

Least-Squares Fitting of Circles and Ellipses*

Walter Gander

Gene H. Golub

Rolf Strebel

Dedicated to Åke Björck on the occasion of his 60th birthday.

Abstract

Fitting circles and ellipses to given points in the plane is a problem that arises in many application areas, e.g. computer graphics [1], coordinate metrology [2], petroleum engineering [11], statistics [7]. In the past, algorithms have been given which fit circles and ellipses in *some* least squares sense without minimizing the geometric distance to the given points [1], [6].

In this paper we present several algorithms which compute the ellipse for which the *sum of the squares of the distances to the given points is minimal*. These algorithms are compared with classical simple and iterative methods.

Circles and ellipses may be represented algebraically i.e. by an equation of the form $F(\mathbf{x}) = 0$. If a point is on the curve then its coordinates \mathbf{x} are a zero of the function F . Alternatively, curves may be represented in parametric form, which is well suited for minimizing the sum of the squares of the distances.

1 Preliminaries and Introduction

Ellipses, for which the *sum of the squares of the distances to the given points is minimal* will be referred to as “best fit” or “geometric fit”, and the algorithms will be called “geometric”.

Determining the parameters of the algebraic equation $F(\mathbf{x}) = 0$ in the least squares sense will be denoted by “algebraic fit” and the algorithms will be called “algebraic”.

We will use the well known Gauss-Newton method to solve the nonlinear least squares problem (cf. [15]). Let $\mathbf{u} = (u_1, \dots, u_n)^T$ be a vector of unknowns and consider the nonlinear system of m equations $\mathbf{f}(\mathbf{u}) = \mathbf{0}$.

If $m > n$, then we want to minimize

$$\sum_{i=1}^m f_i(\mathbf{u})^2 = \min.$$

*This paper appeared in similar form in BIT 34(1994), 558–578.

This is a *nonlinear least squares problem*, which we will solve iteratively by a sequence of linear least squares problems.

We approximate the solution $\hat{\mathbf{u}}$ by $\tilde{\mathbf{u}} + \mathbf{h}$. Developing

$$\mathbf{f}(\mathbf{u}) = (f_1(\mathbf{u}), f_2(\mathbf{u}), \dots, f_m(\mathbf{u}))^T$$

around $\tilde{\mathbf{u}}$ in a Taylor series, we obtain

$$(1.1) \quad \mathbf{f}(\tilde{\mathbf{u}} + \mathbf{h}) \simeq \mathbf{f}(\tilde{\mathbf{u}}) + J(\tilde{\mathbf{u}})\mathbf{h} \approx \mathbf{0},$$

where J is the Jacobian. We solve equation (1.1) as a linear least squares problem for the correction vector \mathbf{h} :

$$(1.2) \quad J(\tilde{\mathbf{u}})\mathbf{h} \approx -\mathbf{f}(\tilde{\mathbf{u}}).$$

An iteration then with the Gauss-Newton method consists of the two steps:

1. Solving equation (1.2) for \mathbf{h} .
2. Update the approximation $\tilde{\mathbf{u}} := \tilde{\mathbf{u}} + \mathbf{h}$.

We define the following notation: a given point P_i will have the coordinate vector $\mathbf{x}_i = (x_{i1}, x_{i2})^T$. The $m \times 2$ matrix $X = [\mathbf{x}_1, \dots, \mathbf{x}_m]^T$ will therefore contain the coordinates of a set of m points. The 2-norm $\|\cdot\|_2$ of vectors and matrices will simply be denoted by $\|\cdot\|$.

We are very pleased to dedicate this paper to professor Åke Björck who has contributed so much to our understanding of the numerical solution of least squares problems. Not only is he a scholar of great distinction, but he has always been generous in spirit and gentlemanly in behavior.

2 Circle: Minimizing the algebraic distance

Let us first consider an algebraic representation of the circle in the plane:

$$(2.1) \quad F(\mathbf{x}) = a\mathbf{x}^T\mathbf{x} + \mathbf{b}^T\mathbf{x} + c = 0,$$

where $a \neq 0$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^2$. To fit a circle, we need to compute the coefficients a , \mathbf{b} and c from the given data points.

If we insert the coordinates of the points into equation (2.1), we obtain a linear system of equations $B\mathbf{u} = \mathbf{0}$ for the coefficients $\mathbf{u} = (a, b_1, b_2, c)^T$, where

$$B = \begin{pmatrix} x_{11}^2 + x_{12}^2 & x_{11} & x_{12} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1}^2 + x_{m2}^2 & x_{m1} & x_{m2} & 1 \end{pmatrix}.$$

To obtain a non-trivial solution, we impose some constraint on \mathbf{u} , e.g. $u_1 = 1$ (commonly used) or $\|\mathbf{u}\| = 1$.

For $m > 3$, in general, we cannot expect the system to have a solution, unless all the points happen to be on a circle. Therefore, we solve the overdetermined system $B\mathbf{u} = \mathbf{r}$ where \mathbf{u} is chosen to minimize $\|\mathbf{r}\|$. We obtain a standard problem (c.f. [3]):

$$\|B\mathbf{u}\| = \min \quad \text{subject to} \quad \|\mathbf{u}\| = 1.$$

This problem is equivalent to finding the right singular vector associated with the smallest singular value of B . If $a \neq 0$, we can transform equation (2.1) to

$$(2.2) \quad \left(x_1 + \frac{b_1}{2a}\right)^2 + \left(x_2 + \frac{b_2}{2a}\right)^2 = \frac{\|\mathbf{b}\|^2}{4a^2} - \frac{c}{a},$$

from which we obtain the center and the radius, if the right hand side of (2.2) is positive:

$$\mathbf{z} = (z_1, z_2) = \left(-\frac{b_1}{2a}, -\frac{b_2}{2a}\right) \quad r = \sqrt{\frac{\|\mathbf{b}\|^2}{4a^2} - \frac{c}{a}}.$$

This approach has the advantage of being simple. The disadvantage is that we are uncertain what we are minimizing in a geometrical sense. For applications in coordinate metrology this kind of fit is often unsatisfactory. In such applications, one wishes to minimize the sum of the squares of the distances. Figure 2.1 shows two circles fitted to the set of points

$$(2.3) \quad \begin{array}{c|ccccc} x & 1 & 2 & 5 & 7 & 9 & 3 \\ \hline y & 7 & 6 & 8 & 7 & 5 & 7 \end{array}.$$

Minimizing the algebraic distance, we obtain the dashed circle with radius $r = 3.0370$ and center $\mathbf{z} = (5.3794, 7.2532)$.

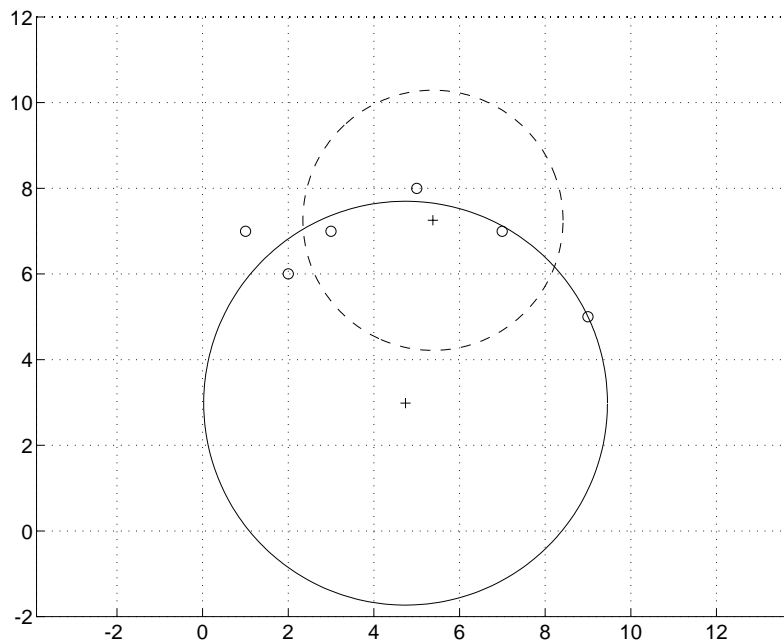


Figure 2.1: algebraic vs. best fit

———— Best fit
 - - - - Algebraic fit

The algebraic solution is often useful as a starting vector for methods minimizing the geometric distance.

3 Circle: Minimizing the geometric distance

To minimize the sum of the squares of the distances $d_i^2 = (\|\mathbf{z} - \mathbf{x}_i\| - r)^2$ we need to solve a nonlinear least squares problem. Let $\mathbf{u} = (z_1, z_2, r)^T$, we want to determine $\tilde{\mathbf{u}}$ so that

$$\sum_{i=1}^m d_i(\mathbf{u})^2 = \min.$$

The Jacobian defined by the partial derivatives $\partial d_i(\mathbf{u})/\partial u_j$ is given by:

$$J(\mathbf{u}) = \begin{pmatrix} \frac{u_1 - x_{11}}{\sqrt{(u_1 - x_{11})^2 + (u_2 - x_{12})^2}} & \frac{u_2 - x_{12}}{\sqrt{(u_1 - x_{11})^2 + (u_2 - x_{12})^2}} & -1 \\ \vdots & \vdots & \vdots \\ \frac{u_1 - x_{m1}}{\sqrt{(u_1 - x_{m1})^2 + (u_2 - x_{m2})^2}} & \frac{u_2 - x_{m2}}{\sqrt{(u_1 - x_{m1})^2 + (u_2 - x_{m2})^2}} & -1 \end{pmatrix}.$$

A good starting vector for the Gauss-Newton method may often be obtained by solving the linear problem as given in the previous paragraph. The algorithm then iteratively computes the “best” circle.

If we use the set of points (2.3) and start the iteration with the values obtained from the linear model (minimizing the algebraic distance), then after 11 Gauss-Newton steps the norm of the correction vector is $2.05E-6$. We obtain the best fit circle with center $\mathbf{z} = (4.7398, 2.9835)$ and radius $r = 4.7142$ (the solid circle in figure 2.1).

4 Circle: Geometric fit in parametric form

The parametric form commonly used for the circle is given by

$$(4.1) \quad x = z_1 + r \cos \varphi$$

$$(4.2) \quad y = z_2 + r \sin \varphi.$$

The distance d_i of a point $P_i = (x_{i1}, x_{i2})$ may be expressed by

$$d_i^2 = \min_{\varphi_i} [(x_{i1} - x(\varphi_i))^2 + (x_{i2} - y(\varphi_i))^2].$$

Now since we want to determine z_1, z_2 and r by minimizing

$$\sum_{i=1}^m d_i^2 = \min,$$

we can simultaneously minimize for z_1, z_2, r and $\{\varphi_i\}_{i=1\dots m}$; i.e. find the minimum of the quadratic function

$$Q(\varphi_1, \varphi_2, \dots, \varphi_m, z_1, z_2, r) = \sum_{i=1}^m [(x_{i1} - x(\varphi_i))^2 + (x_{i2} - y(\varphi_i))^2].$$

This is equivalent to solving the nonlinear least squares problem

$$\begin{aligned} z_1 + r \cos \varphi_i - x_{i1} &\approx 0 \\ z_2 + r \sin \varphi_i - x_{i2} &\approx 0 \quad \text{for } i = 1, 2, \dots, m. \end{aligned}$$

Let $\mathbf{u} = (\varphi_1, \dots, \varphi_m, z_1, z_2, r)$. The Jacobian associated with Q is

$$J = \begin{pmatrix} rS & A \\ -rC & B \end{pmatrix},$$

where $S = \text{diag}(\sin \varphi_i)$ and $C = \text{diag}(\cos \varphi_i)$ are $m \times m$ diagonal matrices. A and B are $m \times 3$ matrices defined by:

$$\begin{array}{llll} a_{i1} & = & -1 & a_{i2} & = & 0 & a_{i3} & = & -\cos \varphi_i \\ b_{i1} & = & 0 & b_{i2} & = & -1 & b_{i3} & = & -\sin \varphi_i. \end{array}$$

For large m , J is very sparse. We note that the first part $\begin{pmatrix} rS \\ -rC \end{pmatrix}$ is orthogonal. To compute the QR decomposition of J we use the orthonormal matrix

$$Q = \begin{pmatrix} S & C \\ -C & S \end{pmatrix}.$$

Multiplying from the left we get

$$Q^T J = \begin{pmatrix} rI & SA - CB \\ O & CA + SB \end{pmatrix}.$$

So to obtain the QR decomposition of the Jacobian, we only have to compute a QR decomposition of the $m \times 3$ sub-matrix $CA + SB = UP$. Then

$$\begin{pmatrix} I & 0 \\ O & U^T \end{pmatrix} Q^T J = \begin{pmatrix} rI & SA - CB \\ O & P \end{pmatrix},$$

and the solution is obtained by backsubstitution. In general we may obtain good starting values for z_1 , z_2 and r for the Gauss-Newton iteration, if we first solve the linear problem by minimizing the algebraic distance. If the center is known, initial approximations for $\{\varphi_k\}_{k=1\dots m}$ can be computed by

$$\varphi_k = \arg((x_{k1} - z_1) + i(x_{k2} - z_2)).$$

We use again the points (2.3) and start the iteration with the values obtained from the linear model (minimizing the algebraic distance). After 21 Gauss-Newton steps the norm of the correction is $3.43E-06$ and we obtain the same results as before: center $\mathbf{z} = (4.7398, 2.9835)$ and radius $r = 4.7142$ (the solid circle in figure 2.1).

5 Ellipse: Minimizing the algebraic distance

Given the quadratic equation

$$(5.1) \quad \mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c = 0$$

with A symmetric and positive definite, we can compute the geometric quantities of the conic as follows.

We introduce new coordinates $\bar{\mathbf{x}}$ with $\mathbf{x} = Q\bar{\mathbf{x}} + \mathbf{t}$, thus rotating and shifting the conic. Then equation (5.1) becomes

$$\bar{\mathbf{x}}^T (Q^T A Q) \bar{\mathbf{x}} + (2\mathbf{t}^T A + \mathbf{b}^T) Q \bar{\mathbf{x}} + \mathbf{t}^T A \mathbf{t} + \mathbf{b}^T \mathbf{t} + c = 0.$$

Defining $\bar{A} = Q^T A Q$, and similarly $\bar{\mathbf{b}}$ and \bar{c} , this equation may be written

$$\bar{\mathbf{x}}^T \bar{A} \bar{\mathbf{x}} + \bar{\mathbf{b}}^T \bar{\mathbf{x}} + \bar{c} = 0.$$

We may choose Q so that $\bar{A} = \text{diag}(\lambda_1, \lambda_2)$; if the conic is an ellipse or a hyperbola, we may further choose \mathbf{t} so that $\bar{\mathbf{b}} = \mathbf{0}$. Hence, the equation may be written

$$(5.2) \quad \lambda_1 \bar{x}_1^2 + \lambda_2 \bar{x}_2^2 + \bar{c} = 0,$$

and this defines an ellipse if $\lambda_1 > 0$, $\lambda_2 > 0$ and $\bar{c} < 0$. The center and the axes of the ellipse in the non-transformed system are given by

$$\begin{aligned} \mathbf{z} &= \mathbf{t} \\ a &= \sqrt{-\bar{c}/\lambda_1} \\ b &= \sqrt{-\bar{c}/\lambda_2}. \end{aligned}$$

Since $Q^T Q = I$, the matrices A and \bar{A} have the same (real) eigenvalues λ_1, λ_2 . It follows that each function of λ_1 and λ_2 is invariant under rotation and shifts. Note

$$\begin{aligned} \det A &= a_{11}a_{22} - a_{21}a_{12} = \lambda_1 \lambda_2 \\ \text{trace } A &= a_{11} + a_{22} = \lambda_1 + \lambda_2, \end{aligned}$$

which serve as a basis for all polynomials symmetric in λ_1, λ_2 . As a possible application of above observations, let us express the quotient $\kappa = a/b$ for the ellipse's axes a and b . With $a^2 = -\bar{c}/\lambda_1$ and $b^2 = -\bar{c}/\lambda_2$ we get

$$\kappa^2 + 1/\kappa^2 = \frac{\lambda_2}{\lambda_1} + \frac{\lambda_1}{\lambda_2} = \frac{\lambda_1^2 + \lambda_2^2}{\lambda_1 \lambda_2} = \frac{(\text{trace } A)^2 - 2 \det A}{\det A} = \frac{a_{11}^2 + a_{22}^2 + 2a_{12}^2}{a_{11}a_{22} - a_{12}^2}$$

and therefore

$$\kappa^2 = \mu \pm \sqrt{\mu^2 - 1}$$

where

$$\mu = \frac{(\text{trace } A)^2}{2 \det A} - 1.$$

To compute the coefficients \mathbf{u} from given points, we insert the coordinates into equation (5.1) and obtain a linear system of equations $B\mathbf{u} = \mathbf{0}$, which we may solve again as constrained least squares problem: $\|B\mathbf{u}\| = \min$ subject to $\|\mathbf{u}\| = 1$.

The disadvantage of the constraint $\|\mathbf{u}\| = 1$ is its non-invariance for Euclidean coordinate transformations

$$\mathbf{x} = Q\bar{\mathbf{x}} + \mathbf{t}, \quad \text{where } Q^T Q = I.$$

For this reason Bookstein [9] recommended solving the constrained least squares problem

$$(5.3) \quad \mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c \approx 0$$

$$(5.4) \quad \lambda_1^2 + \lambda_2^2 = a_{11}^2 + 2a_{12}^2 + a_{22}^2 = 1.$$

While [9] describes a solution based on eigenvalue decomposition, we may solve the same problem more efficiently and accurately with a singular value decomposition

as described in [12]. In the simple algebraic solution by SVD, we solve the system for the parameter vector

$$(5.5) \quad \mathbf{u} = (a_{11}, 2a_{12}, a_{22}, b_1, b_2, c)^T$$

with the constraint $\|\mathbf{u}\| = 1$, which is not invariant under Euclidean transformations. If we define vectors

$$\begin{aligned} \mathbf{v} &= (b_1, b_2, c)^T \\ \mathbf{w} &= (a_{11}, \sqrt{2} a_{12}, a_{22})^T \end{aligned}$$

and the coefficient matrix

$$S = \begin{pmatrix} x_{11} & x_{12} & 1 & x_{11}^2 & \sqrt{2} x_{11} x_{12} & x_{12}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & 1 & x_{m1}^2 & \sqrt{2} x_{m1} x_{m2} & x_{m2}^2 \end{pmatrix},$$

then the Bookstein constraint (5.4) may be written $\|\mathbf{w}\| = 1$, and we have the reordered system

$$S \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} \approx \mathbf{0}.$$

The QR decomposition of S leads to the equivalent system

$$\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} \approx \mathbf{0},$$

which may be solved in following steps:

$$\begin{aligned} R_{22} \mathbf{w} &\approx \mathbf{0} \\ \|\mathbf{w}\| &= 1. \end{aligned}$$

Using the singular value decomposition of $R_{22} = U\Sigma V^T$, finding $\mathbf{w} = \mathbf{v}_3$, and then

$$\mathbf{v} = -R_{11}^{-1} R_{12} \mathbf{w}.$$

Note that the problem

$$(S_1 \ S_2) \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} \approx \mathbf{0} \quad \text{where } \|\mathbf{w}\| = 1$$

is equivalent to the generalized total least squares problem finding a matrix \hat{S}_2 such that

$$\begin{aligned} \text{rank}(S_1 \ \hat{S}_2) &\leq 5 \\ \|(S_1 \ \hat{S}_2) - (S_1 \ S_2)\| &= \inf_{\text{rank}(S_1 \ \bar{S}_2) \leq 5} \|(S_1 \ \bar{S}_2) - (S_1 \ S_2)\|. \end{aligned}$$

In other words, find a best rank 5 approximation to S that leaves S_1 fixed. A description of this problem may be found in [16].

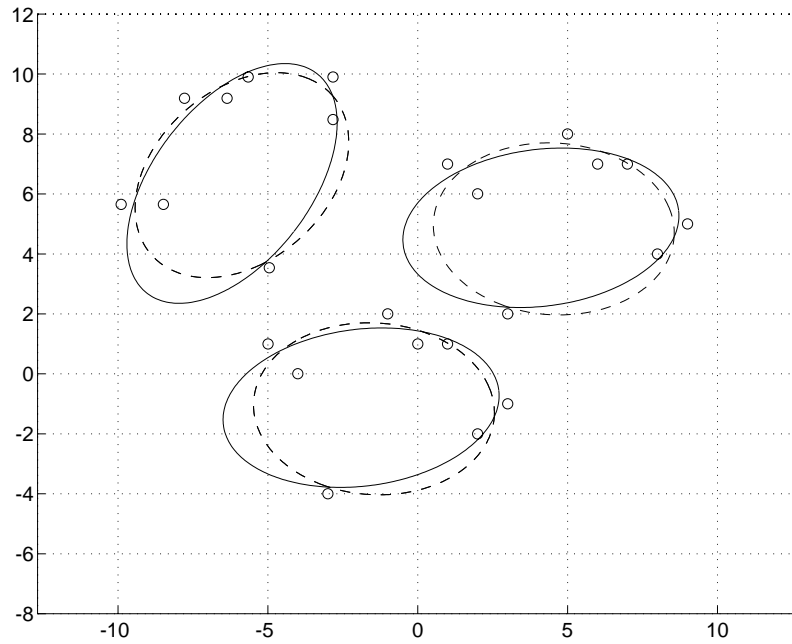


Figure 5.1: Euclidean-invariant algorithms

$$\begin{array}{ll} \text{—————} & \text{Constraint } \lambda_1^2 + \lambda_2^2 = 1 \\ \text{-----} & \text{Constraint } \lambda_1 + \lambda_2 = 1 \end{array}$$

To demonstrate the influence of different coordinate systems, we have computed the ellipse fit for this set of points:

$$(5.6) \quad \begin{array}{c|cccccccc} x & 1 & 2 & 5 & 7 & 9 & 6 & 3 & 8 \\ \hline y & 7 & 6 & 8 & 7 & 5 & 7 & 2 & 4 \end{array},$$

which are first shifted by $(-6, -6)$, and then by $(-4, 4)$ and rotated by $\pi/4$. See figures 5.1–5.2 for the fitted ellipses.

Since $\lambda_1^2 + \lambda_2^2 \neq 0$ for ellipses, hyperbolas and parabolas, the Bookstein constraint is appropriate to fit any of these. But all we need is an invariant $I \neq 0$ for ellipses—and one of them is $\lambda_1 + \lambda_2$. Thus we may invariantly fit an ellipse with the constraint

$$\lambda_1 + \lambda_2 = a_{11} + a_{22} = 1,$$

which results in the linear least squares problem

$$\begin{pmatrix} 2x_{11}x_{12} & x_{22}^2 - x_{11}^2 & x_{11} & x_{12} & 1 \\ \vdots & \vdots & & & \vdots \\ 2x_{m1}x_{m2} & x_{m2}^2 - x_{m1}^2 & x_{m1} & x_{m2} & 1 \end{pmatrix} \mathbf{v} \approx \begin{pmatrix} -x_{11}^2 \\ \vdots \\ -x_{m1}^2 \end{pmatrix}.$$

See the dashed ellipses in figure 5.1.

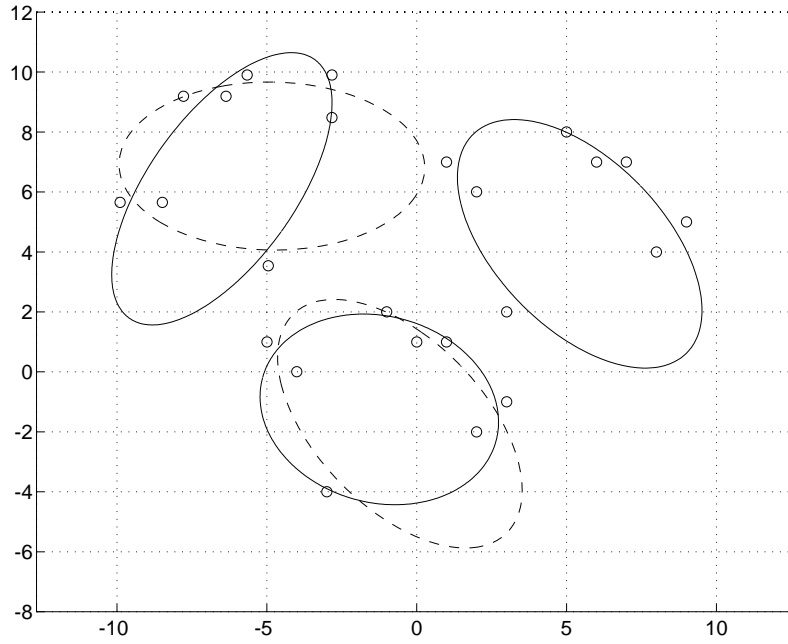


Figure 5.2: Non-invariant algebraic algorithm

————— Fitted ellipses
 - - - - - Originally fitted ellipse after transformation

6 Ellipse: Geometric fit in parametric form

In order to fit an ellipse in parametric form, we consider the equations

$$\mathbf{x} = \mathbf{z} + Q(\alpha)\mathbf{x}', \quad \mathbf{x}' = \begin{pmatrix} a \cos \varphi \\ b \sin \varphi \end{pmatrix}, \quad Q(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}.$$

Minimizing the sum of squares of the distances of the given points to the “best” ellipse is equivalent to solving the nonlinear least squares problem:

$$\mathbf{g}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \end{pmatrix} - \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - Q(\alpha) \begin{pmatrix} a \cos \varphi_i \\ b \sin \varphi_i \end{pmatrix} \approx 0, \quad i = 1, \dots, m.$$

Thus we have $2m$ nonlinear equations for $m + 5$ unknowns: $\varphi_1, \dots, \varphi_m, \alpha, a, b, z_1, z_2$. To compute the Jacobian we need the partial derivatives:

$$\begin{aligned} \frac{\partial \mathbf{g}_i}{\partial \varphi_j} &= -\delta_{ij} Q(\alpha) \begin{pmatrix} -a \sin \varphi_i \\ b \cos \varphi_i \end{pmatrix} \\ \frac{\partial \mathbf{g}_i}{\partial \alpha} &= -\dot{Q}(\alpha) \begin{pmatrix} a \cos \varphi_i \\ b \sin \varphi_i \end{pmatrix} \\ \frac{\partial \mathbf{g}_i}{\partial a} &= -Q(\alpha) \begin{pmatrix} \cos \varphi_i \\ 0 \end{pmatrix} \\ \frac{\partial \mathbf{g}_i}{\partial b} &= -Q(\alpha) \begin{pmatrix} 0 \\ \sin \varphi_i \end{pmatrix} \end{aligned}$$

$$\begin{aligned}\frac{\partial \mathbf{g}_i}{\partial z_1} &= \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\ \frac{\partial \mathbf{g}_i}{\partial z_2} &= \begin{pmatrix} 0 \\ -1 \end{pmatrix}\end{aligned}$$

where we have used the notation

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

Thus the Jacobian becomes:

$$J = \begin{pmatrix} -Q \begin{pmatrix} -as_1 \\ bc_1 \end{pmatrix} & -\dot{Q} \begin{pmatrix} ac_1 \\ bs_1 \end{pmatrix} & -Q \begin{pmatrix} c_1 \\ 0 \end{pmatrix} & -Q \begin{pmatrix} 0 \\ s_1 \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} \\ & \ddots & \vdots & \vdots & \vdots & \vdots \\ & -Q \begin{pmatrix} -as_m \\ bc_m \end{pmatrix} & -\dot{Q} \begin{pmatrix} ac_m \\ bs_m \end{pmatrix} & -Q \begin{pmatrix} c_m \\ 0 \end{pmatrix} & -Q \begin{pmatrix} 0 \\ s_m \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} \end{pmatrix},$$

where we have used as abbreviation $s_i = \sin \varphi_i$ and $c_i = \cos \varphi_i$. Note that

$$\dot{Q}(\alpha) = \begin{pmatrix} -\sin \alpha & -\cos \alpha \\ \cos \alpha & -\sin \alpha \end{pmatrix} \quad \text{and therefore} \quad Q^T \dot{Q} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Since Q is orthogonal, the $2m \times 2m$ block diagonal matrix $U = -\text{diag}(Q, \dots, Q)$ is orthogonal, too, and

$$U^T J = \begin{pmatrix} \begin{pmatrix} -as_1 \\ bc_1 \end{pmatrix} & \begin{pmatrix} -bs_1 \\ ac_1 \end{pmatrix} & \begin{pmatrix} c_1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ s_1 \end{pmatrix} & \begin{pmatrix} c \\ -s \end{pmatrix} & \begin{pmatrix} s \\ c \end{pmatrix} \\ & \ddots & \vdots & \vdots & \vdots & \vdots \\ & \begin{pmatrix} -as_m \\ bc_m \end{pmatrix} & \begin{pmatrix} -bs_m \\ ac_m \end{pmatrix} & \begin{pmatrix} c_m \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ s_m \end{pmatrix} & \begin{pmatrix} c \\ -s \end{pmatrix} \end{pmatrix},$$

where $s = \sin \alpha$ and $c = \cos \alpha$. If we permute the equations, we obtain a similar structure for the Jacobian as in the circle fit:

$$\bar{J} = \begin{pmatrix} -aS & A \\ bC & B \end{pmatrix}.$$

That is, $S = \text{diag}(\sin \varphi_i)$ and $C = \text{diag}(\cos \varphi_i)$ are two $m \times m$ diagonal matrices and A and B are $m \times 5$ and are defined by:

$$\begin{aligned}A(i, 1:5) &= \begin{bmatrix} -b \sin \varphi_i & \cos \varphi_i & 0 & \cos \alpha & \sin \alpha \end{bmatrix} \\ B(i, 1:5) &= \begin{bmatrix} a \cos \varphi_i & 0 & \sin \varphi_i & -\sin \alpha & \cos \alpha \end{bmatrix}.\end{aligned}$$

We cannot give an explicit expression for an orthogonal matrix to triangularize the first m columns of J in a similar way as we did in fitting a circle. However, we can use Givens rotations to do this in m steps.

Figure 6.1 shows two ellipses fitted to the points given by

$$(6.1) \quad \begin{array}{c|cccccccc} x & 1 & 2 & 5 & 7 & 9 & 3 & 6 & 8 \\ \hline y & 7 & 6 & 8 & 7 & 5 & 7 & 2 & 4 \end{array}.$$

By minimizing the algebraic distance with $\|\mathbf{u}\| = 1$ we obtain the large cigar shaped dashed ellipse with $\mathbf{z} = (13.8251, -2.1099)$, $a = 29.6437$, $b = 1.8806$ and residual norm $r = 1.80$. If we minimize the sum of squares of the distances then we obtain the solid ellipse with $\mathbf{z} = (2.6996, 3.8160)$, $a = 6.5187$, $b = 3.0319$ and $r = 1.17$. In order to obtain starting values for the nonlinear least squares problem we used the center, obtained by fitting the best circle. We cannot use the approximation $b_0 = a_0 = r$, since the Jacobian becomes singular for $b = a$! Therefore, we used $b_0 = r/2$ as a starting value. With $\alpha_0 = 0$, we needed 71 iteration steps to compute the “best ellipse” shown in Figure 6.1.

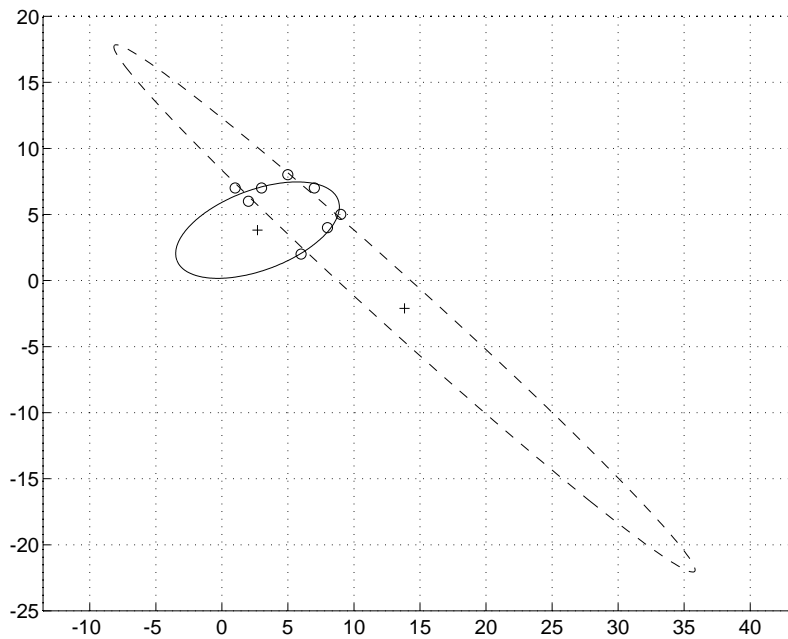


Figure 6.1: algebraic versus best fit

————— Best fit
 - - - - - Algebraic fit ($\|\mathbf{u}\| = 1$)

7 Ellipse: Iterative algebraic solutions

In this section, we will present modifications to the algebraic fit of ellipses. The algebraic equations may be weighted depending on a given estimation—thus leading to a simple iterative mechanism. Most algorithms try to weight the points such that the algebraic solution comes closer to the geometric solution. Another idea is to favor non-eccentric ellipses.

7.1 Curvature weights

The solution of

$$\mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c \approx 0$$

in the least squares sense leads to an equation for each point. If the equation for point (x_{i1}, x_{i2}) is multiplied by $\omega_i > 1$, the solution will approximate this point more accurately. In [6], ω_i is set to $1/R_i$, where R_i is the curvature radius of the ellipse at a point \mathbf{p}_i associated with (x_{i1}, x_{i2}) . The point \mathbf{p}_i is determined by intersecting the ray from the ellipse's center to (x_{i1}, x_{i2}) and the ellipse.

Tests on few data sets show, that this weighting scheme leads to better shaped ellipses in some cases, especially for eccentric ellipses; but it does not systematically restrict the solutions to ellipses. Lets look at the curvature weight solution for two problems. Figure 7.1 shows the result for the data set (6.1) presented earlier: unluckily, the algorithm finds a hyperbola for the weighted equations in the first step. On the other side, the algorithm is successful indeed for a data set close to an

eccentric ellipse. Figure 7.2 shows the large solid ellipse (residual norm 2.20) found by the curvature weights algorithm. The small dotted ellipse is the solution of the unweighted algebraic solution (6.77); the dashed ellipse is the best fit solution using Gauss-Newton (1.66), and the dash-dotted ellipse (1.69) is found by the geometric-weight algorithm described later.

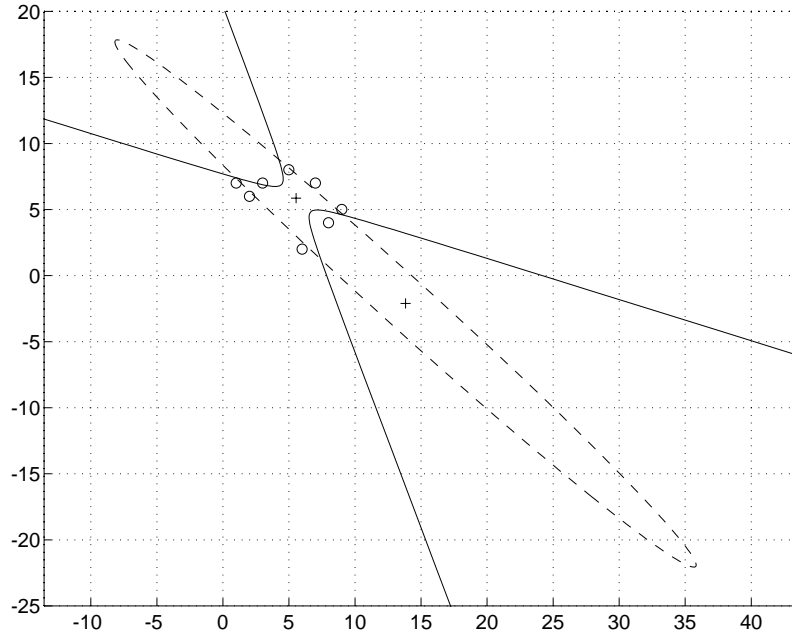


Figure 7.1: algebraic fit with curvature weights

———— Conic after first curvature-weight step
 - - - - Unweighted algebraic fit

7.2 Geometric distance weighting

We are interested in weighting schemes which result in a least square solution for the geometric distance. If we define

$$Q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c ,$$

then the simple algebraic method minimizes Q for the given points in the least squares sense. Q has the following geometric meaning: Let $h(\mathbf{x})$ be the geometric distance from the center to point \mathbf{x}

$$h(\mathbf{x}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2}$$

and determine \mathbf{p}_i by intersecting the ray from the ellipse's center to \mathbf{x}_i and the ellipse. Then, as pointed out in [9]

$$(7.1) \quad Q(\mathbf{x}_i) = \kappa((h(\mathbf{x}_i)/h(\mathbf{p}_i))^2 - 1)$$

$$(7.2) \quad \simeq 2\kappa \frac{h(\mathbf{x}_i) - h(\mathbf{p}_i)}{h(\mathbf{p}_i)} , \quad \text{if } \mathbf{x}_i \simeq \mathbf{p}_i$$

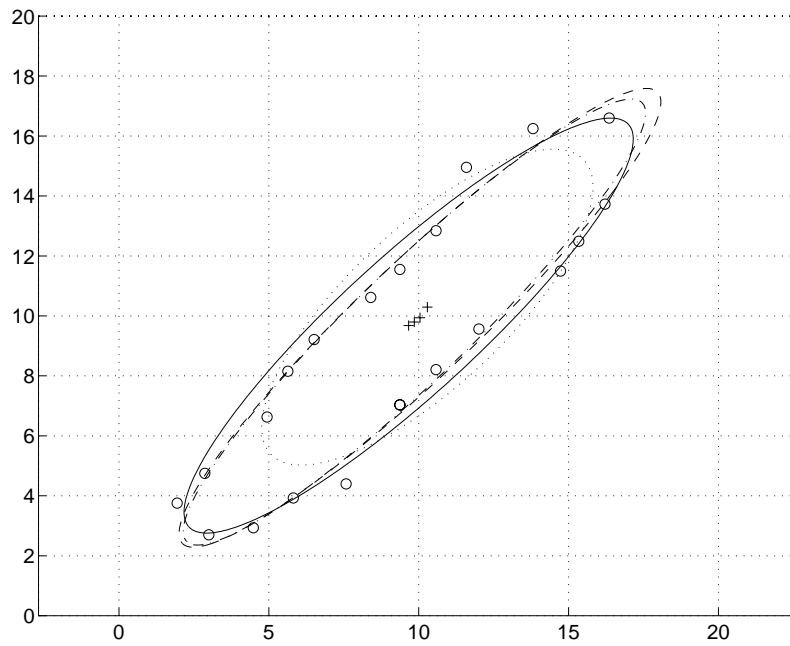
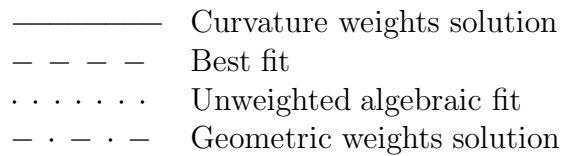


Figure 7.2: comparison of different fits



for some constant κ . This explains why the simple algebraic solution tends to neglect points far from the center.

Thus, we may say that the algebraic solution fits the ellipse with respect to the relative distances, i.e. a distant point has not the same importance as a near point. If we prefer to minimize the absolute distances, we may solve a weighted problem with weights

$$\omega_i = h(\mathbf{p}_i)$$

for a given estimated ellipse. The resulting estimated ellipse may then be used to determine new weights ω_i , thus iteratively solving weighted least squares problems.

Consequently, we may go a step further and set weights so that the equations are solved in the least squares sense for the geometric distances. If $d(\mathbf{x})$ is the geometric distance of \mathbf{x} from the currently estimated ellipse, then weights are set

$$\omega_i = d(\mathbf{x}_i)/Q(\mathbf{x}_i).$$

See the dash-dotted ellipse in figure 7.2 for an example.

The advantage of this method compared to the non-linear method to compute the geometric fit is, that no derivatives for the Jacobian or Hessian matrices are needed. The disadvantage of this method is, that it does not generally minimize the geometric distance. To show this, let us restate the problem:

$$(7.3) \quad \|G(\mathbf{x})\mathbf{x}\|^2 = \min \quad \text{where } \|\mathbf{x}\| = 1.$$

An iterative algorithm determines a sequence (\mathbf{y}_i) , where \mathbf{y}_{k+1} is the solution of

$$(7.4) \quad \|G(\mathbf{y}_k) \mathbf{y}\|^2 = \min \quad \text{where } \|\mathbf{y}\| = 1.$$

The sequence (\mathbf{y}_i) may have a fixed point $\tilde{\mathbf{y}} = \mathbf{y}_\infty$ without solving (7.3), since the conditions for critical points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ are different for the two equations. To show this, we shall use the notation $d\mathbf{z}$ for an infinitesimal change of \mathbf{z} . For all $d\mathbf{x}$ with $\tilde{\mathbf{x}}^T d\mathbf{x} = 0$ the following holds

$$2(\tilde{\mathbf{x}}^T G^T dG\tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T G^T G d\mathbf{x}) = d\|G\tilde{\mathbf{x}}\|^2 = 0$$

for equation (7.3). Whereas for equation (7.4) and $\tilde{\mathbf{y}}^T d\mathbf{y} = 0$ the condition is

$$2(\tilde{\mathbf{y}}^T G^T G d\mathbf{y}) = d\|G\tilde{\mathbf{y}}\|^2 = 0.$$

This problem is common to all iterative algebraic solutions of this kind, so no matter how good the weights approximate the real geometric distances, we may not generally expect that the sequence of estimated ellipses converges to the optimal solution.

We give a simple example for a fixed point of the iteration scheme (7.4), which does not solve (7.3). For $\mathbf{z} = (x, y)^T$ consider

$$G = \begin{pmatrix} 2 & 0 \\ -4y & 4 \end{pmatrix};$$

then $\mathbf{z}_0 = (1, 0)^T$ is a fixed point of (7.4), but $\mathbf{z} = (0.7278, 0.6858)^T$ is the solution of (7.3).

Another severe problem with iterative algebraic methods is the lack of convergence in the general case—especially if the problem is ill-conditioned. We will shortly examine the solution of (7.4) for small changes to G . Let

$$\begin{aligned} G &= U\Sigma V^T \\ \bar{G} &= G + dG = \bar{U}\bar{\Sigma}\bar{V}^T \end{aligned}$$

and denote with $\sigma_1, \dots, \sigma_n$ the singular values in descending order, with \mathbf{v}_i the associated right singular vectors—where \mathbf{v}_n is the solution of equation (7.4) for G . Then we may bound $\|d\mathbf{v}_n\| = \|\bar{\mathbf{v}}_n - \mathbf{v}_n\|$ as follows. First, we define

$$\begin{aligned} \lambda &= \bar{V}^T \mathbf{v}_n \quad \text{thus } \|\lambda\| = 1 \\ \mu &= 1 - \lambda_n^2 \\ \varepsilon &= \|dG\| \end{aligned}$$

and note that

$$\sigma_i - \varepsilon \leq \bar{\sigma}_i \leq \sigma_i + \varepsilon \quad \text{for all } i.$$

We may conclude

$$\|\bar{\Sigma}\lambda\| = \|\bar{U}\bar{\Sigma}\lambda\| = \|\bar{G}\mathbf{v}_n\|$$

and

$$\|\bar{G}\mathbf{v}_n\| \leq \|G\mathbf{v}_n\| + \|dG\mathbf{v}_n\| \leq \sigma_n + \varepsilon,$$

thus

$$\sum_{i=1}^n \lambda_i^2 \bar{\sigma}_i^2 \leq \sigma_n^2 + 2\varepsilon\sigma_n + \varepsilon^2.$$

Using that $\bar{\sigma}_i \geq \bar{\sigma}_{n-1}$ for $i \leq n-1$, and that $\|\lambda\| = 1$, we simplify the above expression to

$$(7.5) \quad (1 - \lambda_n^2)\bar{\sigma}_{n-1}^2 + \lambda_n^2\bar{\sigma}_n^2 \leq \sigma_n^2 + 2\varepsilon\sigma_n + \varepsilon^2.$$

Assuming that $\sigma_n \neq 0$ (otherwise the solution is exact) and $\varepsilon \leq \sigma_n$, we have

$$\sigma_i^2 - 2\varepsilon\sigma_i + \varepsilon^2 \leq \bar{\sigma}_i^2.$$

Applying this to inequality (7.5), we get

$$\mu \leq \frac{4\varepsilon\sigma_n}{\bar{\sigma}_{n-1}^2 - \bar{\sigma}_n^2}.$$

Note that

$$\|\mathbf{v}_n - \bar{\mathbf{v}}_n\|^2 = \|\lambda - (0, \dots, 1)^T\|^2 = (\lambda_n - 1)^2 + (1 - \lambda_n^2) = 2(1 - \lambda_n).$$

Assuming that $\|\mathbf{v}_n - \bar{\mathbf{v}}_n\|$ is small (and thus $\lambda_n \approx 1$, $\mu \approx 0$), then we may write σ for $\bar{\sigma}$ and $\mu = (1 + \lambda_n)(1 - \lambda_n) \approx 2(1 - \lambda_n)$; thus we finally get

$$\|\mathbf{v}_n - \bar{\mathbf{v}}_n\| \leq \sqrt{\frac{4\varepsilon\sigma_n}{\sigma_{n-1}^2 - \sigma_n^2}} \leq \sqrt{\frac{2\varepsilon}{\sigma_{n-1} - \sigma_n}}.$$

This shows that the convergence behavior of iterative algebraic methods depends on how well—in a figurative interpretation—the solution vector $\tilde{\mathbf{v}}_n$ is separated from its hyper-plane with respect to the residual norm $\|G\mathbf{v}\|$. If $\tilde{\sigma}_{n-1} \approx \tilde{\sigma}_n$, the solution is poorly determined, and the algorithm may not converge.

While the analytical results are not encouraging, we obtained solutions close to the optimum using the geometric-weight algorithm for several examples. Figure 7.3 shows the geometric-weight (solid) and the best (dashed) solution for such an example. Note that the calculation of geometric distances is relatively expensive, so a pragmatic way to limit the cost is to perform a fixed number of iterations, since convergence is not guaranteed anyway.

x_i	y_i
2.0143	10.5575
17.3465	3.2690
-8.5257	-7.2959
-7.9109	-7.6447
16.3705	-3.8815
-15.3434	5.0513
-21.5840	-0.6013
9.4111	-9.0697

Table 7.1: Given points close to ellipse

Table 7.1 lists the points to be approximated, resulting in a residual norm of 2.784, compared to 2.766 for the best geometric fit. Since the algebraic method minimizes $w_i Q_i$ in the least squares sense, it remains to check that $w_i Q_i$ is proportional to the geometric distance d_i for the estimated conic. We compute

$$(7.6) \quad M_i = |(w_i Q_i)/d_i|$$

$$(7.7) \quad h_i = 1 - M_i/\|M\|_\infty$$

and find—as expected—that $\|\mathbf{h}\| = 1.1E-5$.

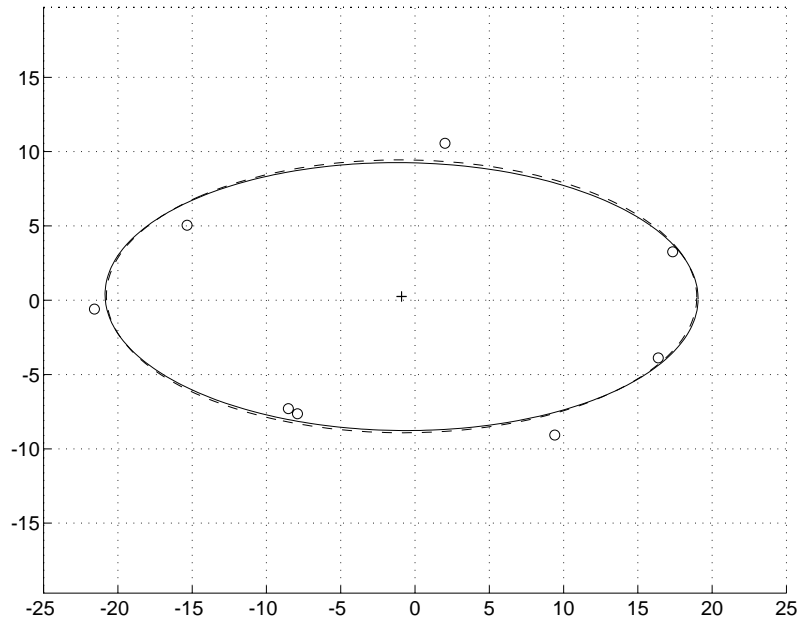


Figure 7.3: geometric weight fit vs. best fit

————— Geometric weight solution
 - - - - - Best fit

7.3 Circle weight algorithm

The main difficulty with above algebraic methods is that the solution may be any conic—not necessarily an ellipse. To cope with this case, we extend the system by weighted equations, which favour circles and non-eccentric ellipses:

$$(7.8) \quad \omega (a_{11} - a_{22}) \approx 0$$

$$(7.9) \quad \omega 2a_{12} \approx 0.$$

Note that these equations are Euclidean-invariant only if ω is the same in both equations, and the problem is solved in the least squares sense. What we are really minimizing in this case is

$$\omega^2((a_{11} - a_{22})^2 + 4a_{12}^2) = \omega^2(\lambda_1 - \lambda_2)^2;$$

hence, the constraints (7.8) and (7.9) are equivalent to the equation

$$\omega (\lambda_1 - \lambda_2) \approx 0.$$

The weight ω is fixed by

$$(7.10) \quad \omega = \epsilon f((\lambda_1 - \lambda_2)^2),$$

where ϵ is a parameter representing the badness of eccentric ellipses, and

$$f: [0, \infty[\rightarrow [0, \infty[\text{ continuous, strictly increasing.}$$

The larger ϵ is chosen, the larger will ω be, and thus the more important are equations (7.8–7.9), which make the solution be more circle-shaped. The parameter ω is determined iteratively (starting with $\omega_0 = 0$), where following conditions hold

$$(7.11) \quad 0 = \omega_0 \leq \omega_2 \leq \dots \leq \omega \leq \dots \leq \omega_3 \leq \omega_1.$$

Thus, the larger the weight ω , the less eccentric the ellipse; we prove this in the following. Given weights $\chi < \omega$ and

$$F = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

then we find solutions \mathbf{x} and \mathbf{z} for the equations

$$\begin{aligned} \left\| \begin{pmatrix} B \\ \chi F \end{pmatrix} \mathbf{x} \right\| &= \min \\ \left\| \begin{pmatrix} B \\ \omega F \end{pmatrix} \mathbf{z} \right\| &= \min \end{aligned}$$

respectively. It follows that

$$\begin{aligned} \|B\mathbf{x}\|^2 + \chi^2 \|F\mathbf{x}\|^2 &\leq \|B\mathbf{z}\|^2 + \chi^2 \|F\mathbf{z}\|^2 \\ \|B\mathbf{z}\|^2 + \omega^2 \|F\mathbf{z}\|^2 &\leq \|B\mathbf{x}\|^2 + \omega^2 \|F\mathbf{x}\|^2 \end{aligned}$$

and by adding

$$\chi^2 \|F\mathbf{x}\|^2 + \omega^2 \|F\mathbf{z}\|^2 \leq \chi^2 \|F\mathbf{z}\|^2 + \omega^2 \|F\mathbf{x}\|^2$$

and since $\omega^2 - \chi^2 > 0$

$$\|F\mathbf{z}\|^2 \leq \|F\mathbf{x}\|^2,$$

so that

$$\bar{\omega} = \varepsilon f(\|F\mathbf{z}\|^2) \leq \varepsilon f(\|F\mathbf{x}\|^2) = \bar{\chi}.$$

This completes the proof, since f was chosen strictly increasing. One obvious choice for f is the identity function, which was used in our test programs.

8 Comparison of geometric algorithms

The discussion of algorithms minimizing the geometric distance is somewhat different from the algebraic distance problems. The problems in the latter are primarily stability and “good-looking” solutions; the former must be viewed by their efficiency, too. Generally, the simple algebraic solution is orders of magnitude cheaper than the geometric counterparts (for accurate results about a factor 10–100); thus iterative algebraic methods are a valuable alternative. But a comparison between algebraic and geometric algorithms would not be very enlightening—because there is no objective criterion to decide which estimate is better. However, we may compare different nonlinear least squares algorithms to compute the geometric fit with respect to stability and efficiency.

8.1 Algorithms

Several known nonlinear least squares algorithms have been implemented:

1. Gauss-Newton (**gauss**)
2. Newton (**newton**)
3. Gauss-Newton with Marquardt modification (**marq**)
4. Variable projection (**varpro**)
5. Orthogonal distance regression (**odr**)

The **odr** algorithm (see [13], [14]) solves the implicit minimization problem

$$\begin{aligned} f(\mathbf{x}_i + \delta_i, \beta) &= 0 \\ \sum_i \|\delta_i\|^2 &= \min \end{aligned}$$

where

$$f(\mathbf{x}, \beta) = \beta_3(x_1 - \beta_1)^2 + 2\beta_4(x_1 - \beta_1)(x_2 - \beta_2) + \beta_5(x_2 - \beta_2)^2 - 1.$$

Whereas the **gauss**, **newton**, **marq** and **varpro** algorithms solve the problem

$$Q(\mathbf{x}, \varphi_1, \varphi_2, \dots, \varphi_m, z_1, z_2, r) = \sum_{i=1}^m \left[(x_{i1} - x(\varphi_i))^2 + (x_{i2} - y(\varphi_i))^2 \right] = \min.$$

For $a \approx b$, the Jacobian matrix is nearly singular, so the **gauss** and **newton** algorithms are modified to apply Marquardt steps in this case. Not surprising, this modification makes all algorithms behave similar with respect to stability if the initial parameters are accurate and the problem is well posed.

8.2 Results

To appreciate the results given in table 8.1, it must be said that the **varpro** algorithm is written for any separable functional and cannot take profit from the sparsity of the Jacobian. The algorithms were tested with example data—each consisting of 8 points—for following problems

1. Special set of points (6.1)
2. Uniformly distributed data in a square
3. Points on a circle
4. Points on an ellipse with $a/b = 2$
5. Points on hyperbola branch

The tests with points on a conic were done both with and without perturbations. For table 8.1, the initial parameters were derived from the algebraically best fitting circle (radius r_a , center z_a); initial center $z_0 = z_a$, axes $a_0 = r_a$, $b_0 = r_a$ and $\alpha_0 = 0$. Note that these initial values are somewhat rudimentary, so they serve to check the algorithms' stability, too. Table 8.1 shows the number of flops (in 1000) for the respective algorithm and problem; the smallest number is underlined. If the algorithm didn't terminate after 100 steps, it was assumed non-convergent and a

	gauss	newton	marq	varpro	odr
Special	146	<u>85</u>	468	1146	◇
Random	◇	◇	◇	<u>2427</u>	◇
Circle	22	22	22	36	<u>7</u>
Circle+	86	<u>67</u>	189	717	69
Ellipse	<u>30</u>	37	67	143	41
Ellipse+	186	◇	633	1977	<u>103</u>
Hyperbola	22	22	22	36	<u>10</u>
Hyperbola+	◇	◇	◇	◇	◇

Table 8.1: Geometric fit with initial parameters of algebraic circle

#flops/1000, minimum is underlined
 ‘◇’ if non-convergence

	gauss	newton	marq	varpro	odr
Special	<u>165</u>	896	566	1506	◇
Random	◇	◇	◇	<u>2819</u>	◇
Circle	32	32	32	102	<u>7</u>
Circle+	76	<u>63</u>	145	574	66
Ellipse	22	22	22	112	<u>7</u>
Ellipse+	161	<u>40</u>	435	1870	74
Hyperbola	◇	◇	◇	<u>1747</u>	◇
Hyperbola+	◇	◇	◇	<u>2986</u>	◇

Table 8.2: Geometric fit with initial parameters of algebraic ellipse

#flops/1000, minimum is underlined
 ‘◇’ if non-convergence

‘ \diamond ’ is shown instead. Table 8.2 contains the results if the initial parameters were obtained from the Bookstein algorithm.

Table 8.2 shows that all algorithms converge quickly with the more accurate initial data for exact conics. For the perturbed ellipse data, it’s primarily the **newton** algorithm which profits from the starting values close to the solution. Note further that the **newton** algorithm does not find the correct solution for the special data, since the algebraic estimation—which serves as initial approximation—is completely different from the geometric solution.

General conclusions from this (admittedly) small test series are

- All algorithms are prohibitively expensive compared to the simple algebraic solution (factor 10–100).
- If the problem is well posed, and the accuracy of the result should be high, the **newton** method applied to the parameterized algorithm is the most efficient.
- The **odr** algorithm—although a simple general-purpose optimizing scheme—is competitive with algorithms specifically written for the ellipse fitting problem. If one takes into consideration further, that we didn’t use a highly optimized **odr** procedure, the method of solution is surprisingly simple and efficient.
- The **varpro** algorithm seems to be the most expensive. Reasons for its inefficiency are that most parameters are non-linear and that the algorithm does not make use of the special matrix structure for this problem.

The programs on which these experiments have been based are given in [17]. The report [17] and the MATLAB sources for the examples are available via anonymous ftp from `ftp.inf.ethz.ch`.

References

- [1] VAUGHAN PRATT, *Direct Least Squares Fitting of Algebraic Surfaces*, ACM J. Computer Graphics, Volume 21, Number 4, July 1987
- [2] M. G. COX, A. B. FORBES, *Strategies for Testing Assessment Software*, NPL Report DITC 211/92
- [3] G. GOLUB, CH. VAN LOAN, *Matrix Computations (second ed.)*, The Johns Hopkins University Press, Baltimore, 1989
- [4] D. SOURLIER, A. BUCHER, *Normgerechter Best-fit-Algorithmus für Freiform-Flächen oder andere nicht-reguläre Ausgleichs-Flächen in Parameter-Form*, Technisches Messen 59, (1992), pp. 293–302
- [5] H. SPÄTH, *Orthogonal Least Squares Fitting with Linear Manifolds*, Numer. Math., 48:441–445, 1986.
- [6] LYLE B. SMITH, *The use of man-machine interaction in data fitting problems*, TR No. CS 131, Stanford University, March 1969 (thesis)
- [7] Y. VLADIMIROVICH LINNIK, *Method of least squares and principles of the theory of observations*, New York, Pergamon Press, 1961
- [8] CHARLES L. LAWSON, *Contributions to the Theory of Linear Least Maximum Approximation*, Dissertation University of California, Los Angeles, 1961
- [9] FRED L. BOOKSTEIN, *Fitting Conic Sections to Scattered Data*, Computer Graphics and Image Processing 9, (1979), pp. 56–71
- [10] G. H. GOLUB, V. PEREYRA, *The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems whose Variables Separate*, SIAM J. Numer. Anal. 10, No. 2, (1973); pp. 413–432
- [11] M. HEIDARI, P. C. HEIGOLD, *Determination of Hydraulic Conductivity Tensor Using a Nonlinear Least Squares Estimator*, Water Resources Bulletin, June 1993, pp. 415–424
- [12] W. GANDER, U. VON MATT, *Some Least Squares Problems*, Solving Problems in Scientific Computing Using Maple and Matlab, W. Gander and J. Hřebíček ed., 251–266, Springer-Verlag, 1993.
- [13] PAUL T. BOGGS, RICHARD H. BYRD AND ROBERT B. SCHNABEL, *A stable and efficient algorithm for nonlinear orthogonal distance regression*, SIAM Journal Sci. and Stat. Computing 8(6):1052–1078, November 1987.
- [14] PAUL T. BOGGS, RICHARD H. BYRD, JANET E. ROGERS AND ROBERT B. SCHNABEL, *User's Reference Guide for ODRPACK Version 2.01—Software for Weighted Orthogonal Distance Regression*, National Institute of Standards and Technology, Gaithersburg, June 1992.
- [15] P. E. GILL, W. MURRAY AND M. H. WRIGHT, *Practical Optimization*, Academic Press, New York, 1981.

- [16] GENE H. GOLUB, ALAN HOFFMANN, G. W. STEWART, *A Generalization of the Eckart-Young-Mirsky Matrix Approximation Theorem*, Linear Algebra and its Appl. 88/89:317–327(1987)
- [17] WALTER GANDER, GENE H. GOLUB, AND ROLF STREBEL, *Fitting of circles and ellipses—least squares solution*, Technical Report 217, Institut für Wissenschaftliches Rechnen, ETH Zürich, June 1994. Available via anonymous ftp from `ftp.inf.ethz.ch` as `doc/tech-reports/1994/217.ps`.

Institut für Wissenschaftliches Rechnen
Eidgenössische Technische Hochschule
CH-8092 Zürich, Switzerland
email: {gander,strebel}@inf.ethz.ch

Computer Science Department
Stanford University
Stanford, California 94305
email: golub@sccm.stanford.edu