Revista INTEGRACIÓN
Universidad Industrial de Santander
Escuela de Matemáticas
Vol. 15, No 1, p. 45–61, enero–junio de 1997

On the euclidean distance from a point to a conic

VICTORIA HERNÁNDEZ MEDEROS*
PABLO BARRERA SÁNCHEZ†
JORGE ESTRADA SARLABOUS*

Abstract

In this paper a new algorithm to compute the euclidean distance from a point to a conic is presented. This algorithm provides good approximations for the euclidean distance, even when the point is not very close to the given conic. Furthermore, the approximations may be improved iteratively to attain a prescribed accuracy. Unlike the most commonly known methods to approximate the euclidean distance, in the proposed method the coordinates of the footpoint for the orthogonal projection of the point on the conic are computed. This particular feature permits to obtain a noteworthy accuracy without increasing too much the computational cost.

A procedure to fit a conic section to a scattered set of points inside a triangle is discussed. The procedure is based on minimizing the sum of squared orthogonal distance of data points from the conic. The approximate orthogonal distances are computed using the previous algorithm.

Keywords. Conics, approximate distance, implicit conic section fitting, least squares. MSC: 65Y25, 51N35.

1. Introduction

At a first glance, it seems very unlikely to be able to say something new concerning such an extensively treated subject

*Centro de Matemática y Física Teórica, ICIMAF

†Facultad de Ciencias, UNAM, México

as the computation of the euclidean distance from a point to a conic (see [Book], [Pav], [Samp] and [Tau2]). Most of the existing methods to deal with this problem avoid the computation of the coordinates of the footpoint of the orthogonal projection of the point on the conic, therefore there is no control on the accuracy of the obtained approximate distance. Moreover, instead of the exact euclidean distance a "suitable" approximation is computed, which usually happens to be a good approximation to the euclidean distance only if the point is very close to the conic.

In this paper a new algorithm to compute the euclidean distance from a point to a conic is presented. This algorithm provides good approximations for the euclidean distance, even when the point is not very close to the given conic. Furthermore, the approximations may be improved iteratively to attain a prescribed accuracy without increasing too much the computational cost. A procedure to fit a conic section to a scattered set of points inside a triangle is discussed, minimizing the sum of squared orthogonal distance of data points from the conic. The approximate orthogonal distances are computed using the previous algorithm.

Let's first study the problem of computing the euclidean distance from a point $q = (x_0, y_0)$ on the plane to an arbitrary conic C with implicit equation

$$f(x,y) = a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + a_{00} = 0\,. \tag{1}$$

By definition, the euclidean distance from q to C, $d(q,C)$, is given by

$$d(q,C) = \min\{\|q-p\| : f(p) = 0\}\,. \tag{2}$$

Thus, to compute the euclidean distance we have to solve a constrained nonlinear minimization problem. More geometrically, the euclidean distance from q to C is attained at a point p on C such that the normal of C at p passes through q. Hence the coordinates (x,y) of p and the euclidean distance d, may be computed as the solution of the following nonlinear polynomial system of equations,

$$\begin{aligned} f_1(x,y,d) &:= (x-x_0)^2 + (y-y_0)^2 - d^2 = 0\,, \\ f_2(x,y,d) &:= f(x,y) = 0\,, \\ f_3(x,y,d) &:= \frac{\partial f}{\partial y}(p)(x-x_0) - \frac{\partial f}{\partial x}(p)(y-y_0) = 0\,. \end{aligned} \tag{3}$$

The efficient numerical solution of nonlinear system (3) requires a good initial approximation of the roots, i.e. of the coordinates of footpoint p and of the euclidean distance d. Up to the moment, general methods for estimating initial approximations for x, y and d are not reported in the literature. Therefore, some other approaches have been studied.

Using elimination theory Kriegman and Ponce [Krie] and Ponce et al [Pon] eliminate the variables x and y (whose initial approximations are more difficult to estimate) and obtain a single polynomial equation on d, $\Phi(d) = 0$, whose minimum positive root d^*, is the euclidean distance from q to C. Unfortunately, the coefficients of $\Phi(d)$ are complicated polynomial expressions in the coefficients of f and in the coordinates of the external point q. Hence, its computation may be expensive and the problem of finding the roots of Φ, using floating point arithmetic, may be numerically unstable.

To overcome these difficulties, other approximations of the euclidean distance have been considered. The simplest is the algebraic distance, $d_a(q, C)$ given by

$$d_a(q, C) = |f(x_0, y_0)|. \tag{4}$$

To compute the algebraic distance is very cheap , but it is a poor approximation of the euclidean distance. In [Pav] Pavlidis proposes other approximations to the euclidean distance from a point to a conic, but they depend on the type of the conic and only when the points are located in some regions the approximations are good.

Taubin presents in [Tau2] several approximations of the euclidean distance from a point to an implicit curve $f(x, y) = 0$, if the function $f(x, y)$ has continuous partial derivatives up to order $k+1$ in a neighborhood of q. Taubin's approximate distance of order k, δ_k, is a lower bound for the euclidean distance from q, to the set of zeros of its Taylor polynomial of order k around q

$$T^k f_q(x, y) = \sum_{h=0}^{k} \sum_{i+j=h} f_{ij}(x - x_0)^i (y - y_0)^j,$$

where

$$f_{ij} = \frac{1}{i!j!} \frac{\partial^{i+j}}{\partial x^i \partial y^j} f(x_0, y_0).$$

More precisely, δ_k, is the unique positive root of the univariate polynomial

$$F_q^k(\delta) = |F_0| - \sum_{h=1}^{k} \|F_h\| \, \delta^h,$$

where F_h is the vector of the normalized coefficients

$$\left\{ f_{ij} / \binom{h}{i} : i + j = h \right\}.$$

In particular, Taubin's approximate distance of first order, δ_1, is the root of the linear polynomial

$$F_q^1 = |F_0| + \|F_1\| \cdot \delta \ ,$$

where

$$F_0 = f_{00} = f(x_0, y_0)\,, \tag{5}$$

$$F_1 = (f_{01}, f_{10}) = \left(\frac{\partial f}{\partial x}(x_0, y_0), \frac{\partial f}{\partial y}(x_0, y_0)\right). \tag{6}$$

Thus,

$$\delta_1 = \frac{|F_0|}{\|F_1\|} = \frac{|f(x_0, y_0)|}{\|\nabla f(x_0, y_0)\|}. \tag{7}$$

Analogously, Taubin's approximate distance of second order, δ_2, is the unique positive root of the quadratic polynomial

$$F_q^2(\delta) = |F_0| - \|F_1\|\,\delta - \|F_2\|\,\delta^2 \ ,$$

where F_0 and F_1 are given by (5) and (6) and

$$F_2 = (f_{02}, f_{11}/2, f_{20}) = \left(\frac{\partial^2 f}{\partial x^2}(x_0, y_0), \frac{1}{2}\frac{\partial^2 f}{\partial x \partial y}(x_0, y_0), \frac{\partial^2 f}{\partial y^2}(x_0, y_0)\right). \tag{8}$$

Hence,

$$\delta_2 = \frac{-\|F_1\| + \sqrt{\|F_1\|^2 + 4\,|F_0|\,\|F_2\|}}{2\,\|F_2\|} \ . \tag{9}$$

Since $T^2 f_q(x, y) \equiv f(x, y)$ when f is a polynomial of degree 2, δ_2 is a lower bound for the euclidean distance from q to the conic. Moreover in this case $\delta_k = \delta_2$ for $k \geq 2$.

In the next section we introduce a new algorithm for computing the euclidean distance from a point q to a conic: the normal distance. While the previous algorithms only compute an approximation of the euclidean distance d , the normal distance provides us with the coordinates of the footpoint p on the conic. On the other hand, since the normal distance is obtained by means of an iterative convergent process from a good initial guess, we may compute the euclidean distance with a previously fixed precision. A direct application of the normal distance is presented in section 3: given a set of points inside a triangle, we fit it by means of a conic section, minimizing the mean square normal distance.

2. Normal distance from a point to a conic

2.1. Computing the normal distance

Let $C = \{(x, y) : f(x, y) = 0\}$ where f is given by (1) an arbitrary conic and $q = (x_0, y_0)$ a point not on C. Then $C_q = \{(x, y) : f(x, y) = f(x_0, y_0)\}$ is also a conic and it is the level curve of the surface $z = f(x, y)$ which passes through the point q. Let us consider a parametric equation $n = (x_1(t), y_1(t))$ of the normal line to C_q at q,

$$\begin{aligned} x_1(t) &= x_0 - \frac{\partial f(x_0, y_0)}{\partial x}\, t \,, \\ y_1(t) &= y_0 - \frac{\partial f(x_0, y_0)}{\partial y}\, t \,. \end{aligned}$$

From (1) we compute the derivatives and substituing in the above equations we obtain

$$x_1(t) = x_0 - t(2a_{20}x_0 + a_{11}y_0 + a_{10}), \qquad (10)$$

$$y_1(t) = y_0 - t(a_{11}x_0 + 2a_{02}y_0 + a_{01}). \qquad (11)$$

If q is close to the conic, the "shape" of C and C_q are very similar (see Fig. 2.1.). Thus, if n intersects the conic C at a point p, the distance between q and p is a good approximation of the euclidean distance from q to C.

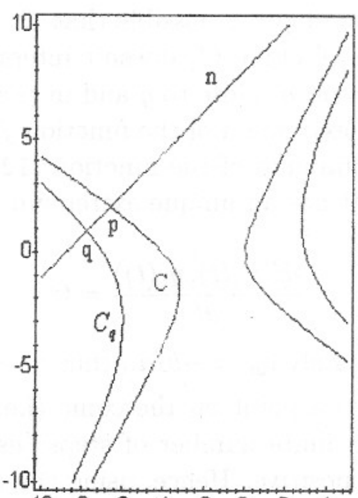

Figure 2.1: Conic C, external point q and level curve C_q.

To compute the intersection between n and C, we substitute $x = x_1(t)$ and $y = y_1(t)$ in (1) . Hence, the intersection is attained at the minimum positive root t^* of the equation

$$f(x_1(t), y_1(t)) = 0 .$$

After some simplifications using MAPLEV, we obtain a quadratic equation in t whose coefficients depend on the coefficients of f , the coordinates of the external point q and on s, i.e.,

$$f(x_1(t), y_1(t)) = at^2 + bt + c , \tag{12}$$

where the expressions for a, b and c are given in the appendix A. If the discriminant $D = b^2 - 4ac$ of the quadratic equation is non negative then

$$t^* = \frac{-2c}{b + \text{sign}(b)\sqrt{D}} .$$

Hence,

$$p = (x_1(t^*), y_1(t^*)) ,$$

and the normal distance from q to C is given by

$$d_n(q, C) = \sqrt{(x_0 - x_1(t^*))^2 + (y_0 - y_1(t^*))^2} \tag{13}$$

Depending on the position of q it is possible that $D < 0$. (see Fig. 2.1.). Then the normal line to the level curve C_q doesn't intersect the conic C. In this case, we look for a new point q_1 close to q and in the positive region of D. To obtain q_1 we compute an extremum of the function f along the normal line to C_q at q, i.e. we seek a minimum of the function (12). Since $f(x_1(t), y_1(t))$ is a quadratic polynomial, it has an unique extremum t_{op}, which is the solution of the linear equation

$$\frac{df(x_1(t), y_1(t))}{dt} = 0 .$$

From (12) we get immediately $t_{op} = -b/a.$, thus $q_1 = (x_1(t_{op}), y_1(t_{op}))$.

This procedure lead us to a point on the conic (i.e. to a point p such that $f(p) = 0$). Then, after a finite number of steps (usually 1 or 2 steps) q_1 is in the region where D is positive. Hence, using the previous method we may compute the intersection p between the normal line to the level curve C_{q_1} at q_1 and the conic C. In the next sections we call p the normal projection of q on C.

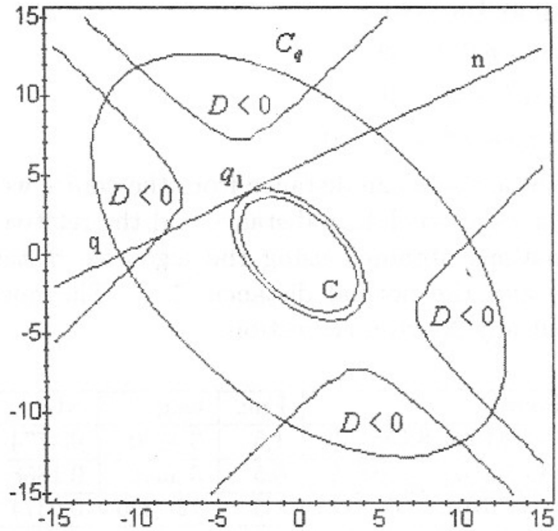

Figure 2.2: Computation of normal distance from q to C.

2.2. Correcting the normal distance

Since the previous algorithm give us not only an approximate value of the euclidean distance, but also the coordinates of the normal projection, we can correct the value of the distance until a previously fixed precision is attained. In fact, using dn and $p = (x_p, y_p)$ as (good) initial guests we may solve the nonlinear system (3) by Newton's method, to correct the value of the euclidean distance and the position of the footpoint . More precisely, the new approximation $v^j = \left(x_p^j, y_p^j, d_n^j\right)^t$ after the jth-step is given by

$$v^j = v^{j-1} + \Delta v^j \,,$$

where Δv^j is the solution of the linear system

$$J(v^j)\Delta v^j = -F(v^j)\,,$$

$F = (f_1, f_2, f_3)$ according with (3) and J is the Jacobian matrix of F.

2.3. Normal distance versus other approximations of the euclidean distance

Now we show some numerical experiments to compare the accuracy of some distances as approximations of the euclidean distance from a point to a conic.

In table 1 we display the conics

parabola: $x^2 - 2x - y + 1 = 0$;

ellipse: $x^2 + xy + y^2 - 4x = 0$;

hyperbola:$-x^2 - xy + y^2 - 4x = 0$,

and the points whose euclidean distance from the conics were computed. Besides, we show the exact euclidean distance and the relative errors associated to the approximations obtained using the algebraic distance (4), Taubin's distances(7), (9) and the normal distance (13). The normal distance was computed without any Newton correction.

Conics	Points	Euc.	ealg	ed_1	ed_2	e_n
parabola	$(-0.41524, 8.65543)$	1.5	3.4350	0.4774	0.2136	0.0175
ellipse	$(4.556152, -1.61575)$	0.5	3.4340	0.1858	0.0872	0.0058
hyperbola	$(-7.73701, 0.95531)$	1.5	12.7396	0.0374	0.2136	0.0274

In the next experiment we consider a set of points inside the "canonical" triangle T whose vertexes are $b_0 = (0,0)$,$b_1 = (0,1)$ and $b_2 = (1,0)$. A conic section C passing through b_0 and b_2 and tangent to the sides of T has the following implicit equation

$$f(x,y) = y^2 - 4w_1^2 x(1 - x - y) = 0\,, \tag{14}$$

where $w_1 \geq 0$ is a free parameter. It can be also written in the Bernstein-Bezier parametric representation as $c(t)$,$0 \leq t \leq 1$,

$$c(t) = \frac{b_0 B_0^2(t) + w_1 b_1 B_1^2(t) + b_2 B_2^2(t)}{B_0^2(t) + w_1 B_1^2(t) + B_2^2(t)}\,, \tag{15}$$

where $B_i^2(t) = \binom{2}{i} t^i (1-t)^{2-i}$ $i = 0, 1, 2$ are the Bernstein's polynomials of degree 2. In this representation the point $S = c(1/2)$ is called the shoulder point of the conic section and holds [Far]

$$w_1 = \frac{||b_m - S||}{||S - b_1||}\,,$$

with b_m the middle point of the segment $b_0 b_2$. Hence, when C is a parabola ($w_1 = 1$) the shoulder point is the middle point of $b_m b_1$. An ellipse corresponds to w_1 in $(0, 1)$ and a hyperbola to w_1 in $(1, \infty)$. Given w_1 we generate 50 points $p_i = (x_i, y_i)$ on the corresponding conic and compute the unit normal n_i at p_i and the points

$$q_i^+ = p_i + \varepsilon\, n_i\,,$$
$$q_i^- = p_i - \varepsilon\, n_i\,,$$

for given ε.

We choose then those $q_i, i = 1, ..., m$ ($m \leq 100$) which are inside of the triangle T and compute

$$\begin{aligned} d_a^i &= d_a(q_i, C)\,, \\ d_{T1}^i &= d_{T1}(q_i, C), \\ d_{T2}^i &= d_{T2}(q_i, C), \\ d_n^i &= d_n(q_i, C), \end{aligned}$$

where d_a, d_{T1}, d_{T2} and d_n represent the algebraic, Taubin's and normal distances given by (4),(7), (9) and (13) respectively.

If ε is small then the euclidean distance from all the points q_i to the conic section is ε. Thus we may compute the relative error associated with each approximation to the euclidean distance and finally a global measure of the relative error given respectively by

$$\begin{aligned} e_a &= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{d_a^i - \varepsilon}{\varepsilon}\right)\,, \\ e_{T1} &= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{d_{T1}^i - \varepsilon}{\varepsilon}\right), \\ e_{T2} &= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{d_{T2}^i - \varepsilon}{\varepsilon}\right), \\ e_n &= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{d_n^i - \varepsilon}{\varepsilon}\right)\,. \end{aligned}$$

Table 2 shows the results of the experiments for $\varepsilon = 0.01, 0.05$, and 0.1 and different values of w_1 that correspond to ellipses, and hyperbolas. The last column is the number of points at which Newton corrections were made in the computation of the normal distance. The best performance is indicated by $(*)$

ε	w_1	ealg	ed_1	ed_2	e_n	m	Nc
0.01	0.2	0.7472	0.0246	0.0321	*0.0004	72	0
	0.6	0.6237	0.0095	0.0070	*0.00003	57	0
	5	107.35	0.0095	0.0083	*0.0001	51	0
	10	431.59	0.0107	0.0114	*0.00005	51	0
0.05	0.2	0.7534	0.1999	0.1272	*0.0080	46	0
	0.6	0.4683	0.0503	0.0330	*0.0011	41	0
	5	94.78	0.0529	0.0376	*0.0017	39	1
	10	381.08	0.0640	0.0431	*0.0015	39	1
0.1	0.2	0.6822	0.3445	0.1636	*0.0182	17	3
	0.6	0.3042	0.1063	0.0695	*0.0060	33	0
	5	82.71	0.1222	0.0753	*0.0107	32	2
	10	342.98	0.1246	0.0622	*0.0085	31	1

From Tables 1 and 2 we conclude that the normal distance is the best approximation to the euclidean distance from a point to a conic. Nevertheless, its computation is more expensive than the remaining approximations.

3. Least squares fitting with conics sections

3.1. The problem and previous works

As geometric models or shape descriptors, conics are used in interactive graphics systems for the automated construction of object models and for building intermediate representations from data during recognition process [Tau1]. The fitting of conic sections have been also used in the approximation of digitized drawing for data compaction and pattern recognition, and in other applications that involve alphanumeric characters [Pav].

Several methods are available for fitting data by arcs of conics [Pav], [Alb], [Book], [Samp], [Gan]. Given a finite set of data points $q_i = (x_i, y_i)$ $i = 1, .., n$ the classical fitting approach consists on determining a conic C that minimizes the mean square distance

$$\frac{1}{n}\sum_{i=1}^{n} d^2(q_i, C)\,, \tag{16}$$

where $d(q_i, C)$ from p_i to C. Since it is not possible to give a closed expression for the euclidean distance from a point to a conic, there are three classes of solutions for problem (16) . The first one uses the parametric representations for conics, consequently one parameter is introduced for each point. Thus, the dimension of the nonlinear least squares problem grows with the number of data points. Some previous works using this approach are [Spa], [Gan]. The second type of solutions to problem (16) is based on minimizing the mean square error

$$\frac{1}{n}\sum_{i=1}^{n} f^2(q_i) \tag{17}$$

with different constraints. To obtain a nontrivial solution of (17) a constraint on the coefficients of conics has to be imposed, since the implicit equation of a conic is homogeneous Some authors have considered linear constraints turning, the minimization of (17) into a linear regression problem, whereas others have proposed quadratic restrictions converting (17) into an eigenvalue problem. Moreover, depending on the constraints, the fitting methods may possess special properties , for instance Taubin's generalized eigenvector fit [Tau3] and the Bookstein algorithm [Book]are invariant under affine transformations, while some other methods include straight lines as possible solutions [Gna],[Pra], [Pat].

The third type of fitting technique consists on approximating the euclidean distance from a point to a conic through a closed formula [Tau2] or an iterative method [Pon]. In the next section we present an algorithm of this type to fit a conic section to a set of points inside a triangle.

3.2. Normal distance fitting

Given a set of points $q_i = (x_i, y_i)$ $i = 1, .., n$ inside a triangle T whose vertexes are the points b_o, b_1, b_2 we want to fit it by means of a conic section C. This is a very important stage when we are fitting data with conic splines. Some previous related works on conic splines are [Baj],[Book], [Pra], [Pav], [Samp].

To construct a C^1 conic spline, each conic section must satisfy the following conditions: C must pass through the vertexes b_0 and b_2 of T and its tangents in these points must be the lines defined by b_0b_1 and b_1b_2 respectively. When data points are in a triangle these assumptions are natural conditions of the problem (see Fig. 3.2.).

Under these conditions there is only one free parameter in the conic [Hoff] which is called w_1. For instance if C is defined in the canonical triangle then its implicit equation is given by (14).

On the other hand, for an arbitrary triangle T, the conic section C satisfies the equation (14) in baricentric coordinates , i.e. if $(u, v, 1-u-v)$ are the baricentric coordinates with respect to b_0, b_1, b_2 of a point (x, y) on the conic then,

$$C : f(x,y) = v^2 - 4w_1^2 u(1-u-v) = 0\,,$$

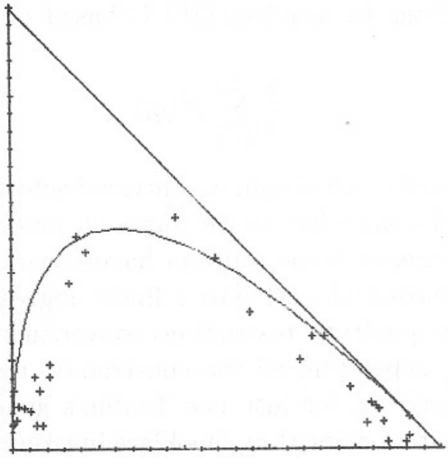

Figure 3.3: Points inside a triangle and fitting conic.

where

$$u = \frac{\begin{vmatrix} x & b_{1x} & b_{2x} \\ y & b_{1y} & b_{2y} \\ 1 & 1 & 1 \end{vmatrix}}{d}, v = \frac{\begin{vmatrix} b_{0x} & x & b_{2x} \\ b_{0y} & y & b_{2y} \\ 1 & 1 & 1 \end{vmatrix}}{d},$$

and $b_0 = (b_{0x}, b_{0y}), b_1 = (b_{1x}, b_{1y}), b_2 = (b_{2x}, b_{2y})$ and d is the determinant of the matrix

$$\begin{bmatrix} b_{0x} & b_{1x} & b_{2x} \\ b_{0y} & b_{1y} & b_{2y} \\ 1 & 1 & 1 \end{bmatrix}$$

associated with the triangle T.

The least squares problem in the normal distance is given by

$$\min_{w_1} \frac{1}{n} \sum_{i=1}^{n} d_n^2(q_i, C)\,, \tag{18}$$

where $d_n(q_i, C)$ is the normal distance from q_i to C.

Since the normal distance from a point to a conic is a nonlinear function of the parameter w_1, the least squares fitting problem (18) is a constrained ($w_1 > 0$) nonlinear minimization problem. To compute the solution of this problem it is important to scale the parameter w_1. In fact, observe that w_1 in $(0, 1]$ corresponds to a conic section whose shoulder point

$$S = \frac{b_0 + 2w_1 b_1 + b_2}{2(1 + w_1)}$$

is located in-the line through $(b_1+b_m)/2$ and b_m, while w_1 in $[1,\infty)$ represents a conic section whose shoulder point is in the line that passes through $(b_1+b_m)/2$ and b_1. Thus, to reduce the parameter interval to $(0,1]$ we compute the maximum value $w_{1\,\max}$ of the parameters w_1^i of conic sections passing through q_i $i=1,..,n$,

$$w_{1\,\max} = \max_{i=1,..,n} w_1^i \ ,$$

where

$$w_1^i = \frac{v_0^i}{2\sqrt{v_0^i(1-u_0^i-v_0^i)}}$$

and u_0^i, v_0^i are the baricentric coordinates of the point q_i

To solve the least squares problem (18) in the new parameter $w_1^* = \frac{w_1}{w_{1\,\max}}$ we use the MATLAB function $f\min$. Additionally, we made up a MATLAB program to compute fitting conics using different approximations of the euclidean distance.

3.3. Numerical examples

In this section we show some examples to compare the quality of the fitting curves obtained minimizing different approximations of the mean square distance. In each experiment, data points were generated by a procedure similar to the described in section 2.3. Given a curve C fitting in some distance the data points q_i $i=1,..,n$, we compute the residual

$$r = \sum_{i=1}^{n} d^2(q_i, C) \ ,$$

where $d(q_i, C)$ is the euclidean distance from q_i to C. To compute the euclidean distance from a point $q=(x_0,y_0)$ to C we may use the Bezier's representation of a conic section (15) and the coordinates of q. In fact, if $x=x(t), y=y(t)$ is the parametric Bezier representation of C, we have to compute the values of the parameter t, such that the normal line to C at the point $(x(t), y(t))$ passes through the external point q,i.e. t, is a solution of

$$x^{'}(t)(x(t)-x_0) + y^{'}(t)(y(t)-y_0) = 0 \ .$$

This equation may be rewritten as a polynomial of degree 4:

$$\Gamma(t) = a(q)t^4 + b(q)t^3 + c(q)t^2 + d(q)t + e(q),$$

whose coefficients depend on the coordinates of q.

If t^* is a real root of $\Gamma(t)$ in $[0,1]$ then $p = c(t^*)$ is a point on the conic section C such that the normal to C at p passes through q. Hence,

$$d(q, C) = \min\{\|q - c(t^*)\|, t^* \in (0,1), \Gamma(t^*) = 0\}.$$

The results corresponding to some data points are shown in table 3. For each data file, the residual r and the optimum value of w_1 associated with different approximations to the euclidean distance are displayed. The first column corresponds to the algebraic distance, while columns 2 and 3 to Taubin's first and second order approximations. The last column represents the normal distance. In Appendix B we include the data points of the examples.

File	Algebraic	Taubin1	Taubin2	Normal
$w01.dat$	$r = 0.3188$ $w_1 = 0.1091$	$r = 0.3943$ $w_1 = 0.1530$	$r = 0.3734$ $w_1 = 0.0674$	$r = 0.3179$ $w_1 = 0.1044$
$w06.dat$	$r = 0.3069$ $w_1 = 0.4311$	$r = 0.2352$ $w_1 = 0.6076$	$r = 0.2329$ $w_1 = 0.6379$	$r = 0.2326$ $w_1 = 0.6552$
$w1.dat$	$r = 0.4205$ $w_1 = 0.6631$	$r = 0.2690$ $w_1 = 0.7893$	$r = 0.2634$ $w_1 = 0.8616$	$r = 0.2629$ $w_1 = 0.8908$
$w3.dat$	$r = 0.6854$ $w_1 = 0.6631$	$r = 0.3046$ $w_1 = 1.4769$	$r = 0.1683$ $w_1 = 2.8294$	$r = 0.1683$ $w_1 = 2.8774$
$w9.dat$	$r = 0.7024$ $w_1 = 0.2990$	$r = 0.3367$ $w_1 = 1.3560$	$r = 0.1953$ $w_1 = 5.0355$	$r = 0.1953$ $w_1 = 5.0391$

From table 3 we observe that the residuals corresponding to normal fitting conics are the smallest, whereas the computation of the normal distance and in consequence the least squares fit in the normal distance, is more expensive than the in others approximations.

Appendix A

The coefficients of the quadratic polynomial (12) are

$$a = z_{20}x_0^2 + z_{11}x_0y_0 + z_{02}y_0^2 + z_{11}x_0y_0 + z_{10}x_0 + z_{01}y_0 + z_{00},$$

where

$$\begin{aligned}
z_{20} &= 4a_{20}^3 + a_{02}a_{11}^2 + 2a_{20}a_{11}^2,\\
z_{11} &= a_{11}(4a_{20}a_{02} + 4a_{20}^2 + a_{11}^2 + 4a_{02}^2),\\
z_{02} &= 4a_{02}^3 + a_{20}a_{11}^2 + 2a_{02}a_{11}^2,\\
z_{10} &= 4a_{20}^2a_{10} + 2a_{11}a_{20}a_{01} + 2a_{02}a_{11}a_{01} + a_{10}a_{11}^2,\\
z_{01} &= 2a_{20}a_{11}a_{10} + 2a_{11}a_{10}a_{02} + 4a_{02}^2a_{10} + a_{11}^2a_{01},\\
z_{00} &= a_{20}a_{10}^2 + a_{11}a_{10}a_{01} + a_{02}a_{01}^2;
\end{aligned}$$

$$b = s_{20}x_0^2 + s_{11}x_0y_0 + s_{02}y_0^2 + s_{11}x_0y_0 + s_{10}x_0 + s_{01}y_0 + s_{00}\,,$$

where

$$\begin{aligned} s_{20} &= -a_{11}^2 - 4a_{20}^2 \quad , \\ s_{11} &= -4a_{11}(a_{20} + a_{02})\,, \\ s_{02} &= -a_{11}^2 - 4a_{02}^2 \quad , \\ s_{10} &= -2a_{11}a_{01} - 4a_{20}a_{10}\,, \\ s_{01} &= -2a_{11}a_{10} - 4a_{02}a_{01}\,, \\ s_{00} &= -a_{10}^2 - a_{01}^2\,; \end{aligned}$$

and

$$c = f(x_0, y_0)\,.$$

Appendix B

Here we include the data points considered in the examples of section 3.3

File *w01.dat*	File *w06.dat*	File *w1.dat*	File *w3.dat*	File *w9.dat*
.1553 .0618	.7421 .2011	.8387 .0624	.0140 .2589	.0858 .8368
.1597 .1383	.0662 .2087	.0704 .0537	.0095 .0644	.0088 .0591
.1683 .1445	.0398 .0741	.6732 .2009	.9014 .0290	.0643 .1078
.1961 .0184	.9096 .0210	.1738 .4455	.9268 .0194	.0429 .0513
.2053 .0212	.0088 .2701	.0048 .0563	.0809 .2541	.9188 .0405
.2191 .0825	.3598 .3804	.0620 .1313	.8602 .0943	.8373 .1467
.2351 .0436	.9387 .0316	.0269 .0904	.8874 .0389	.0444 .0892
.2537 .1103	.0639 .3515	.0871 .1606	.8287 .1579	.0225 .1881
.2651 .1386	.6798 .2911	.0137 .0963	.0403 .4380	.0119 .0981
.3011 .0508	.0638 .0671	.0013 .0524	.9326 .0540	.9391 .0349
.3278 .1278	.8861 .0554	.8371 .1036	.0233 .1386	.8958 .0387
.3645 .0594	.0724 .0858	.5562 .3113	.9089 .0613	.0035 .1485
.3832 .0107	.0241 .0749	.9183 .0136	.2556 .6288	.8188 .1795
.4164 .1136	.0684 .1120	.7896 .1377	.9275 .0721	.8954 .0296
.4355 .0319	.9245 .0224	.0048 .1290	.1497 .7611	.0184 .0731

.5455	.0896	.3353	.4169	.7292	.2523	.7834	.1619	.0347	.1074
.5611	.0404	.8065	.1101	.9261	.0687	.0666	.7137	.9123	.0278
.6019	.1654	.0830	.0817	.8281	.0574	.0441	.6865	.6961	.2226
.6099	.0246	.8629	.1258	.6996	.2544	.8906	.1067	.0285	.3923
.6279	.0593	.6611	.2801	.1342	.3785	.0114	.1828	.8129	.1325
.6616	.1031	.0012	.1699	.3828	.5247	.8942	.0656		
.6774	.1189	.0607	.2111	.8780	.0157				
.6953	.0274	.0154	.0721	.8488	.0949				
.7202	.1432	.9015	.0383	.8566	.0881				
.7208	.0412	.2916	.3667	.0552	.0513				
.7352	.0673	.0574	.0530	.0523	.1603				
.7480	.0054	.0767	.0571	.0881	.1905				
.7690	.0872	.8728	.0957	.8679	.0590				
.7770	.0776	.7264	.1990	.1500	.4826				
.7939	.1299	.0168	.0985						
.7958	.0857								
.8113	.1306								
.8136	.0547								

Acknowledgements

The results contained in the present paper were partially obtained during research stays of the first author at ICTP/Trieste (under a grant of the ICTP Associate and Federation Schemes) and at Facultad de Ciencias, UNAM/México City (thanks a TWAS South-South Fellowship). Herewith she would like to express her gratitude to the ICTP Math. Section for their warm hospitality.

We wish to thank R. Patterson and M. Paluszny for their long lasting support and encouragement and to A. González for the valuable comments and discussions.

References

[Alb] Albano, A. Representation of digitized contours in terms of conic arcs and straight-line segments. Comput. Graphics and Image Processing, Vol. 13, 23-33, 1974.

[Baj] Bajaj, C.L. A-Splines:Local interpolation and approximation using C^k-continuous piecewise real algebraic curves. Computer Science Technical Report, CAPO-92-44, Purdue University, 1992.

[Book] Bookstein, F.L. Fitting conic sections to scattered data. Comput. Vision, Graphics and Image Processing, Vol. 9, 56-71,1979.

[Far] Farin, G. Curves and Surfaces for Computer Aided Geometric Desing, Academic Press, 1992.

[Gan] Gander, W., Golub, G.H., Strebel, R. Least squares fitting of circles and ellipses. BIT 34, 558-578, 1994.

[Gna] Gnanadesikan, R. Methods for Statistical Data Analysis of Multivariate Observations. New York, Wiley, 1977

[Hoff] Hoffmann, C. M., Algebraic and numerical techniques for offsets and blends in Computation of Curves and Surfaces, Kluwer Academic Publishers, 499-528, 1990.

[Krie] Kriegman, D.J, Ponce,J. On recognizing and positioning curved 3D objects from image contours. IEEE Trans. Pattern Anal. Mach. Intell.- 12, December , 1127-1137, 1990.

[Pat] Paton, K. A. Conics sections in automatic chromosomes analysis. Machine Intell., Vol5, 411, 1970

[Pav] Pavlidis, T. Curve fitting with conic splines. ACM Trans. on Graphics, Vol. 2, No 1, January1983.

[Pon] Ponce A.,Hoogs, J. and Kriegman, D. On using CAD models to compute the pose of curved 3D objects. Comput. Vision, Graphics and Image Processing 55,2, 184-197, 1992.

[Pra] Pratt, V. Direct least squares fitting of algebraic surfaces. Comput. Graphics, Vol. 21, No. 4, 145-152, July 1987.

[Samp] Sampson, P.D. Fitting conic sections to very scattered data: an iterative refinement of Bookstein algorithm. Comp. Vision, Graphics, and Image Processing, Vol. 18, 97-108, 1982.

[Spa] Spaeth, H. Orthogonal squared distance fitting with parabolas, Numerical Methods and Error Bounds, Mathematical Research Vol. 89, Akademie Verlag, 261-269, 1996.

[Tau1] Taubin, G. Parameterized families of polynomials for bounded algebraic curve and surface fitting. IEEE Trans. Pattern Anal. Machine Intell.,Vol.16,No.3,287-303, March 1994.

[Tau2] Taubin, G. Distance approximations for rasterizing implicit curves. ACM Trans. on Graphics, Vol. 13, No. 1, January 1994.

[Tau3] Taubin, G. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans. on Pattern Anal. Machine Intell., Vol. 13, No 11, 1115-1138, Nov. 1991.