A FITTING OF PARABOLAS WITH MINIMIZING THE ORTHOGONAL DISTANCE

IK-SUNG KIM

ABSTRACT. We are interested in the problem of fitting a curve to a set of points in the plane, in such a way that the sum of the squares of the orthogonal distances to given data points is minimized. In [1] the problem of fitting circles and ellipses was considered and numerically solved with general purpose methods. Especially, in [2] H. Späth proposed a special purpose algorithm(Späth's ODF) for parabolas $y-b=c(x-a)^2$ and for rotated ones. In this paper we present another parabola fitting algorithm which is slightly different from Späth's ODF. Our algorithm is mainly based on the steepest descent procedure with the view of ensuring the convergence of the corresponding quadratic function $Q(u)$ to a local minimum. Numerical examples are given.

AMS Mathematics Subject Classification : 65D10.
Key words and phrases : orthogonal distance, fitting of parabolas, least squares.

1. Introduction

Let be given a set $\{(x_k, y_k) : k = 1, 2, \ldots, n\}$ of data points in the plane and a model curve in the parametric form

$$x = x(a;t), \quad y = y(b;t), \tag{1.1}$$

where $a = (a_1, a_2, \ldots, a_m)$ and $b = (b_1, b_2, \ldots, b_l)$ are two vectors of unknown parameters. We are interested in the problem of fitting the

Received October 2, 1998. Revised January 22, 1999.
© 1999 Korean Society for Computational & Applied Mathematics and Korean SIG-COAM(Korea Information Processing Society) .

model curve to the given data points in such a way that the sum of the squares of the orthogonal distances from (x_k, y_k) to unknown points $(x(a;t_k), y(b;t_k))$ is minimized. The orthogonal distance d_k of a point (x_k, y_k) can be expressed by .

$$(1.2) \qquad d_k = \sqrt{\min_{t_k}\left[(x_k - x(a;t_k))^2 + (y_k - y(b;t_k))^2\right]}.$$

Thus, to fit the model curve to the given data points we need to determine $a = (a_1, a_2, \ldots, a_m)$ and $b = (b_1, b_2, \ldots, b_l)$ by minimizing $\sum_{k=1}^{n} d_k^2 = \min$. The objective function to be minimized is given by the quadratic function Q defined by

$$(1.3) \qquad Q(u) = \sum_{k=1}^{n} \left[(x_k - x(a;t_k))^2 + (y_k - y(b;t_k))^2\right],$$

where the parameter vector $u = (a_1, \ldots, a_m, b_1, \ldots, b_l, t_1, \ldots, t_n) \in R^{(m+l+n)}$ is dependent upon a, b and the vector $t = (t_1, t_2, \ldots, t_n)$ of additional unknowns $\{t_i\}_{i=1}^{n}$. In addition, a, b and t can be simultaneously determined. To do so, we may either apply minimization algorithms to (1.3) or solve the nonlinear system of $(m + l + n)$ equations for the parameter vector $u = (a_1, \ldots, a_m, b_1, \ldots, b_l, t_1, \ldots, t_n)$ induced by the necessary conditions for a minimum, namely

$$(1.4) \qquad \frac{\partial Q}{\partial a_i} = 0 \quad (i = 1, \ldots, m),$$

$$(1.5) \qquad \frac{\partial Q}{\partial b_j} = 0 \quad (j = 1, \ldots, l),$$

$$(1.6) \qquad \frac{\partial Q}{\partial t_k} = 0 \quad (k = 1, \ldots, n).$$

In this paper, we particularly deal with parabolas as a model curve and propose an iteration algorithm for fitting a parabola(or a rotated parabola) to the given data points in the plane. Let be given any corresponding quadratic function $Q(u)$ for the parameter vector $u = (u_1, u_2, \ldots, u_s)$ which is needed to fit a parabola. Our algorithm for minimizing the given quadratic function $Q(u)$ consists of two well known procedures as follows.

(PROCEDURE-1) Given any r values of $u_1, \ldots, u_r$ $(r < s)$, find the

dCollection

other $(s-r)$ values $u_{r+1}, u_{r+2}, \ldots, u_s$ by solving a constrained nonlinear system of $(s-r)$ equations which is connected with the system of s equations $\dfrac{\partial Q}{\partial u_j} = 0\,(j = 1, \ldots, s)$.

(PROCEDURE-2) By using the steepest descent method for the quadratic function $Q(u)$, determine a new value $u^{(1)} = (u_1^{(1)}, u_2^{(1)}, \ldots, u_s^{(1)}) = u^{(0)} - \alpha\nabla Q(u^{(0)})$ from given an initial approximation $u^{(0)} = (u_1^{(0)}, u_2^{(0)}, \ldots, u_s^{(0)})$. The value of α is given by minimizing the single variable function $h(\alpha) = Q(u^{(0)} - \alpha\nabla Q(u^{(0)}))$. Here, ∇ denotes the gradient operator.

Even if the convergence of the given quadratic function $Q(u)$ to the global minimum may not be guaranteed, our algorithm for fitting parabolas has the advantage of ensuring the convergence of $Q(u)$ to a local minimum. We give some numerical examples for fitting of parabolas and observe the convergence of each quadratic function $Q(u)$.

2. Orthogonal distance fitting of parabolas

Let us now consider parabolas as the model curve (1.1)

(2.1) $$y - b = c(x-a)^2 \quad (c \neq 0)$$

in its parametric form

(2.2) $$x = a + t, \qquad y = b + ct^2.$$

Then the quadratic function to be minimized for fitting a parabola to the given n data points (x_k, y_k) $(k = 1, \ldots, n)$ is given by

(2.3) $$Q(u) = \sum_{k=1}^{n} \left[(x_k - a - t_k)^2 + (y_k - b - ct_k^2)^2\right],$$

where the parameter vector $u = (a, b, c, t_1, t_2, \ldots, t_n) \in R^{(n+3)}$.

We also have the following $(n+3)$ equations induced by the necessary conditions (1.4), (1.5) and (1.6) for a minimum:

(2.4) $$\sum_{k=1}^{n}(x_k - a - t_k) = 0,$$

(2.5) $$\sum_{k=1}^{n}(y_k - b - ct_k^2) = 0,$$

dCollection

(2.6) $$\sum_{k=1}^{n} t_k^2(y_k - b - ct_k^2) = 0,$$

and

(2.7) $$t_i^3 + p_i t_i + q_i = 0 \quad (i = 1, \dots, n)$$

with

(2.8) $$p_i = \frac{1 - 2c(y_i - b)}{2c^2},$$

(2.9) $$q_i = \frac{a - x_i}{2c^2}.$$

The above system of $(n+3)$ equations can be divided into two parts. One is a linear system of the three equations (2.4), (2.5) and (2.6) for a, b and c, whose coefficient matrix depends on $t_1, t_2, \dots, t_n$. The other part is concerned with the n cubic equations (2.7). The ith cubic equation contains just t_i as unknown and its coefficients depend on a, b and c. There are some well-known methods for solving the linear system of the first part and n cubic equations (2.7) respectively. Furthermore, one may be able to propose mixed iteration algorithms which are connected with both well-known methods above. In this connection, in [2] H. Späth proposed an iteration algorithm(Späth's ODF) for fitting of parabolas. Here, we will present another parabola fitting algorithm. In fact, the steepest descent method can be employed as a minimization algorithm for minimizing the quadratic function (2.3). So, the parabola fitting algorithm consists of both the steepest descent method for minimizing $Q(u)$ and the root-finding algorithm for the n cubic equations (2.7). We now describe our iteration algorithm for fitting of parabolas.

Algorithm A:

STEP 0. Let $a^{(0)}, b^{(0)}$ and $c^{(0)}$ be given as three initial approximations for unknowns a, b and c respectively. Set $j := 0$.

STEP 1. Apply PROCEDURE-1 for solving n cubic equations (2.7) with $u_1 = a^{(0)}, u_2 = b^{(0)}$ and $u_3 = c^{(0)}$. In other words, solve the n cubic equations (2.7) for $t_k = t_k^{(0)}$ $(k = 1, \dots, n)$ with respect to $a = a^{(0)}, b = b^{(0)}$ and $c = c^{(0)}$.

dCollection

Each cubic equation may have one or three real roots. Let D_i be the discriminant

$$D_i = (\frac{p_i}{3})^3 + (\frac{q_i}{2})^2. \tag{2.10}$$

In case $D_i > 0$ we have one real root

$$v_1 = A_i - \frac{p_i}{3A_i}, \tag{2.11}$$

where

$$A_i = (-\frac{q_i}{2} + \sqrt{D_i})^{\frac{1}{3}}. \tag{2.12}$$

We then set $t_i^{(0)} := v_1$. For $D_i \leq 0$ we have three real roots:

$$v_1 = B_i - \frac{p_i}{3B_i}, \tag{2.13}$$

$$v_2 = \omega B_i - \frac{\omega^2 p_i}{3B_i}, \tag{2.14}$$

$$v_3 = \omega^2 B_i - \frac{\omega p_i}{3B_i}, \text{ where} \tag{2.15}$$

$$B_i = \sqrt{\frac{|p_i|}{3}} \exp(\frac{\varphi_i}{3} i) \quad \text{with} \quad \cos\varphi_i = -\frac{q_i}{2\sqrt{(\frac{|p_i|}{3})^3}} \text{ and} \tag{2.16}$$

$$\omega = \exp(\frac{2\pi}{3} i). \tag{2.17}$$

In this case we take one of the three roots assigned to $t_i^{(0)}$ in order to reduce the quadratic function $Q(u)$ most. Namely, if

$$\begin{aligned} &(x_i - a^{(0)} - v_l)^2 + (y_i - b^{(0)} - c^{(0)} v_l^2)^2 \\ &= \min_{k=1,2,3} \left[(x_i - a^{(0)} - v_k)^2 + (y_i - b^{(0)} - c^{(0)} v_k^2)^2 \right], \text{ then we} \end{aligned} \tag{2.18}$$

set

$$t_i^{(0)} := v_l. \tag{2.19}$$

We thus get an approximation $u^{(0)} = (a^{(0)}, b^{(0)}, c^{(0)}, t_1^{(0)}, \ldots, t_n^{(0)})$.

STEP 2. Apply PROCEDURE-2 to the problem of determining an approximation $\ddot{u}^{(j+1)} = (\ddot{a}^{(j+1)}, \ddot{b}^{(j+1)}, \ddot{c}^{(j+1)}, \ddot{t}_1^{(j+1)}, \ldots, \ddot{t}_n^{(j+1)})$ from $u^{(j)} = (a^{(j)}, b^{(j)}, c^{(j)}, t_1^{(j)}, \ldots, t_n^{(j)})$.

Let $\nabla Q = (\hat{a}^{(j)}, \hat{b}^{(j)}, \hat{c}^{(j)}, \hat{t}_1^{(j)}, \ldots, \hat{t}_n^{(j)})$ be the gradient of $Q(u)$ at $u = u^{(j)}$ such that

dCollection

$$(2.20)\quad \hat{a}^{(j)} = -2\sum_{k=1}^{n}(x_k - a^{(j)} - t_k^{(j)}),$$

$$\hat{b}^{(j)} = -2\sum_{k=1}^{n}(y_k - b^{(j)} - c^{(j)}{t_k^{(j)}}^2),$$

$$\hat{c}^{(j)} = -2\sum_{k=1}^{n}{t_k^{(j)}}^2(y_k - b^{(j)} - c^{(j)}{t_k^{(j)}}^2)$$

and

$$\hat{t}_k^{(j)} = -4c{t_k^{(j)}}^2(y_k - b^{(j)} - c^{(j)}{t_k^{(j)}}^2) \\ -2(x_k - a^{(j)} - t_k^{(j)})(k = 1, \ldots, n).$$

Then it follows from n equations (2.7) that

$$(2.21)\quad \hat{t}_k^{(j)} = 0 \quad (k = 1, \ldots, n).$$

We also get an approximation

$$(2.22)\quad \ddot{u}^{(j+1)} = u^{(j)} - \alpha\nabla(Q),$$

where α is obtained by minimizing the single variable function

$$(2.23)\quad h(\alpha) = \sum_{k=1}^{n}[(x_k - (a^{(j)} - \alpha\hat{a}^{(j)}) - t_k^{(j)})^2 + (y_k - (b^{(j)} - \alpha\hat{b}^{(j)}) \\ -(c^{(j)} - \alpha\hat{c}^{(j)}){t_k^{(j)}}^2)^2].$$

STEP 3. Apply PROCEDURE-1 with $u_1 = \ddot{a}^{(j+1)}, u_2 = \ddot{b}^{(j+1)}$ and $u_3 = \ddot{c}^{(j+1)}$. That is, solve the n cubic equations (2.7) for $t_k = t_k^{(j+1)}$ $(k = 1, \ldots, n)$ with respect to $a = \ddot{a}^{(j+1)}, b = \ddot{b}^{(j+1)}$ and $c = \ddot{c}^{(j+1)}$. Let $D_i^{(j+1)}$ be the discriminant

$$(2.24)\quad D_i^{(j+1)} = (\frac{p_i^{(j+1)}}{3})^3 + (\frac{q_i^{(j+1)}}{2})^2,$$

where

$$(2.25)\quad p_i^{(j+1)} = \frac{1 - 2\ddot{c}^{(j+1)}(y_i - \ddot{b}^{(j+1)})}{2{\ddot{c}^{(j+1)}}^2},$$

$$(2.26)\quad q_i^{(j+1)} = \frac{\ddot{a}^{(j+1)} - x_i}{2{\ddot{c}^{(j+1)}}^2}.$$

In case $D_i^{(j+1)} > 0$ we set

Collection

$$t_i^{(j+1)} := A_i^{(j+1)} - \frac{p_i^{(j+1)}}{3A_i^{(j+1)}}, \tag{2.27}$$

where

$$A_i^{(j+1)} = \left(-\frac{q_i^{(j+1)}}{2} + \sqrt{D_i^{(j+1)}}\right)^{\frac{1}{3}}. \tag{2.28}$$

When $D_i^{(j+1)} \leq 0$ there are three real roots:

$$v_1 = B_i^{(j+1)} - \frac{p_i^{(j+1)}}{3B_i^{(j+1)}}, \tag{2.29}$$

$$v_2 = \omega B_i^{(j+1)} - \frac{\omega^2 p_i^{(j+1)}}{3B_i^{(j+1)}}, \tag{2.30}$$

$$v_3 = \omega^2 B_i^{(j+1)} - \frac{\omega p_i^{(j+1)}}{3B_i^{(j+1)}}, \tag{2.31}$$

where

$$B_i^{(j+1)} = \sqrt{\frac{|p_i^{(j+1)}|}{3}} \exp\left(\frac{\varphi_i^{(j+1)}}{3} i\right) \tag{2.32}$$

with $\cos\varphi_i^{(j+1)} = -\dfrac{q_i^{(j+1)}}{2\sqrt{\left(\frac{|p_i^{(j+1)}|}{3}\right)^3}}$ and $\omega = \exp\left(\dfrac{2\pi}{3} i\right)$.

In this case we get $t_i^{(j+1)} = v_l$ such that

$$\begin{aligned}(x_i - \ddot{a}^{(j+1)} - v_l)^2 + (y_i - \ddot{b}^{(j+1)} - \ddot{c}^{(j+1)} v_l^2)^2 \\ = \min_{k=1,2,3} \left[(x_i - \ddot{a}^{(j+1)} - v_k)^2 + (y_i - \ddot{b}^{(j+1)} - \ddot{c}^{(j+1)} v_k^2)^2\right].\end{aligned} \tag{2.33}$$

And so, we get an new approximation

$$u^{(j+1)} := (a^{(j+1)}, b^{(j+1)}, c^{(j+1)}, t_1^{(j+1)}, \ldots, t_n^{(j+1)}) \tag{2.34}$$

with $a^{(j+1)} = \ddot{a}^{(j+1)}, b^{(j+1)} = \ddot{b}^{(j+1)}$ and $c^{(j+1)} = \ddot{c}^{(j+1)}$.

STEP 4. If $Q(u^{(j+1)}) < Q(u^{(j)})$, then we set $j := j+1$ and go back to STEP 2.

Due to the descent property of PROCEDURE-1, one may ensure the convergence of **Algorithm A** to a local minimum independent of the initial approximation $u^{(0)}$. That is, it follows

dCollection

(2.35) $$Q(u^{(j+1)}) \le Q(u^{(j)}) \quad (j = 0, 1, 2, \dots).$$

Nevertheless, convergence to the global minimum may not be guaranteed. Unfortunately, it is possible for our algorithm to converge to other than the absolute minimum.

3. Orthogonal distance fitting of rotated parabolas

Now we are interested in the problem of fitting general parabolas (i.e. rotated parabolas) to given data points $(x_k, y_k) \quad (k = 1, \dots, n)$.

Consider the parabolas rotated with unknown angle θ

(3.1) $$\tilde{y} - b = c(\tilde{x} - a)^2 \quad (c \neq 0)$$

with $\tilde{x} = x\cos\theta + y\sin\theta$ and $\tilde{y} = -x\sin\theta + y\cos\theta$.

Then, the parametric form of rotated parabolas with additional parameter θ can be given by

(3.2) $$\begin{aligned} x\cos\theta + y\sin\theta &= a + t \\ -x\sin\theta + y\cos\theta &= b + ct^2, \end{aligned}$$

or

(3.3) $$\begin{aligned} x &= (a+t)\cos\theta - (b+ct^2)\sin\theta \\ y &= (a+t)\sin\theta + (b+ct^2)\cos\theta. \end{aligned}$$

Our problem is to determine the rotated parabolas (3.1) such that the following quadratic function is minimized

(3.4) $$Q(\tilde{u}) = \sum_{k=1}^{n}([x_k - (a+t_k)\cos\theta + (b+ct_k^2)\sin\theta]^2 + [y_k - (a+t_k)\sin\theta - (b+ct_k^2)\cos\theta]^2).$$

The basic idea for fitting the rotated parabolas is as follows:

Let $(\tilde{x}_k, \tilde{y}_k) \quad (k = 1, \dots, n)$ be any new data points given by rotating (x_k, y_k) with unknown angle θ. Namely,

(3.5) $$\begin{aligned} \tilde{x}_k &= x_k\cos\theta + y_k\sin\theta, \\ \tilde{y}_k &= -x_k\sin\theta + y_k\cos\theta. \end{aligned}$$

Consider the problem of fitting the following rotated parabolas to the above data points $(\tilde{x}_k, \tilde{y}_k) \quad (k = 1, \dots, n)$:

(3.6) $$\tilde{y} - b = c(\tilde{x} - a)^2 \quad (c \neq 0)$$

in its parametric form

dCollection

$$\tilde{x} = a + t$$
$$\tilde{y} = b + ct^2.$$

Then a, b, c and additional unknown θ must be computed in order that the following corresponding quadratic function $\tilde{Q}(\tilde{u})$ is minimized:

$$\tilde{Q}(\tilde{u}) = \sum_{k=1}^{n}[(\tilde{x}_k - a - t_k)^2 + (\tilde{y}_k - b - ct_k^2)^2], \tag{3.7}$$

where the parameter vector $\tilde{u} = (a, b, c, \theta, t_1, \ldots, t_n)$.

The necessary conditions for a minimum of $\tilde{Q}(\tilde{u})$ with respect to a, b, c, $t_1, \ldots, t_n$ are the same as for $Q(u)$ with $x_k = \tilde{x}_k$ and $y_k = \tilde{y}_k$ in (2.3). We thus have the following $(n+3)$ equations:

$$\sum_{k=1}^{n}(\tilde{x}_k - a - t_k) = 0, \tag{3.8}$$

$$\sum_{k=1}^{n}(\tilde{y}_k - b - ct_k^2) = 0, \tag{3.9}$$

$$\sum_{k=1}^{n} t_k^2(\tilde{y}_k - b - ct_k^2) = 0 \tag{3.10}$$

and

$$t_i^3 + \tilde{p}_i t_i + \tilde{q}_i = 0 \quad (i = 1, \ldots, n) \tag{3.11}$$

with

$$\tilde{p}_i = \frac{1 - 2c(\tilde{y}_i - b)}{2c^2}, \tag{3.12}$$

$$\tilde{q}_i = \frac{a - \tilde{x}_i}{2c^2}. \tag{3.13}$$

In addition, there is another equation induced by the necessary condition $\dfrac{\partial \tilde{Q}}{\partial \theta} = 0$. Namely,

$$\sum_{k=1}^{n}[(-x_k \sin\theta + y_k \cos\theta)(\tilde{x}_k - a - t_k) - (x_k \cos\theta + y_k \sin\theta)(\tilde{y}_k - b - ct_k^2)] = 0. \tag{3.14}$$

The above equation (3.14) can be solved for the unknown angle θ dependent upon the other unknowns $a, b, c, t_1, \ldots, t_n$.

dCollection

$$\theta = \arctan\left(\frac{\sum_{k=1}^{n}[y_k(a+t_k) - x_k(b+ct_k^2)]}{\sum_{k=1}^{n}[x_k(a+t_k) + y_k(b+ct_k^2)]}\right). \tag{3.15}$$

Thus, using (3.15) and applying **Algorithm A** to the $(n+3)$ equations from (3.8) to (3.11), one may be able to fit a rotated parabola $\tilde{y} - b = c(\tilde{x} - a)^2$ to the given new data points $(\tilde{x}_k, \tilde{y}_k)$ $(k = 1, \ldots, n)$.

So, in view of the modification of **Algorithm A**, we can present an iteration algorithm for solving our problem. We now describe our algorithm for fitting general parabolas (3.1) to the given data points (x_k, y_k) $(k = 1, \ldots, n)$.

Algorithm B:

STEP 0. Let $a^{(0)}, b^{(0)}, c^{(0)}$ and $\theta^{(0)}$ additionally be given as four initial approximations for unknowns a, b, c and θ respectively.
Set $j := 0$.

STEP 1. Apply PROCEDURE-1 for solving n cubic equations (3.11) with $u_1 = a^{(0)}, u_2 = b^{(0)}, u_3 = c^{(0)}$ and $u_4 = \theta^{(0)}$. In other words, compute $(k = 1, \ldots, n)$

$$\begin{aligned} \tilde{x}_k &= x_k \cos\theta^{(0)} + y_k \sin\theta^{(0)}, \\ \tilde{y}_k &= -x_k \sin\theta^{(0)} + y_k \cos\theta^{(0)}, \end{aligned} \tag{3.16}$$

and solve the n cubic equations (3.11) with $t_k = t_k^{(0)}$ $(k = 1, \ldots, n)$ with respect to $a = a^{(0)}, b = b^{(0)}$ and $c = c^{(0)}$.
We then get an approximation $\tilde{u}^{(0)} = (a^{(0)}, b^{(0)}, c^{(0)}, \theta^{(0)}, t_1^{(0)}, \ldots, t_n^{(0)})$.

STEP 2. Determine an angle $\theta^{(j+1)}$ from (3.15). Namely, get

$$\begin{aligned} &\theta^{(j+1)} \\ &= \arctan\left(\frac{\sum_{k=1}^{n}[y_k(a^{(j)} + t_k^{(j)}) - x_k(b^{(j)} + c^{(j)}t_k^{(j)^2})]}{\sum_{k=1}^{n}[x_k(a^{(j)} + t_k^{(j)}) + y_k(b^{(j)} + c^{(j)}t_k^{(j)^2})]}\right) \end{aligned} \tag{3.17}$$

STEP 3. Compute $(k = 1, \ldots, n)$

$$\begin{aligned} \tilde{x}_k &= x_k \cos\theta^{(j+1)} + y_k \sin\theta^{(j+1)}, \\ \tilde{y}_k &= -x_k \sin\theta^{(j+1)} + y_k \cos\theta^{(j+1)} \end{aligned} \tag{3.18}$$

Collection

and the gradient $\nabla\tilde{Q} = (\hat{a}^{(j)}, \hat{b}^{(j)}, \hat{c}^{(j)}, \hat{\theta}^{(j+1)}, \hat{t}_1^{(j)}, \ldots, \hat{t}_n^{(j)})$ of $\tilde{Q}(\tilde{u})$ at $\tilde{u} = \bar{u}^{(j)} = (a^{(j)}, b^{(j)}, c^{(j)}, \theta^{(j+1)}, t_1^{(j)}, \ldots, t_n^{(j)})$.

Here,

$$(3.19)\quad \begin{aligned} \hat{a}^{(j)} &= -2\sum_{k=1}^{n}(\tilde{x}_k - a^{(j)} - t_k^{(j)}),\\ \hat{b}^{(j)} &= -2\sum_{k=1}^{n}(\tilde{y}_k - b^{(j)} - c^{(j)}t_k^{(j)^2}),\\ \hat{c}^{(j)} &= -2\sum_{k=1}^{n}t_k^{(j)^2}(\tilde{y}_k - b^{(j)} - c^{(j)}t_k^{(j)^2}). \end{aligned}$$

Moreover, it follows from n equations (3.11) and the necessary condition $\dfrac{\partial\tilde{Q}}{\partial\theta} = 0$ that

$$(3.20)\qquad \hat{t}_k^{(j)} = 0 \quad (k = 1, \ldots, n),$$

$$(3.21)\qquad \hat{\theta}^{(j+1)} = 0.$$

Apply PROCEDURE-2 for $\tilde{Q}(\tilde{u})$ with $u^{(0)} = \bar{u}^{(j)}$. Find an approximation

$$(3.22)\qquad \begin{aligned} \ddot{u}^{(j+1)} &= (\ddot{a}^{(j+1)}, \ddot{b}^{(j+1)}, \ddot{c}^{(j+1)}, \theta^{(j+1)}, t_1^{(j)}, \ldots, t_n^{(j)})\\ &= \bar{u}^{(j)} - \alpha\nabla\tilde{Q}. \end{aligned}$$

The value of α is determined so as to minimize the single variable function

$$(3.23)\qquad \begin{aligned} \tilde{h}(\alpha) = \sum_{k=1}^{n}[(\tilde{x}_k - (a^{(j)} - \alpha\hat{a}^{(j)}) - t_k^{(j)})^2 + (\tilde{y}_k - (b^{(j)} - \alpha\hat{b}^{(j)})\\ -(c^{(j)} - \alpha\hat{c}^{(j)})t_k^{(j)^2})^2]. \end{aligned}$$

STEP 4. Apply PROCEDURE-1 for solving n cubic equations (3.11) with $u_1 = \ddot{a}^{(j+1)}, u_2 = \ddot{b}^{(j+1)}, u_3 = \ddot{c}^{(j+1)}$ and $u_4 = \theta^{(j+1)}$. In other words, compute $(k = 1, \ldots, n)$

$$(3.24)\qquad \begin{aligned} \tilde{x}_k &= x_k\cos\theta^{(j+1)} + y_k\sin\theta^{(j+1)},\\ \tilde{y}_k &= -x_k\sin\theta^{(j+1)} + y_k\cos\theta^{(j+1)} \end{aligned}$$

and solve the n cubic equations (3.11) with $t_k = t_k^{(j+1)}$ $(k = 1, \ldots, n)$ with respect to $a = \ddot{a}^{(j+1)}, b = \ddot{b}^{(j+1)}$ and $c = \ddot{c}^{(j+1)}$.

We then get a new approximation

$$(3.25)\qquad \tilde{u}^{(j+1)} = (a^{(j+1)}, b^{(j+1)}, c^{(j+1)}, \theta^{(j+1)}, t_1^{(j+1)}, \ldots, t_n^{(j+1)})$$

Collection

with $a^{(j+1)} = \ddot{a}^{(j+1)}, b^{(j+1)} = \ddot{b}^{(j+1)}$ and $c^{(j+1)} = \ddot{c}^{(j+1)}$.

STEP 5. If $Q(\tilde{u}^{(j+1)}) < Q(\tilde{u}^{(j)})$, then we set $j := j+1$ and go back to STEP 2.

4. Numerical examples

To test our algorithms we give some examples for fitting of parabolas. Especially, we observe convergence of the corresponding quadratic function in each case. In fact, it is certain that each quadratic function converges to a local minimum.

Example 1. Let be given seven data points $(-7,24),(-3,0),(-2,-1),(0,3),(1,8),(4,35),(7,80)$, lying exactly on the parabola $y=(x+2)^2-1$ with $a=-2, b=-1$ and $c=1$.

We use **Algorithm A** for fitting a parabola $y-b=c(x-a)^2$ to the above data points. The quadratic function to be minimized is given by

$$Q(u) = \sum_{k=1}^{n} \left[(x_k - a - t_k)^2 + (y_k - b - ct_k^2)^2\right].$$

(i) Using $a^{(0)}=0, b^{(0)}=1$ and $c^{(0)}=2$ as three initial approximations we obtain good approximations a, b and c very close to the exact values in less than 1000 iterations. This good result is visualized as the solid parabola in Figure 1. As seen in Figure 2, the solid line shows the convergence of $Q(u)$ to a local minimum (properly speaking, the global minimum).

(ii) When $a^{(0)}=0, b^{(0)}=1$ and $c^{(0)}=-2$ we get $a=2.13, b=119.13, c=-6.28$ after 1000 iterations. This result is given as the dashed parabola in Figure 1. Also, the dashes in Figure 2 represents that the quadratic function $Q(u)$ converges to a local minimum.

Here we see that one may receive different local minima for different initial approximations. Thus the choice of good initial approximations is needed to get a best approximation (hopely, the global minimum).

Example 2. By rotating the data points of Example 1 with the angle $\varphi=\frac{\pi}{4}$ in the counterclockwise direction we get the data points $(\frac{-31\sqrt{2}}{2}, \frac{17\sqrt{2}}{2}), (\frac{-3\sqrt{2}}{2}, \frac{-3\sqrt{2}}{2}), (\frac{-\sqrt{2}}{2}, \frac{-3\sqrt{2}}{2}), (\frac{-3\sqrt{2}}{2}, \frac{3\sqrt{2}}{2}), (\frac{-7\sqrt{2}}{2}, \frac{9\sqrt{2}}{2}), (\frac{-31\sqrt{2}}{2}, \frac{39\sqrt{2}}{2}), (\frac{-73\sqrt{2}}{2}, \frac{87\sqrt{2}}{2})$, lying exactly on the rotated parabola

Collection

$$x = \frac{\sqrt{2}}{2}((-2+t)-(-1+t^2)),$$
$$y = \frac{\sqrt{2}}{2}((-2+t)+(-1+t^2)).$$

We use **Algorithm B** for fitting a rotated parabola $\tilde{y} - b = c(\tilde{x} - a)^2 \quad (c \neq 0)$ with $\tilde{x} = x\cos\theta + y\sin\theta$ and $\tilde{y} = -x\sin\theta + y\cos\theta$ to the above data points.

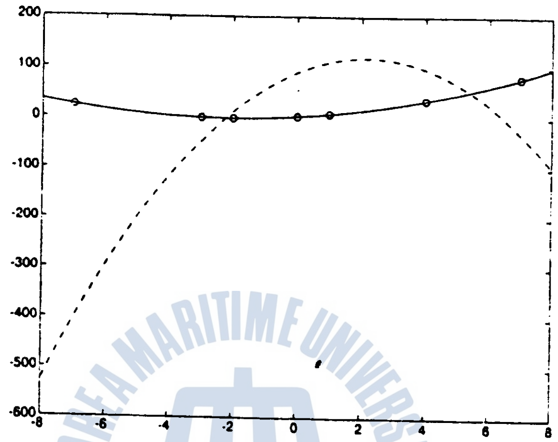

Figure 1

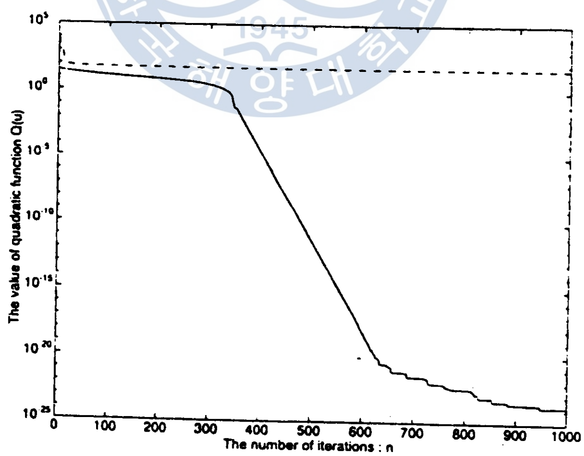

Figure 2

Here the quadratic function $Q(\tilde{u})$ is given by

$$Q(\tilde{u}) = \sum_{k=1}^{n}([x_k - (a+t_k)\cos\theta + (b+ct_k^2)\sin\theta]^2$$

$$+[y_k - (a + t_k)\sin\theta - (b + ct_k^2)\cos\theta]^2),$$

where the parameter vector $\tilde{u} = (a, b, c, \theta, t_1, \dots, t_n)$.

(i) When $a^{(0)} = 0, b^{(0)} = 1$ and $c^{(0)} = 2$ as in Example 1, with $\theta^{(0)} = 0$ as an initial approximation for **Algorithm B** we get $a = -2.85, b = -1.03, c = 1.40$ and $\theta = 0.5642$ after 1000 iterations. We see this result as the dotted parabola fitted to the given data points in Figure 3, and also convergence of the quadratic function $\tilde{Q}(\tilde{u})$ as the dashed line in Figure 4.

(ii) Using $a^{(0)} = 0, b^{(0)} = 1$ and $c^{(0)} = 2$ with $\theta^{(0)} = \frac{\pi}{3}$ for **Algorithm B** we get good approximations near to $a = -2.00, b = -1.00, c = 1.00$ and $\theta = 0.7854$ in less than 1000 iterations. We see this best fit as the solid parabola in Figure 3. The convergence to a local minimum(more precisely, global minimum) is shown by the solid line in Figure 4.

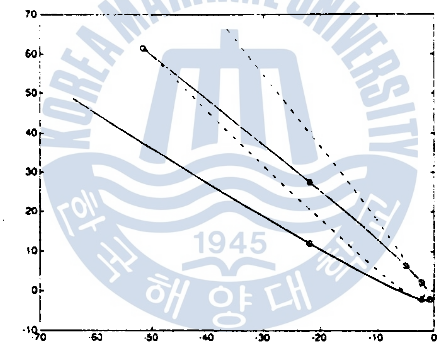

Figure 3

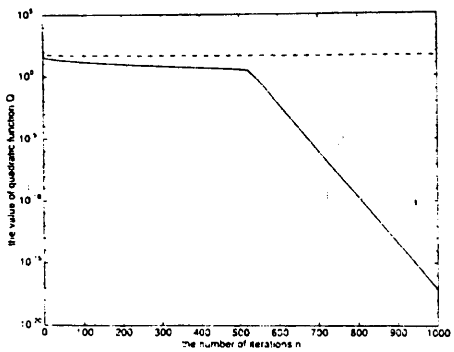

Figure 4

dCollection

Example 3: Given any data points $(-3, 40)$, $(-3, 2)$, $(-2, 30)$, $(-2, 3)$, $(-1, 4)$, $(1, 5)$, $(0, 10)$ we employ **Algorithm A** and **Algorithm B** for fitting a parabola and a rotated parabola respectively. Using $a^{(0)} = 0, b^{(0)} = 1, c^{(0)} = 2$ in common as initial approximations, and additionally $\theta = 0$ for **Algorithm B**, we obtain two parabolas fitted to the given data points. In Figure 5 the dashed parabola and the solid rotated parabola after 2000 iterations are visualized for **Algorithm A** and **Algorithm B** respectively. Also, as can be seen in Figure 6, convergence of each algorithm to a local minimum is guaranteed.

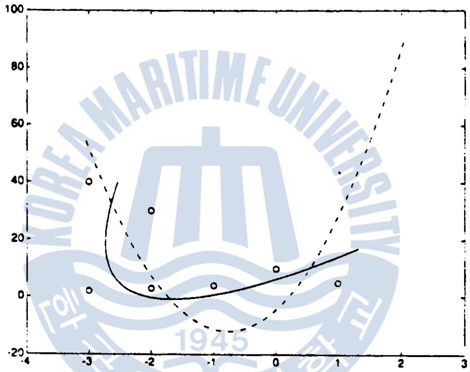

Figure 5

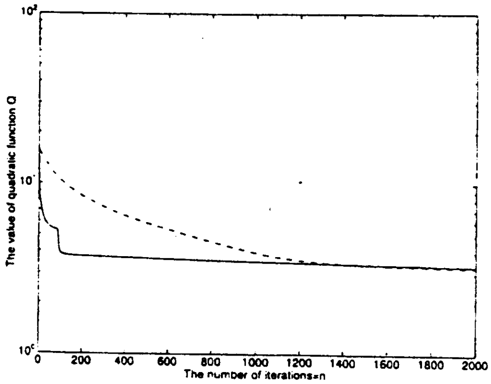

Figure 6

dCollection

References

1. W. Gander, G. H. Golub and R. Strebel, *Least-squares fitting of circles and ellipses*, BIT **34** (1994), 558-578.
2. W. Gander and U. von Matt, *Some least squares problems*, Solving problems in scientific computing using maple and matlab, W. Gander and J. Hřebiček ed., Springer-Verlag, 1993, pp. 251-266.
3. H. Späth, *Orthogonal squared distance fitting with parabolas. Proceedings of IMACS-GAMM International Symposium on Numerical Methods and Error Bounds, Oldenburg*, 1995, pp. 261-269.
4. M. Grossmann, *Parametric curve fitting*, Comp. J. **14** (1971), 169-172.

Ik-Sung Kim
Department of Applied Mathematics
Korea Maritime University
Pusan 606-791, Korea
e-mail:ikim@hanara.kmaritime.ac.kr

dCollection