

General Least-Squares – Direct Solutions and Bundle Adjustments

by David Manthey (manthey@orbitals.com)

Copyright 1999-2005 by David Manthey

Table of Contents

Introduction.....	1
List of Symbols.....	2
General Technique.....	3
Potential Problems	5
Linear Technique	6
Error Residuals, Ellipsoids, and Confidence	7
An Example Problem.....	8
A Linear Solution.....	8
A Nonlinear Solution	10
An Error Ellipse	12
References.....	14

Introduction

The method of least squares is used to solve a set of linear equations having more equations than unknown variables. Since there are more equations than variables, the solution will not be exactly correct for each equation; rather, the process minimizes the sum of the squares of the residual errors. The method is very powerful and can be applied to numerous applications.

In the general case, the least-squares method is often used to solve a set of non-linear equations that have been linearized using a first-order Taylor-series expansion. Solving non-linear equations is an iterative process using Newton's method. The speed of convergence is dependent on the quality of an initial guess for the solution. The non-linear least-squares method is often referred to as a bundle adjustment since all of the values of an initial guess of the solution are modified together (adjusted in a bundle). This technique is also occasionally referred to as the Gauss-Newton method.

The least-squares method is not new; Legendre invented the method in 1805, and reference books have mentioned least squares in their titles as early as the 1870s. However, in most books about least squares, the general method is bound inextricably with the book's primary subject matter. There is little uniformity between different books, or even within a single book, on the designation of different variables. This makes it difficult to understand the method. This paper provides a new description of least squares which hopefully describes the process in a simple and straight-forward manner. A complete derivation of the method is not provided, only a functional description. Familiarity with matrices and partial derivatives is assumed.

An example problem is provided, computing the location of a circle based on a set of points. The example shows a linear solution and an iterative nonlinear solution to this problem as well as some error analysis.

List of Symbols

a_i	coefficients of a linear equation
ε	relative size of adjustment compared to initial value
$F_i(x_1, x_2, x_3, \dots)$	equation based on a set of unknowns. This is not necessarily a function. The equation must be solved for the value (i.e., $F_i(x_1, x_2, x_3, \dots) = k_i$) whose error will be minimized
Fisher(α, ν_1, ν_2)	the Fisher distribution
J	the Jacobian matrix. This is the partial differentials of each equation with respect to each unknown. Sometimes written $\mathbf{J} = \nabla F_i(x_1, x_2, x_3, \dots)$, where ∇ is del, the gradient operator
K	vector of residuals. Each component is the difference between the observation, k_i , and the equation evaluated for the initial guess, x_{i0}
k_i	an observation. The least-squares process minimizes the error with respect to the observations
N	the normal matrix. $\mathbf{N} = \mathbf{J}'\mathbf{W}\mathbf{J}$
Q	cofactor matrix. The inverse of W . This is typically the standard deviations of the measurements
\mathbf{Q}_{xx}	the covariance matrix. Also called the variance-covariance matrix. $\mathbf{Q}_{xx} = \mathbf{N}^{-1}$
\mathbf{Q}'_{xx}	a sub-matrix of the covariance matrix. Used to calculate an error ellipse
$q_{xx_{ij}}$	a value within the covariance matrix
r	the degrees of freedom. This is the number of equations minus the number of unknowns
S_0^2	the reference variance. This is used to scale the covariance matrix
S_{axis}	length of the semi-axis of an error ellipse based on the covariance matrix
$S_{axis\%}$	length of the semi-axis of an error ellipse with a specific confidence level
W	weighting matrix. This is a square symmetric matrix. For independent measurements, this is a diagonal matrix. Larger values indicate greater significance
w_i	the diagonal components of the weighting matrix
X	vector of initial guesses for the unknowns
X'	vector of refined guesses for the unknowns. $\mathbf{X}' = \mathbf{X} + \Delta\mathbf{X}$
x_i	an unknown value. This is solved for in the least-squares process
x_i'	the refined guess for an unknown value. $x_i' = x_{i0} + \Delta x_i$
x_{i0}	the initial guess for an unknown
$\Delta\mathbf{X}$	vector of adjustment values. The initial guesses are adjusted by this amount

Δx_i an adjustment value. This is added to the initial guess to improve the solution
 $\Gamma(v)$ the Gamma distribution

General Technique

The general least squares process can be used to solve a set of equations for a set of unknowns. The only requirement is that there are at least as many equations as there are unknowns.

If the equations are linear, the least-squares process will produce a direct solution for the unknowns. If the equations are not linear, an initial guess of the unknowns is required, and the result is an *adjustment* to the initial parameters. This is repeated until the results converge (the adjustments become very close to zero). The linear case is an adjustment using zero as the initial guess of all parameters.

The process requires a set of equations with the unknowns on one side and some known quantity on the other. Let x_i be the set of unknowns, and let the equations be of the form

$$F_i(x_1, x_2, x_3, \dots) = k_i$$

where k_i is the observation (value) whose least-squares error will be minimized. Since there are more equations than unknowns, the solution of the unknowns will not be exact. Using the solution to compute the equation, $F_i(x_1, x_2, x_3, \dots)$, will not generate the exact observation value, k_i . The square of the difference between the evaluated equation and the observation is minimized.

There is typically one equation for each observation. In photogrammetry, this might be one equation for each x pixel coordinate and one equation for each y pixel coordinate. Each equation is not required to have all of the unknowns in it.

The Jacobian matrix, \mathbf{J} , is the matrix of the partial differentials of each equation with respect to each unknown. That is,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \frac{\partial F_1}{\partial x_3} & \dots \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \frac{\partial F_2}{\partial x_3} & \dots \\ \frac{\partial F_3}{\partial x_1} & \frac{\partial F_3}{\partial x_2} & \frac{\partial F_3}{\partial x_3} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

In general, the height of the Jacobian matrix will be larger than the width, since there are more equations than unknowns.

Furthermore, let the vector \mathbf{K} be the vector of the residuals. A residual is the difference between the observation and the equation calculated using the initial values. That is

$$\mathbf{K} = \begin{bmatrix} k_1 - F_1(x_{10}, x_{20}, x_{30}, \dots) \\ k_2 - F_2(x_{10}, x_{20}, x_{30}, \dots) \\ k_3 - F_3(x_{10}, x_{20}, x_{30}, \dots) \\ \vdots \end{bmatrix}$$

One further parameter is a weighting matrix, \mathbf{W} . This is a matrix which includes the expected confidence of each equation and also includes any dependence of the equations. A larger value in the weighting matrix increases the importance of the corresponding equation (larger values indicate greater confidence). It is a square symmetric matrix with one row per equation. The main diagonal contains the weights of the individual equations, while the off-diagonal entries are the dependencies of equations upon one another. If all of the observations are independent, this will be a diagonal matrix. The cofactor matrix, \mathbf{Q} , is the inverse of the weighting matrix (i.e., $\mathbf{Q} = \mathbf{W}^{-1}$).

Let x_{i0} be an initial guess for the unknowns. The initial guesses can have any finite real value, but the system will converge faster if the guesses are close to the solution. Also, let \mathbf{X} be the vector of these initial guesses. That is

$$\mathbf{X} = \begin{bmatrix} x_{10} \\ x_{20} \\ x_{30} \\ \vdots \end{bmatrix}$$

It is desirable to solve for the adjustment values, $\Delta\mathbf{X}$. This is the vector of adjustments for the unknowns

$$\Delta\mathbf{X} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \vdots \end{bmatrix}$$

where, based on the initial guess, \mathbf{X} , and an adjustment, $\Delta\mathbf{X}$, a set of new values are computed $\mathbf{X}' = \mathbf{X} + \Delta\mathbf{X}$

To solve for $\Delta\mathbf{X}$, (see the references for the reasoning behind this solution)

$$\Delta\mathbf{X} = (\mathbf{J}'\mathbf{W}\mathbf{J})^{-1} \mathbf{J}'\mathbf{W}\mathbf{K}$$

In various texts, the normal matrix, \mathbf{N} , is defined as

$$\mathbf{N} = \mathbf{J}'\mathbf{W}\mathbf{J},$$

and the covariance matrix (sometimes referred to as the variance-covariance matrix), \mathbf{Q}_{xx} , is defined as

$$\mathbf{Q}_{xx} = (\mathbf{J}'\mathbf{W}\mathbf{J})^{-1} = \mathbf{N}^{-1}$$

If the weighting matrix is diagonal, then $\Delta\mathbf{X}$ can be solved by row reduction of the matrix

$$\begin{bmatrix} \sum_i \left(\frac{\partial F_i^2}{\partial x_1} w_i \right) & \sum_i \left(\frac{\partial F_i}{\partial x_1} \frac{\partial F_i}{\partial x_2} w_i \right) & \sum_i \left(\frac{\partial F_i}{\partial x_1} \frac{\partial F_i}{\partial x_3} w_i \right) & \cdots & \sum_i \left(\frac{\partial F_i}{\partial x_1} (k_i - F_i(x_{10}, x_{20}, x_{30}, \dots)) w_i \right) \\ \sum_i \left(\frac{\partial F_i}{\partial x_1} \frac{\partial F_i}{\partial x_2} w_i \right) & \sum_i \left(\frac{\partial F_i^2}{\partial x_2} w_i \right) & \sum_i \left(\frac{\partial F_i}{\partial x_2} \frac{\partial F_i}{\partial x_3} w_i \right) & \cdots & \sum_i \left(\frac{\partial F_i}{\partial x_2} (k_i - F_i(x_{10}, x_{20}, x_{30}, \dots)) w_i \right) \\ \sum_i \left(\frac{\partial F_i}{\partial x_1} \frac{\partial F_i}{\partial x_3} w_i \right) & \sum_i \left(\frac{\partial F_i}{\partial x_2} \frac{\partial F_i}{\partial x_3} w_i \right) & \sum_i \left(\frac{\partial F_i^2}{\partial x_3} w_i \right) & \cdots & \sum_i \left(\frac{\partial F_i}{\partial x_3} (k_i - F_i(x_{10}, x_{20}, x_{30}, \dots)) w_i \right) \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

where w_1, w_2, w_3, \dots are the diagonal elements of the weighting matrix. Note that the left side of the matrix is the normal matrix, \mathbf{N} .

The initial guesses, x_{i0} , are updated using the solution of the adjustment matrix, $\Delta\mathbf{X}$, as follows

$$\mathbf{X}' = \mathbf{X} + \Delta\mathbf{X}$$

or

$$x_i' = x_{i0} + \Delta x_i$$

The process is repeated using the new values, x_i' , as the initial guesses until the adjustments are close to zero.

Practically, an adjustment is close to zero when it is small compared to the absolute magnitude of the value it is adjusting, i.e., $\Delta x_i < x_{i0} \cdot \varepsilon$, where ε is a small value. The actual value for ε can be selected based on the number of decimal digits of precision used in the calculations. Typically, the order of magnitude of ε will be a few less than the number of digits of precision. For example, if the calculations are done on a computer using standard double precision (8-byte) values, the computer can hold around 15 digits of precision; therefore $\varepsilon \approx 10^{-12}$.

Potential Problems

There are conditions where the solution will not converge or will converge to undesirable values. This process finds a local minimum. As such, there may be a better solution than the one found. A solution is dependent on the equations, $F_i(x_1, x_2, x_3, \dots)$, being continuous in x_1, x_2, x_3, \dots . The first and second derivatives do not need to be continuous, but if the equations are not continuous, there is no guaranty that the process will converge. Also, in certain circumstances, even if the equations are continuous, the solution may not converge. This can happen when the first and second derivatives of the equations have significantly different values at the initial values than at the solution.

In any case where the solution does not converge, a solution may still be able to be obtained if different starting values, x_{i0} , are used.

Linear Technique

For sets of linear equations, the least-squares process will produce a direct solution for the unknowns. The linear case is mathematically the same as the general case where an adjustment is performed using zero as the initial guess of all parameters. Only a single iteration is required for convergence.

The equations must be of the form

$$F_i(x_1, x_2, x_3, \dots) = a_1x_1 + a_2x_2 + a_3x_3 + \dots = k_i$$

The Jacobian matrix, \mathbf{J} , is therefore

$$\mathbf{J} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \cdots \\ a_{12} & a_{22} & a_{32} & \cdots \\ a_{13} & a_{23} & a_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where a_{ij} is the i th coefficient of the j th equation.

Since the initial guesses are all zero, the vector of residuals, \mathbf{K} , is

$$\mathbf{K} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ \vdots \end{bmatrix}$$

If the weighting matrix is diagonal, then $\Delta\mathbf{X}$ can be solved by row reduction of the matrix

$$\begin{bmatrix} \sum_i (a_{1i}^2 w_i) & \sum_i (a_{1i} a_{2i} w_i) & \sum_i (a_{1i} a_{3i} w_i) & \cdots & \sum_i (a_{1i} k_i w_i) \\ \sum_i (a_{1i} a_{2i} w_i) & \sum_i (a_{2i}^2 w_i) & \sum_i (a_{2i} a_{3i} w_i) & \cdots & \sum_i (a_{2i} k_i w_i) \\ \sum_i (a_{1i} a_{3i} w_i) & \sum_i (a_{2i} a_{3i} w_i) & \sum_i (a_{3i}^2 w_i) & \cdots & \sum_i (a_{3i} k_i w_i) \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

The final solution will be the adjustment values. That is

$$\mathbf{X} = \Delta\mathbf{X}$$

or

$$x_i = \Delta x_i$$

Error Residuals, Ellipsoids, and Confidence

The covariance matrix, \mathbf{Q}_{xx} , contains the variance of each unknown and the covariance of each pair of unknowns. The quantities in \mathbf{Q}_{xx} need to be scaled by a reference variance. This reference variance, S_0^2 , is related to the weighting matrix and the residuals by the equation

$$S_0^2 = \frac{\mathbf{K}'\mathbf{W}\mathbf{K}}{r}$$

where r is the number of degrees of freedom (i.e., the number of equations minus the number of unknowns).

For any set of quantities, an error ellipse can be calculated. The dimensions and orientations of the ellipse are calculated from the coefficients of the covariance matrix. Only the coefficients of the covariance matrix in the relevant rows and columns are used. This is the appropriate $n \times n$ sub-matrix, where n is the number of dimensions for the error ellipse. The sub-matrix is symmetric.

The ellipse matrix is composed of entries from the covariance matrix. For example, a three-dimensional error ellipsoid is computed from

$$\mathbf{Q}'_{xx} = \begin{bmatrix} q_{xx_{aa}} & q_{xx_{ab}} & q_{xx_{ac}} \\ q_{xx_{ba}} & q_{xx_{bb}} & q_{xx_{bc}} \\ q_{xx_{ca}} & q_{xx_{cb}} & q_{xx_{cc}} \end{bmatrix}$$

where $q_{xx_{ij}}$ are values from the covariance matrix \mathbf{Q}_{xx} , and a , b , and c are the indices for the unknowns for which the ellipse is computed.

The error ellipse semi-axes are given by

$$S_{axis} = \pm \sqrt{S_0^2 \text{eigenvalue}_{axis}(\mathbf{Q}'_{xx})}$$

The orientation of the error ellipse is the column eigenvectors of \mathbf{Q}'_{xx} .

To determine the error to a specific confidence level, the length of the semi-axis is multiplied by a confidence factor based on the Fisher distribution using the formula

$$S_{axis\%} = S_{axis} \sqrt{2 \text{Fisher}(1 - \text{confidence}, \# \text{ of unknowns}, r)}$$

where the *confidence* is a number from 0 to 1, with 1 being complete confidence, and r is the number of degrees of freedom. The Fisher distribution is determined from the equation

$$\alpha = \int_{\text{Fisher}(\alpha, v_1, v_2)}^{\infty} \frac{\Gamma((v_1 + v_2)/2) \left(\frac{v_1}{v_2}\right)^{v_1/2}}{\Gamma(v_1/2)\Gamma(v_2/2) \left(\frac{v_1}{v_2}\right)^{v_1/2}} \frac{x^{(v_1-2)/2}}{[1 + (v_1/v_2)]x^{(v_1+v_2)/2}} dx$$

where the Gamma function, $\Gamma(v)$, is given by

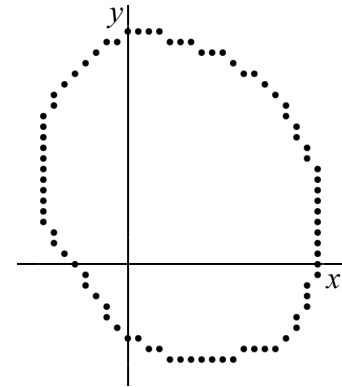
$$\Gamma(v) = \int_0^{\infty} u^{v-1} e^{-u} du$$

An Example Problem

Given a set of two-dimensional points, find the circle best represented by these points.

For this example, there are 82 points, see also the figure to the right:

3,-8	13,-8	18,1	17,11	9,20	-1,21	-8,12	-7,2	1,-7
4,-9	14,-8	18,2	16,12	8,20	-2,21	-8,11	-6,1	2,-8
5,-9	15,-7	18,3	16,13	7,20	-3,20	-8,10	-5,0	
6,-9	16,-6	18,4	15,14	6,21	-4,19	-8,9	-4,-1	
7,-9	16,-5	18,5	15,15	5,21	-5,18	-8,8	-4,-2	
8,-9	17,-4	18,6	14,16	4,21	-6,17	-8,7	-3,-3	
9,-9	17,-3	18,7	13,17	3,22	-7,16	-8,6	-2,-4	
10,-9	17,-2	18,8	12,18	2,22	-7,15	-8,5	-2,-5	
11,-8	18,-1	18,9	11,18	1,22	-8,14	-8,4	-1,-6	
12,-8	18,0	17,10	10,19	0,22	-8,13	-7,3	0,-7	



A set of points used to approximate a circle

Let the points be denoted (x_i, y_i) . That is, $(x_1, y_1) = (3, -8)$, $(x_2, y_2) = (4, -9)$, $(x_3, y_3) = (5, -9)$, ...

A circle can be defined by the equation

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} = r$$

where (x_0, y_0) is the center of the circle and r is the radius of the circle.

Ideally, it is desirable to minimize the square of the distance between the points defining the circle and the circle. That is, minimize

$$\sum_i \left[\left(\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r \right)^2 \right]$$

This is equivalent to performing the least-squares process using the equations

$$F_i(x_0, y_0, r) = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r = 0$$

A Linear Solution

The equation of a circle is not linear in the unknown values x_0 , y_0 , and r . The equation of a circle can be written in the linear form

$$A(x^2 + y^2) + Bx + Cy = 1$$

where

$$A = \frac{1}{r^2 - x_0^2 - y_0^2}$$

$$B = \frac{-2x_0}{r^2 - x_0^2 - y_0^2}$$

$$C = \frac{-2y_0}{r^2 - x_0^2 - y_0^2}$$

This can be rewritten for the original unknowns as

$$x_0 = \frac{-B}{2A}$$

$$y_0 = \frac{-C}{2A}$$

$$r = \frac{\sqrt{4A + B^2 + C^2}}{2A}$$

Note that by using the linear form of the circle equation, the square of the distance from each point to the circle is not the value that is being minimized.

Using the equation $A(x^2 + y^2) + Bx + Cy = 1$ with the unknowns A , B , and C , the Jacobian matrix is

$$\mathbf{J} = \begin{bmatrix} \frac{\partial F_1}{\partial A} & \frac{\partial F_1}{\partial B} & \frac{\partial F_1}{\partial C} \\ \frac{\partial F_2}{\partial A} & \frac{\partial F_2}{\partial B} & \frac{\partial F_2}{\partial C} \\ \frac{\partial F_3}{\partial A} & \frac{\partial F_3}{\partial B} & \frac{\partial F_3}{\partial C} \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} x_1^2 + y_1^2 & x_1 & y_1 \\ x_2^2 + y_2^2 & x_2 & y_2 \\ x_3^2 + y_3^2 & x_3 & y_3 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Placing numerical values in the Jacobian matrix gives

$$\mathbf{J} = \begin{bmatrix} 3^2 + (-8)^2 & 3 & -8 \\ 4^2 + (-9)^2 & 4 & -9 \\ 5^2 + (-9)^2 & 5 & -9 \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 73 & 3 & -8 \\ 97 & 4 & -9 \\ 106 & 5 & -9 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Since all initial guesses are zero, the residual vector is

$$\mathbf{K} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$

Lacking other information, the weighting matrix, \mathbf{W} , is the identity matrix. That is

$$\mathbf{W} = \mathbf{I}$$

The unknowns, A , B , and C , can be solved for using the equation

$$\mathbf{X} = (\mathbf{J}'\mathbf{W}\mathbf{J})^{-1} \mathbf{J}'\mathbf{W}\mathbf{K} = (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'\mathbf{K}$$

This can be simplified to

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_i ((x_i^2 + y_i^2)^2) & \sum_i ((x_i^2 + y_i^2)x_i) & \sum_i ((x_i^2 + y_i^2)y_i) \\ \sum_i ((x_i^2 + y_i^2)x_i) & \sum_i (x_i^2) & \sum_i (x_i y_i) \\ \sum_i ((x_i^2 + y_i^2)y_i) & \sum_i (x_i y_i) & \sum_i (y_i^2) \end{bmatrix}^{-1} \begin{bmatrix} \sum_i (x_i^2 + y_i^2) \\ \sum_i x_i \\ \sum_i y_i \end{bmatrix}$$

Numerically, this is

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 7954653 & 177852 & 234765 \\ 177852 & 9840 & 1242 \\ 234765 & 1242 & 12285 \end{bmatrix}^{-1} \begin{bmatrix} 22125 \\ 438 \\ 497 \end{bmatrix}$$

Solving for the unknowns, this is

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 0.006328442803 \\ -0.06048422085 \\ -0.07436511782 \end{bmatrix}$$

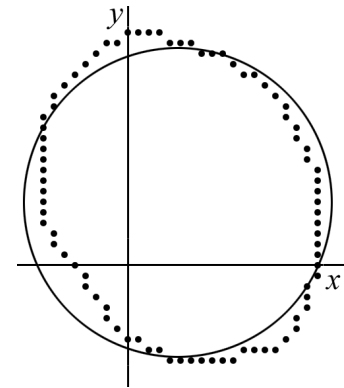
Solving for the circle parameters, this is

$$x_0 = 4.778760172$$

$$y_0 = 5.875467325$$

$$r = 14.67564038$$

This solution is shown superimposed on the original points in the figure to the right. Note that the circle does not fit the data points very well because the solution used the linear form of the circle equations. A much better fit can be obtained using the nonlinear equations.



A circle calculated by a linear least-squares fit

A Nonlinear Solution

Instead of solving for the linear parameters, A , B , and C , it is more desirable to solve for the values x_0 , y_0 , and r using an equation that minimizes the distance from the points to the circle.

The following equation is used

$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r = 0$$

The Jacobian matrix is

$$\mathbf{J} = \begin{bmatrix} \frac{\partial F_1}{\partial x_0} & \frac{\partial F_1}{\partial y_0} & \frac{\partial F_1}{\partial r} \\ \frac{\partial F_2}{\partial x_0} & \frac{\partial F_2}{\partial y_0} & \frac{\partial F_2}{\partial r} \\ \frac{\partial F_3}{\partial x_0} & \frac{\partial F_3}{\partial y_0} & \frac{\partial F_3}{\partial r} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

where

$$\frac{\partial F_i}{\partial x_0} = \frac{x_0 - x_i}{\sqrt{x_i^2 - 2x_0x_i + y_i^2 - 2y_0y_i + x_0^2 + y_0^2}}$$

$$\frac{\partial F_i}{\partial y_0} = \frac{y_0 - y_i}{\sqrt{x_i^2 - 2x_0x_i + y_i^2 - 2y_0y_i + x_0^2 + y_0^2}}$$

$$\frac{\partial F_i}{\partial r} = -1$$

The solution of x_0 , y_0 , and r will require multiple iterations. A starting guess for these values is required. The starting guess can be arbitrary provided the equations are valid. A starting guess of

$$x_0 = 0$$

$$y_0 = 0$$

$$r = 15$$

will be used.

Placing numerical values in the Jacobian matrix gives

$$\mathbf{J} = \begin{bmatrix} -0.3511234415 & 0.9363291775 & -1 \\ -0.4061384660 & 0.9138115486 & -1 \\ -0.4856429311 & 0.8741572761 & -1 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

The residual vector is

$$\mathbf{K} = \begin{bmatrix} 0 - \left(\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} - r \right) \\ 0 - \left(\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} - r \right) \\ 0 - \left(\sqrt{(x_3 - x_0)^2 + (y_3 - y_0)^2} - r \right) \\ \vdots \end{bmatrix} = \begin{bmatrix} 6.455996254 \\ 5.151142198 \\ 4.704369858 \\ \vdots \end{bmatrix}$$

Again, lacking other information, the weighting matrix, \mathbf{W} , is the identity matrix.

The adjustments of the unknowns, Δx_0 , Δy_0 , and Δr , can be solved for using the equation

$$\Delta \mathbf{X} = (\mathbf{J}'\mathbf{W}\mathbf{J})^{-1} \mathbf{J}'\mathbf{W}\mathbf{K} = (\mathbf{J}'\mathbf{J})^{-1} \mathbf{J}'\mathbf{K}$$

Numerically, this is

$$\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \\ \Delta r \end{bmatrix} = \begin{bmatrix} 39.46860791 & -3.660842352 & 17.42846462 \\ -3.660842352 & 42.53139208 & 18.73056460 \\ 17.42846462 & 18.73056460 & 82 \end{bmatrix}^{-1} \begin{bmatrix} 176.5730305 \\ 216.0415308 \\ 37.54002153 \end{bmatrix}$$

Solving for the adjustment values, this is

$$\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \\ \Delta r \end{bmatrix} = \begin{bmatrix} 6.134768609 \\ 6.649105121 \\ -2.364891088 \end{bmatrix}$$

The circle parameters are then adjusted to

$$\begin{aligned} x_0' &= x_0 + \Delta x_0 = 0 + 6.134768609 = 6.134768609 \\ y_0' &= y_0 + \Delta y_0 = 0 + 6.649105121 = 6.649105121 \\ r' &= r + \Delta r = 15 + -2.364891088 = 12.63510891 \end{aligned}$$

This new solution is now used to compute a new Jacobian matrix and a new residual vector, which are then used to again solve for $\Delta \mathbf{X}$. The next iteration is

$$\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \\ \Delta r \end{bmatrix} = \begin{bmatrix} 39.76862489 & -2.956924437 & -1.776388618 \\ -2.956924437 & 42.23137510 & -2.280678045 \\ -1.776388618 & -2.280678045 & 82 \end{bmatrix}^{-1} \begin{bmatrix} -42.60616227 \\ -19.41000442 \\ 132.7928421 \end{bmatrix}$$

$$= \begin{bmatrix} -1.033761936 \\ -0.4464161055 \\ 1.584613997 \end{bmatrix}$$

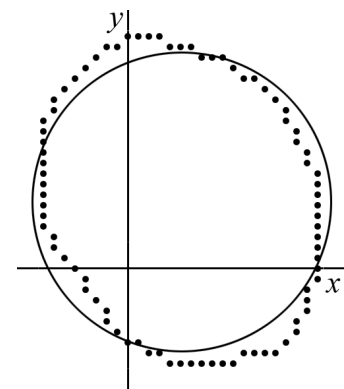
The circle parameters are now adjusted to

$$\begin{aligned} x_0' &= x_0 + \Delta x_0 = 6.134768609 + -1.033761936 = 5.101006672 \\ y_0' &= y_0 + \Delta y_0 = 6.649105121 + -0.4464161055 = 6.202689015 \\ r' &= r + \Delta r = 12.63510891 + 1.584613997 = 14.21972290 \end{aligned}$$

Additional iterations are performed until the adjustment values become close to zero. When performing computations with ten significant figures, after nine iterations, the values have converged to

$$\begin{aligned} x_0 &= 5.155701836 \\ y_0 &= 6.233137797 \\ r &= 14.24203182 \end{aligned}$$

This solution is shown superimposed on the original points in the figure to the right. Note that the circle fits the data points significantly better than the linear solution.



A circle calculated by a nonlinear least-squares fit

An Error Ellipse

To determine the expected error of the nonlinear solution of the circle, the error ellipse can be calculated. To illustrate this, the error ellipse of the circle's center will be computed.

The error ellipse is computed using the covariance matrix

$$\mathbf{Q}_{xx} = (\mathbf{J}'\mathbf{W}\mathbf{J})^{-1} = \begin{bmatrix} 0.02523150611 & 0.001765315825 & -0.000307759723 \\ 0.001765315825 & 0.02385684640 & 0.0002653637522 \\ -0.000307759723 & 0.0002653637522 & 0.01220234392 \end{bmatrix}$$

Since only the error ellipse for the center of the circle, (x_0, y_0) , is desired, the matrix of interest is

$$\mathbf{Q}'_{xx} = \begin{bmatrix} 0.02523150611 & 0.001765315825 \\ 0.001765315825 & 0.02385684640 \end{bmatrix}$$

The matrix \mathbf{Q}'_{xx} is composed of the entries in the first two columns and rows of the matrix \mathbf{Q}_{xx} since x_0 corresponds to the first column and y_0 to the second column of the Jacobian matrix.

The eigenvalues of \mathbf{Q}'_{xx} are

$$\text{eigenvalues}(\mathbf{Q}'_{xx}) = \begin{bmatrix} 0.02643857914 \\ 0.02264977336 \end{bmatrix}$$

The eigenvectors specify the orientation of the ellipse. The eigenvectors are

$$\text{eigenvectors}(\mathbf{Q}'_{xx}) = \begin{bmatrix} 0.8254760367 \\ 0.5644371646 \end{bmatrix}, \begin{bmatrix} -0.5644371646 \\ 0.8254760367 \end{bmatrix}$$

The reference variance is computed based on the residual matrix and on the degrees of freedom. The number of degrees of freedom is the number of observations (82 observations – the number of points used to compute the circle) minus the number of unknowns (3 unknowns – x_0 , y_0 , and r). The reference variance is

$$S_0^2 = \frac{\mathbf{K}'\mathbf{W}\mathbf{K}}{82-3} = \frac{145.8856282}{79} = 1.846653521$$

The error ellipse can be computed for any confidence level. The Fisher distribution needs to be computed for the selected confidence. Although the reference variance was computed based on the total number of unknowns, the Fisher distribution is computed only for the number of unknowns in the ellipse. For 95% confidence

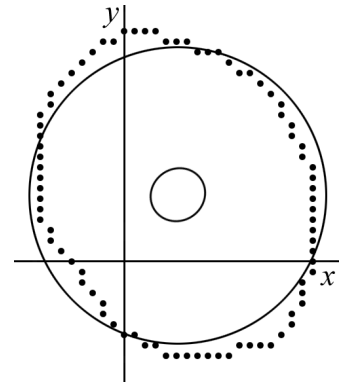
$$\text{Fisher}(1-0.95, 2, 79) = 3.11227$$

Based on this information, the semi-axes of the error ellipse are

$$\begin{aligned} S_{major} &= \sqrt{S_0^2 \text{eigenvalue}} \sqrt{2 \text{Fisher}(1-0.95, 2, 79)} \\ &= \sqrt{1.846653521 \cdot 0.02643857914} \sqrt{2 \cdot 3.11227} = 0.551271 \\ S_{minor} &= \sqrt{1.846653521 \cdot 0.02264977336} \sqrt{2 \cdot 3.11227} = 0.510244 \end{aligned}$$

The 95% confidence error ellipse is shown on the same graph as the circle, see figure to the right. The error ellipse has been magnified by a factor of 5 to make it more visible.

The meaning of the 95% confidence error ellipse is that if more points are added to improve the calculation of the circle, there is 95% probability that the center of the circle will remain within the area specified by the ellipse. As expected, as more points are used to compute the circle, the area of the ellipse becomes smaller. If the points lie closer to the actual circle, this will also reduce the ellipse area.



The error ellipse for the center of the circle magnified by a factor of 5

References

- [1] Nash, Stephan G. and Sofer, Ariela, *Linear and Nonlinear Programming*, McGraw-Hill Companies, Inc, New York, 1996. ISBN 0-07-046065-5.
- [2] Wolf, Paul R. and Ghilani, Charles D., *Adjustment Computations*, John Wiley and Sons, Inc, New York, 1997. ISBN 0-471-16833-5.