Numer Algor (2008) 47:1–14
DOI 10.1007/s11075-007-9141-z

ORIGINAL PAPER

An algorithm for fitting circular arcs to data using the l_1 norm

I. A. Al-Subaihi

Received: 17 April 2007 / Accepted: 31 October 2007 /
Published online: 10 January 2008

Abstract There are many applications of fitting circular arcs to data. We have for example, system control, using a computer controlled cutting machine, approximating hulls of boats, drawing and image techniques. Out of these applications comes the least squares norm to be the most commonly used criterion. This paper examines how the l_1 norm is used which seems to be more appropriate than the use of least squares in the context of wild points in the data. An algorithm and different methods to determine the starting points are developed. However, numerical examples are given to help illustrate these methods.

Keywords Circular arc · Gauss-Newton method · l_1 norm

Mathematics Subject Classification (2000) 65D10

1 Introduction

Curve fitting is an important technique in the analysis of discrete data. Fitting circular arcs and straight line segments to measured data using least squares are considered in Cheng et al. [7], Horng [11], Horng and Johnny [12], Huyuh [13], Joseph [15], Pei and Horng [20], Piegl [21], and Piegl [22]. Moreover, an algorithm, which is based on an explicit solution, is considered in Karimäki [16] to fit the circle curvature, direction and position parameters.

Circular arcs can be used to produce a better representation for a smooth curve [11]. These circular arcs, moreover, has many applications; for

I. A. Al-Subaihi (✉)
Department of Science, King Abdulaziz Military Academy, P.O. Box 73052,
Riyadh 11538, Saudi Arabia
e-mail: alsubaihi@hotmail.com

example, quality inspection in manufacturing curing, processing of technical drawings [17], imaging techniques [16], cutting machine, approximating hulls of boats [6], and identification of particle in tracking chambers of high energy physics [17].

The least square norm is used when the error in the data are thought of to be normally distributed [28–32]. In Huyuh [13], the optimization method, pattern search of Hooke and Jeeves, is used to provide solution for the problem. In addition, the orthogonal distance regression has been used for fitting circular arcs to data in Atieg [5] and Atieg and Watson [6].

The l_1 norm is commonly used for data sets which contain *outliers* or *wild-points*, as the outliers do not have a significant effect upon the best l_1 approximation [1–3, 33]. This article is concerned with using this norm for fitting circular arcs to data. In addition, the work is motivated by the study of the problem in Atieg [5] and Atieg and Watson [6], where the application of the Gauss-Newton method to the problem and orthogonal distance regression, seems to have been first suggested.

In Section 2, the problem is identified. Section 3, deals with the requirements of derivatives for applying Gauss-Newton type method given in Atieg [5] and Atieg and Watson [6], and Gauss-Newton step. Further, a heuristic algorithm is presented in Section 4. Section 5 examines the use of l_2 and l_1 solutions to determine the starting points. Finally, numerical examples are presented for fitting circular arcs to data by the l_1 norm using different initial values in Section 6.

2 The l_1 problem

Suppose that the measured points $\mathbf{x}_i = (x_i, y_i) \in R^2$, $\quad i = 1, \ldots, m$ are given and have been arranged in to I_j subsets, where j is the index of the circular arc C_j, $\quad j = 1, \ldots, q$. Moreover, assume that the points $\mathbf{x}_j$, $\quad j = 1, \ldots, q-1$ are smoothly connecting the arcs C_j and C_{j+1}, so-called *connecting points*, and there is C^1 smoothness at the joint. Determining q circular arcs C_j, with centres $\mathbf{c}_j = (a_j, b_j)$ and radii r_j, $j = 1, \ldots, q$ are required to fit circular arcs to the data $\mathbf{x}_i, i \in I_j$, and to the connecting points $\mathbf{x}_j$, $j = 1, \ldots, q-1$. Different criteria may be used, depending on the actual application.

Define

$$w_i = r_j - \sqrt{(\mathbf{x}_i - \mathbf{c}_j)^T(\mathbf{x}_i - \mathbf{c}_j)}, i \in I_j,\ j = 1, \ldots, q.$$

Then basic fitting problems are to

$$\min \|\mathbf{w}\|_p,\ 1 \le p \le \infty. \tag{1}$$

Take the approximating criterion in the problem (1) to be the l_1 norm, then the problem to be solved can be expressed as a constrained optimization problem,

$$\min \sum_{i=1}^{m} |w_i| \tag{2}$$

subject to

$$(\mathbf{x}_j - \mathbf{c}_j)^T(\mathbf{x}_j - \mathbf{c}_j) = r_j^2 \tag{3}$$

$$(\mathbf{x}_j - \mathbf{c}_{j+1})^T(\mathbf{x}_j - \mathbf{c}_{j+1}) = r_{j+1}^2 \tag{4}$$

$$(x_j - a_j)(y_j - b_{j+1}) = (y_j - b_j)(x_j - a_{j+1}) \tag{5}$$

for $j = 1, \ldots, q-1$. In fact, the problem has a total of $3q$ unknowns a_j, b_j, r_j, where $j = 1, \ldots, q$, and $3(q-1)$ constraints. The first two equality constraints are called *connectedness conditions* between the two circular arcs C_j and C_{j+1}. Whilst the equality constrained (5) is called *smoothness condition* at the connecting point (x_j, y_j) [5, 6].

The connectedness conditions (3) and (4) for the connecting points $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$, $j = 1, \ldots, q-2$ give

$$(\mathbf{x}_{j+1} - \mathbf{c}_{j+1})^T(\mathbf{x}_{j+1} - \mathbf{c}_{j+1}) = (\mathbf{x}_j - \mathbf{c}_{j+1})^T(\mathbf{x}_j - \mathbf{c}_{j+1}).$$

Then

$$b_{j+1} = \alpha_j + \beta_j a_{j+1}, \quad j = 1, \ldots, q-2, \tag{6}$$

where

$$\alpha_j = \frac{x_{j+1}^2 - x_j^2 + y_{j+1}^2 - y_j^2}{2(y_{j+1} - y_j)},$$

and

$$\beta_j = \frac{-(x_{j+1} - x_j)}{y_{j+1} - y_j}.$$

The smoothness condition (5) gives the components $b_{j+1} = g_j(a_{j+1})$, say, $j = 1, \ldots, q-1$,

$$b_{j+1} = y_j - \frac{(y_j - b_j)(x_j - a_{j+1})}{x_j - a_j}. \tag{7}$$

The components $a_{j+1} = f_j(a_j, b_j)$, say, could be found by using (6) and (7)

$$a_{j+1} = \frac{(x_j - a_j)(y_j - \alpha_j) - x_j(y_j - b_j)}{\beta_j(x_j - a_j) - (y_j - b_j)}, \quad j = 1, \ldots, q-2. \tag{8}$$

Moreover, if a_q then $b_q = h(a_q, a_{q-1}, b_{q-1})$, say,

$$b_q = y_{q-1} - \frac{(y_{q-1} - b_{q-1})(x_{q-1} - a_q)}{x_{q-1} - a_{q-1}}. \tag{9}$$

Given $\mathbf{v} = (a_1, b_1, a_q)^T$, then it will be easy to determine the vector $\mathbf{u} = (a_2, b_2, \ldots, a_{q-1}, b_{q-1}, b_q)^T \in R^{2q-3}$, which depends on the value of $\mathbf{v}$. For suitably, the equations (6), (8), and (9) can be rewritten as $\mathbf{z}(\mathbf{u}, \mathbf{v}) \in R^{2q-3}$, or explicitly

$$\begin{aligned} a_2 - f_1(a_1, b_1) &= 0 \\ b_2 - g_1(a_2) &= 0 \\ &\dots \\ a_{q-1} - f_{q-2}(a_{q-2}, b_{q-2}) &= 0 \\ b_{q-1} - g_{q-2}(a_{q-1}) &= 0 \\ b_q - h(a_q, a_{q-1}, a_{q-1}, b_{q-1}) &= 0. \end{aligned}$$

Consequently, the l_1 problem (2)–(5) can be written in other word to minimize

$$F(\mathbf{v}) = \sum_{i \in I_j,\, j=1,\dots,q} |w_i(\mathbf{u}, \mathbf{v})| \quad \text{subject to}$$

$$\mathbf{z}(\mathbf{u}, \mathbf{v}) = 0. \tag{10}$$

where the equations $\mathbf{z}(\mathbf{u}, \mathbf{v}) = 0$ define $\mathbf{u} = \mathbf{u}(\mathbf{v})$ as functions of $\mathbf{v}$, and

$$w_i(\mathbf{u}, \mathbf{v}) = \sqrt{(\mathbf{x}_j - \mathbf{c}_j)^T(\mathbf{x}_j - \mathbf{c}_j)} - \sqrt{(\mathbf{x}_i - \mathbf{c}_j)^T(\mathbf{x}_i - \mathbf{c}_j)},$$

$i \in I_j$, $j = 1, \dots, q-1$ in terms of $\mathbf{v}$. Replacing the connecting point $\mathbf{x}_j$ by $\mathbf{x}_q$ in the above equation, gives $w_i,\ i \in I_q$.

This is a nonlinear problem in $\mathbf{v} \in R^3$. As is normal with nonlinear problems, we look to satisfy first order necessary condition for a minimum or equivalently find a stationary point, such a point may be at best a local minimizer. Standard analysis, for example [18], gives the following theorem.

Theorem 1 $\mathbf{v}^*$ *is a stationary point of* (10) *if there exist numbers* ϕ_i *such that*

$$\sum_{i \in I^*} \phi_i \nabla_{\mathbf{v}^*} w_i(\mathbf{u}(\mathbf{v}^*), \mathbf{v}^*) + \sum_{i \notin I^*} \theta_i \nabla_{\mathbf{v}^*} w_i(\mathbf{u}(\mathbf{v}^*), \mathbf{v}^*) = 0, \quad i = 1, \dots, m, \tag{11}$$

where $I^* = \{i : w_i(\mathbf{u}(\mathbf{v}^*), \mathbf{v}^*) = 0\}$, $\theta_i = \mathrm{sign}(w_i(\mathbf{u}(\mathbf{v}^*), \mathbf{v}^*))$, $i \notin I^*$.

With assumption of non-degeneracy, $|I^*| < 3$. If the set I^* and signs θ_i, $i \notin I^*$ can be identified, then (11), together with $w_i(\mathbf{u}(\mathbf{v}^*), \mathbf{v}^*) = 0$, $i \in I^*$ gives a total of 3 equations (where I^* contains 3 indices) in the 3 unknowns $\mathbf{v}^*, \phi_i, i \in I^*$ [34].

The nonlinear l_1 problem (10) can be solved by, a most popular method, Gauss-Newton type method. It is often used in a form in which the number of variables in the problem is reduced by forcing to hold at each iteration k. Moreover, Levenberg-Marquardt method can be applied to allow the user to control the size of the update step $\mathbf{d}$, see for example, Al-Subaihi and Watson [1–3] and Watson [29, 33, 34].

3 The Gauss-Newton method

At the iteration k and for the given $\mathbf{v} \in R^3$, a Gauss-Newton step for (10) required to minimize $\mathbf{d} \in R^3$

$$\|\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v}) + \nabla_{\mathbf{v}}\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v})\, \mathbf{d}\|_1, \tag{12}$$

where $\nabla_{\mathbf{v}}\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v})$ is the Jacobian matrix at the vector $\mathbf{v}$. In other words

$$\|\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v}) + (\nabla_1\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v})\, \nabla_{\mathbf{v}}\mathbf{u}(\mathbf{v}) + \nabla_2\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v}))\, \mathbf{d}\|_1. \tag{13}$$

The operations ∇_1 and (∇_2) denote the partial derivatives of a function of two sets of variables with respect to the first and second set respectively.

The derivatives $\nabla_1\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v}) \in R^{m\times 2(q-2)+1}$ and $\nabla_2\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v}) \in R^{m\times 3}$, at the iteration k, can be written in the matrix notation

$$\nabla_1\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{0} \\ \frac{\partial \mathbf{w}_2}{\partial a_2} & \frac{\partial \mathbf{w}_2}{\partial b_2} & \ddots & & & & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \frac{\partial \mathbf{w}_j}{\partial a_j} & \frac{\partial \mathbf{w}_j}{\partial b_j} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \frac{\partial \mathbf{w}_{q-1}}{\partial a_{q-1}} & \frac{\partial \mathbf{w}_{q-1}}{\partial b_{q-1}} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{w}_q}{\partial b_q} \end{bmatrix},$$

and

$$\nabla_2\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v}) = \begin{bmatrix} \frac{\partial \mathbf{w}_1}{\partial a_1} & \frac{\partial \mathbf{w}_1}{\partial b_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{w}_q}{\partial a_q} \end{bmatrix}.$$

Note that

$$\frac{\partial \mathbf{w}_j}{\partial a_j} = \frac{x_i - a_j}{\sqrt{(\mathbf{x}_i - \mathbf{c}_j)^T(\mathbf{x}_i - \mathbf{c}_j)}} - \frac{x_j - a_j}{\sqrt{(\mathbf{x}_j - \mathbf{c}_j)^T(\mathbf{x}_j - \mathbf{c}_j)}}, \tag{14}$$

and

$$\frac{\partial \mathbf{w}_j}{\partial b_j} = \frac{y_i - b_j}{\sqrt{(\mathbf{x}_i - \mathbf{c}_j)^T(\mathbf{x}_i - \mathbf{c}_j)}} - \frac{y_j - b_j}{\sqrt{(\mathbf{x}_j - \mathbf{c}_j)^T(\mathbf{x}_j - \mathbf{c}_j)}}, \tag{15}$$

where $i \in I_j$, $j = 1 \dots, q-1$. Replacing the coordinate pairs of the connecting points $\mathbf{x}_j$ by $\mathbf{x}_{q-1}$ and $i \in I_q$ in equations (14) and (15) will give the column vector $\frac{\partial \mathbf{w}_q}{\partial a_q}$ and $\frac{\partial \mathbf{w}_q}{\partial b_q}$ for $i \in I_q$.

Differentiating $\mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v}) = \mathbf{0}$ with respect to $\mathbf{v}$ gives

$$\begin{aligned} \nabla_{\mathbf{v}}\mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v}) &= \nabla_1\mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v})\nabla_{\mathbf{v}}\mathbf{u}(\mathbf{v}) + \nabla_2\mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v}) \\ &= 0. \end{aligned}$$

Note that the nonsingular matrix $\nabla_1 \mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v}) \in R^{(2q-3)\times(2q-3)}$ is given by

$$\begin{bmatrix} 1 & 0 & \dots & & \dots & & \dots & & 0 \\ -\beta_1 & 1 & \ddots & & & & & & \\ -\frac{\partial f_2}{\partial a_2} & -\frac{\partial f_2}{\partial b_2} & \ddots & \ddots & & & & & \vdots \\ 0 & 0 & \dots & \ddots & \ddots & & & & \\ 0 & 0 & & -\beta_j & 1 & 0 & & & \vdots \\ \vdots & & \ddots & -\frac{\partial f}{\partial a_{j+1}} & -\frac{\partial f}{\partial b_{j+1}} & 1 & \ddots & & \\ & & & \ddots & 0 & \dots & \ddots & \ddots & \vdots \\ \vdots & & & & 0 & 0 & -\beta_{q-2} & 1 & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & -\frac{\partial h}{\partial a_{q-1}} & -\frac{\partial h}{\partial b_{q-1}} & 1 \end{bmatrix},$$

$j = 2, \dots, q-3$, and

$$\nabla_2 \mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v}) = \begin{bmatrix} -\frac{\partial f_1}{\partial a_1} & -\frac{\partial f_1}{\partial b_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\frac{\partial h}{\partial a_q} \end{bmatrix} \in R^{(2q-3)\times 3},$$

where the partial derivatives are defined as

$$\begin{aligned} \frac{\partial f_j}{\partial a_j} &= -\frac{(y_j - b_j)(\beta_j x_j - (y_j - \alpha_j))}{(\beta_j(x_j - a_j) - (y_j - b_j)^2}, \quad j = 1, \dots, q-2, \\ \frac{\partial f_j}{\partial b_j} &= -\frac{(x_j - a_j)(-\beta_j x_j + (y_j - \alpha_j))}{(\beta_j(x_j - a_j) - (y_j - b_j)^2}, \quad j = 1, \dots, q-2, \\ \frac{\partial h}{\partial a_{q-1}} &= -\frac{(x_{q-1} - a_q)(y_{q-1} - b_{q-1}))}{(x_{q-1} - a_{q-1})^2}, \\ \frac{\partial h}{\partial b_{q-1}} &= \frac{(x_{q-1} - a_q)}{x_{q-1} - a_{q-1}}, \\ \frac{\partial h}{\partial a_q} &= \frac{(y_{q-1} - b_{q-1})}{x_{q-1} - a_{q-1}}. \end{aligned}$$

Then $\nabla_{\mathbf{v}} \mathbf{u}(\mathbf{v})$ can be defined as

$$\nabla_{\mathbf{v}} \mathbf{u}(\mathbf{v}) = -(\nabla_1 \mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v}))^{-1} \nabla_2 \mathbf{z}(\mathbf{u}(\mathbf{v}), \mathbf{v}).$$

Assume that $\mathbf{v}^{(k)}$ is the vector $\mathbf{v}$ at the iteration k, to put it more simply, in the next discussion of convergence analysis. Similarly for $\gamma^{(k)}$ and $\mathbf{d}^{(k)}$. Let

$0 < \epsilon < 1$ be given, and define $\Pi = \{1, \epsilon, \epsilon^2, \dots\}$. Then, it is possible to choose step length $\gamma^{(k)}$ as the largest element in Π, satisfying

$$\frac{\left\|\mathbf{w}\left(\mathbf{v}^{(k)}\right)\right\|_1 - \left\|\mathbf{w}\left(\mathbf{v}^{(k)} + \gamma^{(k)}\mathbf{d}^{(k)}\right)\right\|_1}{\gamma^{(k)}\left(\left\|\mathbf{w}\left(\mathbf{v}^{(k)}\right)\right\|_1 - \left\|\mathbf{w}\left(\mathbf{v}^{(k)}\right) + \nabla_{\mathbf{v}^{(k)}}\mathbf{w}\left(\mathbf{u}\left(\mathbf{v}^{(k)}\right), \mathbf{v}^{(k)}\right)\mathbf{d}^{(k)}\right\|_1\right)} \geq \sigma, \tag{16}$$

where σ is any fixed value satisfying $0 < \sigma < 1$ [34].

Theorem 2 [34] *Let the Gauss-Newton method generate a sequence of points* $\left\{\mathbf{v}^{(k)}\right\}$, *where* $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \gamma^{(k)}\mathbf{d}^{(k)}$, *where* $\gamma^{(k)}$ *is the largest element in* Π *which satisfies* (16). *Let the sequence of step lengths* $\{\gamma^{(k)}\}$ *be bounded away from zero. Then any limit point of the sequence* $\{\mathbf{v}^{(k)}\}$ *is a stationary point of* (10).

Even though the sequence $\{\gamma^{(k)}\}$ can not be bounded away from zero. "This phenomenon does not inhibit convergence theory in the smooth case, although in practice, one has to take a finite number of steps and place a lower bound restriction on the step length" [27]. The sequence of problems (13) to have solutions is required for fast local convergence. This has been ensured by a strong uniqueness requirement at a limit point for smooth problem, or by the assumption that $\nabla_{\mathbf{v}^{(k)}}\mathbf{w}\left(\mathbf{u}\left(\mathbf{v}^{(k)}\right), \mathbf{v}^{(k)}\right)$ has full rank [27]. If $\nabla_{\mathbf{v}^{(k)}}\mathbf{w}\left(\mathbf{u}\left(\mathbf{v}^{(k)}\right), \mathbf{v}^{(k)}\right)$ has full rank means that there is always a solution to (13) which is

$$w_i\left(\mathbf{u}\left(\mathbf{v}^{(k)}\right), \mathbf{v}^{(k)}\right) + \mathbf{e}_i^T \nabla_{\mathbf{v}^{(k)}}\mathbf{w}\left(\mathbf{u}\left(\mathbf{v}^{(k)}\right), \mathbf{v}^{(k)}\right)\mathbf{d}^{(k)} = 0, \ i \in I^{(k)}, \text{ with } \left|I^{(k)}\right| = 3, \tag{17}$$

where $\mathbf{e}_i$ is the ith coordinate vector, which is non-degeneracy condition. Then full steps ($\gamma^{(k)} = 1$) are associated to fast local convergence. Moreover, the condition $I^{(k)} = I^*$, independent of k, requires the number of zero components of $\mathbf{w}(\mathbf{u}(\mathbf{v}^*), \mathbf{v}^*)$ to be 3. If there is no redundancy in the zero components of $\mathbf{w}(\mathbf{u}(\mathbf{v}), \mathbf{v})$ to be 3, convergence may then be very slow [27, 34].

To sum up, all the required derivatives for applying Gauss-Newton type method can be computed. In fact, the performance of the Gauss-Newton methods for (10), depends primarily on the number of zero components of $\mathbf{w}$, at a limit point. If this number is 3, then a second order rate of convergence is expected.

4 An algorithm

The algorithm of fitting circular arcs to data using the l_1 norm is summarized as follows

STEP 0. Input: The data points $\mathbf{x}_i,\ i = 1, \dots, m$, the connecting points $\mathbf{x}_j,\ j = 1, \dots, q-1$, and a tolerance (**Tol**).

STEP 1. Set $k = 0$, and calculate the initial value of $\mathbf{v}^{(0)} = \left(a_1^{(0)}, b_1^{(0)}, a_m^{(0)}\right)^T$,

STEP 2. Determine:

- The vector $\mathbf{u}^{(k)} = \left(a_2^{(k)}, b_2^{(k)}, \ldots, a_{m-1}^{(k)}, b_{m-1}^{(k)}, b_m^{(k)}\right)^T$.
- The vector $\mathbf{w}\left(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}\right)$.
- The objective function: $F^{(k)} = \left\|\mathbf{w}\left(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}\right)\right\|_1$.
- The Jacobian matrix $\nabla_{\mathbf{v}^{(k)}}\mathbf{w}\left(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}\right)$.

STEP 3. Solve: $\left\|\mathbf{w}\left(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}\right) + \nabla_{\mathbf{v}^{(k)}}\mathbf{w}\left(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}\right)\mathbf{d}^{(k)}\right\|_1$ for $\mathbf{d}^{(k)}$.

- **If** $\left\|\mathbf{d}^{(k)}\right\|_\infty <$ **Tol**, **Then** go to **STEP 5**.
- **Otherwise**, calculate a new estimate $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \gamma^{(k)}\mathbf{d}^{(k)}$ for some suitable chosen step length $\gamma^{(k)}$.

STEP 4. Set $k = k + 1$, Go to **STEP 2**.

STEP 5. Fitting circular arcs to data using the l_1 norm has been completed.

5 Starting points

It is necessary to provide starting values for any iterative algorithm. Sometimes, a set of m data points and the connecting points are given. However, determining the centre $\mathbf{c}_1 = (a_1, b_1)$ and x-coordinate of the last centre $\mathbf{c}_q = (a_q, b_q)$ is what we aim at. In other word, $\mathbf{v}^{(0)} = \left(a_1^{(0)}, b_1^{(0)}, a_q^{(0)}\right)^T$ needs to be determined.

One possibility for the initial value of $\mathbf{v}^{(0)}$ is to use algebraic fitting of circles to the subsets data $I_j = 1, q$ in least quare sense. An algebraic representation of the circle in two dimensions could be defined as

$$\begin{aligned} P(\mathbf{x}, \hat{\mathbf{v}}) &= \hat{a}\mathbf{x}^T\mathbf{x} + \hat{\mathbf{b}}^T\mathbf{x} + \hat{c} \\ &= 0 \end{aligned} \tag{18}$$

where $\hat{\mathbf{v}} = \left(\hat{a}, \hat{b}_1, \hat{b}_2, \hat{c}\right)^T$, $\hat{a} \neq 0$. In fact, the implicit equation (18) is linear in the coefficients vector $\mathbf{v}$.

Assuming that the error is normally distributed, then least squares solutions are appropriate. One can be found by solving

$$\min \|Z\hat{\mathbf{v}}\| \text{ subject to } \|\hat{\mathbf{v}}\| = 1, \tag{19}$$

where

$$Z = \left[\ \|\mathbf{x}\|^2\ x_i\ y_i\ 1\ \right], \quad i \in I_1.$$

This problem is equivalent to finding the right singular vector associated with the smallest singular value of Z [10, 26]. Because of the closed-form solution, this method is probably the most commonly used approach to the problem (19). Moreover, an alternative to (19) is to solve the problem

$$\min \|Z\hat{\mathbf{v}}\| \text{ subject to } \hat{v}_s = 1, \quad 1 \le s \le 3 \tag{20}$$

which is commonly used [10], because of a simpler normalization. However these methods can be lead to a highly biased estimate for small circular arcs with low curvature [9, 35]. A much more accurate methods is the so-called *gradient weighted algebraic fit*. To put it clear, the procedure is summarized in the following form.

The gradient weighted algebraic method is minimizing

$$G = \sum_{i \in I_1} \frac{[P(\mathbf{x}, \hat{\mathbf{v}})]^2}{\|\nabla_1 P(\mathbf{x}, \hat{\mathbf{v}})\|^2}, \tag{21}$$

which can be simplified by Pratt's approximation [23] or Taubin's approximation [25]. The minimum of (21) corresponds to the smallest nonnegative generalized eigenvalue. This root could be determined by using standard matrix method, which tend to be computationally extensive. Also it could be determined by using Newton's iterations which are guaranteed to converge to the root. For more details and algorithms see [8, 9]. Although, it should be born in mind that in the context of data containing outliers, these methods may not be particulary good approximations.

Second possibility for the initial value of $\mathbf{v}^{(0)}$ is to use algebraic fitting of circles to the subsets data $I_j = 1, q$ using l_1 norm, which is an appropriate norm when data has outliers. A natural form which reflects the fact that these wild points also impact on Z is to solve

$$\min \|Z\hat{\mathbf{v}}\|_1 \ \text{ subject to } \ \|\hat{\mathbf{v}}\|_\infty = 1. \tag{22}$$

A normally finite algorithm for this problem is given in [19]. Furthermore, an alternative to this problem is to solve

$$\min \|Z\hat{\mathbf{v}}\|_1 \ \text{ subject to } \ \|\hat{\mathbf{v}}\|_2 = 1, \tag{23}$$

which is treated in [24], and it can also be solved essentially by the method of [19], see [2] for more details and general algorithms. An alternative to these methods is to solve

$$\min \|Z\hat{\mathbf{v}}\|_1 \ \text{ subject to } \ v_s = 1, \quad 1 \le s \le 3. \tag{24}$$

The problems (20) and (24) are equivalent to making the assumption that only the sth column of Z contains errors, and the other column are exact [4, 10, 26].

Expanding (18), rearranging appropriately and completing the squares, we obtain

$$\left(\mathbf{x}_{i1} + \frac{\hat{b}_1}{2\hat{a}}\right)^2 + \left(\mathbf{x}_{i2} + \frac{\hat{b}_2}{2\hat{a}}\right)^2 = \frac{\left\|\hat{\mathbf{b}}\right\|^2}{4\hat{a}^2} - \frac{\hat{c}}{\hat{a}}, i \in I_1 \tag{25}$$

from which the centre $\mathbf{c}_1 = \left(\frac{\hat{b}_1}{2\hat{a}}, \frac{\hat{b}_2}{2\hat{a}}\right)$, and the radius $r_1 = \sqrt{\frac{\left\|\hat{\mathbf{b}}\right\|^2}{4\hat{a}^2} - \frac{\hat{c}}{\hat{a}}}$, provided that the right hand side of (25) is positive. Then the initial centre of the first circular arc C_1 will be $\mathbf{c}_1 = \left(a_1^{(0)}, b_1^{(0)}\right)$. Note that the previous methods for initial approximation point will not normally connect smoothly with its neighbor arcs. However, it dose give an approximate $a_1^{(0)}$ and $b_1^{(0)}$.

The initial value of $a_m^{(0)}$ can be determined by using the same previous strategies and $i \in I_q$. Also it can be determined by using the definition of $\mathbf{z}$ to give arcs $C_2, \ldots, C_{m-1}$. Now we compute $a_m^{(0)}$ so that C_m joints smoothly to C_{m-1} and interpolates one of the last subset of the data points. This would involve solving three nonlinear equations in three unknowns [5, 6].

To summarize, it is clear that a number of methods can be used to generate starting points for the problem (10). In the mean time, a good starting vector for the Gauss-Newton method may be obtained by solving easy linear problems, for example (19) and (24), as we will see in the following examples.

6 Numerical examples

This section gives two examples to illustrate the application of the Gauss-Newton method for fitting circular arcs to data using the l_1 norm. Starting points are determined by using two simple methods (19) and (24). Moreover, the method (23) can be used to provide the final fitted circles and it converges to a stationary point in a finite number of steps [2]. So, one step could be sufficient to determine a starting value. Performance shown in the examples is typical, with a second order rate of convergence.

The two-dimensional m data points were generated by taking a particular curve, and introducing random perturbation. Every example has three starting points $\mathbf{v}^{(0)} = \left(a_1^{(0)}, b_1^{(0)}, a_q^{(0)}\right)$. The initial values of $\mathbf{v}^{(0)}$ in the case (a) is determined by using a least squares sense (19) and SVD. For cases (b) and (c) the initial approximations to $\mathbf{v}^{(0)}$ are determined by (23), (24) and $s = 1$ respectively. These initial values are apparent in the Tables and $k = 0$, where k denotes the iteration number. In addition, $\gamma^{(k)}$ denotes the Gauss-Newton step $\mathbf{d}^{(k)}$'s length. The difference in the initial points for case (b) and (c) is very small, and that is why plotting the initial circular arcs for case (b) has been shown only.

Example 1 Three different starting points are used to fit the 25 points, which have an outlier point. Results are shown in the Tables 1, 2, and 3 and in Fig. 1, where the final $\mathbf{v}^*$ for the three cases are equal, and $I^* = \{1 \in C_1, 10 \in C_2, 19 \in C_4\}$.

Example 2 The three starting points, are shown in Tables 4, 5, and 6, to fit 100 data points, which have five outlier points. Results are shown in the same Tables and in Fig. 2, where the final $\mathbf{v}^*$ for the three cases are equal, and $I^* = \{6 \in C_1, 84 \in C_2, 97 \in C_5\}$.

Table 1 Example 1(a)

k	$a_1^{(k)}$	$b_1^{(k)}$	$a_4^{(k)}$	$F^{(k)}$	$\|\mathbf{d}^{(k)}\|_\infty$	$\gamma^{(k)}$
0	1.9951	1.7453	5.4522	4.478324	4.1777×10^{-1}	1
1	1.5774	2.1329	5.5159	3.213535	1.4317×10^{-2}	1
2	1.5649	2.1186	5.5173	3.183211	1.1349×10^{-3}	1
3	1.5649	2.1186	5.5162	3.178638	9.4216×10^{-7}	

Table 2 Example 1(b)

k	$a_1^{(k)}$	$b_1^{(k)}$	$a_4^{(k)}$	$F^{(k)}$	$\|\mathbf{d}^{(k)}\|_\infty$	$\gamma^{(k)}$
0	1.5212	2.1842	5.4173	5.051484	9.8507×10^{-1}	1
1	1.5635	2.1194	5.5158	3.182639	1.3274×10^{-3}	1
2	1.5649	2.1186	5.5162	3.178633	3.5847×10^{-7}	

Table 3 Example 1(c)

k	$a_1^{(k)}$	$b_1^{(k)}$	$a_4^{(k)}$	$F^{(k)}$	$\|\mathbf{d}^{(k)}\|_\infty$	$\gamma^{(k)}$
0	1.4659	2.1750	5.4749	4.072457	1.0507×10^{-1}	1
1	1.5710	2.1344	5.5177	3.208047	1.5763×10^{-2}	1
2	1.5649	2.1187	5.5162	3.178655	6.9466×10^{-6}	

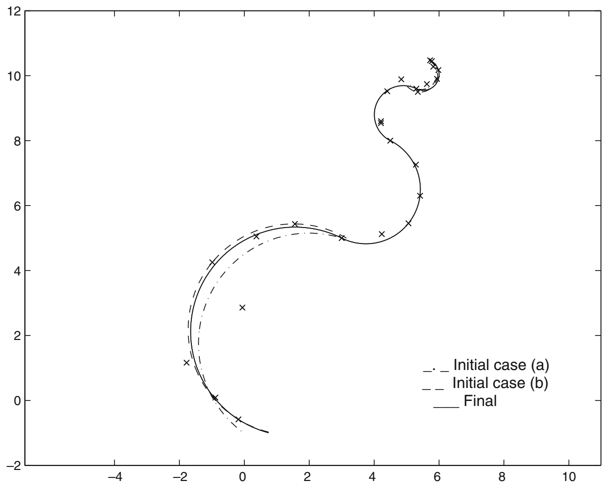


Fig. 1 Fitting circular arcs to 25 data points

Table 4 Example 2(a)

k	$a_1^{(k)}$	$b_1^{(k)}$	$a_4^{(k)}$	$F^{(k)}$	$\|\mathbf{d}^{(k)}\|_\infty$	$\gamma^{(k)}$
0	−5.0681	4.4078	5.4117	42.184815	7.0452×10^{-1}	1
1	−5.7292	3.7033	4.9182	25.182314	3.9835×10^{-1}	1
2	−5.8445	3.5595	4.5198	20.554742	4.0376×10^{-2}	1
3	−5.8849	3.5283	4.5373	15.525043	2.7859×10^{-2}	1
4	−5.8570	3.5477	4.5610	15.486284	2.9566×10^{-4}	1
5	−5.8572	3.5476	4.5607	15.484199	1.4538×10^{-7}	

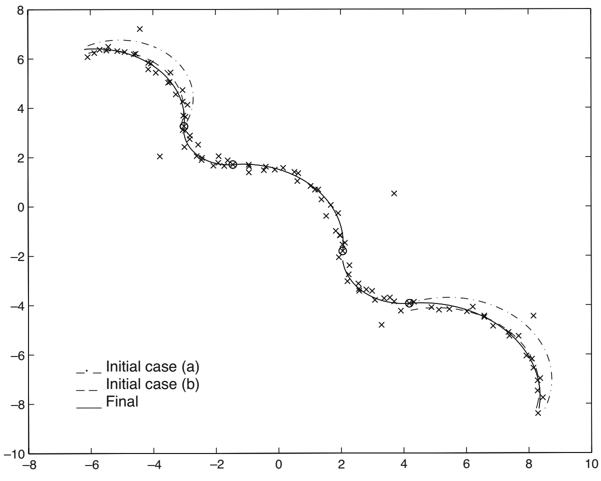
Table 5 Example 2(b)

k	$a_1^{(k)}$	$b_1^{(k)}$	$a_4^{(k)}$	$F^{(k)}$	$\|\mathbf{d}^{(k)}\|_\infty$	$\gamma^{(k)}$
0	−5.3913	3.9548	5.0437	23.066567	3.8596×10^{-1}	1
1	−5.7772	3.6144	4.6646	16.335390	1.0389×10^{-1}	1
2	−5.8618	3.5504	4.5607	15.548162	4.5663×10^{-3}	1
3	−5.8572	3.5475	4.5607	15.484453	6.0231×10^{-4}	1
4	−5.8572	3.5476	4.5607	15.484199	1.8128×10^{-7}	

Table 6 Example 2(c)

k	$a_1^{(k)}$	$b_1^{(k)}$	$a_4^{(k)}$	$F^{(k)}$	$\|\mathbf{d}^{(k)}\|_\infty$	$\gamma^{(k)}$
0	−5.4507	3.8444	4.9862	19.967243	3.7222×10^{-1}	1
1	−5.5821	3.5496	4.6140	17.392253	2.5875×10^{-2}	1
2	−5.8408	3.5485	4.5592	15.618883	1.6478×10^{-3}	1
3	−5.8572	3.5475	4.5606	15.484236	1.2147×10^{-4}	1
4	−5.8572	3.5476	4.5607	15.484199	1.8128×10^{-9}	

Fig. 2 Fitting circular arcs to 100 data points



7 Conclusion

An algorithm of fitting circular arcs to data using the l_1 norm is already presented in this paper. Let's assume that the number of arcs and the connecting points were correct, then fitting circular arcs to the data would be sufficient, and the Gauss-Newton type method or Levenberg-Marquardt method could be applied for using the l_1 norm. Also it is concluded that the main factor in the local convergence is the number of zero distance. Three zero components at a solution were required for second order of convergence, which was illustrated by numerical examples.

Often the use of least squares solutions to determine the initial starting points could be insufficient, especially when data has wild points, as a result of the nature of least square. This was the reason for developing methods that depend on the l_1 norm. The number of iteration on the algorithm was not generally affected by the number of m.

The parametrization in terms of $\mathbf{v} = (a_1, b_1, a_q)^T$ will be singular or ill-conditioned for some arrangements of the connecting points. Consideration of more robust, globally convergent methods, together with the treatment of problems with the parametrization in terms of (a_1, b_1, r_q) will be the subject of future work. The case where the connecting points are free, subject to some regularization, would seem to be more likely to have practical application, but probably more difficult to implement.

Acknowledgements I wish to express profound gratitude to Prof. G.A. Watson for the comments he made on the first version of this paper. Moreover, I thank him deeply for his generous support by offering me the Fortran l_1 solver programme. It is also important to acknowledge the positive comments of Dr Kamal Abodayeh who has exerted immense efforts in revising this paper. Furthermore, I am grateful to the referees for their helpful suggestions.

References

1. Al-Subaihi, I.A., Watson, G.A.: An algorithm for matching point sets using the l_1 norm. Numer. Algorithms **41**, 203–217 (2006)
2. Al-Subaihi, I.A., Watson, G.A.: Algebraic fitting of quadric surfaces to data. Commun. Appl. Anal. **9**, 539–548 (2005)
3. Al-Subaihi, I.A., Watson, G.A.: The use of the l_1 and l_∞ norms in fitting parametric curves and surfaces to data. Appl. Numer. Anal. Comput. Math. 363–376 (2004)
4. Antia, H.M.: Numerical Methods for Scientists and Engineers, 2nd edition. Birkhäuser Verlag, Basel (2002)
5. Atieg, A.: Numerical methods for fitting curves and surfaces to data by orthogonal distance regression and some generalizations. Ph.D. thesis, University of Dundee (2005)
6. Atieg, A., Watson, G.A.: Fitting circular arcs by orthogonal distance regression. Appl. Numer. Anal. Comput. Math. **1**, 66–76 (2004)
7. Cheng, C.C., Duncan, C.A., Goodrich, M.T., Kobourov, S.G.: Drawing planer graphs with circular arcs. Lect. Notes Comput. Sci. **1731**, 117 (2000)
8. Chernov, N., Lesort, C.: Fitting circles and lines by least squares: theory and experiment. http://www.math.uab.edu/cl/cl1 (2007)

9. Chernov, N., Lesort, C.: Least squres fitting of circles. Math. Imaging Vis. **23**, 239–252 (2005)
10. Gander, W., Golub, G.H., Strebel, R.: Fitting of circles and ellipses: least squares solution. BIT **34**, 556–577 (1994)
11. Horng, J.: An adaptive smoothing approach for fitting digital planar curves with line segments and circular arcs. Pattern Recogn. Lett. **22**, 565–577 (2003)
12. Horng, J., Johnny, T.L.: A dynamic programming approach for fitting digital planar curves with line segments and circular arcs. Pattern Recogn. Lett. **22**, 183–197 (2000)
13. Huynh, H.-H.: Least square curve-fitting with circular arcs. Project RMIT (1985)
14. Jittorntrum, K., Osborne, M.R.: Strong uniqueness and second order convergence in nonlinear discrete approximation. Numer. Math. **34**, 439–455 (1980)
15. Joseph, S.H.: Unbiased least square fitting of circular arcs. CVGIP Graph. Models Image Process. **56**(5), 424–432 (1994)
16. Karimäki, V.: Fast code to fit circular arcs. Comput. Phys. Commun. **8**, 133–141 (1991)
17. Lillekjendlie, B.: Circular arcs fitted on a Riemann sphere. Comput. Vis. Image Underst. **67**(3), 311–317 (1997)
18. Osborne, M.R.: Finite Algorithms in Optimization and Data Analysis. Wiley, New York (1984)
19. Osborne, M.R., Watson, G.A.: An analysis of the total approximation problem in separable norms, and an algorithm for the total l_1 problem. SIAM J. Sci. Statist. Comput. **6**, 410–424 (1985)
20. Pei, S., Horng, J.: Fitting digital curve using circular arcs. Pattern Recogn. **28**(1), 107–116 (1994)
21. Piegl, L.A.: Fitting circular arcs to measured data. Int. J. Shape Model. **8**(1), 1–21 (2002)
22. Piegl, L.A.: Curve fitting algorithm for rough cutting. Comput. Aided Des. **18**(2), 79–82 (1986)
23. Pratt, V.: Direct least squares fitting of algebraic surfaces. Comput. Graph. **21**, 145–152 (1987)
24. Späth, H., Watson, G.A.: On orthogonal linear l_1 approximation. Numer. Math. **51**, 531–543 (1987)
25. Taubin, G.: Estimation of planer curves, sufaces and nonplaner space curves defined by implicit equations, with applications to edge and range image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **13**, 1115–1138 (1991)
26. Varah, J.M.: Least squares data fitting with implicit functions. BIT **36**, 842–854 (1996)
27. Watson, G.A.: On the Gauss-Newton method for l_1 orthogonal distance regression. IMA J. Numer. Anal. **22**, 345–357 (2002)
28. Watson, G.A.: Some problems in orthogonal distance and non-orthogonal distance regression. In: Levesley J., Anderson I., Mason J.C. (eds.) Proceedings of Algorithms for Approximation IV, pp. 294–302. University of Huddersfield, Huddersfield (2002)
29. Watson, G.A.: Approximation in normed linear spaces. J. Comput. Appl. Math. **121**, 1–36 (2000)
30. Watson, G.A.: Least squares fitting of parametric surfaces to measured data. Australian National University CMA Technical Report, No 9.044 (1999)
31. Watson, G.A.: Least squares fitting of circles and ellipses to measured data. BIT **39**, 176–191 (1999)
32. Watson, G.A.: Some robust methods far fitting parametrically defined curves or surfaces to measured data. In: The series of conferences on Advanced Mathematical and Computational Tools in Metrology (AMCTM), Oxford 1–17 (1999)
33. Watson, G.A.: Choice of Norms for Data Fitting and Function Approximation, pp. 337–377. Acta Numerica, Cambridge University Press (1998)
34. Watson, G.A.: Approximation Theory and Numerical Methods. Wiley, Chichester (1980)
35. Werghi, N., Fisher, R.B., Ashbrook, A., Robertson, C.: Faithful recovering of quadric surfaces from 3D rang data. In: Proc. 2nd Int. Conf. on 3D Digital Imaging and Modeling, Ottawa, Canada, pp. 280–289 (1999)