive distances. Thus, we may invert our matrix to solve the

equation, which yields:

$$R'_P(0) = \frac{R_P(0)[R_Q(\pi)X'_P(0) - r_P(0)X'_Q(0)]}{R_Q(\pi)R_P(0) + r_P(0)R_Q(0)} \text{ and}$$
$$R'_Q(0) = \frac{R_Q(\pi)[R_Q(0)X'_P(0) + R_P(0)X'_Q(0)]}{R_Q(\pi)R_P(0) + r_P(0)R_Q(0)}.$$

Using identities (1) and (2) again, we also have

$$r'_P(0) = \frac{R_P(0)[R_Q(\pi)X'_P(0) - r_P(0)X'_Q(0)]}{R_Q(\pi)R_P(0) + r_P(0)R_Q(0)} - X'_P(0) \text{ and}$$
$$R'_Q(\pi) = X'_Q(0) - \frac{R_Q(\pi)[R_Q(0)X'_P(0) + R_P(0)X'_Q(0)]}{R_Q(\pi)R_P(0) + r_P(0)R_Q(0)}.$$

The following equations are taken from page 12 of [8] and are ultimately derived from page 85 of [1]:

$$r'_P(\varphi) = r_P(\varphi)\cot(\eta_\varphi - \varphi) \text{ and}$$
$$R'_P(\varphi) = R_P(\varphi)\cot(\omega_\varphi - \varphi),$$

where η_φ is the angle of inclination of the line tangent to ∂K at $(r_P(\varphi), \varphi)$ and ω_φ is the angle of inclination of the tangent line at $(\varphi, R_P(\varphi))$. Using these equations, it is a simple matter to solve for the angles of inclination of the tangent lines at the baseline; that is, when $\varphi = 0$. The angles can also be similarly calculated using an analogous equation with Q as the source point:

$$R'_Q(\psi) = R_Q(\psi)\cot(\nu_\psi - \psi),$$

where ν_ψ is the angle of inclination of the tangent line at $(\psi, R_Q(\psi))$, and taking $\psi = \pi$ for the basepoint nearer to P and $\psi = 0$ for the farther basepoint from P. □

Using the derived formulae on our example from above, we find that the tangent lines at the basepoints of our shadow body are in fact vertical.

4. Computing Curvature at Basepoints

We continue by computing the curvature of the body at the basepoints. The procedure for this computation is borrowed greatly from §6 of [8]. Computing the curvature will not directly impact the process we shall later design to construct an approximation of a shadow body; however, it will allow us to immediately check whether or not our shadow body will have the desired sign of curvature at its basepoints.

Theorem. *Let K be a convex body ($\mathcal{C}^2$ near its basepoints), and let $P \notin K$, $Q \in \operatorname{int} K$ such that P and Q lie on the x-axis with P to the left of Q. Then the curvature of ∂K at each of K's basepoints is computable when $R_P(0)R_Q(\pi) \neq r_P(0)R_Q(0)$.*

Proof. From page 91 of [1] we have the following identity:

$$\mathcal{K}(f+g) = \frac{f+g}{f}\mathcal{K}f + \frac{f+g}{g}\mathcal{K}g - 2fg\left(\frac{f'}{f} - \frac{g'}{g}\right)^2 \tag{10}$$

We can apply this identity to (1) and (2) to obtain

$$\mathcal{K}X_P = \mathcal{K}(R_P - r_P) = \frac{X_P}{R_P}\mathcal{K}R_P - \frac{X_P}{r_P}\mathcal{K}r_P + 2R_P r_P \left(\frac{R'_P}{R_P} - \frac{r'_P}{r_P}\right)^2 \text{ and} \tag{11}$$

$$\mathcal{K}X_Q = \mathcal{K}(R_Q + \widetilde{R}_Q) = \frac{X_Q}{R_Q}\mathcal{K}R_Q + \frac{X_Q}{\widetilde{R}_Q}\mathcal{K}\widetilde{R}_Q - 2R_Q\widetilde{R}_Q \left(\frac{R'_Q}{R_Q} - \frac{\widetilde{R}'_Q}{\widetilde{R}_Q}\right)^2, \tag{12}$$

where $\widetilde{R}_Q(\varphi) = R_Q(\varphi+\pi)$. If we evaluate these equations at 0, we see that this is a system of two equations in four unknowns, namely, $\mathcal{K}R_P(0)$, $\mathcal{K}r_P(0)$, $\mathcal{K}R_Q(0)$, and $\mathcal{K}\widetilde{R}_Q(0) = \mathcal{K}R_Q(\pi)$.

Now, since curvature is independent of parametrization, the curvature of ∂K at a basepoint will be the same whether measured from P or from Q. Taking into account changes in concavity, this fact gives us

$$\frac{\mathcal{K}R_P(0)}{(R_P(0)^2 + R'_P(0)^2)^{\frac{3}{2}}} = \kappa R_P(0) = \kappa R_Q(0) = \frac{\mathcal{K}R_Q(0)}{(R_Q(0)^2 + R'_Q(0)^2)^{\frac{3}{2}}} \text{ and}$$

$$\frac{\mathcal{K}r_P(0)}{(r_P(0)^2 + r'_P(0)^2)^{\frac{3}{2}}} = \kappa r_P(0) = -\kappa R_Q(\pi) = -\frac{\mathcal{K}R_Q(\pi)}{(R_P(\pi)^2 + R'_P(\pi)^2)^{\frac{3}{2}}}.$$

Having evaluated (11) and (12) at 0, we can now make substitutions for $\mathcal{K}R_Q(0)$ and $\mathcal{K}R_Q(\pi)$ to get

$$\mathcal{K}X_P(0) = \frac{X_P(0)}{R_P(0)}\mathcal{K}R_P(0) - \frac{X_P(0)}{r_P(0)}\mathcal{K}r_P(0) + 2R_P(0)r_P(0)\left(\frac{R'_P(0)}{R_P(0)} - \frac{r'_P(0)}{r_P(0)}\right)^2 \text{ and}$$

$$\mathcal{K}X_Q(0) = \frac{X_Q(0)}{R_Q(0)}\mathcal{K}R_P(0)\cdot\left(\frac{R_Q(0)^2 + R'_Q(0)^2}{R_P(0)^2 + R'_P(0)^2}\right)^{\frac{3}{2}} - \frac{X_Q(0)}{R_Q(\pi)}\mathcal{K}r_P(0)\cdot\left(\frac{R_Q(\pi)^2 + R'_Q(\pi)^2}{r_P(0)^2 + r'_P(0)^2}\right)^{\frac{3}{2}}$$
$$- 2R_Q(0)R_Q(\pi)\left(\frac{R'_Q(0)}{R_Q(0)} - \frac{R'_Q(\pi)}{R_Q(\pi)}\right)^2.$$

However, we can simplify this system using the fact that

$$\begin{aligned}\left(\frac{R_Q(0)^2 + R'_Q(0)^2}{R_P(0)^2 + R'_P(0)^2}\right)^{\frac{3}{2}} &= \left(\frac{R_Q(0)}{R_P(0)}\right)^3 \left(\frac{1 + \left(\frac{R'_Q(0)}{R_Q(0)}\right)^2}{1 + \left(\frac{R'_P(0)}{R_P(0)}\right)^2}\right)^{\frac{3}{2}} \\ &= \left(\frac{R_Q(0)}{R_P(0)}\right)^3 \left(\frac{1 + \cot(\nu_0 - 0)^2}{1 + \cot(\omega_0 - 0)^2}\right) \qquad \text{(from the equations of [1])} \\ &= \left(\frac{R_Q(0)}{R_P(0)}\right)^3 \cdot 1 \qquad \text{(since } \nu_0 = \omega_0\text{)}.\end{aligned}$$

We can similarly compute that $\left(\frac{R_Q(\pi)^2 + R'_Q(\pi)^2}{r_P(0)^2 + r'_P(0)^2}\right)^{\frac{3}{2}} = \left(\frac{R_Q(\pi)}{r_P(0)}\right)^3$.

Using these simplifications, we translate the system into matrix equation form:

$$\begin{bmatrix} \frac{X_P(0)}{R_P(0)} & -\frac{X_P(0)}{r_P(0)} \\ \frac{X_Q(0)R_Q(0)^2}{R_P(0)^3} & -\frac{X_Q(0)R_Q(\pi)^2}{r_P(0)^3} \end{bmatrix} \begin{bmatrix} \mathcal{K}R_P(0) \\ \mathcal{K}r_P(0) \end{bmatrix} = \begin{bmatrix} \mathcal{K}X_P(0) - 2R_P(0)r_P(0)\left(\frac{R'_P(0)}{R_P(0)} - \frac{r'_P(0)}{r_P(0)}\right)^2 \\ \mathcal{K}X_Q(0) + 2R_Q(0)R_Q(\pi)\left(\frac{R'_Q(0)}{R_Q(0)} - \frac{R'_Q(\pi)}{R_Q(\pi)}\right)^2 \end{bmatrix}.$$

We can now solve the system for $\mathcal{K}R_P(0)$ and $\mathcal{K}r_P(0)$ using matrix inversion, assuming that

$$\begin{aligned} 0 \neq \det &\begin{pmatrix} \frac{X_P(0)}{R_P(0)} & -\frac{X_P(0)}{r_P(0)} \\ \frac{X_Q(0)R_Q(0)^2}{R_P(0)^3} & -\frac{X_Q(0)R_Q(\pi)^2}{r_P(0)^3} \end{pmatrix} \\ &= \frac{X_P(0)X_Q(0)R_Q(0)^2}{R_P(0)r_P(0)^3} - \frac{X_P(0)X_Q(0)R_Q(\pi)^2}{R_P(0)^3 r_P(0)} \\ &= \frac{X_P(0)X_Q(0)}{R_P(0)r_P(0)}\left(\left(\frac{R_Q(0)}{R_P(0)}\right)^2 - \left(\frac{R_Q(\pi)}{r_P(0)}\right)^2\right). \end{aligned}$$

That is, when

$$\left(\frac{R_Q(0)}{R_P(0)}\right)^2 \neq \left(\frac{R_Q(\pi)}{r_P(0)}\right)^2, \text{ or}$$
$$R_P(0)R_Q(\pi) \neq r_P(0)R_Q(0).$$

Note that we did not solve for curvature itself but rather for the values of the curvature operator at the basepoints of the convex body. However, it is a simple matter to find the signed curvature from these values using the formulae from our definitions section. □

Using the curvature formula just derived to test our example body from above, we see that the signed curvature at the nearside basepoint of the shadow body (as measured from P) is approximately -0.319666 and the signed curvature at the farside basepoint is about 0.321355. This confirms that our shadow body is indeed convex at its basepoints.

5. The Stable Manifold Theorem

Before attempting to create an algorithm which constructs a shadow body, we should consider the mathematical principles that lead us to believe such a construction is possible. Let us begin with the Stable Manifold Theorem:

Theorem. *Let V be an open subset of $\mathbb{R}^2$, and let $f : V \to \mathbb{R}^2$ be a $\mathcal{C}^k$ function with the fixed point $p \in V$. Suppose that the eigenvalues of $Df(p) = \begin{bmatrix} \frac{\partial f_x}{\partial x}(p) & \frac{\partial f_x}{\partial y}(p) \\ \frac{\partial f_y}{\partial x}(p) & \frac{\partial f_y}{\partial y}(p) \end{bmatrix}$ are λ and μ, where $|\lambda| < 1 < |\mu|$.*

Then there exist $\mathcal{C}^k$ curves W_S and W_U, respectively tangent to the eigenspaces E_λ and E_μ of $Df(p)$ at p, and an open subset $V' \subseteq V$ such that for every $x \in W_S \cap V'$, $\lim_{j\to\infty} f^j(x) = p$ and for every $x \in W_U \cap V'$, $\lim_{j\to\infty} f^{-j}(x) = p$, where $f^j(x)$ is the j-fold application of the function f (and f^{-j} that of its inverse) to the point x.

We call W_S the *stable manifold* for f at p, and we call W_U the corresponding *unstable manifold*. Notice that if a curve W is the stable manifold for f at p, then it is also the unstable manifold for f^{-1} at p.

If K is a convex body with $P \notin K$ and $Q \in \text{int}K$, and P is at the origin, consider the following functions defined for vectors $v = (v_x, v_y) \in \mathbb{R}^2 - \{0\}$:

$$T_P^+(v) = v\left(1 + \frac{X_P(\arctan\frac{v_y}{v_x})}{||v||}\right)$$
$$T_P^-(v) = v\left(1 - \frac{X_P(\arctan\frac{v_y}{v_x})}{||v||}\right).$$

Purposely confusing vectors and points for the moment, we notice that each function simply moves a point a distance equal to $X_P(\varphi)$ along the line connecting the point with P, where φ is the angle of inclination of that line. We may also remark that $T_P^- = (T_P^+)^{-1}$ for points farther than $X_P(\varphi)$ from P.

Now, translate the plane so that Q lies at the origin, and consider the following map defined for vectors $v \in \mathbb{R}^2 - \{0\}$:

$$T_Q(v) = v\left(1 - \frac{X_Q(\arctan\frac{v_y}{v_x})}{||v||}\right).$$

This map takes a point and moves it toward (and possibly beyond) Q a distance equal to $X_Q(\psi)$ along the line connecting the point with Q, where ψ is the angle of inclination of that line. T_Q also acts as its own inverse for points within $X_Q(\psi)$ of Q. Indeed, for our purposes we need only consider the intersection of $\{v \in \mathbb{R}^2 \mid \text{dist}(P, v) > X_P(\arctan\frac{v_y}{v_x})\}$ and $\{v \in \mathbb{R}^2 \mid 0 < \text{dist}(Q, v) < X_Q(\arctan\frac{v_y}{v_x})\}$ as the domain for each of these three maps.

Now, consider a point p on the farside boundary of our example body: apply to p the function $f(p) = (T_Q T_P^-)^2(p)$. The point will move towards the basepoint under this map (see Figure 8). Applying f again will move the point closer to the basepoint. Now, since

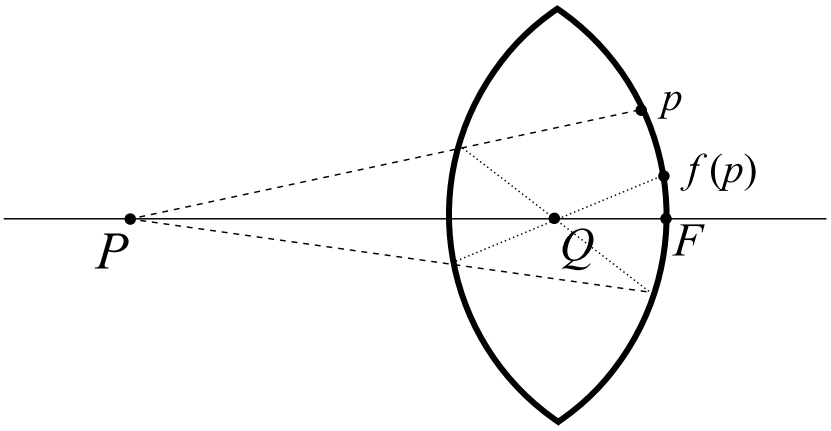

FIGURE 8. Application of $f = (T_Q T_P^-)^2$ to a point on the farside of the body.

the basepoint is itself fixed under this map, we have behavior matching the conclusion of the Stable Manifold Theorem. We now predict $f^{-1}(p) = (T_P^+ T_Q)^2(p)$ will exhibit unstable manifold-like behavior; indeed, points on the farside of the body tend away from the farside basepoint (see Figure 9). Moreover, the vertices of our body act as fixed points under these maps, and points on the farside of the body tend away from and towards the vertices under the maps f and f^{-1}, respectively. Thus, repeated application of f^{-1} to points near the farside basepoint of our body will allow us to generate a good reconstruction of the original body's farside.

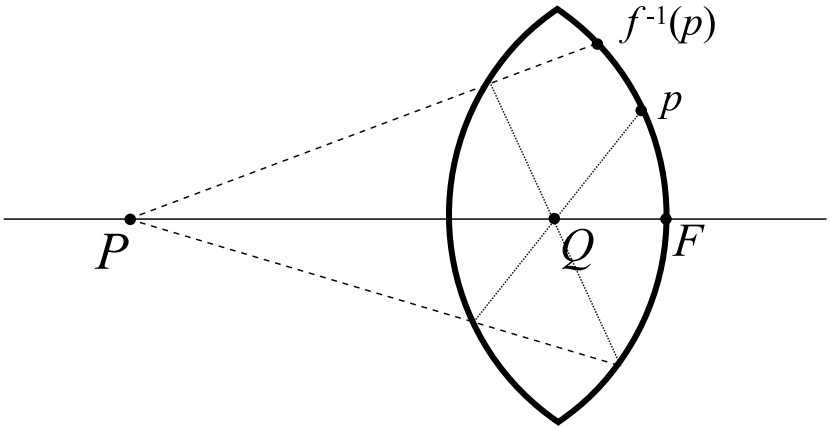

FIGURE 9. Application of $f^{-1} = (T_P^+ T_Q)^2$ to a point on the farside of the body.

Now, suppose there exists a shadow body. It should also have similar behavior under the maps f and f^{-1}. Indeed, as our algorithm will show us, it exhibits the opposite behavior at its farside basepoint: points tend to move toward the basepoint under f^{-1} and away from it under f (this is accounted for by the lack of horizontal symmetry of the shadow body—after constructing the shadow body's approximation, the reader might find it an interesting exercise to apply f and f^{-1} to various points on the approximate body and to observe their behavior under these maps). We could exploit this behavior by applying the map f to points on the shadow body near its farside basepoint, but we do not actually know any points on the shadow body except its basepoint. Therefore, we use a local approximation of the shadow body near the basepoint—the farside tangent line—and apply the function to points on the tangent line within a small distance of the basepoint.

6. Construction of an Approximate Shadow Body

We now attempt to construct an approximation of a shadow body (which we shall call S) along the lines of [8] and [9]. In doing so, we will utilize the example convex body introduced previously. First, we shall create points along the tangent lines of ∂S at each basepoint of S. The points are placed at the intersection of the tangent lines and rays emanating from P at uniformly distributed angles between 0 and ε, an angle chosen beforehand as a program parameter. We operate on each point on every step of the algorithm.

The basic idea of the construction can be briefly described as "chord-chasing"; alternating between the sources P and Q, we move each point across the expected shadow body along the

line connecting it and the source in question, using the function $f(p) = (T_Q T_P^-)^2(p)$ (actually, the algorithm whose code is given below only applies the function f once per iteration, but the end result will be the same). As we iterate this process, our points will start moving away from the basepoints and towards the vertices of the shadow body (as the shadow body's farside boundary turns out to be the unstable manifold of f at the farside basepoint). The iterated points should move along the boundary of the expected shadow body, allowing us to make a rough sketch of the body's shape. We terminate the algorithm when a point moves to or beyond the shadow body's vertex (failure to terminate the algorithm generates interesting results—namely, a reconstruction of the original body—but does not help our attempt to approximate the shadow body). Finally, we use Mathematica's `Interpolation` function to create a twice-differentiable curve that matches the final locations of the points. We then compare this interpolated body with our original body in the hopes of finding only very small discrepancies with our original X-ray data.

In order to generate a similar reconstruction for the nearside, we apply a similar map to points on the tangent line at the nearside basepoint; however, we instead opt to record the locations of the farside points after each application of T_P^-; this will also construct a nice approximation of the shadow body's nearside.

Mathematica code for the algorithm as well as post-algorithmic plots and checks of data can be found in the appendix.

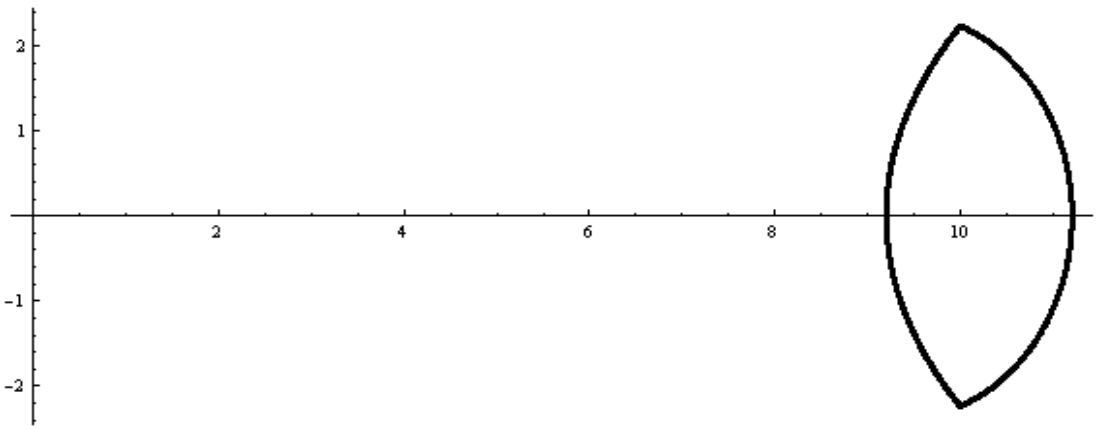

FIGURE 10. The result of running the reconstruction algorithm for 37 iterations with an initial angle of .001 and with $n = 1000$.

A numerical comparison of the X-rays yielded the following results, though with messages indicating the failure of Mathematica's `NIntegrate` function to converge to prescribed accuracy near some points (here, $\hat{X}_P$ and $\hat{X}_Q$ are the X-ray functions of the interpolated shadow

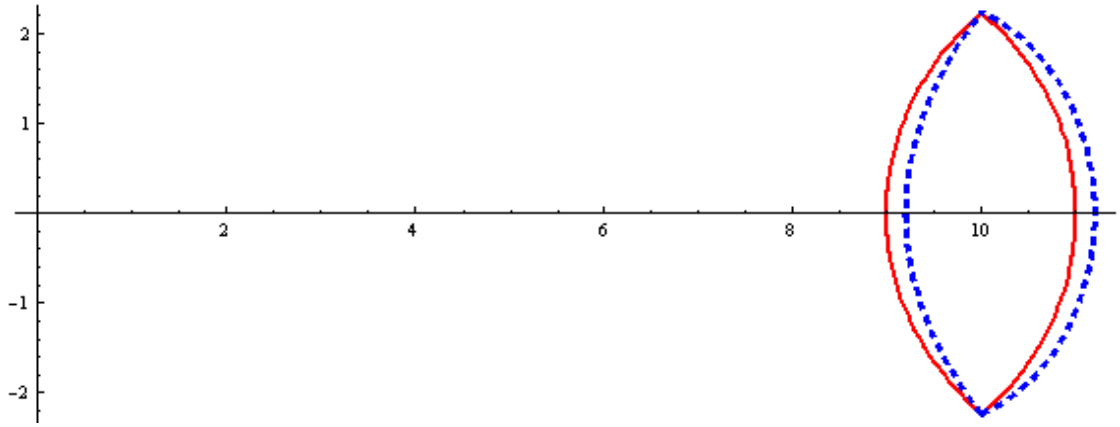

FIGURE 11. The interpolation of the constructed shadow body, in blue, plotted against the original body, in red.

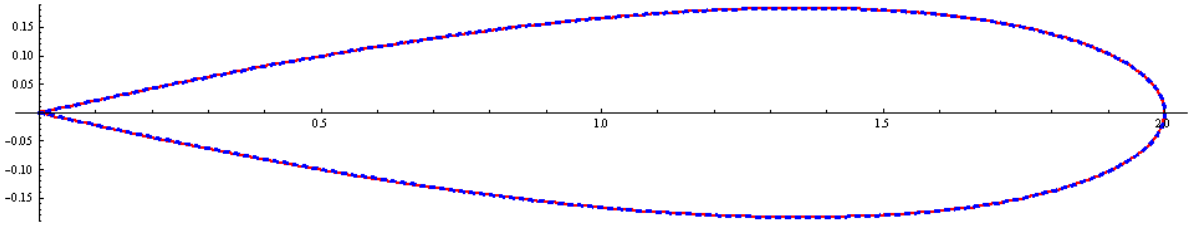

FIGURE 12. The X-ray functions from P of the original body, in red, and the interpolated shadow body, in blue.

body from P and Q, respectively):

$$\int_{-\arccos\left(2\sqrt{\frac{5}{21}}\right)}^{\arccos\left(2\sqrt{\frac{5}{21}}\right)} |X_P(\varphi) - \hat{X}_P(\varphi)| d\varphi \approx 3.09774 \cdot 10^{-6}$$

$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} |X_Q(\psi) - \hat{X}_Q(\psi)| d\psi \approx 0.000012992$$

It is clear from these error estimates that the X-ray bodies are quite similar, though this is far from a proof that there is in fact a shadow body.

We have also plotted the curvature of the near and far sides of the interpolated shadow body. These curvatures were computed by Mathematica using the formula given in the definition of signed curvature. Notice that the plots behave nicely except near angles along which lie the vertices of the body. However, this misbehavior is likely to be an artefact of Mathematica's `Interpolation` function as well as numerical error that comes from the approximation of the basepoints and the use of points on a tangent line for our construction instead of points on the shadow body itself.

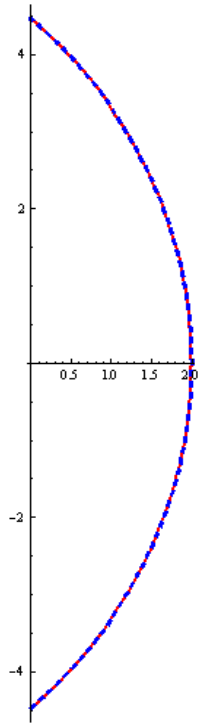

FIGURE 13. The X-ray functions from Q of the original body, in red, and the interpolated shadow body, in blue.

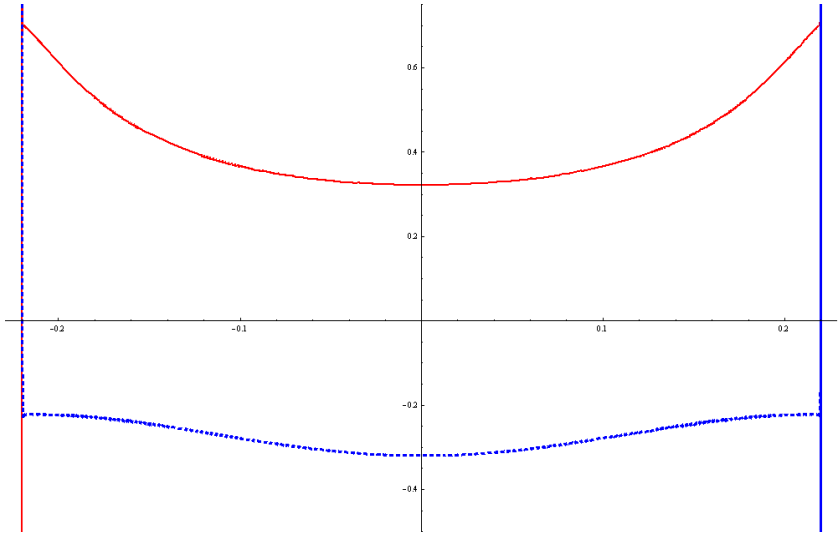

FIGURE 14. The curvatures of the far and near sides of the interpolated shadow body, respectively in red and blue, for $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

7. CONCLUSION

Though we have not produced a rigorous proof of the existence of a shadow body for our example body, we have gathered evidence for the plausibility of such a shadow body's existence. There are a number of directions open to us for future research.

We have only thus far considered a "bottom-up" construction of a shadow body—that is, from the basepoints to the vertices. Ross and Tuite [8] also produced an algorithm for the construction of a shadow body using a "top-down" procedure from the vertices to the basepoints. It is nearly certain that such a method could be devised in the case when one of the point sources is in the interior of the convex body. It would first require the computation of tangent lines and curvature in neighborhoods of the vertices, computations we attempted to produce but did not because of time restraints. Further, one might be able to "link" the constructions generated by the two algorithms in order to avoid the misbehavior of the curvature plot at the vertices (and which would likely be present at the basepoints in a top-down construction). One may even be able to find a numerical convergence to zero of the error between the original X-ray bodies and the X-rays of the constructed shadow body.

Another possible approach would be to find better approximations of the shadow body at its basepoints. For our research we only considered tangent lines of the shadow body, but Siefken and Spargo were able to compute the second derivatives of the shadow body at its basepoints [9], which would allow us to approximate the shadow body with more accurate Taylor polynomial curves. However, there are some obstacles in this approach which must be overcome. The higher-order derivatives quickly become very difficult to compute, though it should be theoretically possible to compute as many as one might desire. Still, we would need the Taylor series in order to find points precisely on the shadow body instead of points very near it; this is problematic in that it is unknown whether such a Taylor series would have a nonzero radius of convergence. Finally, even if an analytic curve is found which is a piece of a shadow body, the algorithm described above as well as those designed and implemented in [8] and [9] may ultimately create an object which is not a compact planar region, let alone a convex region that may be properly called a shadow body.

The field of X-ray tomography has lain open for more than 25 years; the time we had to perform our research lasted barely more than eight weeks. Far from making significant progress in finding a solution to our problem, we have just begun to scratch the surface of the subtleties that abound in this topic of study. We hope to continue our research on this problem and eventually to come to a definitive solution to the question of the uniqueness of a convex body given its point X-rays from two sources.

Appendix A. Code for the Construction Algorithm

The following is Mathematica code that may be used to conduct the algorithm for constructing a shadow body of the example lens from above. Note that this code may also be used to reconstruct the original body as well as to construct other bodies, though it is best suited for the example used in this paper. Note that this code was written on Mathematica 6.0; it will not work unmodified on earlier versions. The functions that will need modification are likely to be the plotting functions.

```
(* Algorithm for the Construction of a Shadow Body from a Lens-shaped
   Convex Body --- by Chris Pryby --- written for Mathematica 6.0 *)

Q = 10; (* x-coordinate of second point *)

(* lens -- intersection of two circles -- only guaranteed to work when Q is
   is centered between the vertices of the lens *)
radius1 = 3;
center1 = 8;
radius2 = 3;
center2 = 12;

init = .001; (* initial angle of approximation - the smaller, the better,
              but you'll need more iterations *)
n = 250; (* there will be 2n+1 points plotted initially *)
iterations = 65; (* the more iterations, the more "reconstructed" the body
                    will become, but too many can lead to a collapse to the
                    stable manifolds near the basepoints and vertices --
                    this is accounted for since the algorithm stops if any
                    point goes past the vertex *)
doshadow = 1; (* 0 for the original body, 1 for the shadow body *)
showall = 1; (* 0 for only final plot, 1 for plots of all iterations *)

(* ---------- only modify the above values! ---------- *)

lensrightP[t_] = center1*Cos[t] + Sqrt[center1^2*Cos[t]^2 - (center1^2 -
                 radius1^2)];
dlensrightP[t_] = D[lensrightP[t], t];
lensleftP[t_] = center2*Cos[t] - Sqrt[center2^2*Cos[t]^2 - (center2^2 -
                radius2^2)];
dlensleftP[t_] = D[lensleftP[t], t];
lensrightQ[t_] = (center1 - Q)*Cos[t] + Sqrt[(center1 - Q)^2*Cos[t]^2 -
                 ((center1 - Q)^2 - radius1^2)];
dlensrightQ[t_] = D[lensrightQ[t], t];
lensleftQ[t_] = (center2 - Q)*Cos[t] - Sqrt[(center2 - Q)^2*Cos[t]^2 -
                ((center2 - Q)^2 - radius2^2)];
```

```
dlensleftQ[t_] = D[lensleftQ[t], t]*Sign[lensleftQ[t]];

(* finds angles on which the vertices lie by solving for the intersection of
   the two circles *)
Print["This is the angle of inclination of the support line from P:"]
supportangle = t /. Quiet[Solve[lensrightP[t] == lensleftP[t], t]][[2]]

(* plots the original body *)
Print["This is the original body with P centered at the origin:"]
PolarPlot[{lensleftP[t], lensrightP[t]}, {t, -supportangle, supportangle}]

(* X-ray functions for lens *)
F[t_] = If[Mod[t,\[Pi]] >= \[Pi]/2, Mod[t,\[Pi]] - \[Pi], Mod[t,\[Pi]]];
lensXP[t_] = lensrightP[t] - lensleftP[t];
dlensXP[t_] = D[lensXP[t], t];
ddlensXP[t_] = D[dlensXP[t], t];
lensXQ[t_] = lensrightQ[t] - lensleftQ[t];
dlensXQ[t_] = D[lensXQ[t], t];
ddlensXQ[t_] = D[dlensXQ[t], t];

Print["This is the distance from P to each vertex:"]
supportradius = lensrightP[supportangle]

(* plots the X-ray bodies, from P and Q, of the original body *)
Print["This is the X-ray function of the original body from P:"]
PolarPlot[lensXP[F[t]], {t, -supportangle, supportangle}]
Print["This is the X-ray function of the original body from Q:"]
PolarPlot[lensXQ[F[t]], {t, -\[Pi]/2, \[Pi]/2}]

(* Falconer's lemma - used for finding basepoints of potential shadow
   body *)
p1 = lensrightP[0];
p2 = lensrightQ[0];
q1 = lensleftP[0];
q2 = lensleftQ[0];
m = p1 - q1;
A = Q;
B = p1*Log[Abs[p1]] - p2*Log[Abs[p2]] - q1*Log[Abs[q1]] +  q2*Log[Abs[q2]];
f[t_] = t*Log[Abs[t]] - (t - m)*Log[Abs[t - m]];
g[t_] = f[t] - f[t - A];
Print["This is the plot of the Falconer g-function against y = B:"]
Plot[{g[t], B}, {t, -Q, 2*Q}, PlotStyle -> {Directive[Red, Thick],
     Directive[Black, Thick, Dashed]}]
(* the result of the FindRoot operation will give the approximate right
```

```
   basepoint for the potential shadow body *)
Print["These are the possible distances for p_1 given by Falconer's Lemma:"]
rightbasepoint1 = t /. FindRoot[G[t] == B, {t, .999*p1}][[1]]
rightbasepoint2 = t /. FindRoot[G[t] == B, {t, 1.25*p1}][[1]]

(* This function of a point will shift the origin of polar coordinates
   measurements for the point; e.g., if P has polar coordinates (t,r) when
   measuring from the cartesian point O=(0,0), and we want to see P's polar
   coordinates when measuring from the cartesian point Q=(2,-5), we would
   use ShiftOrigin[t,r,2,-5]. If the point would end up in the second or
   third quadrants, we add \[Pi] to its resultant angle to account for the
   fact that ArcTan's range is (-\[Pi]/2,\[Pi]/2). *)
ShiftOrigin[t_, r_, h_, k_] = {If[r*Cos[t] - h < 0, \[Pi], 0] +
                               ArcTan[(r*Sin[t] - k)/(r*Cos[t] - h)],
                               Sqrt[(r*Cos[t] - h)^2 + (r*Sin[t] - k)^2]};

(* since the tangent line is vertical, we start by finding the intersection
   of the line x == 9 (which is 9/cos(t)) with angles i*init/n between 0 and
   initial angle init, i from -n to n *)

XP[t_] = lensXP[F[t]];
XQ[t_] = lensXQ[F[t]];
xshift = Q;
yshift = 0;
rightbasepoint = If[doshadow == 0, rightbasepoint1, rightbasepoint2];
leftbasepoint = rightbasepoint - XP[0];

(* sets up initial set of points to manipulate *)
(* the reconstruction collapses if you remove the nearside points and are
   reconstructing the original body; it also collapses if you remove the
   farside points and are constructing the shadow body *)
PointSet = {};

(* initial points on nearside tangent line for original body, farside for
   shadow body *)
If[doshadow == 0,
   For[i = -n, i <= n, i++,
       PointSet = Union[PointSet, {{i*init/n, (rightbasepoint -
            XP[0])Cos[i*init/n]}}]],
   For[i = -n, i <= n, i++,
       PointSet = Union[PointSet, {{i*init/n,
       rightbasepoint*Cos[i*init/n]}}]]
]
```

```
(* initial plot of tangent lines *)
If[showall != 0, Print["This is the initial setup of the points to be
                       manipulated -- they lie on a tangent line of the
                       body:"], 0;]
If[showall != 0, ListPolarPlot[PointSet, PlotRange -> All, PlotStyle ->
                                Directive[Thick, Black]], 0;]
If[showall != 0, Print["These are plots of successive iterations of the
                       algorithm -- every two plots demonstrates a
                       complete iteration:"], 0;]

stop = 0;

(* chord chasing procedure *)
For[i = 1, i <= iterations && stop == 0, i++,

    (* prints the iteration number *)
    If[showall == 1, Print[i], 0];

  (* forces the computer to store points as numerical approximations
     instead of exact expressions -- speeds up the process *)
    PointSet = N[PointSet];

    (* adds (or subtracts) the X-ray data obtained from P -- subtract X_P if
       curvature from P is positive *)
    For[j = 1, j <= Length[PointSet], j++,
        angle = PointSet[[j]][[1]];
        radius = PointSet[[j]][[2]];
        (* determining whether to add or subtract XP is currently
           hardcoded for the lens -- it should be easy if we have a shape
           with two vertices which the support lines from P intersect, but
           otherwise it gets difficult (for the shadow body, that is) *)
        PointSet[[j]] = {angle, radius + If[radius > supportradius, -1, 1]*
                         XP[angle]};
        stop = If[doshadow == 0, If[radius >= supportradius, 1, 0],
               If[radius <= supportradius, 1, 0]]
    ];

    If[showall == 1, Print[ListPolarPlot[PointSet, PlotRange -> All,
                           PlotStyle -> Directive[Thick, Black]]], 0;];

    (* stores PointSet after adding/subtracting from P for future use in
       superimposing reconstructed nearside & farside *)
    PrevPointSet = PointSet;
```

```
    (* shifts origin to Q *)
    For[j = 1, j <= Length[PointSet], j++,
        angle = PointSet[[j]][[1]];
        radius = PointSet[[j]][[2]];
        PointSet[[j]] = ShiftOrigin[angle, radius, xshift, yshift]
    ];

    (* subtracts X-ray data obtained from Q -- always should subtract since
       the points must move inwards towards Q *)
    For[j = 1, j <= Length[PointSet], j++,
        angle = PointSet[[j]][[1]];
        radius = PointSet[[j]][[2]];
        PointSet[[j]] = {angle, radius - XQ[angle]}
    ];

    (* shifts origin back to P *)
    For[j = 1, j <= Length[PointSet], j++,
        angle = PointSet[[j]][[1]];
        radius = PointSet[[j]][[2]];
        PointSet[[j]] = ShiftOrigin[angle, radius, -xshift, -yshift]
    ];

    If[showall == 1, Print[ListPolarPlot[PointSet, PlotRange -> All,
                          PlotStyle -> Directive[Thick, Black]]], 0;];

    (* chord-chasing terminates if point passes vertex - should rerun the
       algorithm with one less iteration to get a more accurate
       interpolation *)
    If[stop != 0, Print["Process stopped because the reconstruction crossed
                        a vertex on iteration ", i], 0;]

](* close for-loop *)

ListPolarPlot[{PointSet, PrevPointSet}, PlotRange -> All, PlotStyle ->
              Directive[Thick, Black]]

(* interpolation of data from P *)
shadowrightP = Interpolation[PointSet, InterpolationOrder -> 2];
dshadowrightP[t_] = D[shadowrightP[t], t];
shadowleftP = Interpolation[PrevPointSet, InterpolationOrder -> 2];
dshadowleftP[t_] = D[shadowleftP[t], t];
shadowXP[t_] = shadowrightP[t] - shadowleftP[t];
dshadowXP[t_] = D[shadowXP[t], t];
```

```
Print["This is the interpolated shadow body plotted against the original
       body."]
Quiet[PolarPlot[{lensrightP[t], lensleftP[t], shadowrightP[t],
      shadowleftP[t]}, {t, -supportangle, supportangle},
      PlotStyle -> {Directive[Red, Thick], Directive[Red, Thick],
      Directive[Blue, Thick, Dashed], Directive[Blue, Thick, Dashed]}]]

Print["This is the X-ray from P of the interpolated shadow body plotted
       against the X-ray of the original body."]
Quiet[PolarPlot[{lensXP[t], shadowXP[t]}, {t, -supportangle,
      supportangle}, PlotStyle -> {Directive[Red, Thick],
      Directive[Blue, Thick, Dashed]}]]

(* shifting data to originate from Q *)
QSet = {};
PrevQSet = {};
For[i = 1, i <= Length[PointSet], i++,
    angle = PointSet[[i]][[1]];
    radius = PointSet[[i]][[2]];
    QSet = Union[QSet, {ShiftOrigin[angle, radius, Q, 0]}]
]
For[i = 1, i <= Length[PrevPointSet], i++,
    angle = PrevPointSet[[i]][[1]];
    radius = PrevPointSet[[i]][[2]];
    PrevQSet = Union[PrevQSet, {ShiftOrigin[angle, radius, Q, 0]}]
]

(* interpolation of data from Q -- domain of left side of shadow body from Q
   was altered, and the leftside data was added instead of subtracted to get
   X_Q, but the idea is what counts *)
shadowrightQ = Interpolation[QSet, InterpolationOrder -> 2];
dshadowrightQ[t_] = D[shadowrightQ[t], t];
tempshadowleftQ = Interpolation[PrevQSet, InterpolationOrder -> 2];
shadowleftQ[t_] = -tempshadowleftQ[t + \[Pi]];
dshadowleftQ[t_] = D[shadowleftQ[t], t]*Sign[shadowleftQ[t]];
shadowXQ[t_] = shadowrightQ[t] - shadowleftQ[t];
dshadowXQ[t_] = D[shadowXQ[t], t];

Print["This is the interpolated shadow body plotted against the original
       body:"]
Quiet[PolarPlot[ {lensrightQ[t], lensleftQ[t], shadowrightQ[t],
      shadowleftQ[t]}, {t, -\[Pi]/2, \[Pi]/2}, PlotStyle -> {Directive[Red,
      Thick], Directive[Red, Thick], Directive[Blue, Thick, Dashed],
      Directive[Blue, Thick, Dashed]}]]
```

```
Print["This is the X-ray from Q of the interpolated shadow body plotted
       against the X-ray of the original body:"]
Quiet[PolarPlot[{lensXQ[t], shadowXQ[t]}, {t, -\[Pi]/2, \[Pi]/2}, PlotStyle
      -> {Directive[Red, Thick], Directive[Blue, Thick, Dashed]}]]

ddshadowleftP[t_] = D[shadowleftP[t], {t, 2}];
ddshadowleftQ[t_] = D[shadowleftQ[t], {t, 2}]*Sign[shadowleftQ[t]];
ddshadowrightP[t_] = D[shadowrightP[t], {t, 2}];
ddshadowrightQ[t_] = D[shadowrightQ[t], {t, 2}];

(* testing the values of the curvatures *)
Print["These are the curvatures of the original body at its basepoints as
       computed by my formula (should be +/- 1/3):"]
matrixA = {{lensXP[0]/lensrightP[0], -lensXP[0]/lensleftP[0]},
        {lensXQ[0]/lensrightQ[0]*((lensrightQ[0]^2 + dlensrightQ[0]^2)/
        (lensrightP[0]^2 + dlensrightP[0]^2))^(3/2), -lensXQ[0]/
        Abs[lensleftQ[0]]*((Abs[lensleftQ[0]]^2 + dlensleftQ[0]^2)/(
        lensleftP[0]^2 + dlensleftP[0]^2))^(3/2)}};
matrixB = {lensXP[0]^2 + 2*dlensXP[0]^2 - lensXP[0]*ddlensXP[0] -
          2*lensrightP[0]*lensleftP[0]*(dlensrightP[0]/lensrightP[0] -
          dlensleftP[0]/lensleftP[0])^2, lensXQ[0]^2 + 2 dlensXQ[0]^2 -
          lensXQ[0] ddlensXQ[0] + 2*lensrightQ[0]*Abs[lensleftQ[0]]*
          (dlensrightQ[0]/lensrightQ[0] - dlensleftQ[0]/
          Abs[lensleftQ[0]])^2};
kops = Inverse[matrixA].matrixB;
Print["Farside curvature:"]
kops[[1]]/(lensrightP[0]^2 + dlensrightP[0])^(3/2)
Print["Nearside curvature:"]
kops[[2]]/(lensleftP[0]^2 + dlensleftP[0])^(3/2)

Print["These are the curvatures of the shadow body at its basepoints as
       computed by my formula:"]
matrixA = {{lensXP[0]/shadowrightP[0], -lensXP[0]/shadowleftP[0]},
        {lensXQ[0]/shadowrightQ[0]*((shadowrightQ[0]^2 +
        dshadowrightQ[0]^2)/(shadowrightP[0]^2 + dshadowrightP[0]^2))^
        (3/2), -lensXQ[0]/Abs[shadowleftQ[0]]*((Abs[shadowleftQ[0]]^2 +
        dshadowleftQ[0]^2)/(shadowleftP[0]^2 + dshadowleftP[0]^2))^(3/2)}};
matrixB = {lensXP[0]^2 + 2*dlensXP[0]^2 - lensXP[0]*ddlensXP[0] -
          2*shadowrightP[0]*shadowleftP[0]*(dshadowrightP[0]/
          shadowrightP[0] - dshadowleftP[0]/shadowleftP[0])^2, lensXQ[0]^2 +
          2*dlensXQ[0]^2 - lensXQ[0]*ddlensXQ[0] + 2*shadowrightQ[0]*
          Abs[shadowleftQ[0]]*(dshadowrightQ[0]/shadowrightQ[0] -
          dshadowleftQ[0]/Abs[shadowleftQ[0]])^2};
```

```
kops = Inverse[matrixA].matrixB;
Print["Farside curvature:"]
kops[[1]]/(shadowrightP[0]^2 + dshadowrightP[0])^(3/2)
Print["Nearside curvature:"]
kops[[2]]/(shadowleftP[0]^2 + dshadowleftP[0])^(3/2)

Print["These are the approximate curvatures of the shadow body at the
       basepoints as computed by Mathematica:"]
kshadowrightP[t_] = (shadowrightP[t]^2 + 2*dshadowrightP[t]^2 -
                     shadowrightP[t]*ddshadowrightP[t])/(shadowrightP[t]^2 +
                     dshadowrightP[t]^2)^(3/2);
kshadowleftP[t_] = (shadowleftP[t]^2 + 2*dshadowleftP[t]^2 - shadowleftP[t]*
                    ddshadowleftP[t])/(shadowleftP[t]^2 + dshadowleftP[t]^2)^
                    (3/2);
kshadowrightQ[t_] = (shadowrightQ[t]^2 + 2*dshadowrightQ[t]^2 -
                     shadowrightQ[t]*ddshadowrightQ[t])/(shadowrightQ[t]^2 +
                     dshadowrightQ[t]^2)^(3/2);
kshadowleftQ[t_] = (Abs[shadowleftQ[0]]^2 + 2*dshadowleftQ[t]^2 -
                    Abs[shadowleftQ[0]]*ddshadowleftQ[t])/
                    (Abs[shadowleftQ[0]]^2 + dshadowleftQ[t]^2)^(3/2);

Print["Farside curvature, parametrized by P:"]
kshadowrightP[0]
Print["Nearside curvature, parametrized by P:"]
kshadowleftP[0]
Print["Farside curvature, parametrized by Q:"]
kshadowrightQ[0]
Print["Nearside curvature, parametrized by Q:"]
kshadowleftQ[0]

Print["Here are plotted the approximate curvatures of the shadow body as
       computed by Mathematica, parametrized by P:"]
Quiet[Plot[{kshadowrightP[t], kshadowleftP[t]}, {t, -supportangle,
           supportangle}, PlotStyle -> {Directive[Thick, Red],
           Directive[Thick, Blue, Dashed]}, PlotRange -> {-.35, .75}]]

Print["Here are some numerical computations of error between the original X-
       rays and the X-rays of the interpolated shadow body:"]
NIntegrate[Abs[shadowXP[t] - lensXP[t]], {t, -ArcCos[2*Sqrt[5/21]],
           ArcCos[2*Sqrt[5/21]]}]
NIntegrate[Abs[shadowXP[t] - lensXP[t]]/Sin[t], {t, -ArcCos[2*Sqrt[5/21]],
           ArcCos[2 Sqrt[5/21]]}]
NIntegrate[Abs[shadowXQ[t] - lensXQ[t]], {t, -\[Pi]/2, \[Pi]/2}]
NIntegrate[Abs[shadowXQ[t] - lensXQ[t]]/Sin[t], {t, -\[Pi]/2, \[Pi]/2}]
```

References

[1] Black, W., J. Kimble, D. Koop, and D. C. Solmon. "Functions that are the Directed X-Ray of a Planar Convex Body." *Rendiconti dell'Istituto di Matematica dell'Universit di Trieste*. XXXV:81–115 (2003).
[2] Butcher, G. L., A. Medin, and D. C. Solmon. "Planar Convex Bodies with a Common Directed X-ray." *Rendiconti dell'Istituto di Matematica dell'Universit di Trieste*. XXXVII:181–206 (2005).
[3] Falconer, K. J. "Hammer's X-ray Problem and the Stable Manifold Theorem." *Journal of the London Mathematical Society*. 28:149–160 (1983).
[4] —. "X-ray Problems for Point Sources." *Proceedings of the London Mathematical Society*. 46:241–262 (1983).
[5] Gardner, R. J. *Geometric Tomography*. 178–194, 199–200 (1995)
[6] —. "Symmetrals and X-rays of planar convex bodies." *Archiv der Mathematik*. 41:183–189 (1983).
[7] Lam, D. and D. C. Solmon. "Reconstructing Convex Polygons in the Plane from One Directed X-Ray." *Discrete Computational Geometry*. 26:105–146 (2001).
[8] Ross, Dusty and Kathleen Tuite. "Investigation of Two Bodies with Equal Point X-rays at Two Sources." *Proceedings of the REU Program in Mathematics at Oregon State University*. 149–176 (2006).
[9] Siefken, Jason and Lena Spargo. "An Algorithm for Reconstruction of a Convex Body from Two Point Sources." *Proceedings of the REU Program in Mathematics at Oregon State University*. (2005).

University of Georgia
E-mail address: cpryby@uga.edu