OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Geometric Least Squares Fitting Of Circles

Abdellatif Bettayeb
Department of General Studies, Jubail Industrial College, Jubail Industrial City, Kingdom of Saudi Arabia
**Corresponding Author: Abdellatif Bettayeb*

ABSTRACT
The problem of Fitting conic sections to given data in the plane is one which is of great interest and arises in many applications, e.g. computer graphics, statistics, coordinate metrology, aircraft industry, metrology, astronomy, refractometry, and petroleum engineering [7, 2, 3]. In this paper, we present several methods which have been suggested for fitting ellipses to data in the plane. We will look particularly at one method, by giving examples and using Matlab to solve these problems.

Date of Submission: 27-04-2018 Date of acceptance: 12-05-2018

I INTRODUCTION

Let a relationship between variables x and y be given by $f(x, y; p) = 0,$ where $p \in R^n$ is a vector of parameters. For example, this could be an ellipse or any conic in the *x*,*y* plane.

Let data points $(x_i, y_i), i = 1,..., m.$ be given. Then ideally we wish to choose p so that
$f(x, y; p) = 0, \quad i = 1,\ldots,m.$

However, this is unlikely to be possible, so we need some other ways of choosing *p* such that the sum of the squared orthogonal distances from each data points to the ellipse is minimal, and this is discussed by Helmuth Spath in [4].

In the other sections we introduce different numerical different numerical examples, with relevant figures and results.

II GEOMETRIC FITTING

Given the model
$f(x, y; \beta) = 0,$

and the data points $(x_i, y_i), i = 1,..., m$ in the plane, another possibility is as follows:

We can choose the sum of the squares of the distances from the data points (x_i, y_i) to the curve $f(x, y; \beta) = 0$ to be minimized. We consider the special case when it is possible to give a parameterization of the curve, using $x = x(t), y = y(t).$ Then we examine here a special algorithm proposed by Helmuth Spath. A more general Gauss-Newton method is considered also to be compared with it. Both methods are applied for ellipses.

III THE METHOD OF SPATH FOR A CIRCLE

Let the data points $(x_k, y_k),\ k = 1,..., m$ be given in the plane , and $x = x(t),\ y = y(t)$ be the parametric representation of a circle

$$x(t) = a + r\cos t, \quad y(t) = b + r\sin t.$$

We have to minimise with respect to β and t

$$S(\beta,t) = \sum_{i=1}^{m}[(x_i - x(t_i))^2 + (y_i - y(t_i))^2], \qquad (3.1)$$

where $\beta = [a\ b\ r]^T$ is the parameter vector which must be determined. Also $t = (t_1,..., t_m)^T$ has to be determined such that distances from the given data points (x_i, y_i) to points $(x(t_i), y(t_i)),\ i = 1,..., m$ on the circle are minimal i.e. the distances are orthogonal.

Notice that t_i only appears in the i-th term i.e. $S(\beta,t)$ is separable with respect to the unknowns $t_1,..., t_m$.

First we fix β and we must determine the minimum t_i which is the zeros of $\frac{\partial S}{\partial t_i}(\beta,t)$, then

$$\frac{\partial S}{\partial t_i}(\beta,t) = 0, \qquad (i = 1,..., m). \qquad (3.2)$$

For each *i*, then we could select the one that globally minimises the *j*-th term of *S*.
Thus we could attain a global minimum of $S(\beta,t) = S_1$.

Now for the circle it turns out that the equations (3.2) already are or can be transformed into *m* polynomial equations of low degree less than or equal to four. For the circle we have two real zeros. Next we fix *t* and take

$$\frac{\partial S}{\partial \beta_j}(\beta,t) = 0, \qquad (j = 1,..., n), \qquad (3.3)$$

which can be interpreted as linear least squares problems, because (β appears linear) this, deliver the global optimum for $S(\beta,t)$ as desired. This process was repeated alternatively w.r.t β and t for many iterations until we get the minimum value of *S*.
Let *r* be the residual vector:

$$r = [r_x \quad r_y]^T,$$

each component of *r* is:

$$r_x = \begin{bmatrix} x_1 - a - r\cos t_1 \\ \vdots \\ x_m - a - r\cos t_m \end{bmatrix},$$

$$r_y = \begin{bmatrix} y_1 - b - r\sin t_1 \\ \vdots \\ y_m - b - r\sin t_m \end{bmatrix},$$

then

$$C\beta + d = r, C \in R^{2m\times n}, \beta \in R^n, d, r \in R^{2m}, \qquad (3.4)$$

where C is a $2m \times n,$ matrix, with n in this case equals to 3, and d is a vector of length $2m$.

$$C = \begin{bmatrix} 1 & 0 & \cos t_1 \\ \vdots & & \vdots \\ 1 & 0 & \cos t_m \\ 0 & 1 & \sin t_1 \\ \vdots & & \vdots \\ 0 & 1 & \sin t_m \end{bmatrix}, d = \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ y_1 \\ \vdots \\ y_m \end{bmatrix},$$

we can put: $A = C^T C \quad b = C^T d \quad and\ b_1 = -b.$

At the end we got the normal equation

$$A\beta + b = 0.$$

Using Matlab, we get the value of corresponding to minimum value of S as below

$\beta = A \setminus b_{1,}$

and

$S_2 = S = r^T r,$

3.1 The general Algorithm

Step 1:

Give initial guess $\beta_{0\,0}$, with tolerance *tol*.

Step 2:

Compute by (Maple) the derivative of S w.r.t. (t)

1. $S_i = (x_i - a - r*\cos(t_i))^2 + (y_i - b - r\sin(t._i))^2,$
2. $eq = diff(S_i, t) =$
   $2*(x_i - a - r*\cos(t_i))*r\sin(t_i) - 2*(y_i - b - r*\sin(t._i))*r\cos(t_i),$

3. $sol = solve(eq,t);$
   $sol = (y_i - b)/(x_i - a) \quad t = a\tan(sol),$
4. Take only the real values of t
5. Substitute these real values in S_i and choose the minimum of S_i and the correspondent t_i.
6. The minimal *S* is $\sum_{i=1}^{n} S_i$, equals to S_1.

Step 3:
Fix *t* and find S_2 the minimum of *S* w.r.t. β

1. Solve the linear least square problem $A*\beta = b$ as we saw above such that $\beta = A\backslash(-b)$.
2. $S_2 = r'*r$, where *r* is the residual defined before.
3. Compute S_1 and S_2 until we get the minimum value of them with tolerance proposed, but there must be at each iteration a decrease in the value of *S*.

Remark:
Direct methods are usually faster and more generally applicable, the usual way to access direct methods in MATLAB is not through the LU or Cholesky factorisation, but rather with the matrix division operator / and \.. If *A* is square, the result of $X = A\backslash B$ is the solution to the linear system $AX = B$. If *A* is not square, then a least squares solution is computed (see [5] and MATLAB version 4.2c, 1994).

IV GAUSS-NEWTON METHOD

On considering the nonlinear least squares problem:

$$\min_{X\in R^{n+m}} S = \sum_{i=1}^{m} r_i^2(X).$$

For the ellipse $X = [\beta, t]^T$, where $\beta = [a \ \ b \ \ p \ \ q]^T$ and $t = [t_1,..., t_m]^T$, n =4,
$r = [r_1, r_2,..., r_m]^T, \quad S = r^T r.$
Differentiating w.r.t. X_i, $(i = 1,..., n+m)$

$$\frac{\partial S}{\partial X_i} = 2\sum_{j=1}^{m} r_j \frac{\partial r_j}{\partial X_i}, \qquad (4.1)$$

$$\nabla S(X) = 2J^T r,$$

where *J* is the Jacobian matrix associated with *S* and is an *m x n* matrix of the form:

$$J_{pq} = \frac{\partial r_p}{\partial X_q}.$$

Hence the *p*-th row is the derivative vector of the *p*-th sub-function r_x w.r.t. each element of *X*.

Differentiate again:

$$\frac{\partial^2 S}{\partial X_i\, \partial X_j} = 2\{\sum_{j=1}^{m} [\frac{\partial r_j}{\partial X_k}.\frac{\partial r_j}{\partial X_i} + r_j \frac{\partial^2 r_j}{\partial X_i \partial X_k}]\}, \qquad (4.2)$$

$$\nabla^2 S(X) = 2(J^T J + B),$$

where $\nabla^2 S$ is the *n x n* symmetric Hessian matrix of *S*. The *n x n* matrix *B* which the error matrix is:

$$B = \sum_{i=1}^{m} r_i \nabla^2 r_i ,$$

where $\nabla^2 S$ symmetric and positive definite $J^T J$ positive semi-definite because $z^T J^T J\ z \equiv y^T y \geq 0,$, thus we neglect B [1].

4.1 The general algorithm for Gauss-Newton Method

Then the G-N Algorithm is:

Step 1:

Choose x^0 initial approximation to x and a maximum value of S let be $S_1 = 10^8$, and a tolerance tol, set k =0.

Step 2:
Compute r^k, J^k, thus $J^{kT} J^k$ and $J^{kT} r^k$.
If $\left\| J^T r \right\| \leq tol,$ stop.
$S = r^{kT} * r^k;$
$d = abs(S_1 - S)$
If d < tol, break, end

Step 3:
Solve the equations by finding , here is the correction vector

$$J^{kT} J^k \delta^k = -J^{kT} r^k .$$

Step 4:

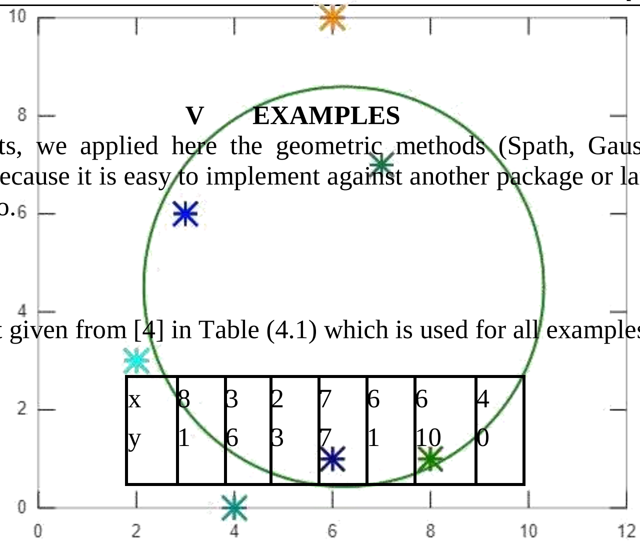
If $\left\| \delta^k \right\| < tol,$ return. Otherwise set $x^{k+1} = x^k + \alpha^k \delta^k$, $k = k+1.$ We put here the step length or the damping factor α equals to 1 (for the circle).

Step 5:

If $S^{k+1} \leq S^k$, return to step 2. where $S = r^T r.$ Otherwise set $\alpha = \frac{\alpha}{2}.$ i.e. halve the step-length , until we get $S^{k+1} \leq S^k$, then return to step 4, (this case appears clearly for the parabola).

The correction vector δ^k is based on local information; the new approximation may have undesirable properties. For example, even though

δ^k is not uphill at x^k, we may still find that $S^{k+1} > S^k$, (the case of the parabola). It is necessary, therefore, to introduce a factor α^k which modifies the norm of the correction vector; it becomes convenient to refer to the latter as a " search direction ". And α^k is usually called a step length, or in the present context, a "damping α factor " (see [1]).



V EXAMPLES

With different set of points, we applied here the geometric methods (Spath, Gauss-Newton) for a circle. MATLAB was used here, because it is easy to implement against another package or language like FORTRAN, and we save a lot of time too.

5.1 Fitting Circles

Consider the 'Spath' data set given from [4] in Table (4.1) which is used for all examples of circles.

x	8	3	2	7	6	6	4
y	1	6	3	7	1	10	0

5.2 Example 1: Geometric Method with Spath Algorithm

Figure 1 shows the circle generated from the data using the Spath method with initial guess $\beta_0 = [6.2804 \quad 4.6459 \quad 4.2230]'$.

And $s = 5.3852$ number of iterations = 19.

The ellipse generated is

$$x = 5.753272 + 3.986094\cos\theta$$

$$y = 4.403979 + 3.986094\sin\theta$$

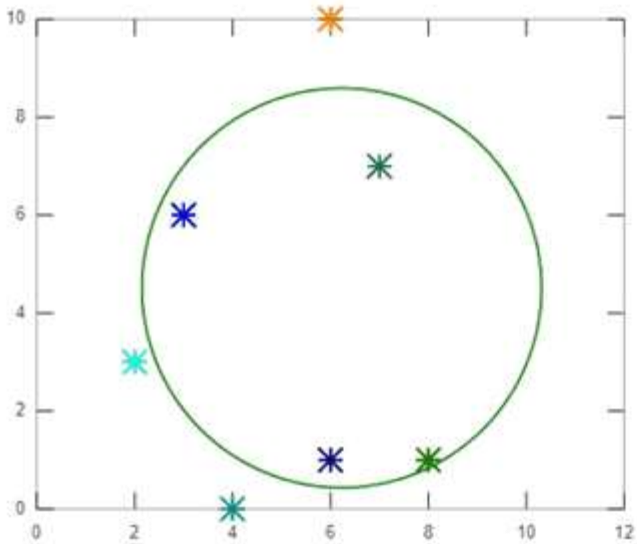


Figure 1: Circle fitting with Spath Method

Example 2: Geometric method with Gauss-Newton

We get the same figure as Figure 1 and we got the results as follow.

s= 5.3852 number of iterations = 13.

The circle generated is

$$x = 5.7335 + 4.1617\cos\theta$$

$$y = 4.9209 + 4.1617\sin\theta$$

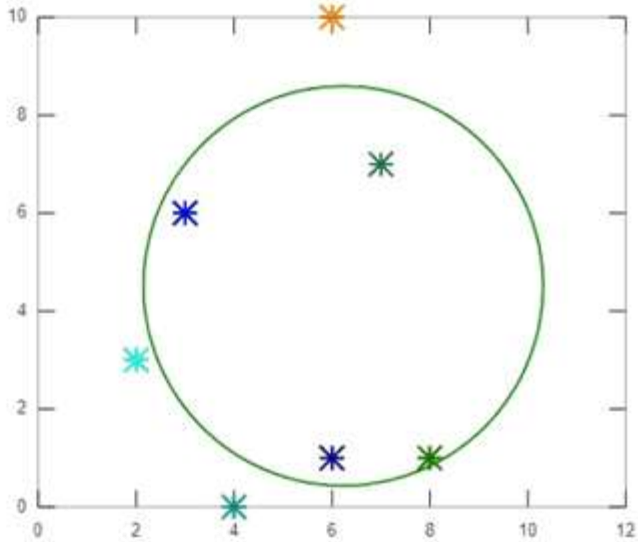


Figure 2: Circle fitting with Gauss-Newton Method.

BIBLIOGRAPHY

[1]. Dixon, L. C. W., Spedicato, E. and Szego, G. P., Nonlinear Optimization: theory and algorithms, Birkhauser Boston, 1980. ISBN 3-7643-3020-1. pp. 91-102.
[2]. Fletcher, R. G., Practical Methods of Optimization, John Wiley and Sons, 1995. ISBN 0 471 91547 5 pp. 111-136.
[3]. Gander, W., Golub, G. H. and Strebel, R., Fitting of circles and ellipses: least square solution,BIT, 34(1994), pp. 556-577.
[4]. Huffel Sabine Van, Recent Advances in total least squares techniques and errors in variables modeling, " Orthogonal Least Squares Fitting by Conic Sections " by Helmuth Spath pp. 259-264, Library of Congress, USA, 1997. ISBN 0-89871-393-5.
[5]. Kolman, B., Elementary linear algebra, 1991. ISBN 0-02-366045-7.
[6]. Schwarz, H. R., Numerical Analysis: A comprehensive introduction, Great Britain, 1989. ISBN 0471 92064 9 pp. 294-329.
[7]. The MathWorks, Inc. MATLAB REFERENCE GUIDE, 1992. PP. 205.

Abdellatif Bettayeb." Geometric Least Squares Fitting Of Circles." International Journal Of Modern Engineering Research (IJMER), vol. 08, no. 05, 2018, pp.64 –70.