# Fast Alternating Fitting Methods for Trigonometric Curves for Large Data Sets

Alessandro Buccini[a], Fei Chen[b], Omar De la Cruz Cabrera[c], Lothar Reichel[c]

[a]*Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy*
[b]*School of Mathematics, Trinity College Dublin, Dublin, Ireland*
[c]*Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA*

**Abstract**

This paper discusses and develops new methods for fitting trigonometric curves, such as circles, ellipses, and dumbbells, to data points in the plane. Available methods for fitting circles or ellipses are very sensitive to outliers in the data, and are time consuming when the number of data points is large. The present paper focuses on curve fitting methods that are attractive to use when the number of data points is large. We propose a direct method for fitting circles, and two iterative methods for fitting ellipses and dumbbell curves based on trigonometric polynomials. These methods efficiently minimize the sum of the squared geometric distances between the given data points and the fitted curves. In particular, we are interested in detecting the general shape of an object such as a galaxy or a nebula. Certain nebulae, for instance, the one shown in the experiment section, have a dumbbell shape. Methods for fitting dumbbell curves have not been discussed in the literature. The methods developed are not very sensitive to errors in the data points. The use of random subsampling of the data points to speed up the computations also is discussed. The techniques developed in this paper can be applied to fitting other kinds of curves as well.

*Keywords:* least-squares approximation, regularization, curve fitting, trigonometric polynomials
*2020 MSC:* 42A10, 65F20, 65F22

## 1. Introduction

Problems that require fitting circles, ellipses, and curves of other shapes such as dumbbell curves to data points in the plane arise in many application areas such as pattern recognition, computer vision, statistics, and data analysis; see, e.g., [1, 2, 3, 4, 5]. This paper is concerned with minimization problems

*Email addresses:* alessandro.buccini@unica.it (Alessandro Buccini), chenf2@tcd.ie (Fei Chen), odelacru@kent.edu (Omar De la Cruz Cabrera), reichel@math.kent.edu (Lothar Reichel)

that arise when fitting curves in the $(x, y)$-plane defined in the standardized parametric form

$$\begin{cases} x(t) = \sum_{i=1}^{3}(a_{2i-1,1}\cos(it) + a_{2i,1}\sin(it)), \\ y(t) = \sum_{i=1}^{3}(a_{2i-1,2}\cos(it) + a_{2i,2}\sin(it)), \end{cases} \tag{1}$$

where $-\pi < t \leq \pi$, and the $a_{ij} \in \mathbb{R}$ are coefficients to be determined with $a_{11}, a_{22} > 0$. For our applications of interest, it suffices to consider the sums (1), however, the methods discussed also can be applied when the sums include higher frequencies. We assume that the curve $t \to (x(t), y(t))$, $-\pi < t \leq \pi$, does not intersect itself, i.e., it is a one-to-one mapping for $-\pi < t \leq \pi$. The curve is assumed to be centered at the origin and to have zero tilt angle, i.e., the major axis of the curve lies on the $x$-axis. Section 3 discusses how to handle the situations when the tilt angle is nonvanishing or the curve is not centered at the origin.

We consider two minimization problems in this paper. Let $(x_i, y_i)$, $i = 1, 2, \ldots, n$, be given data points in the $(x, y)$-plane and define the data matrix

$$Z = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \in \mathbb{R}^{n\times 2}. \tag{2}$$

Introduce the model matrices

$$\tilde{T} = \begin{bmatrix} \cos(t_1) & \sin(t_1) & \cos(2t_1) & \sin(2t_1) & \cos(3t_1) & \sin(3t_1) \\ \cos(t_2) & \sin(t_2) & \cos(2t_2) & \sin(2t_2) & \cos(3t_2) & \sin(3t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cos(t_n) & \sin(t_n) & \cos(2t_n) & \sin(2t_n) & \cos(3t_n) & \sin(3t_n) \end{bmatrix} \in \mathbb{R}^{n\times 6} \tag{3}$$

and

$$T = \begin{bmatrix} \tilde{T} & 0 \\ 0 & \tilde{T} \end{bmatrix} \tag{4}$$

with parameters $t_i \in (-\pi, \pi]$. These parameters define the vector $\boldsymbol{t} = [t_1, t_2, \ldots, t_n]^T \in (-\pi, \pi]^n$, where the superscript $^T$ denotes transposition. We are interested in the situation when a large number of points has to be fitted. Throughout this paper we assume that $n \gg 6$. Our first minimization problem is

$$\min_{\boldsymbol{t},\boldsymbol{a}} \|T\boldsymbol{a} - \boldsymbol{z}\|_2^2, \tag{5}$$

where $\boldsymbol{a} = [a_{11}, a_{21}, \ldots, a_{61}, a_{12}, a_{22}, \ldots, a_{62}]^T$, $\boldsymbol{z} = [x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n]^T \in \mathbb{R}^{2n}$ is the data vector, and $\|\cdot\|_2$ denotes the Euclidean vector norm. We note that (5) minimizes the sum of the squared geometric distances, i.e., the sum of the squared distance between each given data point $z_i = (x_i, y_i)$ and the point $(x(t_i), y(t_i))$ on the fitted curve.

The second problem we consider is the regularized minimization problem,

$$\min_{\boldsymbol{t},\boldsymbol{a}} \left\{ \|T\boldsymbol{a} - \boldsymbol{z}\|_2^2 + \mu \|W\boldsymbol{a}\|_q^q \right\}, \tag{6}$$

where $\|\cdot\|_q$ denotes the vector $\ell_q$-norm for $q \in \{1, 2\}$, $\mu > 0$ is a regularization parameter, and

$$W = \mathrm{diag}[w_{11}, w_{21}, \ldots, w_{61}, w_{12}, w_{22}, \ldots, w_{62}] \in \mathbb{R}^{12\times 12}$$

is a diagonal weighting matrix whose entry $w_{ij} > 0$ is referred to as the weight of $a_{ij}$. The first and second terms of the expression (6) are called the fidelity term and the regularization term, respectively. The regularization parameter $\mu$ determines the relative importance of these terms. Depending on the shape of the curve to be fitted, different choices of weights and parameter $\mu$ may be employed. We discuss these choices in Sections 3.5 and 4.4. In this paper we will choose $q = 1$ since $\ell_1$-regularization promotes sparsity of the computed solution. We illustrate in Section 3.5 that most coefficients of the curves that we are particularly interested in fitting to standardized data vanish.

To the best of our knowledge, dumbbell curve fitting methods have not been considered in the literature. As for circle and ellipse fitting, available methods that minimize the sum of the squared geometric distances are iterative and tend to be time-consuming, in particular when there are many data points; see, e.g., Gander et al. [6]. More efficient methods such as the direct ellipse-specific fitting method [2] and the direct least-algebraic-residuals method [7] are good candidates when the number of data points is large, because they are faster than the iterative method described in [6]; however, the former methods do not minimize the sum of the squared geometric distances and, thus, it is not clear which physical quantities they minimize. This can make it difficult to determine the significance of the determined curve. Moreover, the ellipses determined by the methods described in [2, 7] may be severely affected by outliers among the data points; this is illustrated by examples in Sections 4.1 and 4.3. This paper presents a direct circle fitting method that generates a unique solution to the minimization problem (5), and two alternating iterative methods for fitting ellipses and dumbbell curves, that solve the minimization problem (5) and (6), respectively. These methods are efficient, not very sensitive to outliers, and either minimize the geometric distance (5) or an accurate approximation thereof if the regularization parameter $\mu > 0$ in (6) is small. The techniques developed in this paper also can be applied to fitting curves of higher degrees; then the sum (1) has more terms.

The organization of this paper is as follows. Section 2 briefly reviews methods for fitting circles and ellipses in the current literature. In Section 3, we propose a direct circle fitting method (DCF), an iterative alternating least-squares fitting method (AFM-LS) for ellipse fitting, and another iterative alternating fitting method with soft-thresholding (AFM-FISTA) for dumbbell curve fitting. We also show in this section that the AFM-LS method converges to a solution that is a stationary point of the objective function of the minimization problem (5).

A few related results on the convergence also are provided. Section 4 illustrates the performance of the proposed methods and compares them to some available schemes. The use of random subsampling to speed up the computations is illustrated. Concluding remarks can be found in Section 5.

## 2. Review of Current Methods for Circle and Ellipse Fitting

Available least-squares methods for circle or ellipse fitting can be classified into two categories: i) minimizing the sum of the squared geometric distances, and ii) minimizing the sum of the norms of algebraic residuals. The algebraic residual is also referred to as the algebraic distance. This section reviews some available methods for ellipse fitting; see [6, 8] for circle fitting methods.

### *2.1. Minimizing the Squared Sum of Geometric Distances*

One approach to fit an ellipse to given data points is to minimize the sum of the square of geometric distances between the given point set and the desired ellipse. Consider an ellipse in its parametric form

$$\boldsymbol{z} = \boldsymbol{z}_c + Q(\alpha) \begin{bmatrix} a\cos(t) \\ b\sin(t) \end{bmatrix}, \quad -\pi < t \leq \pi,$$

where $\boldsymbol{z} = [x, y]^T$, $\boldsymbol{z}_c = [x_c, y_c]^T$ is the center of the ellipse, $\alpha$ is the counter-clockwise angle from the horizontal axis to the major axis of the ellipse, and

$$Q(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

is a rotation matrix. To determine the ellipse that minimizes the sum of the squared geometric distances between the given points $(x_i, y_i)$, $i = 1, 2, \ldots, n$, and an ellipse, one solves the minimization problem

$$\min_{x_c, y_c, \alpha, a, b, t_i} \sum_{i=1}^{n} \left\| \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix} - Q(\alpha) \begin{bmatrix} a\cos(t_i) \\ b\sin(t_i) \end{bmatrix} \right\|_2^2;$$

see Gander et al. [6] for details on the geometric least-squares (GLS) algorithm for determining the $n + 5$ unknowns $x_c, y_c, \alpha, a, b, t_1, t_2, \ldots t_n$. Gander et al. [6] illustrate that geometric algorithms in general produce more visually pleasing ellipses than algebraic algorithms to be described below. However, the GLS algorithm is much more computationally expensive than the ones described in the sequel.

### *2.2. Minimizing Algebraic Residuals*

In analytic geometry, an ellipse is defined as a set of points $\boldsymbol{z} = [x, y] \in \mathbb{R}^2$ that satisfy the equation

$$ax^2 + bxy + cy^2 + dx + ey + f = 0, \tag{7}$$

where $f \neq 0$ and $b^2 - 4ac < 0$. Let

$$\boldsymbol{\alpha} = [a, b, c, d, e, f]^T, \quad \boldsymbol{u} = [x^2, xy, y^2, x, y, 1]^T$$

and define

$$r(\boldsymbol{\alpha}, \boldsymbol{u}) = \boldsymbol{\alpha}^T \boldsymbol{u} = ax^2 + bxy + cy^2 + dx + ey + f.$$

Then the algebraic residual of a point $\boldsymbol{z}_i = [x_i, y_i]^T$ to the ellipse $r(\boldsymbol{\alpha}, \boldsymbol{u}) = 0$ is defined as $r(\boldsymbol{\alpha}, \boldsymbol{u}_i)$, where

$$\boldsymbol{u}_i = [x_i^2, x_i y_i, y_i^2, x_i, y_i, 1]^T. \tag{8}$$

The optimal ellipse in the sense of least-algebraic-residuals is determined by the vector $\boldsymbol{\alpha}$ that minimizes

$$\sum_{i=1}^{n} r(\boldsymbol{\alpha}, \boldsymbol{u}_i)^2; \tag{9}$$

see [9].

The ellipse determined by equation (7) is invariant under scaling of the vector $\boldsymbol{\alpha}$. To avoid the trivial solution $\boldsymbol{\alpha} = \boldsymbol{0}$, a constraint such as $\|\boldsymbol{\alpha}\|_2 = 1$ could be imposed. The disadvantage of this particular constraint is that it is not invariant under Euclidean transformation. To overcome this issue, Bookstein [1] proposes the constraint $a^2 + b^2/2 + c^2 = 1$ when fitting ellipses, hyperbolas, and parabolas. When fitting ellipses, Gander et al. [6], Porrill [4], and Rosin [5] impose the constraint $a + c = 1$.

### *2.3. Direct Methods*

The minimization problems in Sections 2.1 and 2.2 are usually solved by iterative methods. For example, Bookstein [1] and Gander et al. [6] apply Gauss-Newton methods for minimizing the algebraic residual norms and for minimizing the squared sum of the geometric distances, respectively. Fitzgibbon et al. [2] propose a direct ellipse-specific fitting (DEF) method, which is significantly faster than the iterative methods, and Bookstein [1] demonstrates that when a quadratic constraint is imposed on the coefficients to avoid the trivial solution $\boldsymbol{\alpha} = \boldsymbol{0}$, the minimization problem (9) can be solved by computing the solution of a generalized eigenvalue problem

$$U^T U \boldsymbol{\alpha} = \lambda C \boldsymbol{\alpha} \quad \text{subject to} \quad \boldsymbol{\alpha}^T C \boldsymbol{\alpha} = 1, \tag{10}$$

where $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]^T$ is the design matrix of the data, $\boldsymbol{u}_i$ is defined as in (8), and the matrix $C \in \mathbb{R}^{6\times 6}$ expresses the quadratic constraint on $\boldsymbol{\alpha}$. The existence of a nontrivial solution $\boldsymbol{\alpha}$ of (10) implies that the the matrix $U^T U - \lambda C$ is rank-deficient. Since the minimization problem (9) with the constraint $b^2 - 4ac < 0$ is difficult to solve, Fitzgibbon and Fisher [3] impose the equality constraint $4ac - b^2 = 1$. The corresponding constraint matrix then is

$$C = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and the constrained ellipse fitting problem is transformed to

$$\min_{\boldsymbol{\alpha}} \|U\boldsymbol{\alpha}\|_2^2 \quad \text{subject to} \quad \boldsymbol{\alpha}^T C \boldsymbol{\alpha} = 1. \tag{11}$$

Differentiating $\|U\boldsymbol{\alpha}\|_2^2$ and introducing the Lagrange multiplier $\lambda$, one can determine a solution of (11) by solving

$$\begin{cases} S\boldsymbol{\alpha} = \lambda C\boldsymbol{\alpha}, & \text{(a)} \\ \boldsymbol{\alpha}^T C\boldsymbol{\alpha} = 1, & \text{(b)} \end{cases} \tag{12}$$

where $S = U^T U$. The solution of (11) is given by $\boldsymbol{\alpha} = \mu\boldsymbol{\beta}$, where

$$\mu = \sqrt{\frac{1}{\boldsymbol{\beta}^T S \boldsymbol{\beta}}}$$

and $(\lambda, \boldsymbol{\beta})$ is a generalized eigenpair of 12(a).

Ohad [7] describes another direct method to fit an ellipse algebraically. Consider the ellipse equation

$$ax^2 + bxy + cy^2 + dx + ey = 1$$

with $b^2 - 4ac < 0$, and let $\circledast$ denote the Hadamard product of two vectors of the same size, i.e.,

$$\boldsymbol{s} \circledast \boldsymbol{t} = [s_1 t_1, s_2 t_2, \ldots, s_n t_n]^T,$$

where $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]^T$ and $\boldsymbol{t} = [t_1, t_2, \ldots, t_n]^T$. Given the data matrix (2), introduce the vectors $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$. The cost function for fitting an ellipse to the points $\boldsymbol{z}_i = [x_i, y_i]^T$, $i = 1, 2, \ldots, n$, is defined as

$$C(\boldsymbol{\alpha}) = (\widetilde{Z}\boldsymbol{\alpha} - \mathbf{1})^T (\widetilde{Z}\boldsymbol{\alpha} - \mathbf{1}), \tag{13}$$

where $\boldsymbol{\alpha} = [a, b, c, d, e]^T$, the matrix

$$\widetilde{Z} = [\mathbf{x} \circledast \mathbf{x}, \mathbf{x} \circledast \mathbf{y}, \mathbf{y} \circledast \mathbf{y}, \mathbf{x}, \mathbf{y}] \in \mathbb{R}^{n \times 5}$$

is assumed to have full column rank, and $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$. The cost function (13) can be treated as the sum of the squared algebraic residuals for all points to the ellipse defined by $\boldsymbol{\alpha}$. At the minimum, the gradient of $C(\boldsymbol{\alpha})$ vanishes, i.e.,

$$\nabla C(\boldsymbol{\alpha}) = 2\boldsymbol{\alpha}^T \widetilde{Z}^T \widetilde{Z} - 2 \cdot \mathbf{1}^T \widetilde{Z} = \mathbf{0}.$$

Thus, $\boldsymbol{\alpha}$ is the solution of the linear system of equations

$$(\widetilde{Z}^T \widetilde{Z})\boldsymbol{\alpha} = \widetilde{Z}^T \mathbf{1}.$$

We refer this method as the direct least-algebraic-residuals method (DLAR). Both direct methods described are efficient and avoid convergence issues that may arise with iterative methods. However, they do not minimize geometric distances and the computed solutions may be sensitive to the presence of outliers in the data.

## 3. Trigonometric Polynomial-Based Methods

We will fit curves with the parametric representation of equation (1) to $n$ data points $\boldsymbol{z}_i = [x_i, y_i]^T$, $i = 1, 2, \ldots, n$. These curves have zero tilt angle and are centered at the origin. If the data points suggest that the tilt angle is nonvanishing or the center of the curve is not expected to be at the origin, then we standardize the data points by rotating and shifting them so that the parametric representation (1) can be employed.

### *3.1. Data Standardization*

The center $\boldsymbol{z}_c = [x_c, y_c]^T$ of the given points $\boldsymbol{z}_i = [x_i, y_i]^T$, $i = 1, 2, \ldots, n$, is the average of the $x_i$ and the $y_i$, i.e.,

$$\boldsymbol{z}_c = \frac{1}{n} \sum_{i=1}^{n} [x_i, y_i]^T.$$

Let

$$\boldsymbol{z}_i^{(c)} = [u_i, v_i]^T := \boldsymbol{z}_i - \boldsymbol{z}_c = [x_i - x_c, y_i - y_c]^T.$$

We determine the tilt angle $\theta$ of the data set based on the idea that the standard deviation $\sigma$ of the $v_i$ is minimal when $\theta = 0$. The standardized data set can be written as

$$Z_s = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} Z_c^T,$$

where

$$Z_s = \begin{bmatrix} x_1^{(s)} & y_1^{(s)} \\ \vdots & \vdots \\ x_n^{(s)} & y_n^{(s)} \end{bmatrix} \quad \text{and} \quad Z_c = \begin{bmatrix} u_1 & v_1 \\ \vdots & \vdots \\ u_n & v_n \end{bmatrix}.$$

The standard deviation of the $y_i^{(s)}$ is

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \left( y_i^{(s)} \right)^2}{n-1}}.$$

Let

$$\begin{aligned} V(\theta) &= (n-1)\sigma^2 \\ &= \sum_{i=1}^{n}(-u_i \sin\theta + v_i \cos\theta)^2 \\ &= \sum_{i=1}^{n} \left[(u_i \sin\theta)^2 + (v_i \cos\theta)^2 - 2u_i v_i \sin\theta \cos\theta\right] \\ &= \sum_{i=1}^{n} \left[(u_i \sin\theta)^2 + (v_i \cos\theta)^2 - u_i v_i \sin 2\theta\right]. \end{aligned}$$

Then

$$\begin{aligned} V'(\theta) &= 2\sin\theta\cos\theta \sum_{i=1}^{n} u_i^2 - 2\sin\theta\cos\theta \sum_{i=1}^{n} v_i^2 - 2\cos 2\theta \sum_{i=1}^{n} u_i v_i \\ &= \sin 2\theta \sum_{i=1}^{n} u_i^2 - \sin 2\theta \sum_{i=1}^{n} v_i^2 - 2\cos 2\theta \sum_{i=1}^{n} u_i v_i \\ &= \sin 2\theta \sum_{i=1}^{n}(u_i^2 - v_i^2) - 2\cos 2\theta \sum_{i=1}^{n} u_i v_i, \end{aligned}$$

and

$$V''(\theta) = 2\cos 2\theta \sum_{i=1}^{n}(u_i^2 - v_i^2) + 4\sin 2\theta \sum_{i=1}^{n} u_i v_i.$$

Setting $V'(\theta) = 0$ yields

$$\theta_0 = \frac{1}{2}\arctan\left(\frac{2\sum\limits_{i=1}^{n} u_i v_i}{\sum\limits_{i=1}^{n}(u_i^2 - v_i^2)}\right).$$

The tilt angle then is

$$\theta_t = \begin{cases} \theta_0, & \text{if } \; V''(\theta_0) > 0; \\ \theta_0 + \pi/2, & \text{otherwise.} \end{cases} \tag{14}$$

We remark that it is common to employ principal component analysis (PCA) to determine the tilt angle $\theta_t$, which usually requires the application of singular value decomposition to the centralized data, i.e., the mean of each variable is zero. We prove in the following theorem that the tilt angle determined by equation (14) is the same as the angle of the first right singular vector of $Z_c$. Since our method avoids computing the singular value decomposition of $Z_c$, it is much faster than PCA. When applied to an experiment data set of 3,528 points, it is more than 35 times faster.

**Theorem 1.** *The angle determined by equation* (14) *for a centralized data set* $Z_c \in \mathbb{R}^{n\times 2}$ *equals the angle between the horizontal axis and the first right singular vector of* $Z_c$.

*Proof.* Assume $Z_c = PSQ^T$ is the reduced singular value decomposition of $Z_c$, where $S \in \mathbb{R}^{2\times 2}$ is a diagonal matrix with singular values $s_1 > s_2 > 0$ on the diagonal, and $P \in \mathbb{R}^{n\times 2}$ and $Q \in \mathbb{R}^{2\times 2}$ are orthonormal matrices whose columns are the corresponding left and right singular vectors respectively. Then

$$(Z_cQ)^T = (PS)^T,$$

$$Q^T Z_c^T = SP^T = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} p_{11} & \cdots & p_{n1} \\ p_{12} & \cdots & p_{n2} \end{bmatrix}, \tag{15}$$

where $p_{ij}$ is the entry of $P$ at the $i^{th}$ row and $j^{th}$ column. Since $Q^{-1} = Q^T$, equation (15) can be represented as the change from the standard base to a new base formed by the columns of $Q$ for the data points in $Z_c$.

For simplicity, we refer to the coordinates with respective to the first base vector as horizontal coordinates and the second coordinates as vertical coordinates. The standard deviation of the new vertical coordinates is

$$\sigma_q = \sqrt{\frac{s_2^2 \sum\limits_{i=1}^{n} p_{i2}^2}{n-1}}$$

Since $P$ is orthonormal,

$$\sum_{i=1}^{n} p_{i2}^2 = 1.$$

Hence,

$$\sigma_q = \frac{s_2}{\sqrt{n-1}}.$$

Suppose the data represented by equation (15) are rotated by an angle $\alpha \in (-\pi, 0) \cup (0, \pi)$,

$$\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} Q^T Z_c^T = \begin{bmatrix} s_1 p_{11}\cos\alpha - s_2 p_{12}\sin\alpha & \cdots & s_1 p_{n1}\cos\alpha - s_2 p_{n2}\sin\alpha \\ s_1 p_{11}\sin\alpha + s_2 p_{12}\cos\alpha & \cdots & s_1 p_{n1}\sin\alpha + s_2 p_{n2}\cos\alpha \end{bmatrix}.$$

The standard deviation of the rotated vertical coordinates is

$$\sigma_\alpha = \sqrt{\frac{s_1^2 \sin^2\alpha \sum\limits_{i=1}^{n} p_{i1}^2 + s_2^2 \cos^2\alpha \sum\limits_{i=1}^{n} p_{i2}^2 + 2 s_1 s_2 \sin\alpha\cos\alpha \sum\limits_{i=1}^{n} p_{i1} p_{i2}}{n-1}}$$

Since $P$ is orthonormal,

$$\sum_{i=1}^{n} p_{i1}^2 = 1, \text{ and } \sum_{i=1}^{n} p_{i1}p_{i2} = 0.$$

Then

$$\sigma_\alpha = \sqrt{\frac{s_1^2 \sin^2\alpha + s_2^2 \cos^2\alpha}{n-1}} = \sqrt{\frac{(s_1 - s_2)^2 \sin^2\alpha + s_2^2}{n-1}}$$

Now that $s_1 > s_2$ and $\sin^2\alpha > 0$,

$$\sigma_\alpha > \sigma_q.$$

Since $Q$ is orthonormal, the translation of $Z_c$ by equation (15) is invariant in Euclidean space. Thus, we have proven the theorem. □

**Algorithm 1:** Algorithm for Standardizing Data

```
Input  : $Z_0 = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$
$(x_c, y_c) = mean(Z_0)$, where $mean$ is the column-wise operation;
$Z_c = Z_0 - (x_c, y_c) = \begin{bmatrix} u_1 & v_1 \\ \vdots & \vdots \\ u_n & v_n \end{bmatrix}$;
$\theta = \frac{1}{2}\arctan\left(\frac{2\sum_{i=1}^{n} u_i v_i}{\sum_{i=1}^{n}(u_i^2 - v_i^2)}\right)$;
if $\cos 2\theta \sum_{i=1}^{n}(u_i^2 - v_i^2) + 2\sin 2\theta \sum_{i=1}^{n} u_i v_i < 0$ then
    $\theta = \theta + \pi/2$;
$Z = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} Z_0^T - \begin{bmatrix} x_c \\ y_c \end{bmatrix}$;
$Z = Z^T$;
Output: The standardized data $Z$, the center $(x_c, y_c)$ and the tilt angle
        $\theta$ of the original data set.
```

*3.2. Direct Circle Fitting Method*

For a circle, the parameters $t_i$ are polar angles of the data points $(x_i, y_i)$, $i = 1, 2, \dots, n$. The $t_i$ can easily be determined as follows,

$$t_i = \begin{cases} \arctan(y_i/x_i) \ \text{ if } \ x_i > 0, \\ \arctan(y_i/x_i) + \pi \ \text{ if } \ x_i < 0 \ \text{ and } \ y_i \geq 0, \\ \arctan(y_i/x_i) - \pi \ \text{ if } \ x_i < 0 \ \text{ and } \ y_i < 0, \\ \pi/2 \ \text{ if } \ x_i = 0 \ \text{ and } \ y_i > 0, \\ -\pi/2 \ \text{ if } \ x_i = 0 \ \text{ and } \ y_i < 0. \end{cases} \tag{16}$$

When fitting a circle, the parametric form of (1) reduces to

$$\begin{cases} x = r\cos(t), \\ y = r\sin(t), \end{cases}$$

where $r$ is the radius to be determined. Let

$$T = [\cos(t_1), \dots, \cos(t_n), \sin(t_1), \dots, \sin(t_n)]^T.$$

The minimization problem (5) then becomes

$$\hat{r} = \arg\min_{r} \|Tr - \boldsymbol{z}\|_2^2. \tag{17}$$

Let $\rho = \|T\|_2$ and $Q = T/\rho$. The unique solution of (17) can be written as

$$\hat{r} = \rho^{-1} Q^T \boldsymbol{z}. \tag{18}$$

We refer to the use of (16) and (18) for fitting a circle as the direct circle fitting (DCF) method.

*3.3. An Alternating Fitting Method*

When the parameters $t_i$ are not polar angles, we propose the application of an alternating fitting method (AFM). We are particularly interested in applying this method to fit ellipses and dumbbell curves. The method is suitable for fitting other kinds of shapes such as olives and pillows as well; see illustrations in Section 3.5.

Given the $t_i$'s, one can solve the minimization problem

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \|T\boldsymbol{a} - \boldsymbol{z}\|_2^2, \tag{19}$$

where the matrix $T$ is defined in (4). If a weighting matrix $W$ is introduced, then one may solve the $\ell_1$-regularized problem instead, i.e.,

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \left\{ \|T\boldsymbol{a} - \boldsymbol{z}\|_2^2 + \mu \|W\boldsymbol{a}\|_1 \right\}, \tag{20}$$

where $\mu > 0$ is a regularization parameter and the nonzero entries of the diagonal matrix $W \in \mathbb{R}^{12\times 12}$ are referred to as weights.

Conversely, given the coefficient vector $\hat{\boldsymbol{a}}$, one can determine a set of parameters $-\pi < t_i \leq \pi$, $1 \leq i \leq n$, by solving

$$\hat{\boldsymbol{t}} = \arg\min_{\boldsymbol{t}} \|T(\boldsymbol{t})\hat{\boldsymbol{a}} - \boldsymbol{z}\|_2^2. \tag{21}$$

Let

$$T_i = \begin{bmatrix} \tilde{T}_i & 0 \\ 0 & \tilde{T}_i \end{bmatrix},$$

where $\tilde{T}_i$ is the $i^{th}$ row of $\tilde{T}$ as in (3). Define

$$\hat{t}_i = \arg\min_{t_i} \|T_i\hat{\boldsymbol{a}} - \boldsymbol{z}_i\|_2^2 \tag{22}$$

It is easy to see that $\hat{\boldsymbol{t}} := [\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_n]^T$ is a solution of (21).

We describe a heuristic way of choosing the matrix $W$ in Section 3.5 and of choosing $\mu$ in Section 4.4. Our reason for applying $\ell_1$-regularization is that the computed solution is less sensitive to outliers in the data than when no regularization or $\ell_2$-regularization are used. Moreover, $\ell_1$-regularization promotes sparsity of the computed solution. The proof of uniqueness of the solution to problem (21) for ellipse fitting is immediate. The uniqueness for dumbbell fitting follows from Theorems 2 and 3 in Section 3.5. The proposed AFM solves the minimization problems (5) or (6) by solving the problems (19) or (20), and problem (21) in an alternating fashion. The convergence of the AFM is shown in Section 3.6.

### *3.4. AFM for Ellipse Fitting*

For an ellipse, the parametric form (1) reduces to

$$\begin{cases} x = a_{11}\cos(t), \\ y = a_{22}\sin(t), \end{cases} \qquad -\pi < t \leq \pi, \tag{23}$$

where $a_{11}, a_{22} > 0$ are coefficients to be determined. The matrix $T$ and vector $\boldsymbol{a}$ in (19) are of the form

$$T = \begin{bmatrix} \cos(t_1) & 0 \\ \cos(t_2) & 0 \\ \vdots & \vdots \\ \cos(t_n) & 0 \\ 0 & \sin(t_1) \\ 0 & \sin(t_2) \\ \vdots & \vdots \\ 0 & \sin(t_n) \end{bmatrix} \quad \text{and} \quad \boldsymbol{a} = \begin{bmatrix} a_{11} \\ a_{22} \end{bmatrix}.$$

The AFM for the solution of the minimization problem (5) alternates between solving the problems (19) and (21): Given the vector $\boldsymbol{t} = [t_1, t_2, \ldots, t_n]^T$ of parameter values, the unique solution of problem (19) can be evaluated by using a QR factorization of the matrix $T$. When, instead, the coefficient vector $\hat{\boldsymbol{a}}$ is given, one may solve problem (21) for $\boldsymbol{t}$ by an interior-point or trust-region method; see, e.g., [10, 11, 12]. Alternatively, one may solve (22) for each $t_i$, $1 \leq i \leq n$, independently so that $\hat{t}_i$ yields the shortest distance between the point $(x_i, y_i)$ and the ellipse defined by $\boldsymbol{a}$. We propose to solve (22) in the following manner. Let $d_i$ be the squared distance between a point $(x_{t_i}, y_{t_i})$ on the defined ellipse, and the given point $(x_i, y_i)$. Then

$$d_i = (x_i - x_{t_i})^2 + (y_i - y_{t_i})^2, \tag{24}$$

where

$$\begin{cases} x_{t_i} = a_{11}\cos(t_i), \\ y_{t_i} = a_{22}\sin(t_i). \end{cases}$$

To minimize (24), we require

$$\frac{\mathrm{d}}{\mathrm{d}t_i} d_i = 2a_{11}\sin(t_i)(x_i - a_{11}\cos(t_i)) - 2a_{22}\cos(t_i)(y_i - a_{22}\sin(t_i)) = 0. \tag{25}$$

When $\cos(t_i) \neq 0$, equation (25) is equivalent to

$$a_{11}x_i\tan(t_i) - (a_{11}^2 - a_{22}^2)\sin(t_i) - a_{22}y_i = 0. \tag{26}$$

It may be convenient to compute the solution of the nonlinear equation (26) by a trust-region algorithm, such as Powell's dogleg method described in [11, 12]. However, both approaches mentioned above for determining $\boldsymbol{t}$ are time-consuming when the number of points $n$ is large. We therefore propose the following fast method for computing an accurate approximate minimizer of (24): Let $(x_{t_i}, y_{t_i})$ be the intersection point of an ellipse and the segment that connects the center of the ellipse and a given point $(x_i, y_i)$. We approximate the minimizer of (24) by testing either an increasing or a decreasing sequence for $t_i$, with the polar angle of $(x_i, y_i)$ as its starting value, until $d$ starts to increase. An outline of this scheme is

1. Calculate $\theta_i$, the polar angle of $(x_i, y_i)$.
2. Calculate the point $(x_{t_i}, y_{t_i})$ on the ellipse with $t_i = \theta_i := \theta$.
3. Calculate $d$, the distance between $(x_i, y_i)$ and $(x_{t_i}, y_{t_i})$.
4. Increase $\theta$ by a small step $\delta > 0$, i.e., let $\theta = \theta_i + \delta$, and update $d$.
5. If $d$ decreases, then compute $d$ for $\theta = \theta_i + n_t\delta$, where $n_t = 2, 3, \cdots$ until $d$ increases.
6. If $d$ increases, then compute $d$ for $\theta = \theta_i - n_t\delta$, where $n_t = 2, 3, \cdots$ until $d$ increases.
7. If necessary, then adjust the value of $\theta$ so that $\theta \in (-\pi, \pi]$.

8. $t_i := \theta$ is an approximate minimizer of equation (24).

Details of the above process are described by Algorithm 2. Timings that compare this algorithm to an interior-point method and a trust-region dogleg method are presented in Table 2 of Section 4. We refer to the ellipse-fitting method that uses Algorithm 2 as AFM-LS since it solves the least-squares problem (19). We show in Section 3.6 that Algorithm 2 is a convergent descent method.

**Algorithm 2:** Fast Algorithm for Determining the Parameter $t$ for a Point on the Ellipses Closest to $(x_0, y_0)$.

```
Input  : A point $(x_0, y_0)$, coefficients $a_{11}$, $a_{22} > 0$ that define the
         ellipse (23), $t_0$ the polar angle of $(x_0, y_0)$, $\delta > 0$ step size for $t$,
         and the squared distance function
         $d(t) = (a_{11}\cos(t) - x)^2 + (a_{22}\sin(t) - y)^2$.
if $x0 = 0$ then
    if $y_0 = 0$ then
        $t_m = \pi/2$;
    else
        $t_m = t_0$;
else
    $d_0 = d(t_0)$;
    $\theta = t_0 + \delta$   A value close to $t_0$;
    $d_1 = d(\theta)$;
    if $d_0 > d_1$ then
        $s = 1$
    else
        $s = -1$
    for $n_t = 0, 1, 2, \cdots$ do
        $h = d(t_0 + s(n_t + 1)\delta) - d(t_0 + s n_t \delta)$;
        if $h > 0$ then
            $N = n_t$;
    $t_m = t_0 + sN\delta$;
Output: $\sqrt{d(t_m)}$ the approximate minimal distance between $(x_0, y_0)$
        and the ellipse, and $t_m$, the approximate value of the
        parameter $t$ for the point on the ellipse closest to $(x_0, y_0)$.
```

### *3.5. AFM for Dumbbell Curve Fitting*

The fitting process for a dumbbell curve is similar to the one for an ellipse. An approximate solution of (20) can be determined by the fast iterative soft-thresholding method (FISTA) described in [13]. More details are provided below.

We assume the data to be standardized, i.e., the dumbbell curve is assumed to be symmetric about both the $x$- and $y$-axes, and the tangent lines of the dips are horizontal. The symmetry requires

$$a_{12} = a_{21} = a_{31} = a_{32} = a_{41} = a_{42} = a_{52} = a_{61} = 0. \tag{27}$$

Therefore, the parametric form (1) reduces to

$$\begin{cases} x(t) = a_{11}\cos(t) + a_{51}\cos(3t), \\ y(t) = a_{22}\sin(t) + a_{62}\sin(3t), \end{cases} \qquad a_{11}, a_{22} > 0, \quad -\pi < t \leq \pi. \tag{28}$$

We require two more constraints on the coefficients $a_{ij}$ to ensure that the fitted curve is dumbbell-shaped. First, we demand that the curve does not self-intersect above or below the $x$-axis.

**Theorem 2.** *Assume that the coefficient $a_{51}$ in* (28) *is nonvanishing. Then the conditions $a_{11}/a_{51} < -9$ and $a_{11}/a_{51} > 3$ suffice to ensure that the curve defined by* (28) *does not self-intersect for $t \in (0, \pi)$ and $t \in (-\pi, 0)$.*

*Proof.* We consider the case when $t \in (0, \pi)$. Since the curve is symmetric about the $x$-axis, a sufficient condition for the curve $t \to (x(t), y(t))$ for $0 < t < \pi$ not to self-intersect is that $x(t)$ is a decreasing function of $t$ for $t \in (0, \pi)$, i.e., that $\frac{\mathrm{d}x}{\mathrm{d}t} < 0$. Thus, we require

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -a_{11}\sin(t) - 3a_{51}\sin(3t) < 0, \quad t \in (0, \pi).$$

Rearranging the terms and using the fact that $\sin(3t) = 3\sin(t) - 4\sin^3(t)$ yields

$$\sin(t)(a_{11} + 9a_{51} - 12a_{51}\sin^2(t)) > 0.$$

Since $\sin(t) > 0$, we obtain

$$a_{51}\left(\frac{a_{11}}{a_{51}} + 9 - 12\sin^2(t)\right) > 0.$$

If $a_{51} > 0$, then the above inequality implies that

$$\frac{a_{11}}{a_{51}} > 12\sin^2(t) - 9. \tag{29}$$

In order for (29) to hold for all $t \in (0, \pi)$, we obtain the necessary condition

$$\frac{a_{11}}{a_{51}} > 3. \tag{30}$$

If, instead, $a_{51} < 0$, then we obtain the inequality

$$\frac{a_{11}}{a_{51}} < 12\sin^2(t) - 9. \tag{31}$$

Since this inequality holds for all $t \in (0, \pi)$, it follows that

$$\frac{a_{11}}{a_{51}} < -9. \tag{32}$$

The proof for $-\pi < t < 0$ follows similarly. □

We also demand that the curve only intersects the $x$-axis for $t = 0$ and $t = \pi$.

**Theorem 3.** *Assume that $a_{62} \neq 0$. Then either one of the conditions $a_{22}/a_{62} < -3$ or $a_{22}/a_{62} > 1$ secure that the curve defined by (28) only intersects the $x$-axis at $t = 0$ and $t = \pi$.*

*Proof.* Assume that $y(t) = a_{22} \sin(t) + a_{62} \sin(3t) = 0$ for some $0 < t < \pi$. Then using the same manipulations as in the proof of Theorem 2 shows that this equation is equivalent to

$$\sin(t)(a_{22} + a_{62}(3 - 4\sin^2(t))) = 0.$$

We have to ensure that

$$a_{22}/a_{62} + (3 - 4\sin^2(t)) = 0$$

has no solution for $0 < t < \pi$. This is the case when $4\sin^2(t) = a_{22}/a_{62} + 3 < 0$ or $4\sin^2(t) = a_{22}/a_{62} + 3 > 4$ Hence,

$$a_{22}/a_{62} < -3 \text{ or } a_{22}/a_{62} > 1. \tag{33}$$

The proof for $-\pi < t < 0$ follows similarly. □

Theorems 2 and 3 ensure that the curve does not self-intersect. In addition, since the dips of the desired dumbbell curve $t \to (x(t), y(t))$ are symmetric about the $x$-axis, the equation

$$\frac{\mathrm{d}y}{\mathrm{d}t} = a_{22}\cos(t) + 3a_{62}\cos(3t) = 0 \tag{34}$$

has exactly 6 solutions. Using similar manipulations as in the proof of Theorem 2, we obtain that equation (34) is equivalent to

$$a_{22}\cos(t) + 3a_{62}(4\cos^3(t) - 3\cos(t)) = 0,$$

which can be written as

$$\cos(t)(a_{22} + 3a_{62}(4\cos^2(t) - 3)) = 0.$$

The factor $\cos(t)$ vanishes for $t = \pm\pi/2$ and provides two solutions, and the other factor yields the remaining four solutions. In detail, using similar manipulations as above, the second factor gives

$$4\cos^2(t) - 3 = -a_{22}/3a_{62}.$$

Thus, we need the relation

$$4\cos^2(t) = 3 - a_{22}/3a_{62} \in (0, 4)$$

to hold to secure the existence of the other four solutions. Therefore, we require

$$-a_{22}/a_{62} \in (-9, 3),$$

which can be expressed as

$$-3 < a_{22}/a_{62} < 9. \tag{35}$$

Combining (33) and (35), we obtain the constraints

$$1 < a_{22}/a_{62} < 9. \tag{36}$$

We illustrate in Figure 1 different shapes of curves with different ratios of $a_{11}/a_{51}$ and $a_{22}/a_{62}$. Note that when $a_{11}/a_{51} < -9$, the curve appears more like a pillow than a dumbbell; see Figure 1(b). Thus, in our experiments for fitting dumbbell curves, we only consider the case when $a_{11}/a_{51} > 3$. We also note that the closer the ratio $a_{22}/a_{62}$ is to 1, the deeper are the dips of the dumbbell curve. When the ratio is closer to 9, the dips are shallow and the curve appears elliptical; see Figure 1(e). When the ratio keeps increasing above 9, the places where the dips are become "bumps" and an olive-shaped curve begins to form; see Figure 1(f). The constraint (33) in Theorem 2 indicates that when solving the minimization problem (6), the weights for $a_{51}$ and $a_{62}$ should be large enough to secure that the desired inequalities hold. The upper bound of constraint (36) suggests that the weight for $a_{62}$ should not be too large. We describe a heuristic way of choosing weights for $a_{51}$ and $a_{62}$ after introducing the fast iterative soft-thresholding algorithm (FISTA) for solving problem (20).

The AFM for solving problem (6) for dumbbell curve fitting alternates between solving the minimization problems (20) and (21). The former problem is equivalent to

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \left\{ \|TW^{-1}W\boldsymbol{a} - \boldsymbol{z}\|_2^2 + \mu\|W\boldsymbol{a}\|_1 \right\}. \tag{37}$$

Let $T_w = TW^{-1}$ and $\boldsymbol{a}_W = W\boldsymbol{a}$. Then, (37) becomes

$$\hat{\boldsymbol{a}}_W = \arg\min_{\boldsymbol{a}} \left\{ \|T_w\boldsymbol{a}_W - \boldsymbol{z}\|_2^2 + \mu\|\boldsymbol{a}_W\|_1 \right\}, \tag{38}$$

and we solve this problem instead of (37). The solution is computed by the fast iterative soft-thresholding algorithm (FISTA) [13].

FISTA-type methods are designed to solve convex optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \{\mathcal{J}(x) : \mathcal{J}(x) := f(x) + g(x)\},$$

with $f : \mathbb{R}^n \to \mathbb{R}$ a smooth convex function of type $C^{1,1}$, i.e.,

$$\|\nabla f(x) - \nabla f(y)\|_2 \le l_f\|x - y\|_2, \ \forall x, y \in \mathbb{R}^n,$$

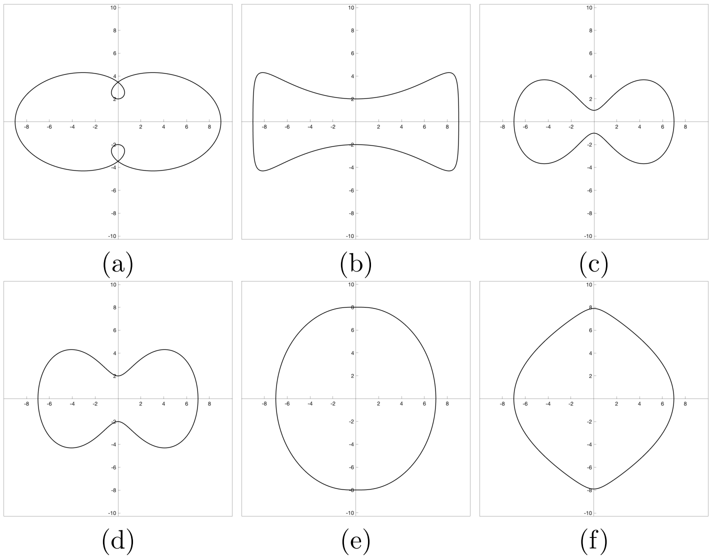


Figure 1: Curves with different coefficients: (a) $a_{11} = 6, a_{51} = 3, a_{22} = 4, a_{62} = 2$, (b) $a_{11} = 10, a_{51} = -1, a_{22} = 4, a_{62} = 2$, (c) $a_{11} = 6, a_{51} = 1, a_{22} = 3, a_{62} = 2$, (d) $a_{11} = 6, a_{51} = 1, a_{22} = 4, a_{62} = 2$, (e) $a_{11} = 6, a_{51} = 1, a_{22} = 9, a_{62} = 1$, (f) $a_{11} = 6, a_{51} = 1, a_{22} = 8, a_{62} = 0.1$. All other coefficients vanish.

where $l_f > 0$ is a Lipschitz constant for the gradient $\nabla f$ and $g : \mathbb{R}^n \to \mathbb{R}$ is a continuous, possibly nonsmooth, convex function. The general idea of FISTA is that, after having determined the point $x^{(k)} \in \mathbb{R}^n$ at iteration $k$, an additional point $u^{(k)} \in \mathbb{R}^n$ is chosen as a linear combination of $x^{(k)}$ and the previous point $x^{(k-1)} \in \mathbb{R}^n$. Then

$$x^{(k+1)} = T_{l_f}\left(u^{(k)}\right) := \arg\min_{x \in \mathbb{R}^n} \left\{ Q_{l_f}(x, u^{(k)}) + g(x) \right\},$$

i.e., $x^{(k+1)}$ is the unique minimizer of $Q_{l_f}(x, u) + g(x)$ at $u = u^{(k)}$, where

$$x \to Q_{l_f}(x, u) := f(u) + \langle x - u, \nabla f(u) \rangle + \frac{l_f}{2} \|x - u\|_2^2$$

is a quadratic approximation of $f(x)$. Since the term $f(u^{(k)})$ in $T_{l_f}(u^{(k)})$ is independent of $x$, we have

$$\begin{aligned} x^{(k+1)} &= T_{l_f}(u^{(k)}) \\ &= \arg\min_{x \in \mathbb{R}^n} \left\{ Q_{l_f}(x, u^{(k)}) + g(x) \right\} \\ &= \arg\min_{x \in \mathbb{R}^n} \left\{ \langle x - u^{(k)}, \nabla f(u^{(k)}) \rangle + \frac{l_f}{2} \|x - u^{(k)}\|_2^2 + g(x) \right\} \\ &= \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{l_f}{2} \left\| x - (u^{(k)} - \frac{1}{l_f} \nabla f(u^{(k)})) \right\|_2^2 + g(x) \right\}. \end{aligned}$$

Let

$$f(x) = \|Ax - b\|_2^2 \quad \text{and} \quad g(x) = \mu\|x\|_1, \tag{39}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. Let $\lambda_{\max}(A^TA)$ denote the largest eigenvalue of $A^TA$. Then $l_f = 2\lambda_{\max}(A^TA)$ is the smallest Lipschitz constant of $\nabla f$. The quotient

$$L_f = \frac{l_f}{s}$$

serves as step size, where $0 < s \leq 1$ is a scalar step size. Beck and Teboulle [13] show that with constant step size $L_f = l_f$, FISTA achieves a convergence rate of $\mathcal{O}(1/k^2)$, where $k$ denotes the number of iterations performed. The operator $T_{L_f}(u)$ may be considered a proximal regularization operator for the non-smooth $\ell_1$-regularized problem

$$\min_{x\in\mathbb{R}^n} \left\{\|Ax - b\|_2^2 + \mu\|x\|_1\right\}. \tag{40}$$

Since the $\|x\|_1$-term in (40) is separable, we obtain

$$x^{(k+1)} = T_{L_f}(u^{(k)}) = \Phi_{s\mu/l_f}\left(u^{(k)} - \frac{s}{l_f}\nabla f(u^{(k)})\right),$$

where $\Phi_\alpha(x)$, for $\alpha > 0$, denotes the soft-thresholding operator, i.e.,

$$\Phi_\alpha(x) = [\phi_\alpha(x_1), \phi_\alpha(x_2), \ldots, \phi_\alpha(x_n)]^T \in \mathbb{R}^n$$

with $x = [x_1, x_2, \ldots, x_n]^T$ and

$$\phi_\alpha(x) := \begin{cases} 0 & \text{if } |x| \leq \alpha, \\ \operatorname{sign}(x)(|x| - \alpha) & \text{if } |x| > \alpha; \end{cases}$$

see [13] for details. For the functions $f(x)$ and $g(x)$ given by (39), the iterations of FISTA, with $x^{(0)} = x^{(1)} = 0$ and $t_0 = t_1 = 1$, can be written as

$$\left\{\begin{array}{l} z^{(k)} = x^{(k)} + \left(\frac{t_{k-1}-1}{t_k}\right)\left(x^{(k)} - x^{(k-1)}\right), \\ u^{(k)} = z^{(k)} - \frac{2s}{l_f}A^T(Az^{(k)} - b), \\ x^{(k+1)} = \Phi_{s\mu/l_f}(u^{(k)}), \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \end{array}\right. \qquad k = 1, 2, \ldots\ ;$$

see [13]. The computations are described in Algorithm 3. problem (40) becomes (38) when $x = \boldsymbol{a}_W, A = T_w$ and $b = \boldsymbol{z}$. Through experiments, we found that when employing FISTA, the weight of each edge should be set to the reciprocal value of the ratio of the corresponding constraints (30) and (36). For instance, when fitting a dumbbell curve, the coefficients $a_{11}$ and $a_{22}$ are not related by a constraint. Therefore, their weights should be the same, i.e., we may set $w_{11} = w_{22} = 1$. Since

$$\frac{a_{11}}{a_{51}} = \frac{5}{1}$$

satisfies (30), and

$$\frac{a_{22}}{a_{62}} = \frac{2}{1}$$

satisfies (36), we set $w_{51} = 5$ and $w_{62} = 2$. We specify in Section 4 how the parameters are chosen in the computed examples.

```
Algorithm 3: Fast Iterative Soft-Thresholding Algorithm (FISTA)
Input : $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $\mu > 0$, $l_f = 2\lambda_{\max}(A^TA)$, $s \in (0,1]$,
        $t_0 = t_1 = 1$, $tol > 0$, and the soft-thresholding operator
        $\Phi_{s\mu/l_f}(\cdot)$.
$x^{(0)} = x^{(1)} = 0 \in \mathbb{R}^n$;
for $k = 1, 2, \ldots$ do
    $z^{(k)} = x^{(k)} + \frac{t_{k-1} - 1}{t_k}(x^{(k)} - x^{(k-1)})$;
    $u^{(k)} = z^{(k)} - \frac{2s}{l_f}A^T(Az^{(k)} - b)$;
    $x^{(k+1)} = \Phi_{s\mu/l_f}(u^{(k)})$;
    if $\|x^{(k+1)} - x^{(k)}\|_2 \le tol\|x^{(k)}\|_2$ then
        exit;
    $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$;
Output: Approximate solution $x^{(k+1)}$.
```

We employ Algorithm 1 to standardize the data. When applied to dumbbell curve fitting, the scheme for estimating the $t$'s for ellipses does not guarantee a global minimal distance between the point and the fitted curve, but only a local one; see the discussion in Section 3.6. We therefore propose a simple sampling method as an alternative; see Algorithm 5. We refer to the algorithm for fitting a curve with weights via FISTA as AFM-FISTA, which is described in Algorithm 4.

We also note that, for an ellipse or dumbbell curve centered at the origin with no tilt, a solution of (22) $\hat{t}_i \in (-\pi, \pi]$ is unique when $\boldsymbol{z}_i$ is on neither of the axes. In the case when $\boldsymbol{z}_i$ is on either one of the axes, one may choose to only keep the smallest solution of $\hat{t}_i$ such that algorithms such as Algorithm 2 and Algorithm 5 generate unique solutions $\hat{t}_i$'s, and thus determine a unique solution $\hat{\boldsymbol{t}} := [\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_n]^T$ of (21) even though the solutions to those problems themselves are not necessarily unique.

### *3.6. Convergence of Iterative Alternating Methods*

This subsection shows several results on the convergence of the proposed iterative alternating methods.

**Algorithm 4:** Iterative Alternate Method with FISTA (AFM-FISTA) for Weighted Curve Fitting

```
Input  : $Z_0 = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}^T$, soft-thresholding parameter $\alpha$, weighting
         matrix $W$, tolerance of the relative difference of error $\tau$, and
         the initial relative difference of error $r^{(0)} = \infty$.
Apply Algorithm 1 on $Z_0$ for the center $(center_x, center_y)$, tilt angle $\theta$
  of the curve, and the standardized data $Z$;
Calculate $T^{(0)}$ with $Z$ via eq. (16);
Let $\boldsymbol{z}$ be the column-stacked vector determined by the columns of $Z$;
for $k = 1, 2, 3 \ldots$ do
    $T_w^{(k-1)} = T^{(k-1)} W^{-1}$;
    $\mu = 2\alpha\lambda_{max}(T_w^{(k-1)T} T_w^{(k-1)})$;
    Apply FISTA to problem (38) to compute $\boldsymbol{a}_W^{(k)}$, an estimate of $\boldsymbol{a}_W$;
    $\boldsymbol{a}^{(k)} = W^{-1}\boldsymbol{a}_W^{(k)}$;
    for $i = 1, \ldots, n$ do
        Apply Algorithm 5 for $t_i^{(k)}$ and $d_i^{(k)}$ with $\boldsymbol{a}^{(k)}$ and $T^{(k-1)}$;
    $T^{(k)}$ is obtained;
    Fitting error $e^{(k)} = \sum_{i=1}^{n} d_i^{(k)}$;
    if $r^{(k)} = \| \frac{e^{(k)} - e^{(k-1)}}{e^{(k-1)}} \| \leq \tau$ then
        $\boldsymbol{a} = \boldsymbol{a}^{(k)}$ and $e = e^{(k)}$;
Form $A$ from $\boldsymbol{a}$;
Output: Fitting error $e$ and the parametric form of the best fit
        dumbbell curve

        $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} center_x \\ center_y \end{bmatrix} + \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$
            $\left( \begin{bmatrix} \cos(t) & \sin(t) & \cos(2t) & \sin(2t) & \cos(3t) & \sin(3t) \end{bmatrix} A \right)^T$.
```

**Theorem 4.** *Let $\Gamma$ be a curve in the $(x, y)$-plane that is symmetric about both the $x$- and $y$-axes. Consider a point $P_M = (x_M, y_M)$ on $\Gamma$, and let the point $P_0 = (x_0, y_0) \in \mathbb{R}^2$ be in a different quadrant than $P_M$. Then, there exists a point $P_m = (x_m, y_m)$ on $\Gamma$ in the same quadrant as $P_0$ such that $\overline{P_0P_m} < \overline{P_0P_M}$, where $\overline{UV}$ denotes the distance between the points $U$ and $V$.*

*Proof.* We may without loss of generality assume that $P_0$ is in the first quadrant, i.e., $x_0, y_0 > 0$. Consider the following three cases:

**Case 1:** Let $P_M$ be in the second quadrant, i.e., $x_M < 0$ and $y_M > 0$. Define $x_m = -x_M$ and $y_m = y_M$. Then $P_m = (x_m, y_m)$ is in the first quadrant. Since $\Gamma$ is symmetric about the $y$-axis, $P_m$ is also on $\Gamma$. We have

$$\begin{aligned}\overline{P_0P_M}^2 &= (x_0 - x_M)^2 + (y_0 - y_M)^2,\\ \overline{P_0P_m}^2 &= (x_0 - x_m)^2 + (y_0 - y_m)^2\\ &= (x_0 + x_M)^2 + (y_0 - y_M)^2.\end{aligned}$$

It follows that

$$\overline{P_0P_M}^2 - \overline{P_0P_m}^2 = -4x_0x_M > 0$$

and, therefore,

$$\overline{P_0P_m} < \overline{P_0P_M}.$$

**Case 2:** Let $P_M$ be in the third quadrant, i.e., and $x_M < 0$ and $y_M < 0$. Define $x_m = -x_M$ and $y_m = -y_M$. Then $P_m$ is on $\Gamma$ and in the first quadrant. It follows that

$$\begin{aligned}\overline{P_0P_m}^2 &= (x_0 - x_m)^2 + (y_0 - y_m)^2\\ &= (x_0 + x_M)^2 + (y_0 + y_M)^2\end{aligned}$$

and, therefore,

$$\overline{P_0P_M}^2 - \overline{P_0P_m}^2 = -4x_0x_M - 4y_0y_M > 0,$$

which shows that

$$\overline{P_0P_m} < \overline{P_0P_M}.$$

**Case 3:** Let $P_M$ be in the fourth quadrant, i.e., $x_M > 0$ and $y_M < 0$. Define $x_m = x_M$ and $y_m = -y_M$. Then $P_m$ is on $\Gamma$ and in the first quadrant. We have

$$\begin{aligned}\overline{P_0P_m}^2 &= (x_0 - x_m)^2 + (y_0 - y_m)^2\\ &= (x_0 - x_M)^2 + (y_0 + y_M)^2.\end{aligned}$$

Hence,

$$\overline{P_0P_M}^2 - \overline{P_0P_m}^2 = -4y_0y_M > 0,$$

and consequently

$$\overline{P_0P_m} < \overline{P_0P_M}.$$

The proofs for the cases when $P_0$ is in one of the other quadrants are similar. □

**Theorem 5.** *Let the point $P_0 = (x_0, y_0)$ be in one of the quadrants of the $(x, y)$-plane. The closest point to $P_0$ on a non-self-intersecting curve $\Gamma$ defined by equation* (28) *is in the same quadrant.*

*Proof.* Suppose that $P_0$ is in the first quadrant, i.e., $x_0, y_0 > 0$. Let $P(t) = (x(t), y(t))$ for $-\pi < t \leq \pi$ be a parametric representation of $\Gamma$ and define the squared distance function

$$d(t) = ((x(t) - x_0)^2 + (y(t) - y_0)^2, \quad -\pi < t \leq \pi. \tag{41}$$

Then

$$\begin{aligned} d'(t) = & \, 2(a_{11}\cos(t) + a_{51}\cos(3t) - x_0)(-a_{11}\sin(t) - 3a_{51}\sin(3t)) \\ & + 2(a_{22}\sin(t) + a_{62}\sin(3t) - y_0)(a_{22}\cos(t) + 3a_{62}\cos(3t)). \end{aligned}$$

Hence,

$$d'(0) = -2y_0(a_{22} + 3a_{62})$$

and

$$d'(\pi/2) = 2x_0(a_{11} - 3a_{51}).$$

If $a_{62} = 0$, then $d'(0) < 0$, and if $a_{51} = 0$, then $d'(\pi/2) > 0$. Moreover, if $a_{62} \neq 0$, then the constraint (33) ensures that $d'(0) < 0$. Finally, if $a_{51} \neq 0$, then the constraints (30) and (32) ensure that $d'(\pi/2) > 0$. Since $d'(t)$ is a continuous function, there is some $t_0 \in (0, \pi/2)$ such that

$$d'(t_0) = 0$$

and

$$d(t_0) \leq d(t), \quad \forall t \in (0, \pi/2).$$

Theorem 4 ensures that for any point

$$P_M = (x(t_M), y(t_M)), \quad t_M \in (-\pi, -\pi/2) \cup (-\pi/2, 0) \cup (\pi/2, \pi)$$

on $\Gamma$, one can find a point

$$P_m = (x(t_m), y(t_m)), \quad t_m \in (0, \pi/2),$$

on $\Gamma$ in the first quadrant such that $\overline{P_0P_m} < \overline{P_0P_M}$, i.e., $d(t_m) < d(t_M)$. It follows that

$$d(t_0) \leq d(t_m) < d(t_M), \quad \text{where} \;\; t_0, t_m \in (0, \pi/2).$$

The proofs for the cases when $P_0$ is in one of the other three quadrants are similar. □

**Theorem 6.** *With the notation of Algorithm 2 it holds that the algorithm stops after a finite number of steps and, if $n_t < N$,*

$$d(t_0 + s(n_t + 1)\delta) < d(t_0 + sn_t\delta).$$

*Therefore, Algorithm 2 is a descent method. Moreover, let $t_m^\delta$ denote the output of Algorithm 2 with parameter $\delta$, and let $t_m^*$ be the global minimizer of the squared distance function* (41)*. Then it holds*

$$\lim_{\delta \to 0} t_m^\delta = t_m^*.$$

*Proof.* Algorithm 2 details how we determine the closest point on an ellipse centered at the origin, with zero tilt angle to a given point $P_0 = (x_0, y_0)$ in the $(x, y)$-plane. The algorithm considers two situation:

(i) $P_0$ is in one of the four quadrants;

(ii) $P_0$ is on one of the axes or at the origin.

The proof below is divided into four cases. Case 1 is for the situation (i), and Cases 2 to 4 are for the situation (ii).

We express the ellipse (23) in standard form,

$$\frac{x^2}{a_{11}^2} + \frac{y^2}{a_{22}^2} = 1.$$

Without loss of generality, we may assume that $a_{11} > a_{22} > 0$. Consider for now the piece of the ellipse where $0 \leq x \leq a_{11}$ and $y \geq 0$. Expressing $y$ in terms of $x$ gives

$$y = \sqrt{a_{22}^2 - \frac{a_{22}^2}{a_{11}^2} x^2}.$$

Then the squared distance between $P_0$ and a point on this piece of the ellipse is

$$d(x) = (x - x_0)^2 + \left( \sqrt{a_{22}^2 - \frac{a_{22}^2}{a_{11}^2} x^2} - y_0 \right)^2.$$

The first and second derivatives of $d(x)$ are

$$\begin{aligned} d'(x) &= 2(x - x_0) + 2 \left( \sqrt{a_{22}^2 - \frac{a_{22}^2}{a_{11}^2} x^2} - y_0 \right) \frac{1}{2} \left( a_{22}^2 - \frac{a_{22}^2}{a_{11}^2} x^2 \right)^{-1/2} \left( -2 \frac{a_{22}^2}{a_{11}^2} x \right) \\ &= 2x - 2x_0 - 2 \frac{a_{22}^2}{a_{11}^2} x \left( 1 - y_0 \left( a_{22}^2 - \frac{a_{22}^2}{a_{11}^2} x^2 \right)^{-1/2} \right) \\ &= 2x - 2x_0 - 2 \frac{a_{22}^2}{a_{11}^2} x + 2y_0 \frac{a_{22}}{a_{11}^2} x \left( 1 - \frac{x^2}{a_{11}^2} \right)^{-1/2} \end{aligned}$$

and

$$\begin{aligned} d''(x) &= 2 - 2\left(\frac{a_{22}}{a_{11}}\right)^2 + 2y_0\frac{a_{22}}{a_{11}}\left[\left(1-\frac{x^2}{a_{11}^2}\right)^{-1/2} - \frac{x}{2}\left(1-\frac{x^2}{a_{11}^2}\right)^{-3/2}(-2x)\right] \\ &= 2 - 2\left(\frac{a_{22}}{a_{11}}\right)^2 + 2y_0\frac{a_{22}}{a_{11}}\left[\left(1-\frac{x^2}{a_{11}^2}\right)^{-1/2} - x^2\left(1-\frac{x^2}{a_{11}^2}\right)^{-3/2}\right] \\ &= 2 - 2\left(\frac{a_{22}}{a_{11}}\right)^2 + 2y_0\frac{a_{22}}{a_{11}}\left(1-\frac{x^2}{a_{11}^2}\right)^{-1/2}\left(1+\frac{x^2}{1-\frac{x^2}{a_{11}^2}}\right), \end{aligned}$$

respectively.

In the following, we show that there is a unique global minimizer of $d(x)$.

**Case 1:** Let $P_0$ be in the first quadrant, i.e., $x_0, y_0 > 0$. Consider the last factor of the last term of $d'(x)$. One has

$$\lim_{x\to a_{11}^-}\left(1-\frac{x^2}{a_{11}^2}\right)^{-1/2} = \infty \tag{42}$$

and, therefore,

$$\lim_{x\to a_{11}^-} d'(x) = \infty > 0. \tag{43}$$

Moreover,

$$\lim_{x\to 0} d'(x) = d'(0) = -2x_0 < 0. \tag{44}$$

Thus, there exists some $x_m \in (0, a_{11})$ such that $d'(x_m) = 0$.

Now consider $d''(x)$. It follows from $a_{11} > a_{22}$ that

$$2 - 2\left(\frac{a_{22}}{a_{11}}\right)^2 > 0,$$

and, therefore,

$$\left(1-\frac{x^2}{a_{11}^2}\right)^{-1/2} > 0 \quad \text{and} \quad 1+\frac{x^2}{1-\frac{x^2}{a_{11}^2}} > 0.$$

Thus, $d''(x) > 0$ for $0 < x < a_{11}$. It follows that $d'(x)$ is monotonically increasing for $0 < x < a_{11}$, $x_m$ is a unique minimizer of $d(x)$, and $d(x)$ is monotonically decreasing for $0 \le x \le x_m$. Since the points

$$P_m = \left(x_m, \sqrt{a_{22}^2 - \frac{a_{22}^2}{a_{11}^2}x_m^2}\right)$$

and $P_0$ are in the same quadrant, Theorem 5 ensures that $x_m$ is a global minimizer of $d(x)$.

**Case 2:** Let $P_0$ be on the positive $y$-axis, i.e., $x_0 = 0$ and $y_0 > 0$. The limits (42) and (43) are still valid, and the limit (44) becomes

$$\lim_{x \to 0} d'(x) = d'(0) = -2x_0 = 0. \tag{45}$$

Since the expression for $d''(x)$ is independent of $x_0$, it still holds that $d''(x) > 0$. This property together with (45) shows that $d'(x)$ is positive on this interval. Consequently, $d(x)$ is monotonically increasing for $0 \le x \le a_{11}$. Hence, $x_m = 0$ is a unique minimizer of $d(x)$.

It is easy to show that $x_m = 0$ also is a global minimizer. Let the point $P_3 = (x', y')$ be on the ellipse in the third quadrant. Then $P_2 = (x', -y')$ is on the ellipse in the second quadrant and $P_1 = (-x', -y')$ is on the ellipse in the first quadrant. Since $P_0 = (0, y_0)$ is on the positive $y$-axis, we have

$$\overline{P_0P_3} > \overline{P_0P_2} = \overline{P_0P_1} = \sqrt{d(-x')} \ge \sqrt{d(x_m)} = \overline{P_0P_m}.$$

**Case 3:** Let the point $P_0 = (x_0, y_0)$ be on the positive $x$-axis, i.e., $x_0 > 0$ and $y_0 = 0$. Then

$$d'(x) = 2x - 2x_0 - 2\frac{a_{22}^2}{a_{11}^2}x = 2\left[x\left(1 - \frac{a_{22}^2}{a_{11}^2}\right) - x_0\right].$$

Thus,

$$\lim_{x \to 0} d'(x) = d'(0) = -2x_0 < 0,$$

and

$$\lim_{x \to a_{11}^-} d'(x) = 2\left[a_{11}\left(1 - \frac{a_{22}^2}{a_{11}^2}\right) - x_0\right].$$

Moreover, since $y_0 = 0$, one has

$$d''(x) = 2\left(1 - \frac{a_{22}^2}{a_{11}^2}\right) > 0.$$

To show the existence of a unique minimizer $x_m \in [0, a_{11}]$ of $d(x)$, we consider three subcases:

**Subcase 3a:** Assume that $0 < x_0 < a_{11}\left(1 - \dfrac{a_{22}^2}{a_{11}^2}\right)$. Then

$$\lim_{x \to a_{11}^-} d'(x) > 0.$$

Similarly to the argument in Case 1, there is a unique minimizer $x_m \in (0, a_{11})$ of $d(x)$, $d(x)$ is monotonically decreasing for $0 \le x \le x_m$, and $d'(x_m) = 0$.

**Subcase 3b:** Assume that $x_0 = a_{11}\left(1 - \dfrac{a_{22}^2}{a_{11}^2}\right)$. Then

$$\lim_{x \to a_{11}^-} d'(x) = 0.$$

Since $d''(x) > 0$ and $d'(x)$ is continuous, $d'(x) < 0$ on $[0, a_{11})$. Thus, $d(x)$ is strictly decreasing on $[0, a_{11})$ until $x \to a_{11}^-$. It follows that $x_m = a_{11}$ is a unique minimizer.

**Subcase 3c:** Assume that $x_0 > a_{11}\left(1 - \dfrac{a_{22}^2}{a_{11}^2}\right)$. Then

$$\lim_{x \to a_{11}^-} d'(x) < 0.$$

Since $d''(x) > 0$ and $d'(x)$ is continuous, $d'(x) < 0$ on $[0, a_{11})$. Thus, $d(x)$ is strictly decreasing on $[0, a_{11}]$. It follows that $x_m = a_{11}$ is a unique minimizer.

It is easy to show that $x_m = a_{11}$ also is a global minimizer of all subcases of Case 3. Let the point $P_2 = (x', y')$ be in the second quadrant on the ellipse. Then $P_1 = (-x', y')$ is on the ellipse in the first quadrant. Since $P_0 = (x_0, 0)$ is on the positive $x$-axis, it follows that

$$\overline{P_0P_2} > \overline{P_0P_1} = \sqrt{d(-x')} \geq \sqrt{d(x_m)} = \overline{P_0P_m}.$$

**Case 4:** Let the point $P_0 = (x_0, y_0)$ be at the origin, i.e., $x_0 = y_0 = 0$. Then

$$d'(x) = 2x - 2\frac{a_{22}^2}{a_{11}^2}x = 2x\left(1 - \frac{a_{22}^2}{a_{11}^2}\right).$$

It follows that $d'(x) = 0$ if and only if $x = 0$. Since $d''(0) > 0$, $x_m = 0$ is the unique minimizer of $d(x)$. By a similar argument as in Case 3, $x_m = 0$ also is a global minimizer.

Algorithm 2 determines the exact global minimizer for Cases 2 to 4. For Case 1, the algorithm searches for a point that reduces the squared distance function $d(t)$, see (41), and terminates once $d(t)$ starts to increase. It is immediate to see that, since $\delta > 0$ and the function $d(t)$ is monotonically increasing or decreasing on each quadrant, the algorithm stops after a finite number of iterations, $N^\delta$, that depends on $\delta$. Let $t_m^*$ denote the global minimizer of $d(t)$, and let $t_m^\delta$ be the approximation obtained with the algorithm. By construction, we have

$$|t_m^\delta - t_m^*| \leq 2\delta.$$

Therefore,

$$0 \leq \lim_{\delta \to 0} |t_m^\delta - t_m^*| \leq \lim_{\delta \to 0} 2\delta = 0,$$

which concludes the proof. □

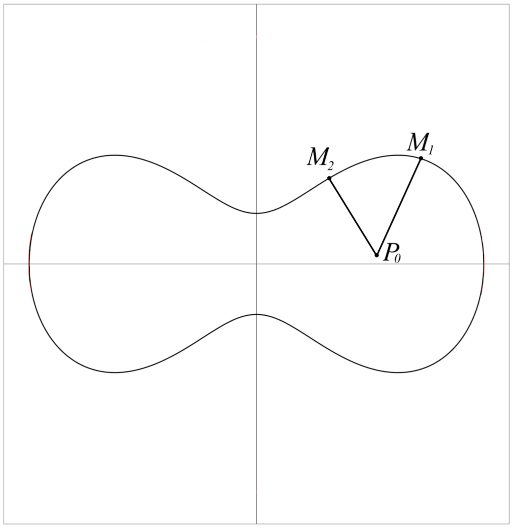


Figure 2: $P_0 = (x_0, y_0)$ is a point in the $(x, y)$-plane, and $M_1$ and $M_2$ are two points on a dumbbell curve that correspond to local minimizers of the squared distance function (46) in the first quadrant.

When replacing the squared distance function by

$$d(t) = (a_{11}\cos(t) + a_{51}\cos(3t) - x_0)^2 + (a_{22}\sin(t) + a_{62}\sin(3t) - y_0)^2 \quad (46)$$

for dumbbell curves, Algorithm 2 is not guaranteed to determine a global minimize, but only a local one. This is illustrated by Figure 2: Let $P_0 = (x_0, y_0)$ be a point in the first quadrant of the $(x, y)$-plane, and let $M_1$ and $M_2$ be two points on the dumbbell curve that correspond to local minima of (46). Algorithm 2 terminates once it has determined $t_m$, an approximation of the parameter value $t$ for the point $M_1$. However, $\overline{P_0M_1} > \overline{P_0M_2}$. Therefore, $t_m$ is not a global minimizer for (46).

To determine an approximation of a global minimum, we apply a simple sampling method. The idea is to sample points $(x(t), y(t))$ on the dumbbell curve for equidistant values of $t$ in $-\pi < t \leq \pi$, see (28), in one of the four quadrants, possibly including a point on the axes, and to choose the value of the parameter $t$ that produces the smallest squared distance to a given data point $(x_0, y_0)$; see Algorithm 5 for details. As the number of sample points increases, Algorithm 5 will produce more accurate approximations of a global minimizer of the squared distance function.

To show convergence of AFM, we first present the 2-block Gauss-Seidel (GS) method described in [14]. Suppose that $g(a_1, \ldots, a_m, t_1, \ldots, t_n) : \mathbb{R}^{m+n} \to \mathbb{R}$ is a continuously differentiable function. Let $\boldsymbol{a} = [a_1, \ldots, a_m]$ and $\boldsymbol{t} = [t_1, \ldots, t_n]$ be the two components, i.e., the two blocks, of the variables of $g$. The 2-block GS method minimizes the objective function $g$ by alternating between fixing one

**Algorithm 5:** Fast Algorithm of Estimation of the Parameter $t$ for Dumbbell Curves

```
Input  : A point $(x_0, y_0)$, coefficients $a_{11}$, $a_{51}$, $a_{22}$, and $a_{62}$ that define
         the dumbbell curve (28), $n_t \geq 2$ the number of sample points
         of $t$, the squared distance function $d(t) =$
         $(a_{11}\cos(t) + a_{51}\cos(3t) - x_0)^2 + (a_{22}\sin(t) + a_{62}\sin(3t) - y_0)^2$.
$\delta = \pi/2(n_t - 1)$;
if $x_0 \geq 0$ and $y_0 \geq 0$ then
    $t_0 = 0$;
if $x_0 \geq 0$ and $y_0 < 0$ then
    $t_0 = -\pi/2$;
if $x_0 \leq 0$ and $y_0 \leq 0$ then
    $t_0 = -\pi$;
if $x_0 \leq 0$ and $y_0 > 0$ then
    $t_0 = \pi/2$;
for $i = 0, 1, \cdots, n_t - 1$ do
    $d_i = d(t_0 + i\delta)$
Find $m$ such that $d_m$ is the minimum of all $d_i$'s;
$t_m = t_0 + m\delta$;
Output: $\sqrt{d(t_m)}$ the approximate minimal distance between $(x_0, y_0)$
        and the dumbbell curve, and $t_m$ the approximate solution of
        the parameter.
```

block of variables at a time; see Algorithm 6 for details. The proposed AFM-LS scheme is a 2-block GS method since the objective function in problem (5) can be written as

$$f(\boldsymbol{t},\boldsymbol{a}) = f(t_1,\dots,t_n,a_{11},\dots,a_{62}) = \sum_{i=1}^{n}(Z_{x,i}^2 + Z_{y,i}^2), \tag{47}$$

where

$$Z_{x,i} = \sum_{j=1}^{3}(a_{2j-1,1}\cos(jt) + a_{2j,1}\sin(jt)) - x_i,$$

$$Z_{y,i} = \sum_{j=1}^{3}(a_{2j-1,2}\cos(jt) + a_{2j,2}\sin(jt)) - y_i,$$

and $(x_i, y_i)$ is the $i^{th}$ given data point in the $(x,y)$-plane. It is clear that the function $f(\boldsymbol{t},\boldsymbol{a})$ is continuously differentiable.

The minimization problem

$$\arg\min_{\boldsymbol{t},\boldsymbol{a}}\{F(\boldsymbol{t},\boldsymbol{a}) = f(\boldsymbol{t},\boldsymbol{a}) + \mu\|W\boldsymbol{a}\|_1\},$$

where $\boldsymbol{t} = [t_1, t_2, \dots, t_n]^T \in (-\pi,\pi]^n$ and $\boldsymbol{a} = [a_{11}, a_{21}, \dots, a_{61}, a_{12}, a_{22}, \dots, a_{62}]^T$, is equivalent to problem (5) when $\mu = 0$ and to problem (6) when $\mu > 0$. The following result is a rewording of [14, Theorem6.3(ii)].

**Theorem 7.** *Suppose that global minimization of the function $g(\boldsymbol{a},\boldsymbol{t})$ with respect to each component of the argument vectors is well defined. Then the 2-block GS method generates an infinite sequence $\{\boldsymbol{a}^{(k)},\boldsymbol{t}^{(k)}\}_{k=0}^{\infty}$ such that if the level set of $g$ corresponding to the initial point $[\boldsymbol{a}^{(0)},\boldsymbol{t}^{(0)}]$,*

$$\mathcal{L} = \{[\boldsymbol{a},\boldsymbol{t}] \in \mathbb{R}^{m+n} : g(\boldsymbol{a},\boldsymbol{t}) \le g(\boldsymbol{a}^{(0)},\boldsymbol{t}^{(0)})\},$$

*is compact, then $\lim_{k\to\infty}\nabla g(\boldsymbol{a}^{(k)},\boldsymbol{t}^{(k)}) = 0$ and there exists at least one limit point that is a stationary point of $g$.*

**Algorithm 6:** 2-block Gauss-Seidel Method

```
Input  : $g([\boldsymbol{a},\boldsymbol{t}])$ and initial guess $[\boldsymbol{a}^{(0)},\boldsymbol{t}^{(0)}] \in \mathbb{R}^{m+n}$.
for $k = 1, 2, \dots$ do
    $\boldsymbol{a}^{(k)} = \arg\min_{\boldsymbol{a}} g(\boldsymbol{a}, t^{(k-1)})$;
    $\boldsymbol{t}^{(k)} = \arg\min_{\boldsymbol{t}} g(\boldsymbol{a}^{(k)}, \boldsymbol{t})$;
Output: $[\boldsymbol{a}^{(k)},\boldsymbol{t}^{(k)}] \in \mathbb{R}^{m+n}$.
```

To apply Theorem 7 to AFM-LS, we need the following result.

**Proposition 1.** *Let the function $f$ be defined by* (47)*. Then the level set*

$$\mathcal{L} = \Big\{[\boldsymbol{a}, \boldsymbol{t}] \in \mathbb{R}^{12+n} : f(\boldsymbol{a}, \boldsymbol{t}) \le f(\boldsymbol{a}^{(0)}, \boldsymbol{t}^{(0)})\Big\}$$

*of $f$ corresponding to the point $[\boldsymbol{a}^{(0)}, \boldsymbol{t}^{(0)}]$ is compact, where we assume that the level set is non-empty.*

*Proof.* The proposition follows from the continuity of the function (47). □

**Proposition 2.** *AFM-LS generates a sequence $[\boldsymbol{a}^{(k)}, \boldsymbol{t}^{(k)}]$, $k = 0, 1, \dots$ , that converges to a stationary point $[\boldsymbol{a}^{(K)}, \boldsymbol{t}^{(K)}]$ of $f$ as in* (47)*, which is another form of the objective function of* (5)*.*

*Proof.* Problems (19) and (21) are minimization problems with respect to the two components of problem (5). Thus, the first assumption of Theorem 7 is satisfied.

By Proposition 1, the level set $\mathcal{L}$ of $f$ corresponding to the initial point $[\boldsymbol{a}^{(0)}, \boldsymbol{t}^{(0)}]$ is compact. Hence, the second assumption of Theorem 7 is satisfied.

The function $f$ is continuously differentiable. Therefore, by Theorem 7,

$$\lim_{k\to\infty} \nabla f(\boldsymbol{a}^{(k)}, \boldsymbol{t}^{(k)}) = 0.$$

The proposition is thus established. □

Recall that, for the model matrix $\tilde{T} \in \mathbb{R}^{n\times 6}$ in (3), we assume that the parameters $t_i$ are in the interval $(-\pi, \pi]$. Moreover, we assume that the fitted curve does not intersect itself. These two assumptions secure that any point on the curve corresponds to a unique parameter value $t_i$. This is trivial for an ellipse, and it is not hard to arrive at the same conclusion for a curve defined by (28) by using Theorem 2. Hence, under the condition that there are at least 6 points, whose closest points on the curve are pair-wise distinct, the matrix $\tilde{T}$ has full column rank. In addition, when these points on the curve are not too close to each other, $\tilde{T}$ is fairly well-conditioned. In this case,

$$T = \begin{bmatrix} \tilde{T} & 0 \\ 0 & \tilde{T} \end{bmatrix}$$

also has full column rank and is fairly well-conditioned. Since we seek to fit a curve to a large number, $n$, of points, one has $n \gg 6$. The two conditions mentioned above are typically satisfied. Therefore, for the remainder of this section, we assume that the iterates $T^{(k)}$ generated by AFM have full column rank and their condition numbers are uniformly bounded.

**Theorem 8.** *Let $\boldsymbol{z} = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]^T \in \mathbb{R}^{2n}$, where $(x_i, y_i)$, $i = 1, 2, \dots, n$, are the $n$ given points, and $n \gg 6$. Suppose that $[\boldsymbol{a}^*, T^*]$ is an optimal solution of* (5)*. Let the matrix $T^{(k)}$ be determined by $\boldsymbol{t}^{(k)}$ as in* (4) *and assume the condition numbers of $T^{(k)}$ are uniformly bounded. Let*

*$\boldsymbol{a}^{(K)} := \lim_{k\to\infty} \boldsymbol{a}^{(k)}$ and $T^{(K)} := \lim_{k\to\infty} T^{(k)}$ be the two components of the stationary point generated via AFM-LS. Then,*

$$\|d_K - d^*\| \le \|\theta \boldsymbol{a}^*\|, \tag{48}$$

*where $d_K = \|T^{(K)}\boldsymbol{a}^{(K)} - \boldsymbol{z}\|$, $d^* = \|T^*\boldsymbol{a}^* - \boldsymbol{z}\|$, and $\theta = T^* - T^{(K)}$.*

*Proof.* Let

$$d' = \|T^{(K)}\boldsymbol{a}^* - \boldsymbol{z}\|.$$

Given $T^{(K)}$, $\boldsymbol{a}^{(K)}$ is the unique solution of

$$\min_{\boldsymbol{a}} \|T^{(K)}\boldsymbol{a} - \boldsymbol{z}\|_2^2.$$

Thus,

$$d_K \le d'.$$

The definition of $d^*$ guarantees

$$d^* \le d_K \le d'. \tag{49}$$

By the Cauchy-Schwarz inequality, we have

$$d' = \|(T^* - \theta)\boldsymbol{a}^* - \boldsymbol{z}\| = \|T^*\boldsymbol{a}^* - \boldsymbol{z} - \theta\boldsymbol{a}^*\| \le d^* + \|\theta\boldsymbol{a}^*\|. \tag{50}$$

If $d' \le d^* + \|\theta\boldsymbol{a}^*\| < d_K$, then $d_K > d'$, contradicting inequality (49). If $d' \le d^* + \|\theta\boldsymbol{a}^*\| = d_K$, then via (49),

$$d' = d_K.$$

Then

$$d' = d^* + \|\theta\boldsymbol{a}^*\| = d_K.$$

It follows that

$$d_K - d^* = \|\theta\boldsymbol{a}^*\|.$$

If $d^* + \|\theta\boldsymbol{a}^*\| > d_K$, then via (49) and (50),

$$d_K \le d' \le d^* + \|\theta\boldsymbol{a}^*\|.$$

Therefore,

$$0 \le d_K - d^* \le \|\theta\boldsymbol{a}^*\|.$$

□

As for AFM-FISTA ($\mu > 0$), since the $\ell_1$-regularization term is not continuously differentiable, we cannot apply Theorem 7. Nevertheless, we have the following results.

**Theorem 9.** *The objective function values of problem* (6) *with* $q = 1$ *generated by the iterates of AFM-FISTA form a strictly monotonically decreasing sequence which converges to a positive value, assuming any two consecutive iterates of* $\boldsymbol{a}$ *are not equal and the condition numbers of all iterates* $T^{(k)}$ *are uniformly bounded.*

*Proof.* Define the objective function value at the $k^{th}$ iteration to be

$$d^{(k)} = \|T^{(k)}\boldsymbol{a}^{(k)} - \boldsymbol{z}\|_2^2 + \mu\|W\boldsymbol{a}^{(k)}\|_1, \quad k = 1, 2, \dots , \tag{51}$$

where $T^{(k)}$ and $\boldsymbol{a}^{(k)}$ are the $k^{th}$ determined by AFM-FISTA. These iterates are calculated at the end of the $k^{th}$ iteration.

Since the $d^{(k)}$ are bounded below by zero, to prove the convergence, we only need to show that the sequence is strictly monotonically decreasing. Based on the analysis of the basic approximation model of the iterative soft-thresholding algorithm (see [13, Section 2.3]), the unique minimizer of problem (20) with the given matrix $T^{(k)}$ is

$$\boldsymbol{a}^{(k+1)} = \arg\min_{\boldsymbol{a}} \left\{ \mu\|W\boldsymbol{a}\|_1 + \frac{L}{2}\left\| \boldsymbol{a} - \left(\boldsymbol{a}^{(k)} - \frac{1}{L}\nabla f(\boldsymbol{a}^{(k)})\right)\right\|^2 \right\},$$

where $f(\boldsymbol{a}) = \|T^{(k)}\boldsymbol{a} - \boldsymbol{z}\|_2^2$, $\nabla f$ denotes the gradient of $f$, and $L = 2\lambda_{max}({T^{(k)}}^T T^{(k)})$. Since $\boldsymbol{a}^{(k+1)} \neq \boldsymbol{a}^{(k)}$, one has

$$d^{(k)} > d_0^{(k+1)} := \|T^{(k)}\boldsymbol{a}^{(k+1)} - \boldsymbol{z}\|_2^2 + \mu\|W\boldsymbol{a}^{(k+1)}\|_1.$$

Let $t_{true_i}^{(k+1)}$ be a true solution that minimizes the distance function $d(t)$ in Algorithm 5 defined by $\boldsymbol{a}^{(k+1)}$ and $\boldsymbol{z}_i$. We have that $t_i^{(k+1)} \to t_{true_i}^{(k+1)}$ as the number of samples increases to infinity.

Since Algorithm 5 determines the unique solution $T^{(k+1)}$ of (21), given $\boldsymbol{a}^{(k+1)}$, one has

$$d_0^{(k+1)} \geq d^{(k+1)}.$$

Hence,

$$d^{(k)} > d^{(k+1)}, \quad k = 1, 2, \dots .$$

□

One can show the following theorem similarly as Theorem 9.

**Theorem 10.** *The objective function values of problem* (5) *generated by the iterates of AFM-LS form a strictly monotonically decreasing sequence, which converges to a positive value, under the assumption that any two consecutive iterates* $\boldsymbol{a}^{(k)}$ *are not equal and the condition numbers of all iterates* $T^{(k)}$ *are uniformly bounded.*

The following theorem gives conditions when a stationary point of AFM is found.

**Theorem 11.** *Define $d^{(k)}$ as in (51) and assume that the condition numbers of all iterates $T^{(k)}$ are uniformly bounded. Consider AFM-LS ($\mu = 0$) and AFM-FISTA ($\mu > 0$). If $T^{(K-1)} = T^{(K)}$ or $\boldsymbol{a}^{(K-1)} = \boldsymbol{a}^{(K)}$ for some $K$, then for all $k \geq K$, one has*

$$d^{(k)} = d^{(k+1)}, \quad T^{(k)} = T^{(k+1)}, \quad \boldsymbol{a}^{(k)} = \boldsymbol{a}^{(k+1)}.$$

*Proof.* We divide the proof of the theorem into two parts.

**Part 1:** Assume that $T^{(K-1)} = T^{(K)}$. Then

$$\begin{aligned} \boldsymbol{a}^{(K)} &= \arg\min_{\boldsymbol{a}} \|T^{(K-1)}\boldsymbol{a} - \boldsymbol{z}\|_2^2 + \mu\|W\boldsymbol{a}\|_1 \\ &= \arg\min_{\boldsymbol{a}} \|T^{(K)}\boldsymbol{a} - \boldsymbol{z}\|_2^2 + \mu\|W\boldsymbol{a}\|_1 \\ &= \boldsymbol{a}^{(K+1)}. \end{aligned}$$

Since the Algorithms 2 and 5 determine a unique solution to problem (21), we have

$$T^{(K)} = T^{(K+1)}. \tag{52}$$

It follows by (51) that $d^{(k)} = d^{(k+1)}$, for $k \geq K$.

**Part 2:** Assume that $\boldsymbol{a}^{(K-1)} = \boldsymbol{a}^{(K)}$. Similarly to the process of reaching (52), one has $T^{(K-1)} = T^{(K)}$. The rest of the proof is the same as in Part 1. □

## 4. Computed Examples

In this section, we seek to fit a curve to the edge of the black hole at the center of galaxy M87 [15], the edge of the Andromeda galaxy [16], and the edges of two different brightnesses of MyCn18 [17], a young planetary nebula; see the first column of Figure 3 for the images of these celestial bodies. Each of the original color images from NASA.org has three color channels, referred to as red, green and blue. For our experiments, we extract data points only from the red channel; see the second column of Figure 3. The pixel values from the red channel vary from 0 to 255, with 0 meaning no red color to 255, meaning the brightest red.

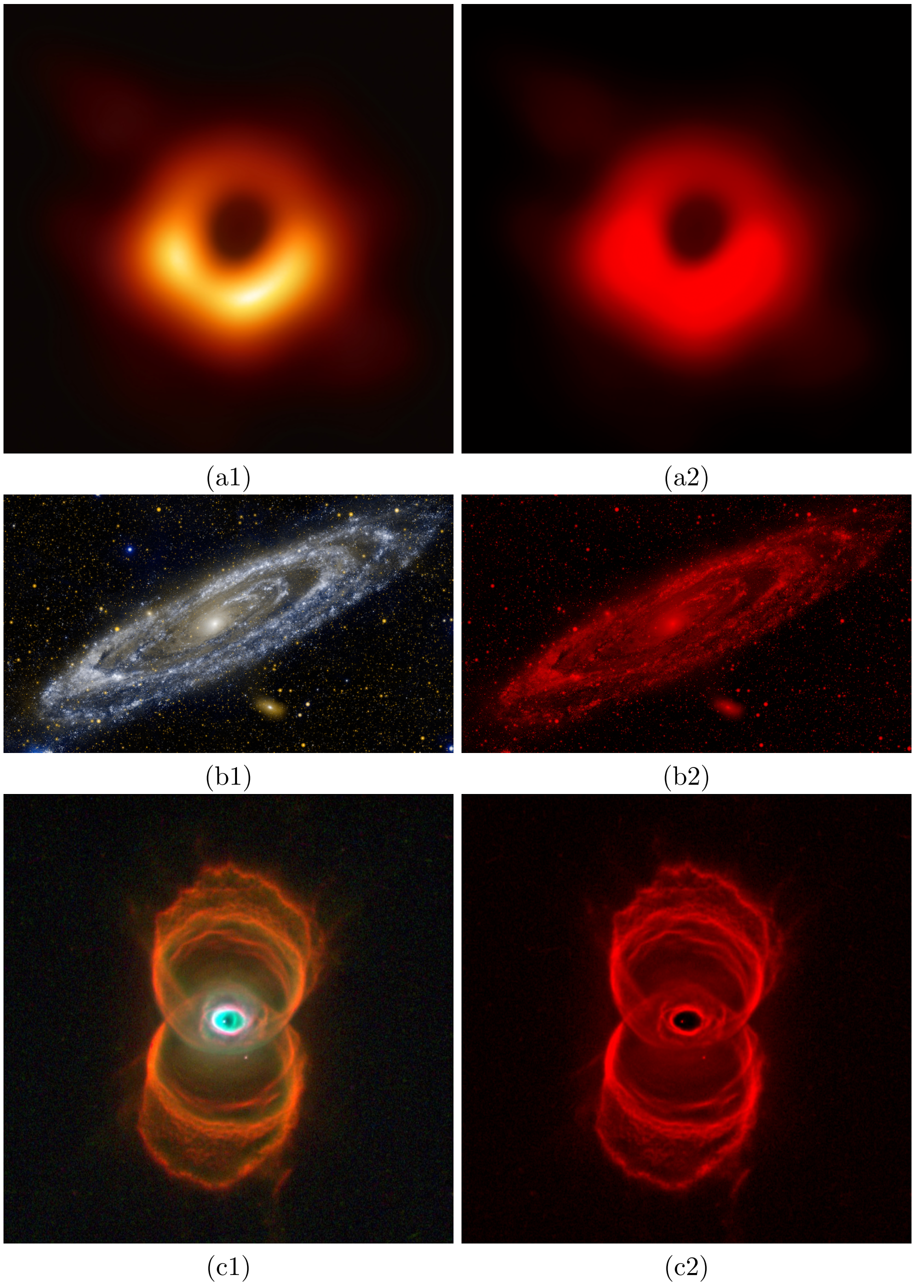
Define the fitting error of the fitted curve generated at the $k^{th}$ iteration as

$$e^{(k)} = \sum_{i=1}^{n} d_i^{(k)}, \tag{53}$$

where $d_i^{(k)}$ is the shortest distance between the point $(x_i, y_i)$ and the curve generated in the iteration. Since the $e^{(k)}$ decrease monotonically with $k$ according to Theorems 9 and 10, we let the stopping criterion for AFM be

$$\frac{e^{(k-1)} - e^{(k)}}{e^{(k-1)}} \leq 10^{-3}.$$

For Algorithm 2, we set the step size to $\delta = \pi/1080$, and for Algorithm 5 the number of sample points is $n_t = 540$.

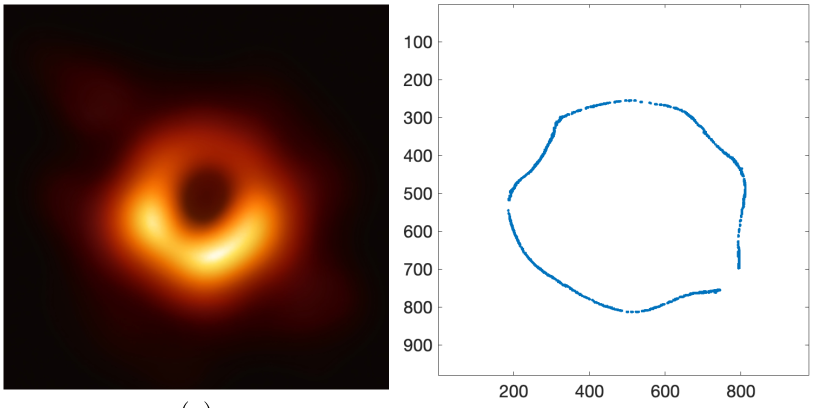


(a1) (a2)

(b1) (b2)

(c1) (c2)

Figure 3: (a1) The black hole at the center of galaxy M87 and (a2) its red channel image; (b1) The Andromeda galaxy and (b2) its red channel image; (c1) MyCn18 and (c2) its red channel image.



(a)

(b)

Figure 4: (a) The black hole at the center of M87, and (b) The points where the pixel value in the red channel equals 60.

The experiments are run on an Apple MacBook Air laptop with an M2 8-Core 3.49 GHz chip and 8 GB of RAM, using macOS Ventura 13.0 and 64-bit arithmetic. All computations are carried out in Rosetta-translated MATLAB R2022b with about 15 significant decimal digits.

### *4.1. Circle Fitting*

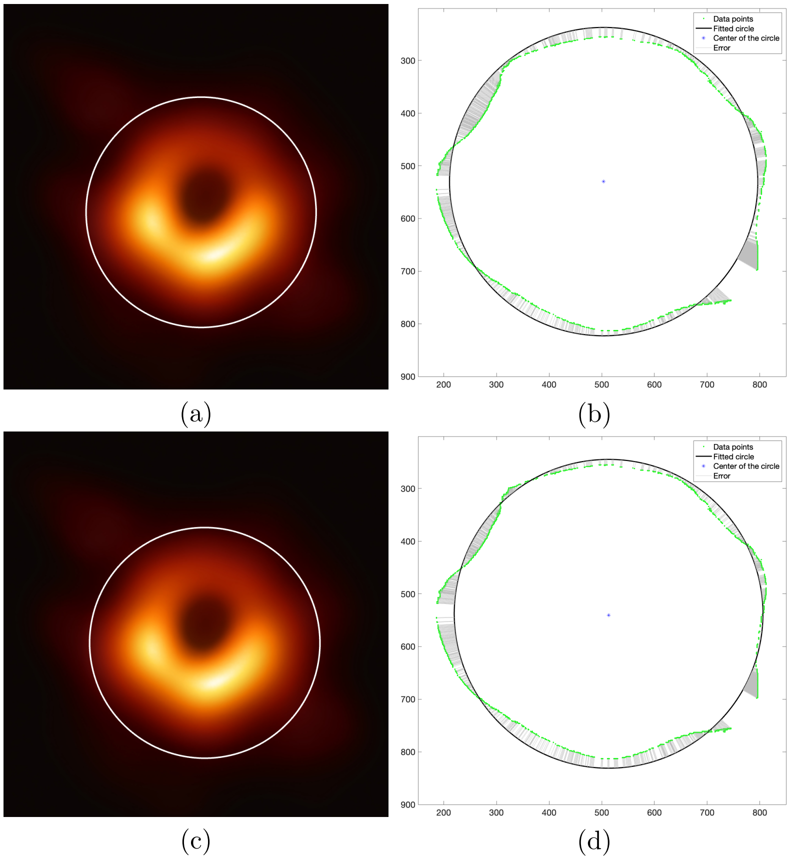
Figure 4 (a) is the central portion of an image of the black hole at the center of galaxy M87 taken by the Event Horizon Telescope. The original resolution is 2330 × 4000 pixels, from which we downsize the image to about 25% of the original size and crop it to its central region. The resulting resolution is 980 × 980 pixels. We seek to fit a circle to points whose pixel value in the red channel equals 60; see panel (b). There are a total of 1,946 such points. Results for the direct circle-fitting method (DCF), the iterative least-algebraic-residuals (LAR) method, the iterative geometric least-squares (GLS) method described in [6], and the direct geometric least-squares (DGLS) method proposed in [8] are listed in Table 1. DCF and DGLS are at least more than 100 times faster than LAR and GLS while DCF produces the smallest fitting error, which is more than 12% smaller than the next smallest fitting error produced by GLS. The fitted circles and their corresponding individual fitting errors, i.e., the shortest distance from a point to the fitted circle for CDF and GLS are shown in Figure 5. These experiments demonstrate that, compared with other methods, DCF is efficient and accurate.

### *4.2. Methods of Computing the $t_i$'s*

Figure 6(a) shows the main portion of a two-color composite image of the Andromeda galaxy from NASA's Galaxy Evolution Explorer. Here blue represents far-ultraviolet light and orange represents near-ultraviolet light. The

Table 1: Circle fitting test for 1,946 points in Figure 4(b): Method, CPU time in seconds, fitting error, center and radius.

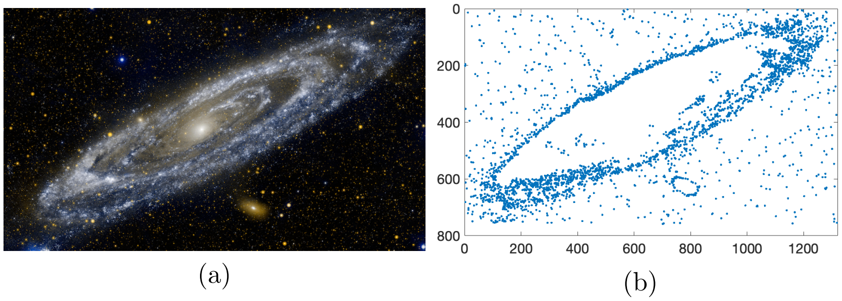
| Method | Time | Error | Center | Radius |
|---|---|---|---|---|
| DCF | $\leq 0.0001$ | 32,022 | (503.5, 529.95) | 292.78 |
| LAR | 0.0109 | 37,804 | (512.95, 540.25) | 293.38 |
| GLS | 0.0091 | 36,556 | (512.87, 537.83) | 293.18 |
| DGLS | $\leq 0.0001$ | 36,801 | (512.2, 538.78) | 293.64 |



(a) (b)

(c) (d)

Figure 5: Fitted circles for data points in Figure 4(b) by (a,b) DCF, and (c,d) GLS. Each gray line segment represents the shortest distance between a point and the fitted circle.



(a)

(b)

Figure 6: (a) The Andromeda galaxy, and (b) the points whose pixel value in the red channel equals 45.

Table 2: Computation of $t_i$'s for the standardized $3,528$ points in Figure 6(b). The major and minor axes of the ellipse are listed in the first line of Table 3. The CPU time is measured in seconds.

| Method | Time | Error |
|---|---|---|
| Algorithm 2 | 0.0117 | 200,556 |
| Interior points | 2.3085 | 529,429 |
| Trust-region | 19.6524 | 729,690 |

resolution of the original image is $7,000 \times 9,400$ pixels, which we downsize to a smaller image about 4% of the original size and crop to show the central region of the original large image. The resulting image has $760 \times 1,320$ pixels. We filter out the points whose pixel value in the red channel equals 45; see panel (b). There are $3,528$ such points, which we first standardize by using Algorithm 1. We then apply Algorithm 2, the interior point method, and the trust-region method to compare their performances.

Table 2 lists results for the computed $t_i$'s, where the ellipse is defined by $a_{11} = 622.26$ and $a_{22} = 163.39$. For each method, the initial guess of the parameter value $t_i$ associated with a given point is obtained by (16). The step size of Algorithm 2 is set to $\delta = \pi/1080$. We measure the error of the computed solution by

$$\sum_{i=1}^{n} \sqrt{(a_{11}\cos(t_i) - x_i)^2 + (a_{22}\sin(t_i) - y_i)^2}.$$

Both the interior point method and the trust-region method terminate the computations prematurely, because they exceed the default upper bound of the number of function evaluations in MATLAB, $8,500$ and $352,800$, respectively. Table 2 shows Algorithm 2 to be faster than the other methods in our comparison and to determine the most accurate approximations of the $t_i$'s.

### *4.3. Ellipse Fitting*

We compare several methods for fitting ellipses to the set of data points shown in Figure 6(b). The test results are listed in Table 3. AFM-LS shows its

Table 3: Ellipse fitting test for the 3,528 points in Figure 6(b): Method, CPU time in seconds, fitting error, center, major axis, minor axis, and the tilt angle of the fitted ellipse in degrees.

| Method | Time | Error | Center | Major Axis | Minor Axis | Tilt |
|---|---|---|---|---|---|---|
| AFM-LS | 0.0417 | 200,556 | (672.43, 379.94) | 622.26 | 163.39 | 24.27 |
| LAR | 6.3654 | 325,955 | (617.44, 366.47) | 480.45 | 244.22 | 163.28 |
| GLS | - | - | - | - | - | - |
| DEF | $\leq 0.0001$ | 314,228 | (655.74, 373.65) | 516.58 | 258.62 | 15.48 |
| DLAR | 0.0013 | 381,297 | (654.96, 373.41) | 613.59 | 328.14 | 8.48 |

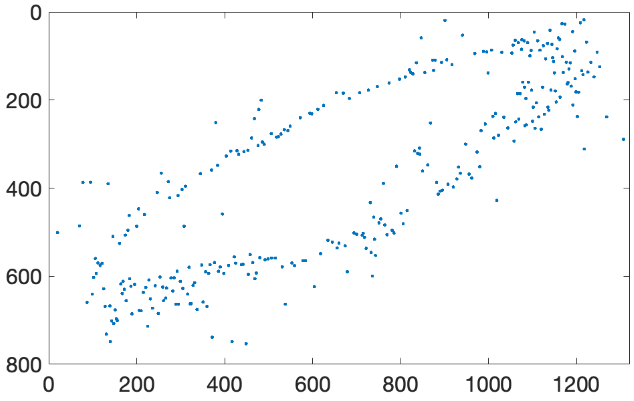


Figure 7: The 312 selected points from Figure 6(b)

superiority over the other methods in our comparison with regard to efficiency, fitting error, and robustness against outliers. LAR is applied with the constraint $a + c = 1$ as proposed in [6]. Due to the amount of noise in the data, GLS fails to detect an ellipse. DEF and DLAR are the fastest of the methods, but the computed ellipses are strongly affected by the outliers and these methods produce large fitting errors.

To compare the fitting results obtained with GLS to those of other methods, we rotate all the points in Figure 6(b) around its center (672.43, 379.94) by 24.27 degrees clockwise, and discard points whose $x$- or $y$-coordinates are more than 650 and 250, respectively, away from the center to reduce the number of outliers. We then rotate all the remaining 3,112 points around the center counterclockwise by 24.27 degrees. GLS still fails to converge within 30 minutes. We therefore reduce the data set by picking every 10th point; see Figure 7 for a display of the resulting 312 points. The results obtained when fitting ellipses to this small set of points are listed in Table 4. GLS produces the smallest fitting error, but AFM-LS is more than 70 times faster and produces a similarly visually appealing ellipse; see Figure 7.

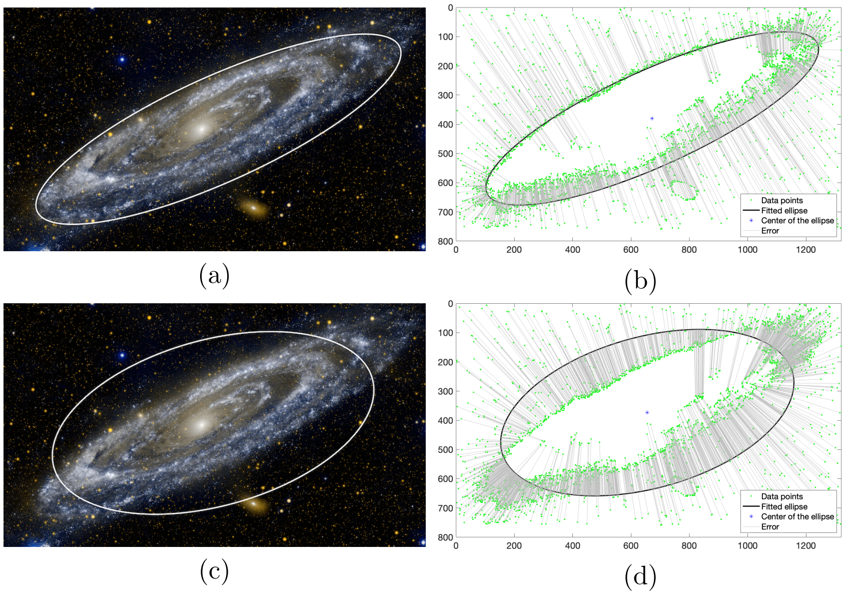


Figure 8: Fitted ellipses for data points in Figure 6(b) by (a,b) AFM-LS, and (c,d) DEF with trace constraint. Each gray line segment shows the distance between the point and the fitted ellipse.

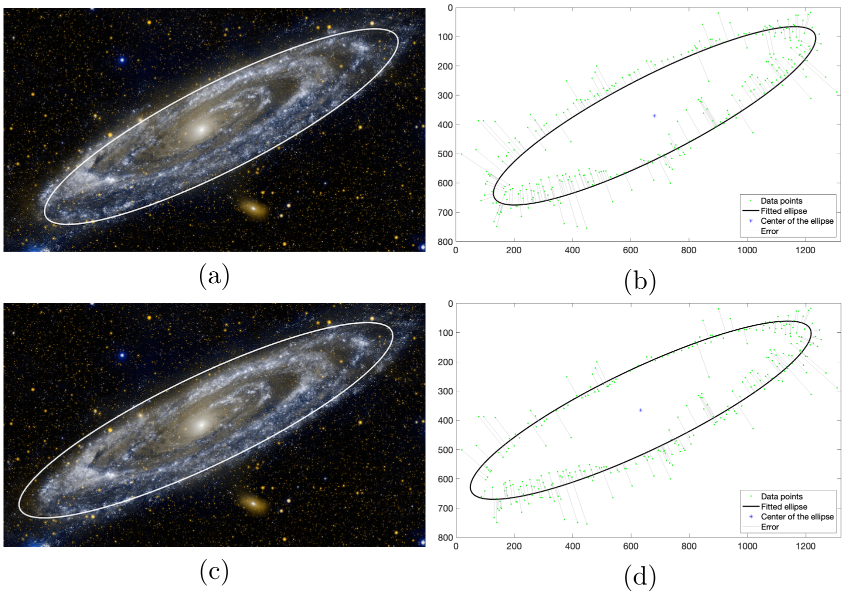


Figure 9: Fitted ellipses for data points in Figure 7 by (a,b) AFM-LS, and (c,d) GLS. Each gray line segment shows the distance between the point and the fitted ellipse.

Table 4: Ellipse fitting test for the 312 points in Figure 7: Method, CPU time in seconds, fitting error, detected center, major axis, minor axis, and the tilt angle of the fitted ellipse in degrees.

| Method | Time | Error | Center | Major Axis | Minor Axis | Tilt |
|---|---|---|---|---|---|---|
| AFM-LS | 0.0039 | 11,519 | (681.95, 370.31) | 616.84 | 135.09 | 26.92 |
| LAR | 0.0076 | 12,718 | (638.65, 355.58) | 533.57 | 149.77 | 153.94 |
| GLS | 0.2755 | 9,466 | (633.63, 365.26) | 644.15 | 139.24 | 154.49 |
| DEF | $\leq 0.0001$ | 12,570 | (647.63, 368.76) | 549.97 | 169.95 | 24.99 |
| DLAR | $\leq 0.0001$ | 13,821 | (648.31, 368.53) | 588.22 | 188.35 | 24.38 |

### *4.4. Dumbbell Curve Fitting*

Figure 10(a) is a Hubble telescope snapshot of MyCn18, a young planetary nebula. The resolution of the image is $1,280 \times 1,280$ pixels. We form two sets of data points from the image. The first and second sets consist of the points whose pixel values in the red channel equal 46 and 160, respectively; see panel (b) and (c). There are $3,528$ points in the first set and $1,601$ in the second.

This example uses FISTA to determine a suitable dumbbell curve. We set the step size in FISTA to $s = 1$ for the fastest global convergence; see [13]. The regularization parameter $\mu$ is defined as

$$\mu = \alpha L_f,$$

where $L_f = 2\lambda_{max}(T_w^T T_w)$ and $\alpha > 0$ is a user-specified parameter. This parameter also is used to define soft thresholding, which is carried out in each iteration of FISTA. Let $\boldsymbol{a}_W$ be the available computed approximate solution and let $\phi_\alpha(\cdot)$ denote the soft-thresholding operator for a specified $\alpha > 0$,

$$\phi_\alpha(w_{ij}a_{ij}) := \begin{cases} 0 & \text{if } |w_{ij}a_{ij}| \leq \alpha, \\ \operatorname{sign}(w_{ij}a_{ij})(|w_{ij}a_{ij}| - \alpha) & \text{if } |w_{ij}a_{ij}| > \alpha. \end{cases}$$

Here $w_{ij}$ is the weight for the coefficient $a_{ij}$. We have found that letting $500\alpha$ be the half-length of the major axis of the dumbbell curve to be a suitable choice. For example, if the half-length is about 5, then we set $\alpha = 0.01$.

In our first experiment, we seek to fit a dumbbell curve to the points in Figure 10(b). The simplest fitting method is to solve the least-squares problem (19) by QR factorization of $T$. We obtain the solution

$$A_{LS} = \begin{bmatrix} 421.8357 & 2.5434 \\ 3.2423 & 309.4641 \\ -13.5107 & -9.1222 \\ 22.8378 & 4.0918 \\ 49.1261 & -5.1334 \\ -2.3615 & 73.0413 \end{bmatrix}.$$

All the coefficients are nonvanishing. The computed dumbbell therefore is not symmetric about its center, because symmetry requires (27) to hold. The computed solution is displayed in Figure 11(a).

We turn to an application of FISTA. Following the heuristics for determining weights for the coefficients discussed at the end of Section 3.5, we set $w_{11} = w_{22} = 1$, $w_{51} = 3.1$, and $w_{62} = 2$. To force the rest of the coefficients to vanish to preserve symmetry, we set their weights to $w_{ij} = 10$. The half-length of the major axis is about 500. We therefore set $\alpha = 1$. The iterations with FISTA are terminated when the relative difference of two consecutive iterates is smaller than $10^{-3}$. We use the same weights and $\alpha$-value for fitting dumbbell curves to points in Figure 10(b) and all the remaining AFM-FISTA fitting tests. For each test, we report the ratio $a_{22}/a_{62}$ as the deepness of the dips of a dumbbell curve. The smaller this ratio is, the deeper are the dips. We list the results in Table 5. Figure 11(c) demonstrates that AFM-FISTA determines dumbbell curves that capture the edge of the nebula and preserve symmetry is preserved. The latter is also evident from the zero entries of the computed solution

$$A_{FISTA} = \begin{bmatrix} 425.7607 & 0 \\ 0 & 307.6677 \\ 0 & 0 \\ 0 & 0 \\ 44.0641 & 0 \\ 0 & 69.9298 \end{bmatrix}.$$

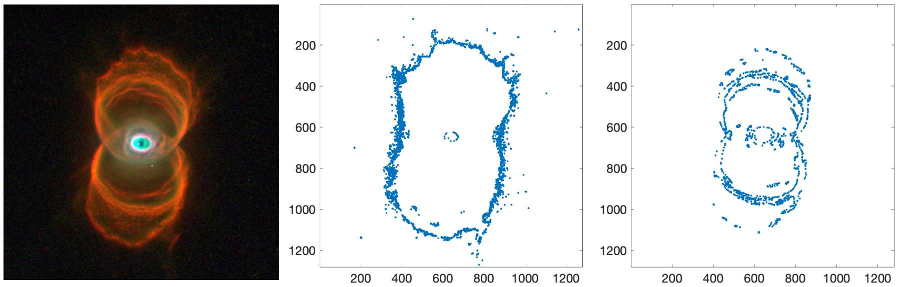
In our second experiment, we seek to fit a dumbbell curve to the points in Figure 10(c). Straightforward solution of the least-squares problem (19) by QR factorization of $T$ gives a self-intersecting curve; see Figure 11(a). All coefficients of the solution are nonvanishing, similarly as in the previous example. AFM-FISTA generates a non-self-intersecting dumbbell curve that is symmetric about the center and captures the brighter edge of the nebula. The solutions determined by this method have many vanishing coefficients,

$$A_{FISTA} = \begin{bmatrix} 253.3355 & 0 \\ 0 & 162.5876 \\ 0 & 0 \\ 0 & 0 \\ 74.9386 & 0 \\ 0 & 94.1630 \end{bmatrix}.$$

### *4.5. Curve Fitting to Subsampled Points*

For the first test in this subsection, we randomly subsampled 10% of the data points in Figure 6(b) five times and fit ellipses to them using AFM-LS. The results demonstrate that the subsampling method is up to more than 10 times faster and generates well-fitted ellipses; see Figure 13 and Table 6.

In the next experiment, we fit dumbbell curves to randomly subsampled 10% of the data points in Figure 10(b) five times using AFM-LS and AFM-FISTA



(a) (b) (c)

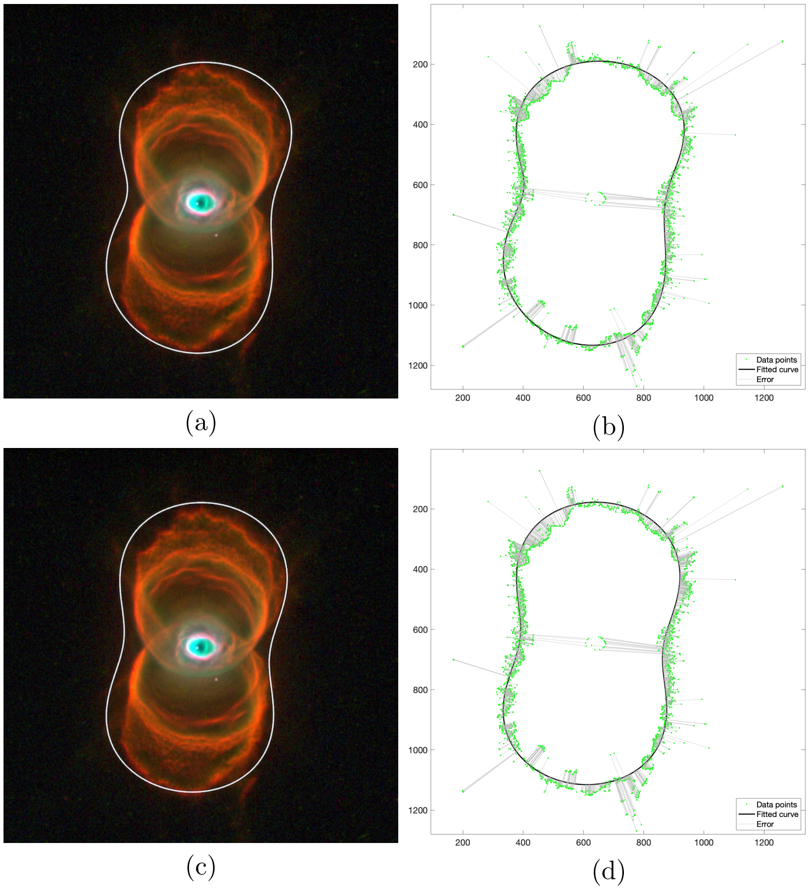
Figure 10: (a) MyCn18, (b) The points where the pixel value in the red channel equals 46, and (c) The points where the pixel value in the red channel equals 160.

Table 5: Results for dumbbell curve fitting to data points in Figure 10(b) and (c): Panel of the figure, number of data points, method, CPU time in seconds, number of iterations, fitting error, center, tilt angle in degrees, and deepness of the dips.

| Panel | Points | Method | Time | Iter. | Error | Center | Tilt | Dip |
|---|---|---|---|---|---|---|---|---|
| (b) | 3,975 | AFM-LS | 2.5558 | 9 | 70,901 | (627.09, 646.39) | 84.78 | 4.24 |
| (b) | 3,975 | AFM-FISTA | 1.3878 | 4 | 79,093 | (627.09, 646.39) | 84.78 | 4.40 |
| (c) | 1,601 | AFM-LS | 3.8086 | 33 | 48,803 | (658.67, 649) | 86.93 | 1.18 |
| (c) | 1,601 | AFM-FISTA | 3.6097 | 26 | 51,589 | (658.67, 649) | 86.93 | 1.73 |

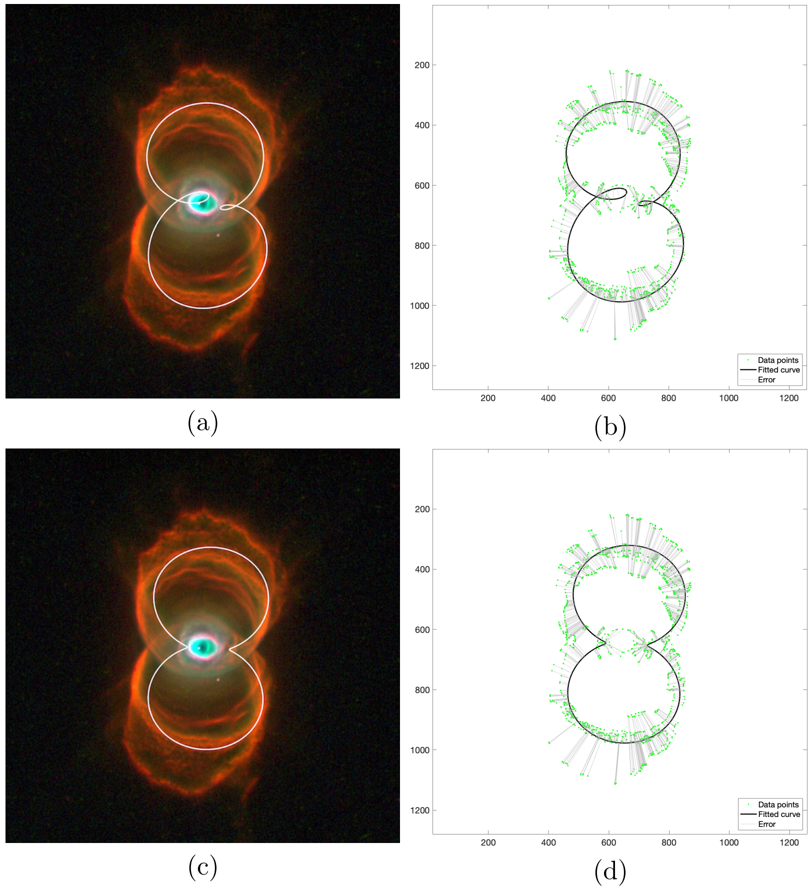
Table 6: Results for fitting ellipses 10% of randomly subsampled points, and to all data points in Figure 6(b) using AFM-LS. The number of data points, CPU time in seconds, number of iterations, fitting error, center of the curve, major axis, minor axis, and the tilt angle in degrees are shown. The first row shows the result when fitting all the data points.

| Points | Time | Iter | Error | Center | Major Axis | Minor Axis | Tilt |
|---|---|---|---|---|---|---|---|
| 3,528 | 0.0417 | 14 | 200,556 | (672.43, 379.94) | 622.26 | 163.39 | 24.27 |
| 353 | 0.0079 | 18 | 194,929 | (692.65, 361.69) | 643.63 | 158.06 | 23.78 |
| 353 | 0.0044 | 14 | 210,027 | (650.76, 390.22) | 626.98 | 173.62 | 22.83 |
| 353 | 0.0050 | 16 | 192,947 | (654.64, 381.15) | 597.23 | 162.19 | 24.41 |
| 353 | 0.0044 | 14 | 235,286 | (705.16, 385.48) | 628.76 | 162.79 | 24.44 |
| 353 | 0.0037 | 11 | 213,995 | (664.87, 393.36) | 603.02 | 160.89 | 24.76 |



(a)
(b)
(c)
(d)

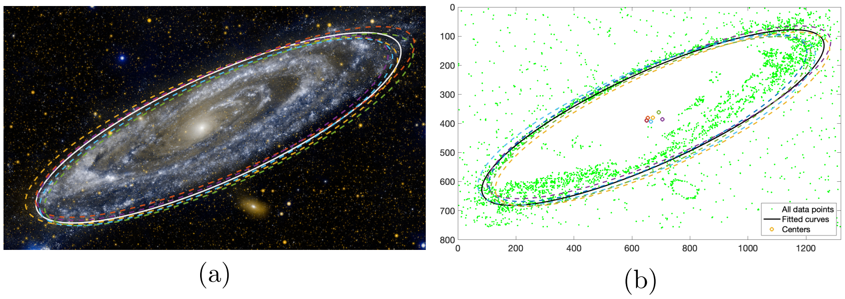
Figure 11: Fitted curves for data points in Figure 10(b) by (a,b) AFM-LS, and (c,d) AFM-FISTA. Each gray line segment represents the shortest distance between a point and the fitted curve.



(a) (b)

(c) (d)

Figure 12: Fitted curves for data points in Figure 10(c) by (a,b) AFM-LS, and (c,d) AFM-FISTA. Each grey line segment represents the shortest distance between a point and the fitted curve.



(a) (b)

Figure 13: Results for fitting ellipses to 10% of randomly subsampled points, and to all data points in Figure 6(b) using AFM-LS. The solid curves and black asterisk are respectively the fitted ellipse and its centers for all the points. The dotted curves and hollow dots are respectively the fitted ellipses and their centers for the subsampled points.

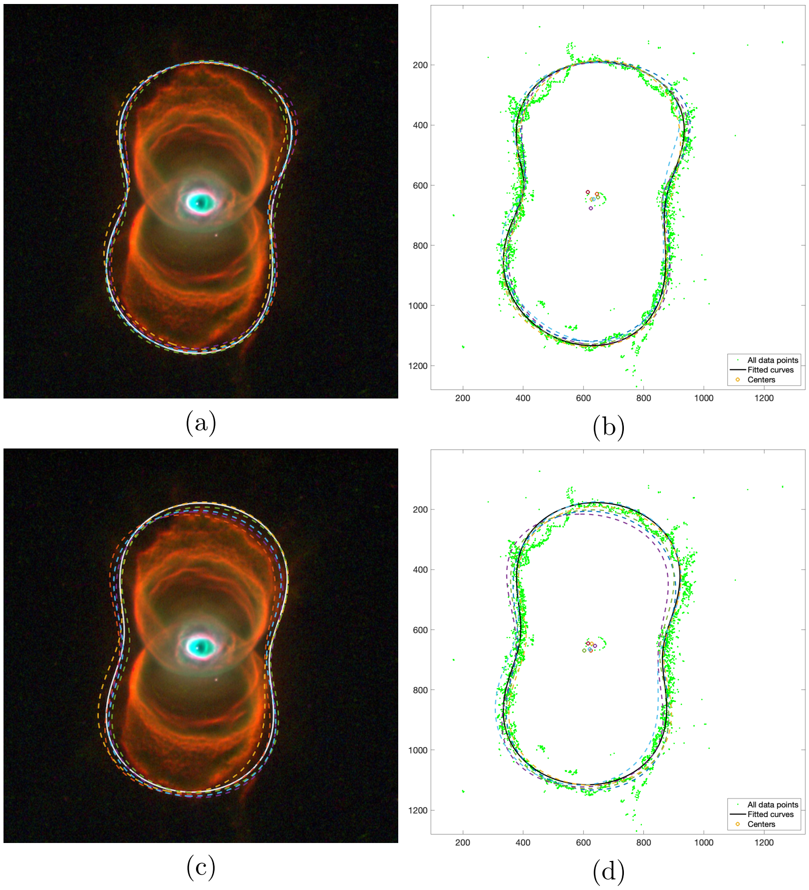
Table 7: Dumbbell curve fitting to randomly subsampled 10% of the data points and all the data points in Figure 10(b). The table shows the number of data points, the method, CPU time in seconds, number of iterations, fitting error, center of the curve, tilt angle in degrees, and the deepness of the dips. The first and the seventh row display result for fitting all the data points.

| Points | Method | Time | Iter. | Error | Center | Tilt | Dip |
|---|---|---|---|---|---|---|---|
| 3,975 | AFM-LS | 2.5558 | 9 | 70,901 | (627.09, 646.39) | 84.78 | 4.24 |
| 398 | AFM-LS | 0.4866 | 14 | 75,198 | (618.40, 663.78) | 79.35 | 4.28 |
| 398 | AFM-LS | 0.2741 | 8 | 68,080 | (639.88 645.00) | 83.19 | 4.26 |
| 398 | AFM-LS | 0.3827 | 11 | 73,198 | (624.11, 661.13) | -86.92 | 4.71 |
| 398 | AFM-LS | 0.5181 | 18 | 76,860 | (613.61, 655.80) | -89.63 | 4.81 |
| 398 | AFM-LS | 0.4601 | 16 | 79,767 | (614.92, 627.94) | 83.57 | 4.58 |
| 3,975 | AFM-FISTA | 1.3878 | 4 | 79,093 | (627.09, 646.39) | 84.78 | 4.40 |
| 398 | AFM-FISTA | 0.1769 | 5 | 94,007 | (626.98, 622.60) | 82.16 | 4.61 |
| 398 | AFM-FISTA | 0.1724 | 5 | 88,197 | (630.18, 666.08) | -87.62 | 4.88 |
| 398 | AFM-FISTA | 0.1140 | 4 | 90,915 | (624.62, 643.87) | -87.86 | 4.77 |
| 398 | AFM-FISTA | 0.1381 | 4 | 105,224 | (620.72, 627.45) | 89.50 | 4.30 |
| 398 | AFM-FISTA | 0.1378 | 4 | 93,980 | (626.11, 662.86) | -84.01 | 5.24 |

with the same weighting matrix $W$ and soft-thresholding value $\alpha$ as in Section 4.4. The subsampling combined with AFM-LS or AFM-FISTA is up to more than 12 times faster than using all points, and generates curves that capture the edge of the nebula; see Figure 14 and Table 7.

## 5. Conclusion

The direct circle-fitting (DCF) method and the iterative alternating methods AFM-LS proposed in this paper minimize the sum of the geometric distances between the generated curves and the given data points. They are efficient methods for fitting curves to a large number of data points. The experimental results show that they are robust against noise in the data. We provide a heuristic technique for choosing weights and the regularization parameter for fitting dumbbell curves using AFM-FISTA by exploring the relationship between the coefficients. We show that AFM-LS converges to a solution that is a stationary point of the objective function. We also demonstrate that the objective function values generated by iterates of AFM form a strictly monotonically decreasing convergent sequence. Compared with other methods for fitting circles and ellipses, DCF and AFM-LS are fast, outlier-resistant, and produce smaller fitting errors. The experimental results for subsampling methods illustrate that they are competitive with regard to CPU time and generally determine suitable curves.



(a) (b)

(c) (d)

Figure 14: Fitted dumbbell curves for 10% of randomly subsampled points and all the data points in Figure 10(b) by (a,b) AFM-LS, and (c,d) AFM-FISTA. The solid curves and black asterisks are respectively the fitted dumbbell curves and their centers for all the points. The dotted curves and hollow dots are respectively the fitted dumbbell curves and their centers for the subsampled points.

## Acknowledgments

The authors would like to thank the referees for comments that lead to an improved presentation. A.B. is a member of the GNCS group of INdAM that partially financed their research with the INdAM-GNCS 2023 Project "Tecniche numeriche per lo studio dei problemi inversi e l'analisi delle reti complesse" (CUP_E53C22001930001). A.B. is partially supported by Fondazione di Sardegna, Progetto biennale bando 2021, "Computational Methods and Networks in Civil Engineering (COMANCHE)".

## References

[1] F. L. Bookstein, Fitting conic sections to scattered data, Computer Graphics and Image Processing 9 (1979) 56–71.

[2] A. W. Fitzgibbon, M. Pilu, R. B. Fisher, Direct least square fitting of ellipses, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 476–480.

[3] A. H. Fitzgibbon, R. B. Fisher, A buyer's guide to conic fitting, in: Proceedings of the British Machine Vision Conference (BMVC'95), Birmingham, England, 1995, pp. 265–271.

[4] J. Porrill, Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter, Image and Vision Computing 8 (1990) 37–41.

[5] P. L. Rosin, A note on the least squares fitting of ellipses, Pattern Recognition Letters 14 (1993) 799–808.

[6] W. Gander, G. H. Golub, R. Strebel, Least-squares fitting of circles and ellipses, BIT Numerical Mathematics 34 (1994) 558–578.

[7] O. Gal, Least-squares criterion for estimation of the best fit to an ellipse, https://www.mathworks.com/matlabcentral/fileexchange/3215-fit_ellipse, 2003. [MATLAB Central File Exchange. Retrieved April 17, 2022].

[8] A. Beck, Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB, SIAM, Philadelphia, 2014.

[9] R. M. Haralick, L. G. Shapiro, Computer and Robot Vision, Addison-Wesley Longman, London, 1992.

[10] R. H. Byrd, J. C. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, Mathematical Programming 89 (2000) 149–185.

[11] J. E. Dennis, R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, Philadelphia, 1996.

[12] M. J. Powell, A Fortran subroutine for solving systems of nonlinear algebraic equations, Technical Report, Atomic Energy Research Establishment, Harwell, United Kingdom, 1968.

[13] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (2009) 183–202.

[14] L. Grippof, M. Sciandrone, Globally convergent block-coordinate techniques for unconstrained optimization, Optimization Methods and Software 10 (1999) 587–637.

[15] R. Garner, What are black holes, National Aeronautics and Space Administration, `https://www.nasa.gov/vision/universe/starsgalaxies/black_hole_description.html`, 2020.

[16] National Aeronautics and Space Administration, The galaxy next door, `https://www.nasa.gov/mission_pages/galex/pia15416.html`, 2017.

[17] National Aeronautics and Space Administration, Hubble finds an hourglass nebula around a dying star, `https://science.nasa.gov/image-detail/amf-pia14442`, 1996.