



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>
Handle ID: <http://hdl.handle.net/10985/8647>

To cite this version :

Nadim EL HAYEK, Hichem NOUIRA, Nabil ANWER, Mohamed DAMAK, Olivier GIBARU - A new method for aspherical surface fitting with large-volume datasets - Precision Engineering - Vol. 38, n°4, p.935-947 - 2014

Any correspondence concerning this service should be sent to the repository

Administrator : scienceouverte@ensam.eu



A new method for aspherical surface fitting with large-volume datasets

N. El-Hayek^{a,b,*}, H. Nouria^a, N. Anwer^c, O. Gibaru^b, M. Damak^{b,d}

^a Laboratoire Commun de Métrologie (LCM), Laboratoire National de Métrologie et d'Essais (LNE), 1 Rue Gaston Boissier, 75015 Paris, France

^b Arts et Métiers de Lille (ENSAM), Laboratory of Information Sciences and Systems (LSIS), 8 Boulevard Louis XIV, 59046 Lille, France

^c ENS de Cachan, The University Research Laboratory in Automated Production, 61 Avenue du Président Wilson, 94235 Cachan, France

^d GEOMNIA, 3D Metrology Engineering and Software Solutions, 165 Avenue de Bretagne, EuraTechnologies, 59000 Lille, France

A B S T R A C T

In the framework of form characterization of aspherical surfaces, European National Metrology Institutes (NMIs) have been developing ultra-high precision machines having the ability to measure aspherical lenses with an uncertainty of few tens of nanometers. The fitting of the acquired aspherical datasets onto their corresponding theoretical model should be achieved at the same level of precision. In this article, three fitting algorithms are investigated: the Limited memory-Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), the Levenberg-Marquardt (LM) and one variant of the Iterative Closest Point (ICP). They are assessed based on their capacities to converge relatively fast to achieve a nanometric level of accuracy, to manage a large volume of data and to be robust to the position of the data with respect to the model. Nevertheless, the algorithms are first evaluated on simulated datasets and their performances are studied. The comparison of these algorithms is extended on measured datasets of an aspherical lens. The results validate the newly used method for the fitting of aspherical surfaces and reveal that it is well adapted, faster and less complex than the LM or ICP methods.

Keywords:

Aspherical surface fitting

Form metrology

Large data

Limited memory BFGS

1. Introduction

Due to their superior performance, aspherical and other freeform optics have replaced spherical components in almost all optical systems that exist today. This trend of extensive use of complex optical elements is found in a wide range of applications, such as, digital cameras, medical laser systems, ophthalmologic and endoscopic systems, multi-media projectors, car head-up displays, astronomical telescopes and optical microscopes. However, a key issue of these optical elements is their surface quality, especially form which is crucial for the optical performance and functionality. Therefore, it is a must to track form deviations all the way from the design phase up to the operational phase and evaluate form defects owing to manufacturing. With the advance of technology and ultra-precise manufacturing capabilities [1,2], design

specifications have become more and more challenging, reaching a nanometric scale of precision. While current techniques allow for manufacturing arbitrary optical surfaces and provide correction at the nanometer scale, high precision measurement of optical surfaces as well as data processing is still a challenge in industry.

In this regard, a three-year project has been launched by the European Metrology Research Program (EMRP) and encompasses a multitude of European National Metrology Institutes (NMIs), industrial partners and academic partners [3]. It aims at developing methods for high-precision measurement of aspherical optics and characterizing their surface form. A recent and exhaustive research on the processing and manufacturing of aspherical and freeform optics is found in [1]. The LNE's high-precision profilometer, the Zeiss F25, the μ CMM, the NANOMEFOS and the ISARA400 measuring machines achieve ultra-high-precision positioning and measurement on aspherical artifacts as described in [4]. These machines respect the Abbe principle, apply the dissociated metrology structure principle and incorporate high-precision mechanical guiding systems. Motions are performed under the control of sub-nanometric-resolution laser interferometers which are traceable to the SI meter definition and positioned and aligned such that they satisfy the Abbe principle. The thermal expansion of

* Corresponding author at: Laboratoire Commun de Métrologie (LCM), Laboratoire National de Métrologie et d'Essais (LNE), 1 Rue Gaston Boissier, 75015 Paris, France. Tel.: +33 642582545.

E-mail addresses: nadimhayek@gmail.com (N. El-Hayek), hichem.nouria@ine.fr (H. Nouria).

the selected materials is negligible firstly because the surrounding temperature is generally controlled within a range of $20 \pm 0.2^\circ\text{C}$ and secondly because all the major parts of the machines are made of Invar and Zerodur having a very low thermal expansion coefficient. The performance of these machines depends on both the performance of the scanning probe and the stability of the metrology loop.

Once the measurements are performed and the data are recorded, the aspherical form should be evaluated by fitting the appropriate model to the measured data with a nanometric accuracy. Many fitting techniques exist in literature, but very few discuss the fitting of aspheres. Chen et al. [5] propose an aspherical lens characterization by means of a 2D profile fitting. The dataset used is a profile measured using stylus profilometry and the reference model is the corresponding asphere profile. The fitting is done using the Levenberg–Marquardt (LM) algorithm for its quick convergence in regard to ultra-precise examination [6]. Similar works have been published and also deal with aspherical profile identification and conic sections fitting [7,8]. Sun et al. [9] perform fitting of aspherical curves and surfaces on simulated data with vertical distance minimization using a Gauss–Newton algorithm. In fact, they assume that the model and the data are both defined in the same reference frame. Other works involve approximating aspheres with NURBS surfaces in order to generate CAD models for manufacturing purposes [10].

A different, yet classical, approach involves the Iterative Closest Point (ICP) algorithm which is a registration algorithm that finds a spatial transformation to align different point-sets [11]. It is designed for discrete data, making it a reference algorithm with negligible storage. It can be used for fitting applications when the reference element is a point-set model or a mesh model. The algorithm relies on a point matching operation between the dataset and the reference model and tries to minimize the distance separating the matched points. This operation is computationally expensive with the classical ICP and that is why efficient variants of the algorithm were developed [12]. ICP is very sensitive to initial alignment between the dataset and the model, and tends to converge to a local minimum because the optimization is of non-convex nature [13].

The Limited memory–Broyden–Fletcher–Goldfarb–Shanno (L-BFGS), introduced by Jorge Nocedal, is a quasi-Newton method for solving unconstrained non-linear problems [14]. Recent works involving this algorithm have published promising results. Zheng et al. [15] show that, unlike classical methods, the L-BFGS algorithm is robust and fast pertaining to the fitting of geometric B-Spline curves to points sampled from freeform shapes.

Until this date, no particular work has given interest to the fitting of large datasets, except in the case of registration methods based on ICP which is an algorithm that uses discrete elements [16]. This paper provides a comparison of L-BFGS, LM and one particular variant of the ICP algorithm. The methods are all based on the same sequential algorithmic structure. They iteratively compute, in a nested loop, the footpoint of each data point and then minimize the sum of the squared distances between the data and the reference model. A footpoint is the projection of a data point onto the reference model and can either be an orthogonal projection or a vertical projection. The algorithms, and especially the newly used L-BFGS, are evaluated in conformance with ISO 10360-Part 6 [17]. This standard describes the procedure for the evaluation of software that performs a characterization of the residual errors and form deviations of measured data for canonical features. The algorithms should therefore be tested on simulated datasets (D_m) to which systematic and random errors are added. Finally, the algorithms are compared on measured datasets of an aspherical lens using the LNE’s high precision profilometer when it is equipped with a tactile probe. The obtained results are reported and discussed.

2. Preliminaries

The form evaluation of aspheres can be done by performing the association of the aspherical model to the measured data. This process, defined in ISO 17450-Part 1 [18], is the operation of fitting an ideal feature (the model) to a non-ideal feature (the data points) according to a criterion (such as Least-Squares). It is usually done by optimizing for transformation parameters and model parameters, but, in this paper, only transformation parameters are calculated since the model parameters are known. Three fitting algorithms are compared based on their capacities to converge quickly, to manage a large volume of data and to be robust to the initial position of the dataset with respect to the model. The process goes by iteratively optimizing for five transformation parameters, the rotation about z being redundant here. Reference models such as a point-set model or a mesh model can be used when the problem needs to be expressed in discrete form. In these two cases, the very well-known ICP algorithm is used and distances are calculated on a point-to-point basis and on a point-to-triangle basis, respectively. Otherwise, the reference model is a mathematical equation and distances are calculated on a point projection basis. The evoked fitting algorithms follow the same structure and sequentially compute for footpoints and transformation parameters (\mathbf{R} : rotation matrix, \mathbf{T} : translation vector). After being evaluated on simulated datasets, the algorithms are tested on measured surfaces with large volumes of data. Aspherical surface fitting onto simulated small-sized datasets can be achieved by applying a Gauss–Newton optimization with vertical distance minimization [9]. However, the usage of vertical distance is limited to cases where both entities to fit are defined in the same reference frame. Nonetheless, orthogonal distance calculation guarantees the freedom to have the measured dataset and the model in different coordinate systems [19].

2.1. The aspherical model

The mathematical formulation of aspherical surfaces is detailed and standardized in ISO 10110-Part 12 [20]. In our case, the aspherical form to be tested $F(r, z; c, k, \mathbf{a})$ is an axis-symmetric surface defined in implicit form such as in Eq. (1)

$$F(r, z; c, k, \mathbf{a}) = z - f(r) = 0, \quad (1)$$

$r = \sqrt{x^2 + y^2}$ and z are the Cartesian coordinates with a change of variables applied on x and y ; c is the curvature at the apex, κ is the conic constant, $\mathbf{a} = (a_2, a_4, a_6, a_8, a_{10})$ is the vector of the 10th-order aspherical deviation parameters. For a given r , the height in z is independent of the angle $\theta = \tan^{-1}(y/x)$, and $f(r)$ can be written as in Eq. (2):

$$f(r) = - \left(\frac{cr^2}{1 + \sqrt{1 - (1 + \kappa)c^2r^2}} + \sum_{j=1}^5 a_{2j}r^{2j} \right). \quad (2)$$

For any $(r, z) \in \mathbb{R}^2$, $F(r, z; c, k, \mathbf{a})$ is twice differentiable. The aspherical model considered here is not exactly expressed as in ISO 10110-Part 12 [20] due to the presence of the a_2 term. In this ISO standard, a_2 is mingled with the curvature parameter c . The coordinate system is defined such that the origin coincides with the apex of the asphere and the z -axis is always oriented upwards. The parameters of the asphere are then defined accordingly. For the considered model, the curvature at the apex is positive and equal to $c = 10^{-20} \text{ mm}^{-1}$, the conic constant $\kappa = -1$ (Fig. 1) and the asphere parameters are $\alpha_2 = 0.0223$, $\alpha_4 = 7.293 \times 10^{-6}$, $\alpha_6 = 4.52 \times 10^{-9}$, $\alpha_8 = -1.061 \times 10^{-11}$, $\alpha_{10} = 9.887 \times 10^{-15}$. The asphere has a sag $S = 3.217 \text{ mm}$ of and a clear aperture $CA = 11.74 \text{ mm}$. This lens was manufactured by a Single Point Diamond Turning (SPDT) process

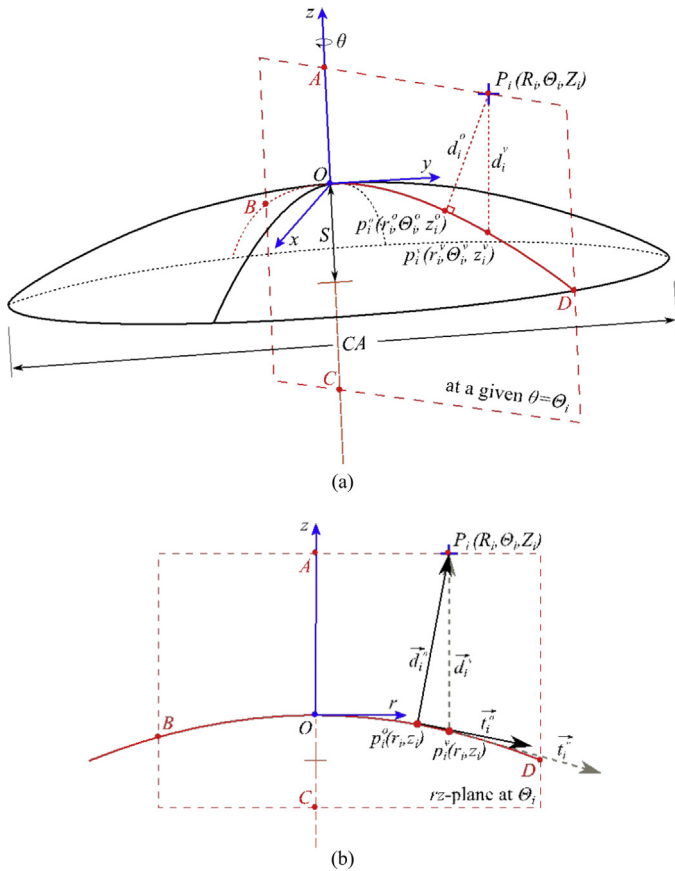


Fig. 1. The orthogonal/vertical distance vector from a data point P_i to its footpoint p_i^o/p_i^v on an asphere is contained in the rz -plane at an angle Θ_i passing through the axis of symmetry (all red points are coplanar). (a) a 3D view of the asphere and the data point projection in the given rz -plane (red); (b) a planar view of the distance projection where \vec{d}_i^o and \vec{d}_i^v are the distance vectors and \vec{t}_i^o and \vec{t}_i^v denote the tangent vectors.

[21] and finished with a high precision polishing process and glass coating.

2.2. Optimization algorithms

The Newton–Raphson method [22] is used in many optimization problems that are not highly non-linear or complex. It is therefore used here for the computation of footpoints which are required for the orthogonal distance minimization problem. At every iteration, the method which has a simple algorithmic structure aims at finding an approximation of the roots of a real-valued function $S : \mathbb{R}^3 \rightarrow \mathbb{R}$. The goodness of the approximation depends on the stop criterion and on the quality of the initial guess (relative position of the data and the model should be close to the optimal solution) [22,23]. If the initial guess is close enough and the derivative of S exists everywhere, the algorithm guarantees convergence to the roots of S . For the problem of aspheres which have low curvatures, the vertical projection point p_i^v is taken as an initial guess and is actually close to the orthogonal projection point p_i^o . Then, the Newton–Raphson method iterates until p_i^o is accurately approximated (stop criterion: $x_{i+1} - x_i \leq \varepsilon_1 = 10^{-16}$).

As for the minimization of the non-linear squared distances problem, d_i^o for orthogonal and d_i^v for vertical, it is dealt with using other optimization algorithms such as L-BFGS, LM and ICP. LM [6,24] is a well-known optimization algorithm that is based on an interpolation between a Gauss-Newton approach and the gradient descent (Eq. (3)). It has been evaluated and approved by the National Institute of Standards and Technology (NIST) for

metrology applications that require fitting simple curves and surfaces in 3D [25]. This algorithm converges reasonably quickly and accurately for a wide range of initial guesses that are close to the optimal solution [6]

$$x_{j+1} = x_j - (H(x_j) + \lambda_j \cdot \text{diag}[H(x_j)])^{-1} \cdot \nabla f_{obj}(x_j), \quad (3)$$

where x_j is the solution vector at iteration j , H_j is the Hessian matrix at iteration j , $\text{diag}[H(x_j)]$ is the diagonal elements of the Hessian and ∇f_{obj} is the gradient of the objective function. λ is the parameter that can be assimilated to the step coefficient of the gradient descent. λ changes at each iteration such as when it is large, the gradient descent predominates the optimization process and when it is small, Gauss-Newton predominates. Shakarji defines a procedure for choosing an initial value for λ and then updating it at each iteration [25]. In an application to freeform surfaces, Jiang et al. [13] propose another way of determining the value of λ based on the smallest singular value of the Hessian matrix.

In [13] a method to fit freeform surfaces to small numbers of data in two steps is described. A coarse fitting is firstly performed by using a structured region signature procedure and secondly, a LM optimization is applied for fine fitting in which a Jacobian matrix J needs to be calculated. The LM algorithm further necessitates the computation of the Hessian matrix $H = J^T J$ by linear approximation and its inverse at each iteration. Although matrix inversion is taken care of by some efficient pseudo-inverse techniques such as SVD or QR decomposition, it is still the major issue of this kind of algorithms. The complexity of matrix inversion is $O(n^3)$ with n being the number of variables in the system. For a large n , the cost of inverting H is considerable.

For a very large number of variables or unconstrained non-linear problems, iterative quasi-Newton methods such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method can be more convenient [26]. Like any minimization algorithm, BFGS preferably requires a twice-differentiable objective function f_{obj} whose gradient must be zero at optimality. The method approximates the inverse Hessian of the function by cumulating information from previous iterations, therefore, a sequence of matrices is constructed throughout. This sequence occupies a very large memory space which eventually comes to saturation when all the matrices are stored [15,27]. Subsequently, Nocedal describes an improved method called L-BFGS which keeps updating the Hessian matrix using a limited amount of storage [14]. At every iteration, the Hessian is approximated using information from the last m iterations with each time, the new approximation replacing the oldest one in the queue. L-BFGS is an enhanced BFGS optimization algorithm for reducing memory usage when storage is critical and is suitable for applications involving large volumes of data and variables. Furthermore, Zheng et al. [15] propose a L-BFGS algorithm to perform B-Spline curve fitting and show that, unlike traditional methods, L-BFGS can perform optimization of control points and footpoints simultaneously. Additionally, neither formulating nor solving linear equations is needed, making the algorithm very efficient and faster than other methods such as the one in [28]. Zheng et al. [15] have also studied the complexity of the algorithm and showed that it is linear in the number of data.

The L-BFGS algorithm goes as in Eq. (4). Consider the problem in which the objective function f_{obj} has gradient g and Hessian H and returns a scalar value, $\phi(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$

$$f_{obj} = \min_x (\phi(x)). \quad (4)$$

2.2.1. Initialization

The initialization consists of 4 main settings: (1) make an initial guess x_0 , (2) choose m for the number of iterations to be considered for the inverse Hessian update (preferably $m > 5$), (3) set two real

numbers β' and β such that $0 < \beta' < \frac{1}{2}$ and $\beta' < \beta < 1$ and (4) choose a symmetric, positive definite starting inverse Hessian matrix H_0^{-1} .

2.2.2. Iterations

Perform a *line search* routine to compute the quasi-Newton direction. By being positive definite, the Hessian ensures that the optimization direction d_j is a descent direction (Eq. (5))

$$d_j = -H_{j-1}^{-1} \Delta f(x_{j-1}) = -H_{j-1}^{-1} g(x_{j-1}). \quad (5)$$

Determine the *step size* α (for example by backtracking line search). α_j should satisfy the Wolfe conditions (Eq. (6)) at each iteration. β' and β are parameters to control the accuracy of the line search routine. β' defines a tolerance on the function f_{obj} and β a tolerance on its gradient g

$$\begin{aligned} f_{obj}(x_j + \alpha_j \cdot d_j) &\leq f_{obj}(x_j) + \beta' \alpha_j \cdot g(x_j)^T \cdot d_j. \\ g(x_j + \alpha_j \cdot d_j)^T d_j &\geq \beta g(x_j)^T \cdot d_j. \end{aligned} \quad (6)$$

The Wolfe conditions along with exact *line search* routine require a large number of function and gradient evaluations. Nevertheless, inexact search allows to determine a step length α at minimal cost while adequately reducing the f_{obj} and making a reasonable progress in minimization. Then the update rule for the solution x is indicated in Eq. (7)

$$x_{j+1} = x_j + \alpha_j d_j. \quad (7)$$

2.2.3. Update H_j^{-1}

The inverse Hessian at step $j+1$ is the inverse Hessian at step j plus a certain variation Δh which is expressed in function of the change in the variables Δx_j , and the change in the gradient Δg_j (Eq. (8)). Due to the iterative process, an inverse Hessian matrix at an iteration m can be written in function of H_0^{-1} and the vectors Δx_j and Δg_j of all iterations up to m only. The update becomes as follows:

$$\begin{aligned} H_m^{-1} &= H_0^{-1} + \Delta h(\Delta x_0, \Delta g_0, H_0^{-1}) + \Delta h(\Delta x_1, \Delta g_1, H_1^{-1}) + \dots \\ &+ \Delta h(\Delta x_{m-1}, \Delta g_{m-1}, H_{m-1}^{-1}) \end{aligned} \quad (8)$$

The complexity of the update for L-BFGS is of the order of $O(nm)$ as compared to its predecessor BFGS which complexity is $O(n^2)$.

3. Evaluation and comparison of fitting algorithms

The newly used L-BFGS as well as the LM fitting algorithms and ICP, are evaluated in by referring to the standards defined in ISO 10360-Part 6 [17] and the works provided by Lin et al. [29]. Different tests are done on simulated datasets which are derived from the theoretical model equation of the asphere and modified by means of the addition of Gaussian noise and form deviations.

The fitting process, illustrated in Fig. 2, is the process in which the model parameters (c , κ and \mathbf{a}) are fixed and the only variables to optimize are the transformation parameters often called motion parameters. Since the aspherical model is axis-symmetric, the rotation about z is a redundant degree of freedom, thus, only five motion parameters out of six are required to determine the best fit of the dataset with respect to the model: a rotation matrix \mathbf{R}_β about the x -axis, a rotation matrix \mathbf{R}_γ about the y -axis and three translations t_x , t_y and t_z in x -, y - and z - directions, respectively.

The objective function f_{obj} to minimize is the sum of the squared distances between the data points $\mathbf{P}_i(X_i, Y_i, Z_i)$ and their respective footpoints $\mathbf{p}_i(x_i, y_i, z_i)$ on the model. Eq. (9) is an expression of this

functional in which $\mathbf{T} = (t_x, t_y, t_z)$ denotes the translation vector and β and γ are the rotation angles about x and y , respectively:

$$f_{obj} = (\beta, \gamma, \mathbf{T}) = \min_{\beta, \gamma, t_x, t_y, t_z} \sum_{i=1}^N \left\| \mathbf{p}_i - (\mathbf{R}_{\beta, \gamma} \cdot \mathbf{P}_i + \mathbf{T}) \right\|^2, \quad (9)$$

N being the number of points in the dataset and $\mathbf{R}_{\beta, \gamma} = \mathbf{R}_\beta \mathbf{R}_\gamma$ the combined rotation matrix about x and y .

L-BFGS, LM and ICP optimization algorithms are compared based on whether they return the same residual errors (RMS and PV) and corresponding motion parameters as the simulated ones. Orthogonal distance-based fitting and vertical distance-based fitting are also studied.

3.1. Implementation of the L-BFGS and LM algorithms

L-BFGS and LM algorithms are implemented in sequential computation of footpoints and transformation parameters. Vertical footpoints are the direct projection of each corresponding data point. However, orthogonal footpoints need to be calculated in a nested minimization loop within the main transformation parameters minimization loop using the Newton-Raphson method (Fig. 1). The asphere is an axis-symmetric surface that only depends on the variable $r = \sqrt{x^2 + y^2}$. Finding the 3D orthogonal projection of a data point can be seen as a 2D problem. In fact, the rz -plane that contains the data point to which a projection point must be found, is oriented by an angle θ_i , and definitely contains the projection point as well as the axis of symmetry (Fig. 1). Usually, the projection of a point \mathbf{P}_i on a surface requires the optimization of two parameters, here r_i and θ_i . But, since the model of the asphere (Eq. (1)) is independent of the θ -coordinate, θ_i is directly determined from the corresponding data point provided that it is expressed in its cylindrical coordinates.

Each Cartesian data point $\mathbf{P}_i^{Car}(X_i, Y_i, Z_i)$ is written in cylindrical coordinates $\mathbf{P}_i^{Cyl}(R_i = \sqrt{X_i^2 + Y_i^2}, \Theta_i = \tan^{-1}(Y_i/X_i), Z_i)$. Then, as shown in Fig. 1b, the footpoint \mathbf{p}_i^v (vertical) or \mathbf{p}_i^o (orthogonal) is computed in the selected rz -plane at $\theta_i = \Theta_i$.

For the vertical distance projection problem, the r -coordinate of the vertical footpoint \mathbf{p}_i^v (Fig. 1) is taken to be that of the data point, $r_i^v = R_i$. The z -coordinate $z_i^v = f(r_i^v)$ is calculated following Eq. (2).

For the orthogonal distance projection problem, the r -coordinate of the footpoint is initially taken to be that of the data point, $r_{ini}^o = R_i$. Then, the Newton-Raphson method iteratively optimizes r_i^o , and $z_i^o = f(r_i^o)$ is calculated following Eq. (2). The asphere profile is of simple geometry, therefore, the Newton-Raphson method converges to a global minimum inevitably. Indeed, finding the orthogonal footpoint by Newton-Raphson's method requires setting the dot product $S_i = \vec{d}_i \cdot \vec{t}_i$ to zero. \vec{d}_i is the distance vector from \mathbf{P}_i^{Cyl} to $\mathbf{p}_i(r_i, z_i)$ in the corresponding θ_i -oriented rz -plane (Fig. 1). \vec{t}_i is the tangent vector at \mathbf{p}_i and is expressed as in Eq. (10), by differentiating the implicit function F (Eq. (1)) of the surface at \mathbf{p}_i . The initial distance vector input to Eq. (11) is the vertical distance vector $\vec{d}_i^v = (r_i^v - R_i, z_i^v - Z_i)$

$$\vec{t}_i = \begin{pmatrix} \left. \frac{\partial F(r, z)}{\partial z} \right|_{r=r_i, z=f(r_i)} \\ \left. \frac{-\partial F(r, z)}{\partial r} \right|_{r=r_i, z=f(r_i)} \end{pmatrix} \quad (10)$$

The function S_i is iteratively solved until the dot product between \vec{d}_i and \vec{t}_i falls below a certain predefined threshold $\varepsilon_1 = 10^{-16}$.

$$S_i = (r_i - R_i) \frac{\partial F(r_i, z_i)}{\partial z} + (z_i - Z_i) \frac{\partial F(r_i, z_i)}{\partial r} = 0,$$

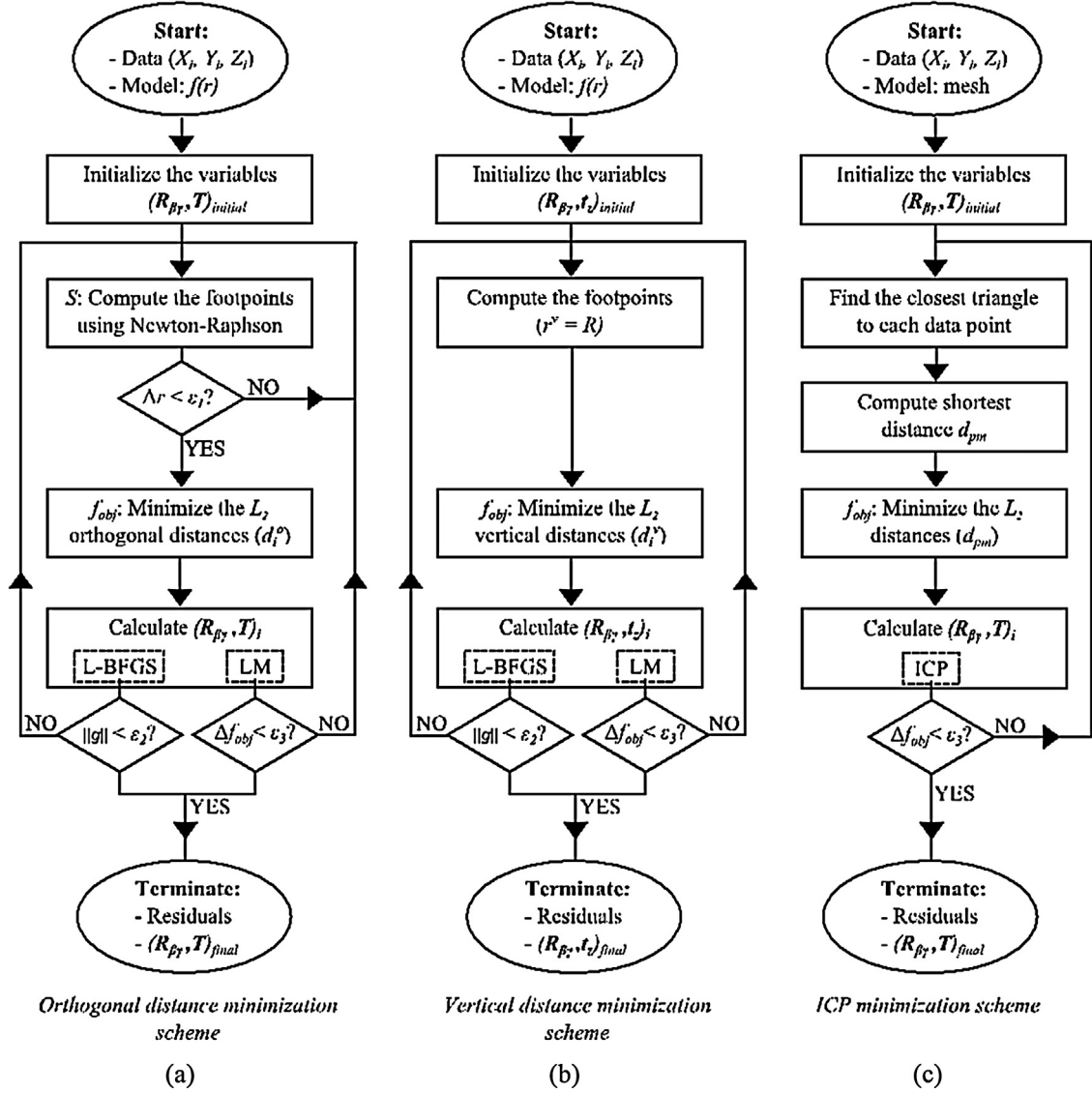


Fig. 2. Sequential algorithm for the fitting problem. L_2 : least-squares norm; $\mathbf{R}_{\beta,\gamma}$: combined rotation matrix and \mathbf{T} : translation vector; f_{obj} : change in the objective function value; $\|g\|$: norm of the gradient of f_{obj} ; $\varepsilon_1, \varepsilon_2, \varepsilon_3$: the algorithms stop criteria.

with

$$z_i = f(r_i) = - \left(\frac{cr_i^2}{1 + \sqrt{1 - (1+k)c^2r_i^2}} + \sum_{j=1}^5 a_{2j} r_i^{2j} \right). \quad (11)$$

The orthogonal footpoint calculated in cylindrical coordinates is then transformed back to 3D Cartesian coordinates, i.e. $\mathbf{p}_i^o = (r_i^o \cdot \cos \Theta_i, r_i^o \cdot \sin \Theta_i, f(r_i^o))$.

With the coordinates of each footpoint, vertical $\mathbf{p}_i^v(x_i^v, y_i^v, z_i^v)$ or orthogonal $\mathbf{p}_i^o(x_i^o, y_i^o, z_i^o)$, obtained from the inner loop, it becomes possible to plug in the distances and solve for the transformation parameters. The outer loop is executed repeatedly until convergence conditions are met. The norm of the gradient is minimized with L-BFGS with a stop criterion $\varepsilon_2 = 10^{-16}$. A stop criterion $\varepsilon_3 = 10^{-15}$ is set to be the objective function's tolerance for LM and ICP since it can be assimilated to a distance tolerance in both cases (Fig. 2b and c). ε_3 measures the relative error desired in the sum of squared distances.

3.2. Implementation of the variant of the ICP algorithm

The ICP is a registration algorithm which finds a spatial transformation to align two point-sets and follows the procedure illustrated

in Fig. 2c. It is the mostly used algorithm in point-set registration because it is designed for discrete data, making it a relatively fast algorithm with very low storage. ICP is based on two main operations, point identification and point matching which are usually computationally expensive. An iterative loop identifies pairs of points and matches them across both entities. The matching phase results in a transformation matrix that brings one point-set to the other with residual error. If this error is larger than the threshold value, point identification and matching restarts until the two point-sets are closely aligned. In order to have fine precision on the results, it is preferable that the size of both sets be equal.

The variant of ICP that is proposed here is designed to be used in fitting applications where one of the point-sets is a theoretical mesh model. All points are involved in the point-matching phase so that accuracy is preserved. A mesh model offers the advantage of obtaining a more accurate distance calculation than a point model does. The mesh is a regular triangular mesh built from axis-symmetric theoretical points simulated on the asphere's surface and reconstructed using a Delaunay triangulation technique [30]. To guarantee accurate residual errors, equivalent to those found by L-BFGS and LM, the chord error C_{err} between the mesh triangles and the theoretical surface is taken to be very small: $C_{err} \leq 2$ nm.

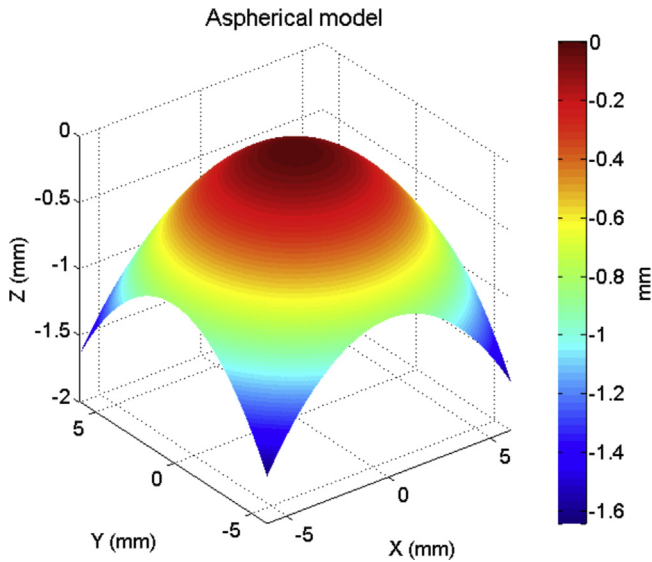


Fig. 3. Simulated asphere model.

The value of C_{err} is determined according to the precision sought for aspherical lens characterization and is assured by increasing or decreasing the sampling density.

With the proposed discrete approach, the objective function to minimize is the sum of the squared distances d_{pm} (Eq. (12)). d_{pm} is the shortest distance separating a data point p from its footpoint situated on the closest triangle in the mesh, i.e. $d_{pm} = \min(d_{pv}, d_{pe}, d_{pt})$ and d_{pv} , d_{pe} and d_{pt} are the point-to-vertex, point-to-edge and point-to-triangle distances respectively. The variables disregard the angle about z , since the points are axis-symmetric

$$f_{obj}(\beta, \gamma, \mathbf{T})_{ICP} = \min_{\beta, \gamma, t_x, t_y, t_z} \sum_{i=1}^N \|d_{pm_i}\|^2. \quad (12)$$

3.3. Applications to simulated datasets

The aspherical model is simulated based on Eq. (2) by generating symmetrically distributed points around the asphere's axis over a rectangular grid (Fig. 3). The model parameters c , k , a_2 , a_4 , a_6 , a_8 , a_{10} are equal to the values given in Section 2.1. To

evaluate the robustness and correctness of the developed fitting algorithms in conformance with ISO 10360: Part 6 [17], two simulations are performed: one with added random errors to characterize surface roughness and another with combined random and systematic errors to characterize both surface roughness and form errors owing to the manufacturing processes. The simulations do not take into account measurement errors (noise and probing random errors) but these will not be disregarded. The specification of form errors and surface roughness is that they act in the orthogonal direction to the aspheres' surface.

3.3.1. Random errors

The first test involves generating Gaussian noise with controlled mean and standard deviation ($\mu = 0$, $\sigma = 8$ nm). This value is coherent with areal surface roughness that can manifest on real datasets due to manufacturing defects such as tool wear, tool mark and other asynchronous motion errors [31]. A Matlab function ("randn") is used to generate the noise (Fig. 4) which is added to the theoretical data of Fig. 3 in the orthogonal direction at each data point.

The execution of this function returns actual standard deviations that slightly differ from the imposed value. The aim of this simulation is to study the robustness of the algorithms when confronted to repeated random datasets of equal magnitude. L-BFGS, LM and one variant of ICP are then used to fit the noisy data and their robustness is qualified. The RMS of the residual errors and their PV are both evaluated and compared to the simulated ones since they transcribe the form specifications of aspheres [32]. The machine used for the tests is an Intel core i7/ $\times 64$ platform with 8 Gb of RAM and a 2.0 GHz processor.

The obtained residual PV values, reported in Table 1, are quasi-identical for all three algorithms and are sufficiently close to the simulated PV values to about some hundredths of nanometers. The RMS values are also coherent, and this remains unchanged for the repeated test.

3.3.2. Combined random and systematic errors

The test involves superposing systematic errors onto the previous random errors. Systematic errors F_h are referred to as form deviations and are generated using Fourier harmonics according to Eq. (13) and ISO10360-Part 6 standard for simple shapes. Fig. 5 shows the combined systematic and random errors which can in reality be due to the manufacturing process errors and especially the synchronous (systematic) motion errors of the mechanical

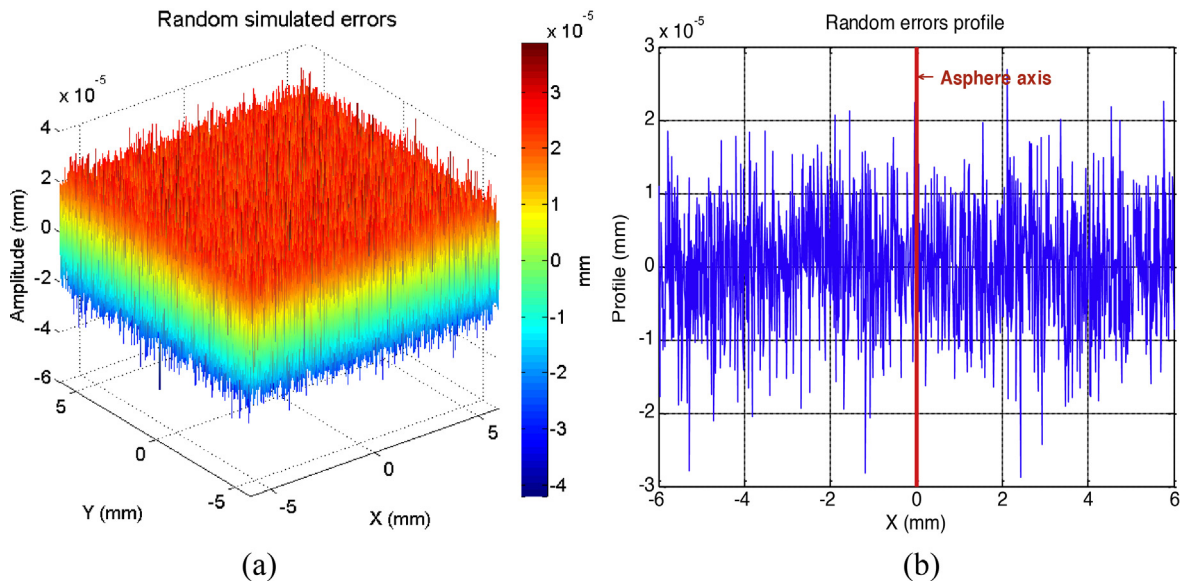


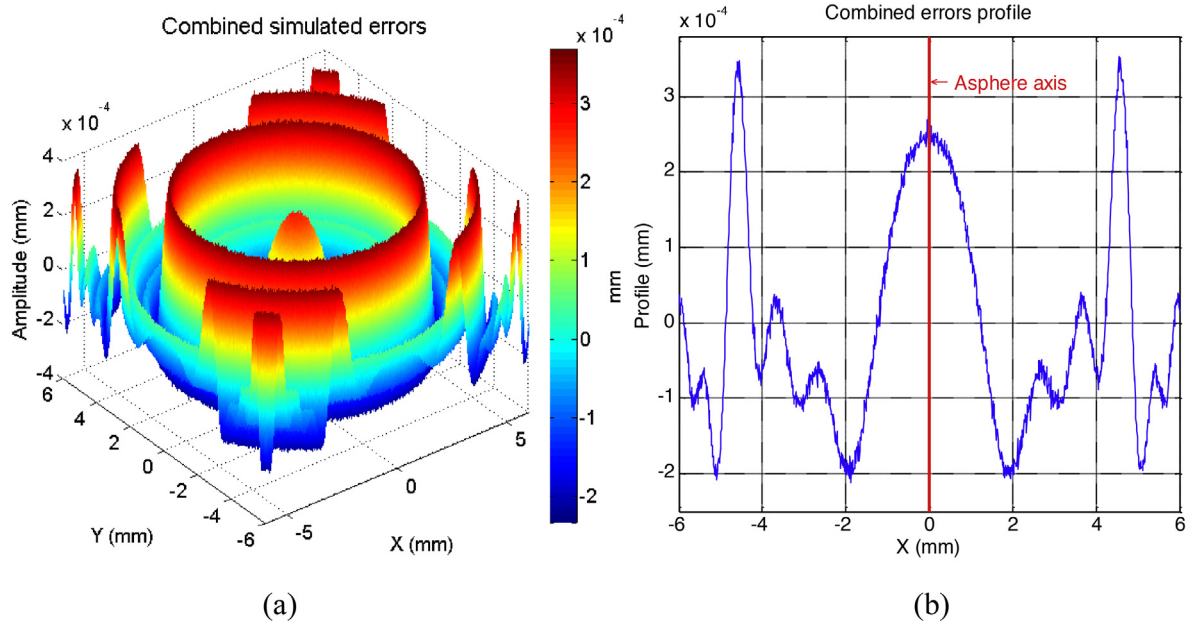
Fig. 4. Simulated Gaussian noise ($\mu = 0$, $\sigma = 8.0$ nm). (a) 3D plot and (b) 2D distribution at the middle section in the xz -plane.

Table 1

Fitting of 5 random error datasets with Gaussian noise of 0 mean and 8.0 standard deviation; 1,500,000 points are used.

Gen. err. (nm)		L-BFGS (nm)			LM (nm)			ICP (nm)		
RMS	PV	RMS	PV	Δ^{PV} (%)	RMS	PV	Δ^{PV} (%)	RMS	PV	Δ^{PV} (%)
8.003	77.325	8.003	77.334	0.017	8.003	77.329	0.005	8.004	77.374	0.063
8.001	79.875	8.001	79.881	0.008	8.001	79.880	0.006	8.003	79.945	0.088
7.996	81.992	7.996	81.938	-0.066	7.996	81.942	-0.061	7.998	82.041	0.060
8.002	83.252	8.002	83.299	0.056	8.002	83.301	0.059	8.003	83.387	0.162
8.008	88.871	8.008	88.888	0.019	8.008	88.889	0.020	8.009	88.379	-0.554

Gen. err., generated errors.

**Fig. 5.** Combined errors: simulated form deviations with added random noise ($\sigma = 8.0$ nm). (a) 3D plot; (b) profile view at the middle section in the xz -plane.

guiding systems [31,33]. The generated PV value of the Fourier harmonics (≈ 609 nm) corresponds to a realistic form error on aspherical lenses

$$F_h(\tau) = \sum_{k=1}^n A_k \sin(2\pi k w_0 Z) + B_k \cos(2\pi k w_0 Z), \quad (13)$$

where k is the index of the harmonic cosine wave, w_0 is the fundamental frequency, A_k and B_k are the user-specified partial amplitudes of the k th harmonic and Z is the z -coordinate to which the form defects are applied.

The systematic errors are taken as axis-symmetric (Fig. 5) because the major manufacturing process of aspheres is a turning process that combines a rotation of the substrate and a functional motion of the tool [34]. They occur in the orthogonal direction to the surface and that is why an orthogonal distance fitting is applied.

Table 2Fitting using least-squares orthogonal distance minimization for the combined systematic and random errors dataset. N is the number of points.

N	Gen. err. (nm)		L-BFGS (nm)		LM (nm)		ICP (nm)	
	RMS	PV	RMS	PV	RMS	PV	RMS	PV
100			137.437	642.155	137.436	642.142	138.621	642.875
1000			136.490	616.270	136.490	616.204	137.246	616.894
10,000			136.480	609.031	136.456	609.022	137.250	609.623
100,000	136.482	609.593	136.455	609.321	136.455	609.347	137.231	609.945
1,000,000			136.455	609.403	136.455	609.403	137.232	609.810
5,000,000			136.455	609.372	136.455	609.390	137.232	609.786
10,000,000			136.455	609.364	136.455	609.375	137.232	609.771

Gen. err., generated errors.

All three fitting methods return the same RMS and PV value as the simulated ones (Table 2). The RMS and PV values remain quasi-unchanged whichever the number of points in the dataset is as long as it is not too small (< 1000 points). The points are selected from the data file following the order of the recorded data and by picking 1 point every x points, where x is an integer number representing a ratio by which the number of points will be reduced by.

Table 3 reports the estimated motion parameters for two different initial positions of the simulated datasets. The first imposed initial position (IP_1) is generated with an offset of 1 mm in all three directions with respect to the model while keeping rotations at zero and IP_2 is generated with rotations by 30° about x - and y -directions with respect to the model while keeping translations to zero. The results of the motion parameters estimation show that both L-BFGS and LM are accurate for both imposed initial alignments as they return accurately enough good estimates. ICP fails

Table 3
Fitting using orthogonal distance minimization for the simulated combined errors dataset with two theoretical initial alignments.

		Gen. alignment	L-BFGS	LM	ICP
IP_1	β	0°	0.00000094°	0.0000056°	0.0000046°
	γ	0°	0.000033°	0.000097°	0.000029°
	t_x	1 mm	-1.0000124 mm	-1.0000368 mm	0.0000346 mm
	t_y	1 mm	-1.0000004 mm	-0.9999979 mm	0.0000369 mm
	t_z	1 mm	-0.9999852 mm	-0.9999866 mm	-0.541437 mm
IP_2	β	30°	-30.000034°	-30.000082°	
	γ	30°	-30.000066°	-30.000106°	
	t_x	0 mm	-0.0000141 mm	-0.0000277 mm	×
	t_y	0 mm	0.0000033 mm	0.0000226 mm	
	t_z	0 mm	0.0000150 mm	0.0000150 mm	

Gen. alignment, generated alignment.

Table 4
Comparison of vertical and orthogonal distance minimization for different dataset sizes and for combined simulated errors: reporting RMS values. N is the number of points.

N	Gen. RMS err. (nm)	Orthogonal (nm)		Vertical (nm)	
		L-BFGS	LM	L-BFGS	LM
1000		136.490	136.490	139.991	139.964
10,000		136.480	136.456	139.832	139.832
100,000	136.482	136.455	136.455	139.865	139.865
1,000,000		136.455	136.455	139.868	139.868
10,000,000		136.455	136.455	139.862	139.862

Gen. RMS err., generated RMS error.

to return estimations for the case of IP_2 as the algorithm diverges when the angle is more than 15° .

3.3.3. Vertical versus orthogonal distance minimization

The evaluation of the effect of fitting data based on vertical and orthogonal distance minimization is achieved here by using the previously simulated datasets and the results are compared.

Table 4 shows a comparison between vertical and orthogonal Least Squares distance minimizations using the L-BFGS algorithm (LM giving the similar results) with the added combined errors. Note that for the vertical distance case, the input datasets are perfectly aligned in x and y directions with respect to the model. The results show that orthogonal distance minimization is more accurate than vertical distance minimization and that this remains true irrespective of the number of points in the input dataset. This is clearly due to the fact that errors were added to the data in the orthogonal direction. It is to note that vertical distance minimization fitting time is extremely fast (<3 s for 10^6 points).

Another simulation has been performed to reproduce optical probing noise which can occur during a measurement process. Such errors occur in the vertical direction since measurement is done along this direction. For the current case, vertical distance minimization is more appropriate. Nevertheless, the error made by choosing to perform orthogonal distance minimization is less than $<1\%$ for the PV and RMS values. Since this error is negligible as compared to form and roughness errors which predominate, choosing orthogonal distance minimization is not very detrimental.

3.3.4. Algorithmic complexity

The L-BFGS, LM and ICP algorithmic complexities are analyzed based on two criteria, the units of memory used and the computational time expressed as Central Processing Unit time (CPU time). Fig. 6 shows that the time complexities of L-BFGS and LM are linear in the number of points for the case of added random errors. However, it can be observed based on the given implementations of L-BFGS, LM and ICP, that ICP is slow as compared to L-BFGS and LM and that LM is about twice slower than L-BFGS ($t_{L-BFGS} \approx 50\% \times t_{LM}$) especially when the number of points exceeds 10^5 points.

Regarding memory storage, all algorithms use less than 1Gb in general. L-BFGS stores the least memory space among all because of its limited memory feature (less than 0.2 Gb). As the number of variables grow, the Jacobian of the function to minimize in the case of LM grows very large and its inversion costs more. ICP needs to store a massive triangular mesh of the model. The time complexity of the algorithms is compared for the case of combined errors too (Fig. 7) and confirms that the time complexity of the L-BFGS and LM algorithms is linear in the number of data points.

Some additional tests are performed on the simulated combined errors dataset in order to analyze the sensitivity of the time complexity and residual errors to the L-BFGS parameters, such as, the limited number of iterations m and the coefficients β' and β . One parameter is changed at a time while the others remain fixed. The evolution of the variables and the objective function over the iterations are also discussed. This analysis is only performed on L-BFGS

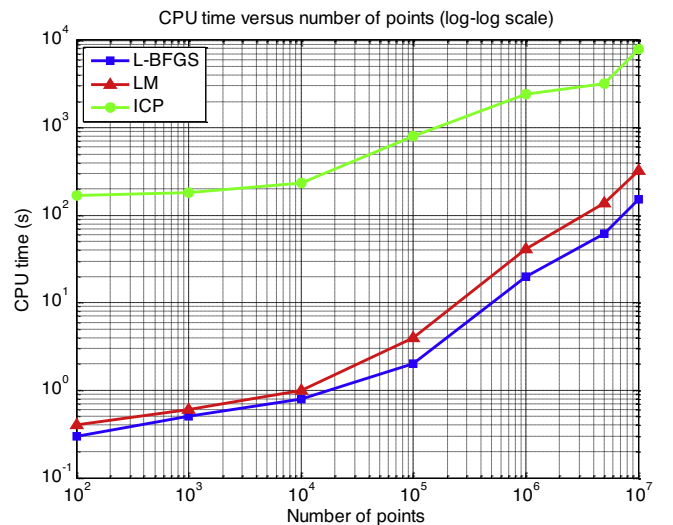


Fig. 6. Time performances in seconds of the algorithms versus the number of points for a simulated dataset with added Gaussian noise of 0 mean and 8.0 standard deviation.

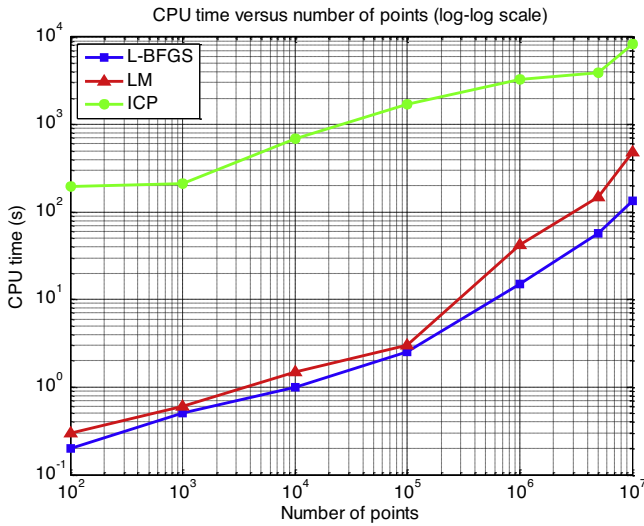


Fig. 7. Time performances in seconds of the algorithms versus the number of points for a simulated dataset with added random and systematic errors. 3.3.5 Sensitivity of L-BFGS intrinsic parameters.

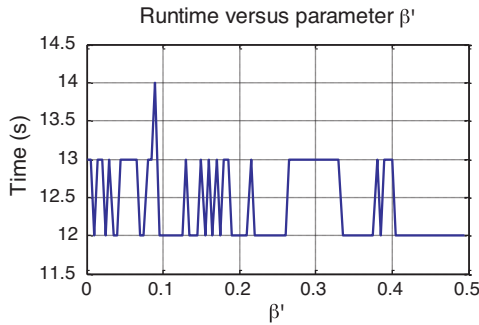


Fig. 8. The influence of β' on the computational time performance of the algorithm for the simulated combined errors dataset.

parameters because a similar analysis has been done for LM parameters [25].

The time performance of the L-BFGS algorithm for the dataset with the added combined errors versus the parameter β' over its entire range of possible values is shown in Fig. 8. The time complexity of L-BFGS is not affected by the value of β' as only a fluctuation of 2 s is observed. Fig. 9 shows the influence of this parameter on the residual errors of the fit and reveals that they are independent of β' . In both cases, m and β are fixed to default values of 11 and 0.9, respectively. From these plots, it can be concluded that β' has no major impact on either of the algorithm's complexity or the residual errors.

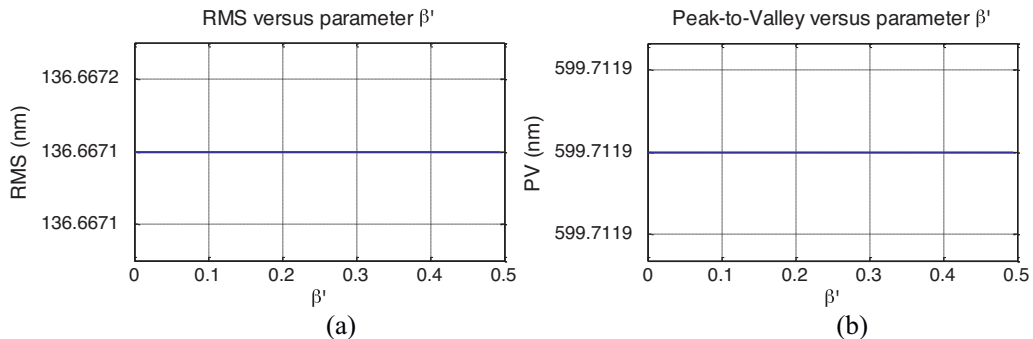


Fig. 9. The influence of β' on the residuals for the simulated combined errors dataset. (a) RMS residuals; (b) PV residuals.

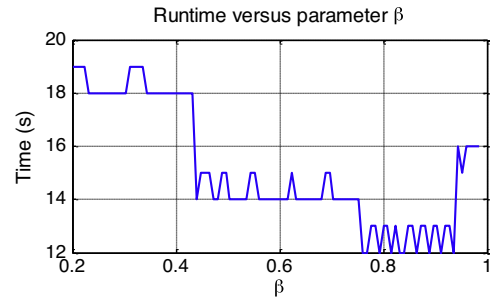


Fig. 10. The influence of β on the computational time performance of the algorithm for the simulated combined errors dataset.

Figs. 10 and 11 illustrate the influence of the parameter β on the time complexity and residual errors while m and β' are fixed to 11 and 0.0001, respectively. Here again, the residual errors are unaffected by the choice of β , nonetheless, convergence time is clearly influenced and can vary by up to 58%.

Fig. 12 illustrates the effect of changing the number of iterations m to be taken into account for the limited memory criterion. The graph reveals that a value of $m > 5$ should be generally considered. Below this value, computational time becomes significant and the advantage of using a limited memory is depreciated. It is to note that the value of m does not affect residual errors.

Based on the above, the intrinsic L-BFGS parameters have a very low impact on residual errors. Nonetheless, computational time is sensitive to all three parameters.

For one specific set of parameter values, the evolution of the objective function value and the transformation parameters are illustrated in order to show how they vary along the iterations and provide an idea about convergence rate. Fig. 13 is a plot of the evolution of the objective function's value over the iterations of the fitting algorithm. Starting from around the 4th iteration, the objective function value stabilizes with a soft decrease whilst the motion parameters still delicately adjust until a minimum is found and convergence conditions are met around the 17th iteration.

The sum of squared distances evolution depicts the rate of convergence of the L-BFGS algorithm and confirms that it is relatively fast.

4. Application to a measured aspherical lens

After the validation of the algorithms on simulated datasets, the aim is to tackle real measured data with the three proposed algorithms in order to compare them and show the efficiency of L-BFGS.

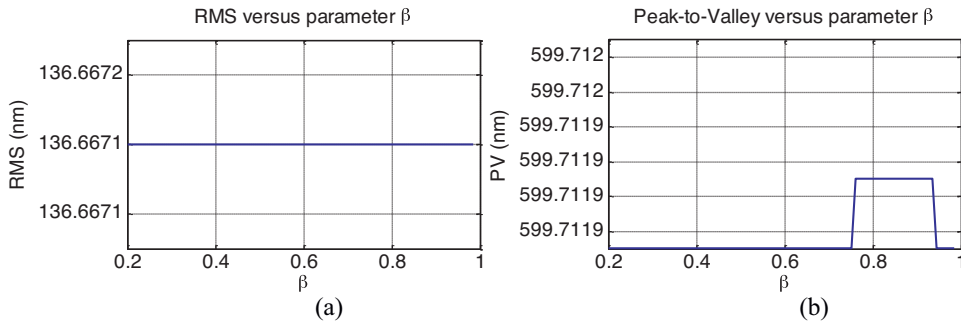


Fig. 11. The influence of β on the residuals for the simulated combined errors dataset. (a) RMS residuals; (b) PV residuals.

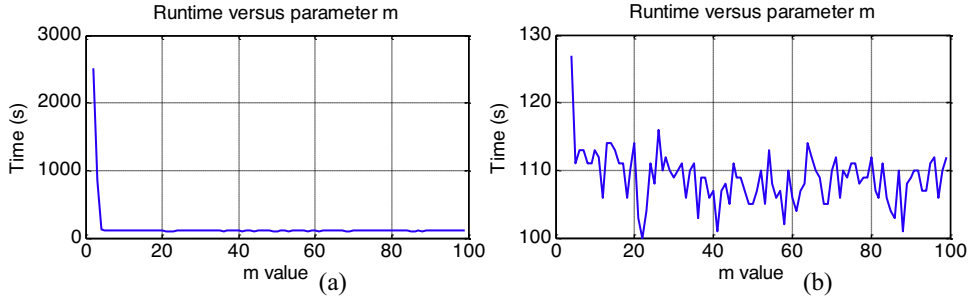


Fig. 12. The influence of m on the time performance. (a) $m > 1$; (b) $m > 3$.

4.1. LNE's high precision profilometer description and measurement

LNE's high precision profilometer is a measurement machine (Fig. 14) capable of performing independent motions in all x , y and z directions using three independent high-precision mechanical guiding systems. While x and y motions are controlled by sub-nanometer resolution laser interferometers, the z motion is controlled by a differential laser interferometer that allows to shorten the metrology loop and maintain a sub nanometric accuracy. The supporting structure is made of massive granite and carries the guiding elements. The metrology frame is made of Invar for minimal sensitivity to environmental influence [35,36]. The machine was configured to hold both optical and tactile single scanning probes. On this measuring machine and other systems involving some optical measurement techniques, the new paradigm leans toward the generation of large volumes of data of more than 10^5 points.

The asphere is posed on the Zerodur table (Fig. 14) and is only measured by a tactile single point scanning probe which has been previously calibrated in situ.

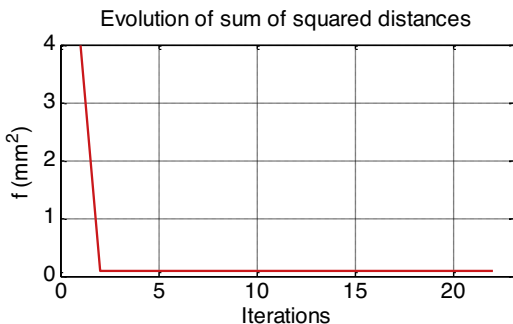


Fig. 13. The evolution of the objective function's value over the iterations.

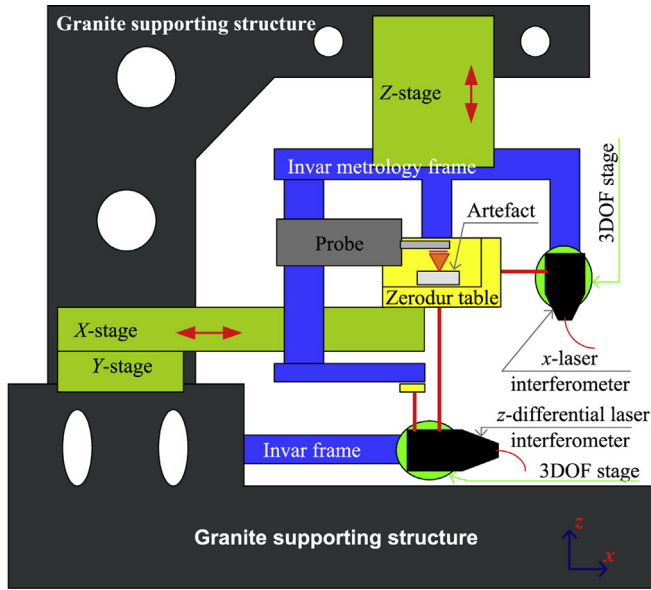
On this machine, it is not possible to exactly align the asphere's axis of symmetry with the z -axis of the measurement (Fig. 15), however, an approximation of the apex position can be done by estimating the cusp of the surface. For this matter, the surface is scanned once in the x -direction and once in the y -direction and a peak is computed. This peak represents an approximation of the cusp around which a symmetrical measurement is performed in x - and y -directions.

The measurement process is automatic and the data are recorded and reported in Cartesian (X, Y, Z) coordinates. The surface is firstly scanned over an area of $6 \text{ mm} \times 6 \text{ mm}$, giving a grid of 1225×1225 points denoted as Scan1. A portion of that scan is extracted to be non-centered and a fitting is performed again and this scan is denoted as Scan2. Scan2 is about $3.5 \text{ mm} \times 3.5 \text{ mm}$ in area and is constituted of a grid of about 1385×343 points

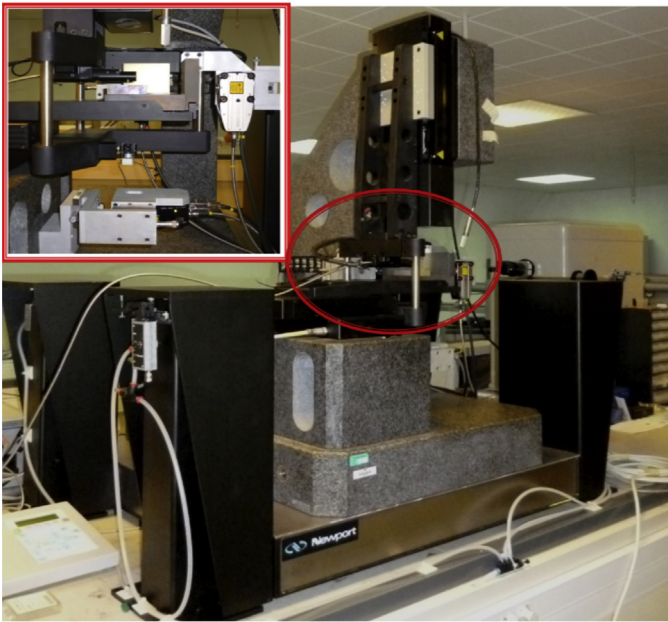
4.2. Comparison of the fitting algorithms

On the LNE high precision profilometer, the alignment of the measurement coordinate system with the model coordinate system is manually achieved. Therefore, vertical distance minimization is not valid since it inhibits horizontal translations (t_x and t_y), and by that, generates additional errors in the fitting process. From another perspective, the vertical synchronous motion errors of the mechanical high precision guiding elements of the profilometer are compensated by the vertical laser interferometer. The asynchronous motion errors are supposed negligible compared to form errors which act in the orthogonal direction, hence, it is wiser to apply orthogonal distance fitting in this case. The results of the L-BFGS, LM based on Least-Squares orthogonal distance minimization are compared to ICP for the two experimental datasets, Scan1 and Scan2. Table 5 for Scan1 and Table 6 for Scan2 report the residual errors for different sizes of the datasets.

The residual errors are identical for all three algorithms. The larger scan, Scan1, returns a RMS value of about 217.2 nm and the smaller scan, Scan2, returns a RMS of about 44.1 nm. Fig. 16 illustrates the computational time in CPU seconds for the fitting of the



(a)



(b)

Fig. 14. Design (a) and photography (b) of the high precision profilometer.

Table 5

Fitting using orthogonal distance minimization for the first scan. N denotes the number of points.

	L-BFGS		LM		ICP	
	RMS (nm)	PV (μm)	RMS (nm)	PV (μm)	RMS (nm)	PV (μm)
$N = 75,000$	217.18	2.1725	217.18	2.1727	217.29	2.1982
$N = 200,000$	217.18	2.1725	217.18	2.1726	217.29	2.1992
$N = 500,000$	217.18	2.1728	217.18	2.1728	217.29	2.1979
$N = 1,500,000$	217.18	2.1726	217.18	2.1732	217.29	2.1983

Table 6

Fitting using orthogonal distance minimization for the second scan. N denotes the number of points.

	L-BFGS		LM		ICP	
	RMS (nm)	PV (μm)	RMS (nm)	PV (μm)	RMS (nm)	PV (μm)
$N = 75,000$	44.18	0.8186	44.18	0.8183	44.21	0.8173
$N = 160,000$	44.18	0.8187	44.18	0.8184	44.21	0.8171
$N = 500,000$	44.18	0.8186	44.18	0.8185	44.21	0.8175

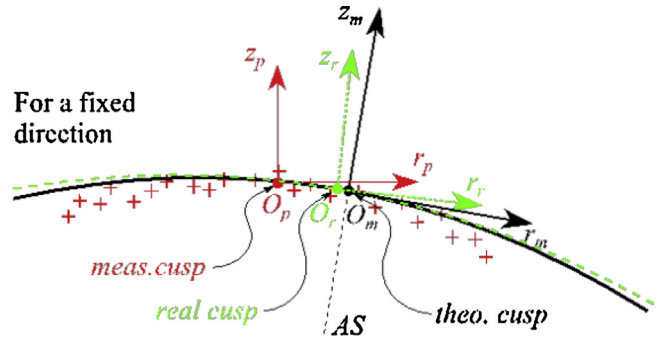


Fig. 15. Illustration of the problem with the alignment of the measurement coordinate system. Indices m , p and r stand for model, measured point-set and real surface, respectively; AS stands for theoretical axis of symmetry.

measured points of Scan1 for different dataset sizes. It shows a linear complexity for L-BFGS with respect to the number of points.

Fig. 17 plots the evolution of the sum of squared distances (f_{obj}) over the iterations of the L-BFGS and LM algorithms for the case of Scan1. The algorithms are remarkably comparable with respect to the objective function's evolution. A rapid decay can be observed along the first 3 iterations for L-BFGS and 7 iterations for LM. A more or less stable solution starts at iterations 60 and 55 for L-BFGS and LM, respectively. The error maps of the residual errors resulting from the fitting of each of the datasets are plotted in Fig. 18 using L-BFGS, and show that the residual errors distribution depend on the measured zone. The plots are shown only for the clear apertures of 1.75 mm and 3 mm, respectively.

5. Discussion

The L-BFGS, LM and the presented variant of ICP algorithms are validated for the fitting of aspherical surfaces by means of tests performed on simulated datasets. The simulations include systematic and random errors represented by Gaussian noise and Fourier harmonics. The fitting results return accurate estimates of the motion parameters, except for ICP, as well as the same RMS and PV value as the simulated ones whether in the case of added random errors or combined random and systematic errors. The tests have shown that, unlike ICP, both LM or L-BFGS are robust to initial position of the point-set with respect to the model. The results remain unchanged for a large span of dataset sizes ranging from about 1000 points up to 10,000,000. From this observation, it can be deduced that different point sampling strategies do not have a remarkable impact on the accuracy of the results. Since the proposed sampling strategy is not a filtration but some sort of a technique to pick

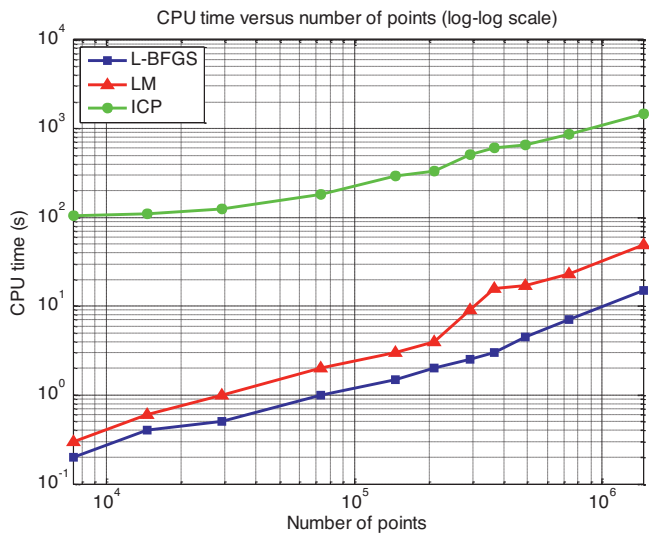


Fig. 16. Computational time of the algorithms versus dataset size.

points from all around the scanned surface, it is auspicious for the processing of large volumes of data.

The Least-Squares vertical distance minimization and orthogonal distance minimization are compared on the simulated datasets and show that orthogonal distance minimization is more accurate and appropriate than vertical distance minimization. In any case, the only condition for the correct use of vertical distance minimization is to have both the dataset and the model perfectly aligned in x and y directions, which is impossible in practice unless ultra-high precision measuring machines are used. In this case, vertical distance minimization can be used to perform the fitting extremely fast (<3 s for 10^6 points), however, to the expense of losing in residual errors accuracy (up to 1%). The algorithmic complexities of L-BFGS and LM are further assessed and show that L-BFGS is almost 3–4 times faster than LM when the number of points becomes considerable ($>10^6$ points). Moreover, ICP is much slower than any of L-BFGS or LM. The analysis of the L-BFGS algorithm parameters shows that their value does not considerably affect residual errors. However, a very small value of m (<6), amplifies computational time.

Then, the comparison of the three algorithms on real datasets with respect to residual errors is performed. For a given number of points, the residual values returned by L-BFGS, LM and ICP are very similar. This remains true for a large span of dataset sizes and provided that a relatively good initial alignment is taken into

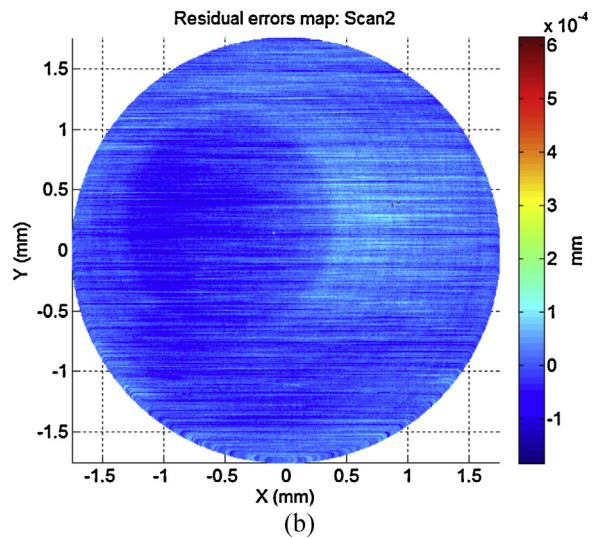
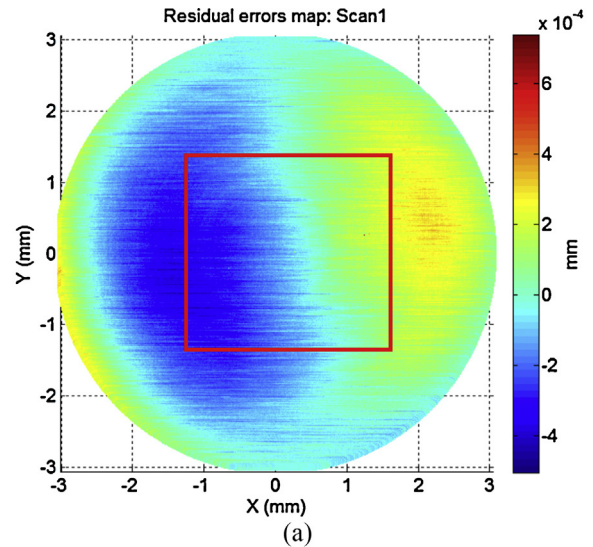


Fig. 18. 3D residual errors map characterizing form defects. (a) Scan1, (b) Scan2.

consideration. All three optimization algorithms present very low memory storage and their time complexities are linear in the number of points. L-BFGS and LM enable the processing of very large amounts of data of more than 100,000 points within few seconds. For instance, for Scan1 containing 1,500,000 points, the running

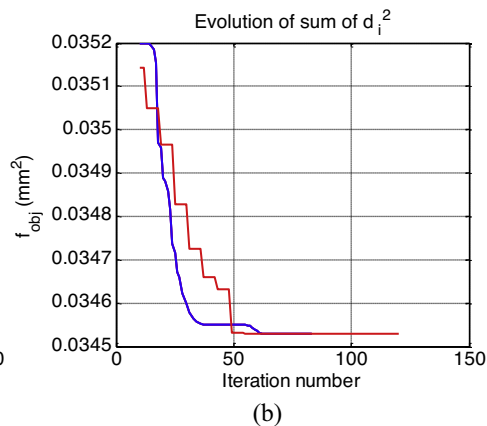
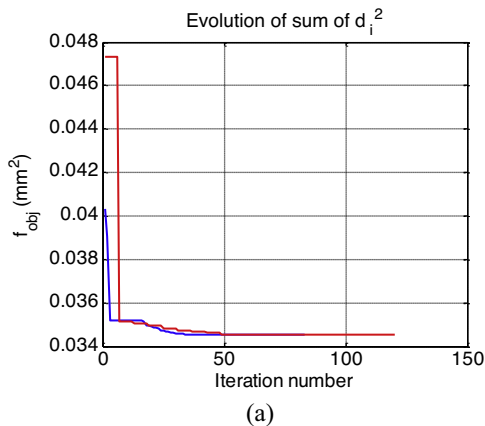


Fig. 17. Evolution of the objective function value through the iterations for the case of Scan1 (L-BFGS: blue, LM: red). (a) All iterations; (b) Starting from iteration 10.

time of both L-BFGS and LM is in the range between 30 s and 100 s. Memory was only of a few tens of Megabytes. Since the complexity of both algorithms is linear in the number of points, and the memory storage is negligible, a simple extrapolation suggests that about 10,000,000 points can be eventually processed within 6 to 7 minutes on real datasets.

6. Conclusion

This paper deals with the fitting of aspherical surfaces in the aim of characterizing their form errors. The L-BFGS method is proposed and compared to the classically used LM and ICP algorithms on both simulated and measured data. Two cases are proposed for the simulations: a simulation with random errors and another with combined random and systematic errors. The three algorithms return similar results regarding the simulated PV and RMS values. ICP fails to converge accurately for relatively far initial alignments of the data with respect to the model. The impact of the number of points is investigated and does not influence the obtained results. Nonetheless, L-BFGS shows linear time complexity with respect to the number of points and runs faster than LM and largely faster than ICP.

As for the comparison on measured data, two datasets are measured with the LNE's high precision profilometer using a tactile probe. 500,000 and 1,500,000 points are recorded respectively and the above algorithms are applied for the fitting of the data. Again, the RMS and PV values are independent of the size of the dataset and are similar across all algorithms.

As a future work, authors would like to investigate the performance of the L-BFGS algorithm for the fitting of quadric and freeform surfaces. Then, another aspect to be explored would be the fitting using the MinMax criterion. Finally, it was shown that vertical distance minimization is sometimes more suitable than orthogonal distance minimization, hence, a more clever way of considering this issue shall be studied in later publications.

Acknowledgements

This work is part of EMRP Joint Research Project-IND10 "Optical and tactile metrology for absolute form characterization" project. The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union.

References

- [1] Fang F, Zhang X, Weckenmann A, Zhang G, Evans C. Manufacturing and measurement of freeform optics. *CIRP Ann Manuf Technol* 2013;62(2).
- [2] Yi A, Huang C, Klocke F, Brecher C, Pongs G, Winterschladen M, et al. Development of a compression molding process for three-dimensional tailored free-form glass optics. *Appl Opt* 2006;45:6511–8.
- [3] "EURAMET" [Online]. Available: <http://www.emrponline.eu/> [accessed 01.14].
- [4] Nouira H, Bergmans R, Küng A, Piree H, Henselmans R, Spaan H. Ultra-high precision CMMs as well as tactile and optical single scanning probes evaluation in dimensional metrology. In: *CAFMET*. 2014.
- [5] Chen Z-L, Guo Z-D, Mi Q, Yang Z-Q, Bai R-B. Research of fitting algorithm for coefficients of rotational symmetry aspheric lens. In: 4th international symposium on advanced optical manufacturing and testing technologies: optical test and measurement technology and equipment. 2009.
- [6] Moré J. The Levenberg–Marquardt algorithm: implementation and theory. In: *Numerical analysis*. Springer; 1978. p. 105–16.
- [7] Aceves-Campos H. Profile identification of aspheric lenses. *Appl Opt* 1998;37:8149–50.
- [8] Zhang Z. Parameter estimation techniques: a tutorial with application to conic fitting. *Im Vis Comp* 1997;15:59–76.
- [9] Sun W, McBride J-W, Hill M. A new approach to characterizing aspheric surfaces. *Precis Eng* 2010;34:171–9.
- [10] Park H. A solution for NURBS modelling in aspheric lens manufacture. *Int J Adv Manuf Technol* 2004;23:1–10.
- [11] Besl P, McKay N. Method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell* 1992;14:239–56.
- [12] Rusinkiewicz S, Levoy M. Efficient variants of the ICP algorithm. *3D Dig Im Mod* 2001:145–52.
- [13] Jiang X, Zhang XSP. Template matching of freeform surfaces based on orthogonal distance fitting for precision metrology. *Meas Sci Technol* 2010;21:045101.
- [14] Nocedal J. Updating quasi-Newton matrices with limited storage. *Math Comp* 1980;35:773–82.
- [15] Zheng W, Bo P, Liu Y, Wang W. Fast B-spline curve fitting by L-BFGS. *Comp Aided Geom Des* 2012;29:448–62.
- [16] Pomerlau F, Colas F, Siegwart R, Magnenat S. Comparing ICP variants on real-world data sets. *Auton Robots* 2013;34:133–48.
- [17] I. 10360-6. Geometrical Product Specifications (GPS) – Acceptance and re-verification tests for coordinate measuring machines (CMM) – Part 6: Estimation of errors in computing Gaussian associated features; 2002.
- [18] I. 17450-1. Geometrical Product Specifications (GPS) – General concepts – Part 1: Model for geometrical specification and verification; 2012.
- [19] Ahn SJ. Least squares orthogonal distance fitting of curves and surfaces in space. Berlin Heidelberg; Springer; 2004.
- [20] I. 10110-12. Optics and photonics – Preparation of drawings for optical elements and systems – Part 12: Aspheric surfaces; 2007.
- [21] Kim H-S, Kim E-J, Song B-S. Diamond turning of large off-axis aspheric mirrors using a fast tool servo with on-machine measurement. *J Mater Process Technol* 2004;146(3):349–55.
- [22] Lancaster P. Error analysis for the Newton-Raphson method. *Numeri Math* 1966;9:55–68.
- [23] Süli E, Mayers D. An introduction to numerical analysis. Cambridge University Press; 2003.
- [24] Transtrum M, Sethna J. Improvements to the Levenberg–Marquardt algorithm for nonlinear least-squares minimization; 2013 arXiv:1201.5885.
- [25] Shakarji C. Least-squares fitting algorithms of the NIST algorithm testing system. *J Res Natl Inst Stan Technol* 1998;103:633–41.
- [26] Nocedal J, Wright S. Numerical optimization. New York: Springer; 1999.
- [27] Liu D, Nocedal J. On the limited memory BFGS method for large-scale optimization. *Math Prog* 1989;503–28.
- [28] Speer T, Kuppe M, Hoschek J. Global reparametrization for curve approximation. *Computer Aided Geometric Design* 1998;15:869–77.
- [29] Lin Y, Damodharan K, Shakarji C. Standardised reference data sets generation for coordinate measuring machine software assessment. *Int J Adv Manuf Technol* 2001:819–30.
- [30] El-Hayek N, Nouira H, Anwer N, Damak M, Gibaru O. Reconstruction of freeform surfaces for metrology. In: *Met. & Props*. 2013.
- [31] Marsh E. Precision spindle metrology. DEStech Publishing; 2010.
- [32] Zhang X, Jiang X, Scott P. A minimax fitting algorithm for ultra-precision aspheric surfaces. *J Phys* 2011;311:12–31.
- [33] Henselman R. Non-contact measurement machine for freeform optics. Eindhoven; 2009. ISBN 978-90-386-1607-0.
- [34] Davis E, Roblee J. Comparison of freeform manufacturing techniques in the production of monolithic lens arrays. *SPIE* 2009;7426.
- [35] El-Hayek N, Nouira H, Anwer N, Gibaru O, Damak M. Comparison of tactile and chromatic confocal measurements of aspherical lenses for form metrology. *Int J Precis Eng Manuf* 2014;15:1–9.
- [36] Nouira H, Salgado J, El-Hayek N, Ducourtieux S, Delvallée A, Anwer N. Setup of a high-precision profilometer and comparison of tactile and optical. *Meas Sci Technol* 2014;25.