IMPLICIT SURFACE FITTING USING DIRECTIONAL CONSTRAINTS *

M. GULLIKSSON¹, I. SÖDERKVIST², and G.A. WATSON³

¹*Department of Computing Science, Umeå University
S-901 87, Umeå, Sweden. email: marten@cs.ume.se*

²*Department of Mathematics, Technical University of Luleå
S-971 87, Luleå, Sweden. email: inge@sm.luth.se*

³*Department of Mathematics, University of Dundee
Dundee DD1 4HN, Scotland. email: gawatson@maths.dundee.ac.uk*

AMS subject classification: 65D10.

Key words: Gauss-Newton, Least squares, Metrology, Surface fitting,

Abstract.

A commonly used technique for fitting curves and surfaces to measured data is that known as orthogonal distance regression, where the sum of squares of orthogonal distances from the data points to the surface is minimized. An alternative has recently been proposed for curves and surfaces which are parametrically defined, which minimizes the sum of squares in given directions which depend on the measuring process. In addition to taking account of that process, it is claimed that this technique has the advantage of complying with traditional fixed-regressor assumptions, enabling standard inference theory to apply. Here we consider extending this idea to curves and surfaces where the only assumption made is that there is an implicit formulation. Numerical results are given to illustrate the algorithmic performance.

1 Introduction

The problem of fitting a curve or surface to data is a very important one with many applications, for example in computer graphics, engineering, statistics, metrology, astronomy, reflectometry, etc. An important application is to metrology, the science of measurement, and there it is often the case that the data are obtained by taking readings from the surface of a manufactured part. This normally involves the use of a co-ordinate measuring machine, where a probe moving in a certain direction makes contact with the part to obtain the measured points $\tilde{x}_i, i = 1, \dots, m$. A common technique for fitting the surface to these data is that known as orthogonal distance regression: the shortest distance from the measured points to the model is obtained, and the sum of squares of these distances is minimized [9], [10], [11], [12], [16]. Under mild assumptions, the shortest distances are orthogonal projections from the observed data onto

---

*Received xxxx 20000 . Communicated by xxx xxx.

the surface, and this explains the terminology. It follows that this type of approximation process is entirely independent of the way in which the data points are obtained, and in particular makes no use of the probe directions, even if these are available.

It has been argued by Hulting [7] that making explicit use in the approximation method of the actual probe directions (with respect to a specific frame of reference) can lead to a process which not only makes use of the measurement design, but complies with traditional fixed-regressor assumptions, enabling standard inference theory to apply. An algorithm for fitting circles and ellipses to measured data which incorporates the probe directions was given by Watson [13], and this was extended to parametrically defined surfaces in [14]. The use of the l_1 rather than the l_2 norm was considered in [15].

Suppose that the points $\tilde{x}_i$ are determined by using a co-ordinate measuring machine with a touch probe, and the probe directions, with respect to a particular frame of reference, are given by v_i , $i = 1, \dots, m$. It is assumed that these are known precisely, and are normalized so that $v_i^T v_i = 1$. It follows that the points corresponding to $\tilde{x}_i$ where the probe makes contact with the model curve or surface are given by $x_i = \tilde{x}_i + \delta_i v_i$ for some numbers δ_i , whose magnitude measures the error in the i th point. A feature of the methods mentioned in the previous paragraph is the assumption that a parameterization of the surface is available, so that a point on the surface can be represented as

$$x = x(t, \alpha)$$

with t a parameter, in $\mathbb{R}^{n-1}$ if $x \in \mathbb{R}^n$.

The appropriate minimization problem is then

$$(1.1) \quad \begin{aligned} & \min_{\alpha, \delta_i} \sum_i \delta_i^2 \\ & \text{such that } \tilde{x}_i + \delta_i v_i = x(t_i, \alpha), \quad i = 1, \dots, m. \end{aligned}$$

This is a nonlinear least squares problem with mn nonlinear constraints. The key to efficient solution of the problem (1.1) is to regard each δ_i as a function of α by satisfying exactly the corresponding constraint. Thus the problem becomes one in α alone, which can be solved by the Gauss-Newton method.

We are concerned here with the problem of fitting to given measured data $\tilde{x}_i \in \mathbb{R}^n$, $i = 1, \dots, m$, a surface (or curve) represented by

$$(1.2) \quad f(x, \alpha) = 0, \quad f : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R},$$

where the parameters forming the vector $\alpha \in \mathbb{R}^p$ determine the nature of the surface. These parameters may be the same ones as would occur in (1.1), if such a representation exists, but even in this case, they need not be. A basic assumption is that f is a continuously differentiable function of its parameters. It is the purpose of this paper to extend the ideas give above to this sort of *implicitly defined* surface, in other words to surfaces for which there may be no obvious or convenient parameterization. Therefore we will assume no more than

that we have a representation of the surface as defined by (1.2). Our problem can then be formulated as a constrained minimization problem

$$(1.3) \quad \begin{aligned} & \min_{\alpha, \delta_i} \sum_i \delta_i^2 \\ & \text{such that } f(\tilde{x}_i + \delta_i v_i, \alpha) = 0, \quad i = 1, \dots, m. \end{aligned}$$

This is a nonlinear least squares problem with m nonlinear constraints.

While an explicit or parametric form may be expressed in the form (1.2), the converse does not necessarily hold. For example a model curve which is naturally implicit is the hyperbolic model given by

$$x_1^2 x_2^2 = a,$$

(see [1]), where $a \in \mathbb{R}$, and where (here) x_1 and x_2 are the components of x . Many other curves and surfaces, including for example spline curves and surfaces, may also be expressed in this form, although parametric forms may sometimes also be available. Our intention here, therefore, is to increase the generality of the underlying approach by showing how to work directly with the form (1.3). As another simple example we may take the sphere represented implicitly as $\|x - c\|^2 = r^2$ where c is the centre and r is the radius. Then we have the problem

$$\begin{aligned} & \min_{\alpha, \delta_i} \sum_i \delta_i^2 \\ & \text{such that } \|\tilde{x}_i + \delta_i v_i - c\|^2 = r^2, \quad i = 1, \dots, m. \end{aligned}$$

As an example of the technique, this problem will be considered in more detail in Section 3. Some experiments are also carried out using an ellipse and an hyperquadric curve. A feature of working with parametric surfaces or curves (like ellipses) by solving (1.1) is that special arrangements may have to be made when they are not in “normal” position with respect to the axes. This can be done by for example working with rotated data. We will see that this aspect can be dealt with in a more natural and potentially straightforward manner using the present formulation. A final comment here about working with (1.3) as opposed to (1.1) is that solving for δ_i from a single equation may be easier than solving for δ_i, t_i from n equations.

2 A Gauss-Newton algorithm

2.1 Calculating the Jacobian

We aim to solve (1.3) by using a Gauss-Newton method and, as before, we will try to stay on the constraint surface. This implies that δ_i is to be seen as a function of α implicitly defined by (1.2). Thus, we formulate our unconstrained problem as

$$(2.1) \quad \min_{\alpha} \|\delta(\alpha)\|_2^2$$

where $\delta = [\delta_1, \dots, \delta_m]^T$ and $\delta_i(\alpha)$ is defined by the equation

$$(2.2) \quad f(x_i, \alpha) = 0.$$

where

$$x_i = \tilde{x}_i + \delta_i v_i, \quad i = 1, \dots, m.$$

To derive a Gauss-Newton algorithm we need the Jacobian

$$J = \nabla_\alpha \delta.$$

Since (2.2) holds identically in α , we get

$$(2.3) \quad \nabla_{x_i} f \nabla_\alpha x_i + \nabla_\alpha f = 0.$$

Since $x_i = \tilde{x}_i + \delta_i v_i$ we have $\nabla_\alpha x_i = v_i \nabla_\alpha \delta_i$ which, inserted in (2.3), gives

$$(2.4) \quad \nabla_{x_i} f v_i \nabla_\alpha \delta_i + \nabla_\alpha f = 0.$$

Solving for $\nabla_\alpha \delta_i$ in (2.4) we get

$$(2.5) \quad \nabla_\alpha \delta_i = -(\nabla_{x_i} f v_i)^{-1} \nabla_\alpha f,$$

provided the inverse exists, and J follows using

$$J = \begin{bmatrix} \nabla_\alpha \delta_1 \\ \vdots \\ \nabla_\alpha \delta_m \end{bmatrix}.$$

THEOREM 2.1. *The Jacobian J exists at α if and only if, for each $i = 1, \dots, m$, v_i is not tangent to the surface at the point $x_i(\alpha) = \tilde{x}_i + \delta_i(\alpha)v_i$.*

PROOF. Since we have assumed differentiability of f we only need to assure that the inner products $\nabla_{x_i} f v_i$ in (2.5) are not zero. Since $\nabla_{x_i} f$ is the normal to the surface defined implicitly by $f(x_i, \alpha) = 0$, this is equivalent to v_i not perpendicular to $\nabla_{x_i} f$. $\square$

LEMMA 2.2. *When there is a parameterization of the surface as in (1.1), and the same parameter vector is used in (1.2), then the Jacobian matrix J is identical to that used in [13], [14], when both are defined.*

PROOF. Let a point on the surface $f(x, \alpha) = 0$ be parameterized by $x = x(t, \alpha)$. Then (dropping the subscript i),

$$\tilde{x} + \delta v = x(t, \alpha)$$

identically in α . Thus

$$v \nabla_\alpha \delta = \nabla_\alpha x + \nabla_t x \nabla_\alpha t,$$

so that

$$\nabla_\alpha \delta = e_1^T [v \quad -\nabla_t x]^{-1} \nabla_\alpha x.$$

Now

$$f(x(t, \alpha), \alpha) = 0,$$

identically in t and α . Therefore,

$$\nabla_x f \nabla_t x = 0,$$

in addition to (2.3). Now

$$\begin{aligned}\nabla_x f v e_1^T &= (\nabla_x f v, 0) \\ &= (\nabla_x f v, -\nabla_x f \nabla_t x) \\ &= \nabla_x f [v : -\nabla_t x].\end{aligned}$$

Therefore

$$\nabla_x f v e_1^T [v : -\nabla_t x]^{-1} = \nabla_x f,$$

or

$$e_1^T [v : -\nabla_t x]^{-1} = (\nabla_x f v)^{-1} \nabla_x f.$$

Thus

$$\begin{aligned}e_1^T [v : -\nabla_t x]^{-1} \nabla_\alpha x &= (\nabla_x f v)^{-1} \nabla_x f \nabla_\alpha x \\ &= -(\nabla_x f v)^{-1} \nabla_\alpha f,\end{aligned}$$

which completes the proof. $\square$

For v_i given from real measurements it is highly unlikely that v_i is tangent to the surface. It is perhaps more likely that v_i does not cut the surface at all, and that would require adjusting the parameter α . For a reasonably defined problem, and reasonably good starting points, this should not be necessary. Therefore, in practice we may assume that the Jacobian $\nabla_\alpha \delta$ exists throughout. However, a really robust method might be expected to be able to at least handle rank deficient Jacobians, which might occur anytime, but possibly far away from a solution. That could be done by incorporating a trust region into the calculation for the step length. Indeed this might be appropriate from other considerations. However, we will not consider such an algorithmic development here.

2.2 Algorithmic details

The main algorithm can be summerised as:

ALGORITHM 2.1.

Set the parameters Tol and Maxiter.

$k := 1$; $Close := \mathbf{false}$; $\beta_k := 10 Tol$

Initiate the start value for α .

while $\|J^T \delta\| > Tol$ **and** $k < Maxiter$

1. *Determine $\delta(\alpha)$ by solving $f(\tilde{x}_i + \delta_i v_i, \alpha) = 0$ for $\delta_i, i = 1, \dots, m$.*

2. *Determine the Jacobian J .*

3. *Compute the GN direction, $d = -J^\dagger \delta$.*

4. $\beta_{k+1} := \|Jd\|$; $Rate := \beta_{k+1}/\beta_k$

5. *If Close then*

$\gamma = 1$

else

Determine the step length γ .

end

6. $\alpha := \alpha + \gamma d$, $k := k + 1$; $Close := (0.5\beta_k < \|\delta(\alpha)\|)$

end

Let us make a few comments on Algorithm 2.1. Firstly, from Lemma 2.2, it follows that when a parameterization is available, and the same α is used, then the same Jacobian matrix is being used, and so the algorithms of [13], [14] generate identical directions of progress to those calculated here.

The determination of δ at step 1. is generally a nonlinear problem. In the case of the sphere and ellipse (see Section 3) δ_i is easily found by solving a second order algebraic equation. Considering convex surfaces there are two solutions if the line $l_i(s) = \tilde{x}_i + sv_i$ crosses the the surface and one if $l_i(s)$ is a tangent to the surface. If the sign of v_i is chosen such that v_i points in the direction from an outside measurement device towards the surface, we choose the smallest solution. That implies that $x_i = \tilde{x}_i + \delta_i v_i$ is the intersection between $l_i(s)$ and the surface that is closest to the measurement device. That makes sense in practical situations because, from the device's point of view, we can only touch the "front" of an object. It is of course possible to use other choices, e.g., the solution with smallest absolute value. This choice can however cause problems when measuring thin objects which are thinner than twice the magnitude of the measurement errors.

In step 4. we use β_k to measure the closeness to a local minimum, as in [4]. A first order estimate of the local convergence rate is attained by β_{k+1}/β_k , (see [5]) and we terminate the iterations if the size of the gradient is less than Tol .

The step length algorithm used in step 5. is a standard cubic interpolation scheme, as given in [3]. The merit function is

$$\Phi(\gamma) = \|\delta(\alpha + \gamma d)\|_2^2 / 2.$$

Note that in order to evaluate $\Phi(\gamma)$ we need to solve $f(\tilde{x}_i + \delta_i v_i, \alpha + \gamma d) = 0$ for δ_i . If $Close = 1$ we use a full step, $\gamma = 1$, since we are close to the solution: more details are given in [4].

3 Computational tests

In order to investigate the numerical performance of the algorithm, we consider fitting spheres, ellipses, and hyperquadric curves. Although the sphere and the ellipse have a readily available parametric form, we treat them in the form appropriate to the algorithmic ideas considered here.

3.1 The sphere in implicit form

A sphere is implicitly defined by the equation

$$f(x, \alpha) = \|x - c\|_2^2 - r^2 = 0,$$

where c is the centre and r is the radius. The centre and the radius are the unknown parameters and we may define

$$\alpha = \begin{bmatrix} r \\ c \end{bmatrix} \in \mathbb{R}^4.$$

The interesting derivatives of a sphere are

$$\nabla_{\alpha} f(x_i, \alpha) = -2 [r, (x_i - c)^T]$$

and

$$\nabla_{x_i} f(x_i, \alpha) = 2(x_i - c)^T,$$

which give the i 'th row in the Jacobian as (dropping the arguments of f)

$$\nabla_{\alpha} \delta_i(x_i, \alpha) = -(\nabla_{x_i} f v_i)^{-1} \nabla_{\alpha} f = \frac{[-(x_i - c)^T, r]}{(x_i - c)^T v_i}.$$

In this case we can easily see that v_i has to be non-tangent to the sphere if the Jacobian is to be well defined.

At every new iteration point we need to compute δ_i by solving

$$\|\tilde{x}_i + \delta_i v_i - c\|_2^2 - r^2 = 0.$$

By introducing $y_i = \tilde{x}_i - c$ the roots are given by

$$\delta_i = -y_i^T v_i \pm \sqrt{(y_i^T v_i)^2 - (y_i^T y_i - r^2)}.$$

The roots are always real if the line $l_i(s) = \tilde{x}_i + s v_i$ crosses the sphere at any point. According to Sec. 2.2 we use the smallest root.

3.2 The ellipse in implicit form

The ellipse is defined by the quadratic equation

$$f(x, \alpha) = (x - c)^T A (x - c) - 1 = 0,$$

where

$$A = L L^T = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} \\ 0 & l_{22} \end{bmatrix},$$

is positive definite and c is the centre of the ellipse. The axis and rotation of the ellipse can easily be found by the eigenvalues and eigenvectors of A . The unknown parameters are defined as $\alpha = [l_{11}, l_{21}, l_{22}, c^T]^T \in \mathbb{R}^5$.

As for the sphere, the i 'th row in the Jacobian follows from (2.5), where (here taking x_1 and x_2 to be the components of x),

$$\begin{aligned} \nabla_{\alpha} f(x, \alpha) &= 2 [(x_1 - c_1)^2 l_{11} + (x_1 - c_1)(x_2 - c_2) l_{21}, \\ &\quad (x_1 - c_1)(x_2 - c_2) l_{11} + (x_2 - c_2)^2 l_{21}, \\ &\quad -2a_{11}(x_1 - c_1) - 2a_{12}(x_2 - c_2), (x_2 - c_2)^2 l_{22}, -(x - c)^T A], \\ \nabla_x f(x, \alpha) &= 2(x - c)^T A. \end{aligned}$$

We find δ_i by solving the equation

$$(\tilde{x}_i + \delta_i v_i - c)^T A (\tilde{x}_i + \delta_i v_i - c)^T = 1,$$

or by using $y_i = \tilde{x}_i - c$

$$\delta_i^2 (v_i^T A v_i) + 2(y_i^T A v_i) \delta_i + y_i^T A y_i - 1 = 0.$$

As for the sphere we always choose the smallest root.

3.3 A convex hyperquadric curve

A convex hyperquadric curve with two terms can implicitly be defined as

$$f(x, \alpha) = |a^T x + b|^{e^2+1} + |c^T x + d|^{f^2+1} - 1 = 0,$$

see, e.g., [2]. This is a curve where a parametrization is not readily available. The representation of the exponents, $e^2 + 1 \geq 1$ and $f^2 + 1 \geq 1$, forces the curve to be convex. We define the unknown parameter vector as

$$\alpha = [a^T, b, c^T, d, e, f]^T \in \mathbb{R}^8.$$

The above form does not imply a unique representation for a particular curve. For example, the parameter vectors

$$\begin{aligned} \alpha_A &= [1, 0, 0, 0, 1, 0, \infty, \infty]^T \quad \text{and} \\ \alpha_B &= [1/2, -1/2, 0, 1/2, 1/2, 0, 0, 0]^T \end{aligned}$$

both represent the same square with sides $x = \pm 1$, $y = \pm 1$.

Introducing $s_1 = a^T x + b$ and $s_2 = c^T x + d$, the derivatives needed to compute the Jacobian are

$$\begin{aligned} \nabla_\alpha f(x, \alpha) &= 2 [\text{sign}(s_1)(e^2 + 1)|s_1|^{e^2} x^T, \text{sign}(s_1)(e^2 + 1)|s_1|^{e^2}, \\ &\quad \text{sign}(s_2)(f^2 + 1)|s_2|^{f^2} x^T, \text{sign}(s_2)(f^2 + 1)|s_2|^{f^2}, \\ &\quad 2e|s_1|^{e^2+1} \log(|s_1|), 2f|s_2|^{f^2+1} \log(|s_2|)] \\ \nabla_x f(x, \alpha) &= \text{sign}(s_1)(e^2 + 1)s_1^{e^2} a^T + \text{sign}(s_2)(f^2 + 1)s_2^{f^2} c^T. \end{aligned}$$

For each $i = 1, \dots, m$, we find δ_i by solving the equation

$$(3.1) \quad g(\delta_i) = f(\tilde{x}_i + \delta_i v_i, \alpha) = 0,$$

using the standard Newton iteration

$$\delta_i^{(k)} = \delta_i^{(k-1)} + g(\delta_i^{(k-1)})/g'(\delta_i^{(k-1)}).$$

The derivative

$$g'(\delta_i) = \nabla_{x_i} f(\tilde{x}_i + \delta_i v_i, \alpha) v_i$$

is nonzero provided that v_i is not tangent to the hyperquadric curve at the point $x_i = \tilde{x}_i + \delta_i v_i$. The function $g(\delta_i)$ is convex and there exist two solutions to (3.1) if the line $l_i(s) = \tilde{x}_i + s v_i$ crosses the hyperquadric curve. Since we want to compute the smallest solution (see Sect. 2.2), we start the iteration with $\delta_i^{(0)}$ satisfying $g(\delta_i^{(0)}) \geq 0$ and $g'(\delta_i^{(0)}) < 0$. That makes the Newton iteration converge to the smallest solution monotonically from the left. The situation is analogous to applying the Newton iteration to the equation $z^2 - 1 = 0$, and starting with $z^{(0)} \leq -1$.

3.4 Test results

Here we present some results for fitting a sphere, an ellipse, and a hyperquadric curve to some different sets of data. Throughout the tests we have used Algorithm 2.1 with the stopping criterion $Tol = 0.0001$. The results of the tests are summarized in Table 3.1 and in Figures 3.1-3.5.

For the ellipse we have used three data sets. These data sets were used in [13], were named Späth, triangle and GGS, and are exactly as given there. They are plotted in Figures 3.1-3.3 together with the initial ellipses and the corresponding solution ellipses. It is assumed here that the initial ellipses are chosen such that the situation with at least one of the lines $l_i(s) = \tilde{x}_i + sv_i$ not crossing the ellipse does not occur. In [13], this was not assumed because failure of the directions to cut the initial ellipse triggered off a strategy for adjustment of the parameters to expand the ellipse to remove the problem.

When the axes of the ellipse are not aligned with the coordinate axes, then the algorithm of [13] works with rotated data, rather than incorporating the rotation into the model. However, no corresponding rotation was incorporated into the directions v_i , and this means that the results are not directly comparable. If the algorithm of [13] is modified to include rotation of the directions v_i , then the same problems are being solved, although the algorithms are not equivalent because the vectors of parameters used are different. Indeed it turns out that rather good initial approximations are needed for adequate performance of the method in that case, and the method given here is superior for these examples. This reflects the fact that working with the parameter vector α as here seems easier than working with a vector of parameters corresponding to the centre and semi-axes of the ellipse, augmented by a rotation angle.

The data sets for the sphere and the hyperquadric curve were created in order to simulate real measurement data. That means that for given directions $v_1, \dots, v_{16}$, the points $\tilde{x}_i$, $i = 1, \dots, 16$, were generated as $\tilde{x}_i = x_i + \varepsilon_i v_i$, where $\varepsilon_i \sim N(0, 0.5)$ and each point x_i satisfies $f(x_i, \alpha) = 0$ for a particular value of α .

The measurement directions used when fitting the hyperquadric were

$$v_i = (p_i - x_i) / \|p_i - x_i\|_2, \quad i = 1, \dots, 16,$$

where

$$\begin{aligned} p_1 = \dots = p_4 &= [7.5, -2.5]^T, & p_5 = \dots = p_8 &= [-7.5, 12]^T, \\ p_9 = \dots = p_{12} &= [-7.5, -2.5]^T, & p_{13} = \dots = p_{16} &= [7.5, 12]^T, \end{aligned}$$

represent four different positions of the measurement device, see Figure 3.4. Experience with fitting hyperquadrics has shown that in order to avoid situations with non-intersecting directions (see the discussion about ellipses above) it is very important to use good initial values for the parameters. This is particularly true when fitting to data sets with square-like shapes. Such data sets make the Jacobian matrices become ill-conditioned.

For the sphere, we generated the data using the measurement directions $v_i = \pm[0, 0, 1]^T$, $i = 1, \dots, 16$. The sign depends on whether the point x_i is on the

Table 3.1: Test results for different fitting examples

object(data set)	Initial α^T Final α^T	$\sum \delta_i^2$	iter
ellipse(Spth)	[1/4, 0, 1/4, 3, 3] [0.347, -0.011, 0.205, 5.166, 4.689]	3.355	19
ellipse(triangle)	[1/3, 0, 1/3, 3, 2] [0.609, 0.145, 1.104, 3.073, 1.677]	1.604	10
ellipse(GGS)	[1/5, 0, 1/5, 5, 3] [0.217, -0.098, 0.317, 3.999, 4.400]	1.546	17
hyperquadric	[0.14, 0, -0.2, 0, 0.21, -0.9, 0.8, 0.7] [0.20, 0.00, -0.10, -0.00, 0.30, 0.16, 0.68, 0.40]	0.471	22
sphere(z-dir)	[6, 1.2, 2.1, 3.2] [4.1330, 99321.956, 2.833]	2.864	4
sphere(orth-dir)	[6, 1.2, 2.1, 3.2] [4.175, 0.891, 2.010, 2.780]	2.341	4

higher(-) or lower(+) part of the sphere. Points x_i that were close to the equator of the sphere were avoided. This test case is labeled “sphere(z-dir)” in Table 3.1.

With the sphere we have also made a comparison between the effect of using these measurement directions and ones which are orthogonal to the correct physical sphere. The results are then likely to be similar to the use of orthogonal distance regression. Therefore, using the same measurement points as above, with measurement errors only in the z-direction, we fit the sphere using directions $v_i = (x_i - c)/r, i = 1, \dots, 16$. The case is labeled “sphere(orth-dir)” in Table 3.1.

The correct physical sphere has the parameters $\hat{\alpha} = [4, 1, 2, 3]^T$. By computing the vectors of the moduli of the differences between components, that is

$$\begin{aligned} |\alpha_{\text{orth-dir}} - \hat{\alpha}| &= [0.1751, 0.1089, 0.0105, 0.2201]^T, \\ |\alpha_{\text{z-dir}} - \hat{\alpha}| &= [0.1330, 0.0068, 0.0444, 0.1668]^T, \end{aligned}$$

we can observe that all but the third of the parameters are closer to $\hat{\alpha}$ when using the first set of error directions compared to when using directions orthogonal to the correct physical sphere. Of course the square sum of residuals, $\sum \delta_i^2$, reduces as the directions v_i tend to the orthogonal directions.

Figure 3.1: Ellipse fitting to Späth data

Figure 3.2: Ellipse fitting to triangle data

Figure 3.3: Ellipse fitting to GGS data

Figure 3.4: Fitting of a hyperquadric curve

Figure 3.5: A sphere fitted to the (z-dir) data

4 Conclusions

We have developed an algorithm for fitting implicitly defined surfaces to measured data, using the least squares norm, which takes specific account of the probe directions used to generate the data. The process is a natural generalization of a method given for parametrically defined surfaces. However, as well as allowing greater generality in the kind of surfaces being fitted, it also has other advantages. For example, the absence of the extra position parameters defining the points can make the calculation of the Jacobian matrix simpler. In addition, the orientation of the surface does not need special treatment and there is evidence that this can lead to computational savings.

REFERENCES

1. H. I. Britt and R. H. Luecke, *The estimation of parameters in nonlinear, implicit models*, Technometrics, Vol.15, 233–247, 1973.
2. I. Cohen and L. Cohen, *A hybrid hyperquadric model for 2-D and 3-D data fitting*, Computer Vision and Image Understanding, Vol. 63, 527–541, 1996.
3. J. E. Dennis and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice Hall, 1983.
4. M. E. Gulliksson and I. Söderkvist, *Surface fitting and parameter estimation with nonlinear least squares*, Opt. Meth. and Soft., Vol. 5, 247–269, 1995.

5. M. E. Gulliksson, I. Söderkvist and P.-Å. Wedin, *Algorithms for constrained and weighted nonlinear least squares*, SIAM J. Opt., Vol. 7, No. 1, 208–224, 1997.
6. F. L. Bookstein, *Fitting conic sections to scattered data*, Computer Graphics and Image Processing Vol. 9, 56–71, 1979.
7. F. L. Hulting, Discussion contribution to the paper by M M Dowling, P M Griffin, K-L Tsui and C Zhou, *Statistical issues in geometric feature inspection using coordinate measuring machines*, Technometrics, Vol. 39, 18–20, 1997.
8. V. Pratt, *Direct least-squares fitting of algebraic surfaces*, Computer Graphics, Vol. 21, 145–152, 1987.
9. D. M. Sourlier, *Three-dimensional Feature-Independent Bestfit in Coordinate Metrology*, PhD Dissertation, Swiss Federal Institute of Technology Zurich, 1995.
10. R. Strebel, D. Sourlier and W. Gander, *A comparison of orthogonal least squares fitting in coordinate metrology*, in Recent Advances in Total Least Squares and Errors-in-Variables Techniques, ed S. Van Huffel, SIAM, Philadelphia, 249–258, 1997.
11. D. A. Turner, *The Approximation of Cartesian Co-ordinate Data by Parametric Orthogonal Distance Regression*, PhD Thesis, University of Huddersfield, 1999.
12. D. A. Turner, I. J. Anderson, J. C. Mason, M. G. Cox and A. B. Forbes, *An efficient separation-of-variables approach to parametric orthogonal distance regression*, in Advanced Mathematical and Computational Tools in Metrology IV, eds P. Ciarlini, A. B. Forbes, F. Pavese and D. Richter, Series on Advances in Mathematics for Applied Sciences, Volume 53, World Scientific, Singapore, 246–255 (2000).
13. G. A. Watson, *Least squares fitting of circles and ellipses to measured data*, BIT, Vol. 39, 176–191, 1999.
14. G. A. Watson, *Least squares fitting of parametric surfaces to measured data*, University of Dundee Technical Report NA/195 (1999).
15. G. A. Watson, *Some robust methods for fitting parametrically defined curves or surfaces to measured data*, in Advanced Mathematical and Computational Tools in Metrology IV, eds P. Ciarlini, A. B. Forbes, F. Pavese and D. Richter, Series on Advances in Mathematics for Applied Sciences, Volume 53, World Scientific, Singapore, 256–272 (2000).
16. D. S. Zwick, *Applications of orthogonal distance regression in metrology*, in Recent Advances in Total Least Squares and Errors-in-Variables Techniques, ed S. Van Huffel, SIAM, Philadelphia, 1997.