Article

Generation of Reference Softgauges for Minimum Zone Fitting Algorithms: Case of Aspherical and Freeform Surfaces

Amine Chiboub [1], Yassir Arezki [1,2], Alain Vissiere [1] , **Charyar Mehdi-Souzani [2]** , **Nabil Anwer [2]** , **Bandar Alzahrani [3], Mohamed Lamjed Bouazizi [3]** and **Hichem Nouira [1,***

[1] Laboratoire Commun de Métrologie (LNE-CNAM), 1 Rue Gaston Boissier, 75015 Paris, France; amine.chiboub@external.hec.fr (A.C.); yassir.arezki@gmail.com (Y.A.); alain.vissiere@lecnam.net (A.V.)
[2] Université Paris-Saclay, Université Sorbonne Paris Nord, ENS Paris-Saclay, LURPA, 91190 Gif-sur-Yvette, France; charyar.souzani@ens-paris-saclay.fr (C.M.-S.); nabil.anwer@ens-paris-saclay.fr (N.A.)
[3] Department of Mechanical Engineering, College of Engineering, Prince Sattam bin Abdulaziz University (PSAU), Alkharj 16273, Saudi Arabia; ba.alzahrani@psau.edu.sa (B.A.); my.bouazizi@psau.edu.sa (M.L.B.)
* Correspondence: hichem.nouira@lne.fr

Abstract: Optical aspherical lenses with high surface quality are increasingly demanded in several applications in medicine, synchrotron, vision, etc. To reach the requested surface quality, most advanced manufacturing processes are used in closed chain with high precision measurement machines. The measured data are analysed with least squares (LS or L_2-norm) or minimum zone (MZ) fitting (also Chebyshev fitting or L_∞-norm) algorithms to extract the form error. Performing data fitting according to L_∞-norm is more accurate and challenging than L_2-norm, since it directly minimizes peak-to-valley (PV). In parallel, reference softgauges are used to assess the performance of the implemented MZ fitting algorithms, according to the *F1 algorithm measurement standard*, to guarantee their traceability, accuracy and robustness. Reference softgauges usually incorporate multiple parameters related to manufacturing processes, measurement errors, points distribution, etc., to be as close as possible to the real measured data. In this paper, a unique robust approach based on a non-vertex solution is mathematically formulated and implemented for generating reference softgauges for complex shapes. Afterwards, two implemented MZ fitting algorithms (HTR and EPF) were successfully tested on a number of generated reference pairs. The evaluation of their performance was carried out through two metrics: degree of difficulty and performance measure.

Keywords: fitting algorithm validation; reference softgauges generation; minimum zone fitting; Chebyshev fitting; complex surfaces

1. Introduction

Conformance assessment of manufactured parts to design tolerance specification is a major activity in the quality control process. Traceable ultra-high precision coordinate measuring machines (CMMs) to the SI metre definition are usually deployed to generate a set of data points lying on the artefact's surface [1]. The recorded data will be processed in order to infer information about the measured surface. This task becomes difficult when it comes to the measurement of aspherical and freeform surfaces due to their complex geometry. Different factors could greatly affect the machine uncertainty, namely, the work piece, the hardware of the CMMs, the sampling strategy, the fitting algorithms, etc. [1–3]. Fitting could be defined as the process of determining parameters of the geometric features that best describe the measured data according to a defined criterion. This geometric element is called "*associated feature*" [4]. Different criteria could be used, including least squares (LS) (or Gaussian, or L_2-norm fitting), minimum zone (MZ) (or minimax, Chebyshev, L_∞-norm fitting) and one-sided measures, such as minimum circumscribed (MC)

or maximum inscribed (MI) elements. While the two least criteria are well adapted for circular or cylindrical features, the former ones could be applied for almost all shapes. Least squares criterion, which is more adapted when random measurement errors predominate, originates from maximum likelihood theory [5]. It is the most widely used criterion in industry and most fitting commercial software relies on it. However, in the accept/reject process, LS fitting may not give complete information and may cause the rejection of conforming parts. Therefore, the use of MZ fitting reflects the functional requirements of measured parts. This criterion is appropriate when measuring errors are small compared to manufacturing ones. Moreover, MZ is considered as the default fitting criterion in the ISO geometrical product specification (GPS) [6,7]. Once the MZ value is found, a measure of variability must be set and the part will be accepted if the measure is below design limits. The commonly used measure is the peak-to-valley (PV), which is the difference between maximum and minimum form deviations. Unlike LS fitting methods, the implementation of MZ fitting methods is challenging, since it directly minimizes the peak-to-valley (PV) [8].

For metrology of aspherical and freeform surfaces, a few reference fitting algorithms have recently been developed. Key characteristics of reference fitting algorithms were outlined in [8]. Thus, it is stated that they must:

- perform well for representative data
- work sensibly for unrepresentative data and be able to detect extreme cases
- perform efficiently in poor cases

Despite its importance, performance in terms of execution time is not the first characteristic sought for reference MZ fitting algorithms. However, as reported in [2] the algorithm must be stable and robust. The stability means that the underlying numerical operations are numerically stable. For example, small perturbations in input data must only result in small perturbations in output. Robustness refers to the algorithm ability to handle extreme cases.

To establish the traceability chain for datasets analysis with a small uncertainty (below the nanometre level), the validation of implemented reference MZ fitting algorithms becomes indispensable in order to make sure that the returned values are correct. There exist two methods to assess the correctness of the values returned by MZ fitting algorithms [9–12], namely *type F1 algorithm measurement standard* (using reference softgauges pair) and *type F2 algorithm measurement standard* (using reference algorithm) [11].

For complex geometries, an approach based on *type F1* was presented in [13]. However, the proposed method generates reference softgauges with vertex solution only. Meanwhile, non-vertex solution occurs in practice as reported in [1,14].

In this article, the validation of MZ fitting algorithms using reference softgauges is discussed. Section 2 introduces the MZ fitting optimisation problem, and Section 3 investigates the validation procedures. The problem is mathematically formulated in Section 4, while a general approach is given in Section 5 on the generation of reference softgauges based on a non-vertex solution for the case of complex shapes. Numerical validation on two implemented fitting algorithms (exponential penalty function (EPF) and hybrid trust region (HTR)) is carried out in Section 6. Two metrics to determine the degree of difficulty and the performance measure are given in Section 7, so as to assess the performance of MZ fitting algorithms.

2. Data Fitting

LS or MZ fitting are considered as an optimisation problem. Given a set of N data points, $\{P_i\}_{1 \leq i \leq N}$, let $f(\hat{X}, s) = 0$ is the generic equation describing the shape of the measured surface where $\hat{X}$ is the space vector ($\hat{X} = (\hat{x}, \hat{y}, \hat{z})$ in Cartesian coordinates) and s denotes the shape parameters. For each point P_i, one could associate its orthogonal projection on the nominal surface denoted by Q_i. $d_i = \|P_i - Q_i\|$, where $\|.\|$ is the Euclidean norm, which defines the orthogonal distance between the measured data points P_i and the nominal surface, known as form deviation (Figure 1).
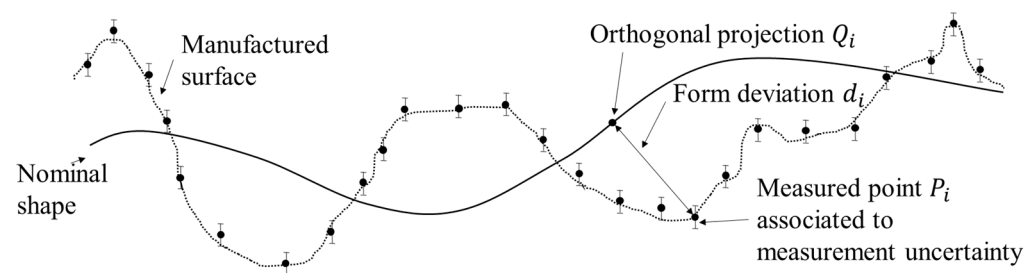
Figure 1. Definition of the form errors (or deviation errors).

The objective function to minimize for LS (resp. MZ) is given in (1) (resp. (2)). $x = (m, s) \in \mathbb{R}^n$ could be either the set of intrinsic shape parameters s, the motion parameters m: rotation and translation applied to $\{P_i\}$, or both.

$$\min_{x} \sum_{i=1}^{N} (\|P_i - Q_i\|)^2 \tag{1}$$

$$\min_{x} \max_{1 \leq i \leq N} \|P_i - Q_i\| \tag{2}$$

The formulations (1) and (2) represent different mathematical properties. Those problems were extensively studied for simple geometries and a number of techniques were developed, as reported in [15–18]. Existing methods like variants of the Gauss–Newton [8] could be applied to solve the smooth least squares objective function. The objective function in the second MZ problem (2) is not differentiable; thus, a number of differentiable optimization techniques cannot be applied. Numerous MZ fitting methods were suggested for the case of straightness, flatness, roundness and cylindricity tolerances [19–21]; however, rare methods were proposed for the MZ fitting of complex shapes. Nevertheless, some attempts were made in [22–24], where techniques such as smoothing functions, primal-dual interior point methods or differential evolution algorithms were adapted to aspherical shapes.

3. Validation of MZ Fitting Algorithms

The need for fitting algorithm validation in metrology was initiated and supported in [3,12]. The aim is to ensure that MZ fitting algorithms return correct values. One methodology to assess the results returned by metrology algorithms is to use a reference pair. This methodology known as "*Type F1 algorithm measurement standards*" is defined in ISO 5436-2 2012 [11]. Even if this *Type F1 standard* is common in surface texture domain, its concept could be extended to MZ fitting algorithms. *Type F1 standards* could be considered as a numerical representation of the measured part to which we associate a reference measure and value known with a given uncertainty. For the evaluation, reference softgauges are inputted to the metrology algorithm under test, the returned value is compared to the reference measure, and then a decision could be made as to whether the metrology algorithm is accepted or rejected (Figure 2).
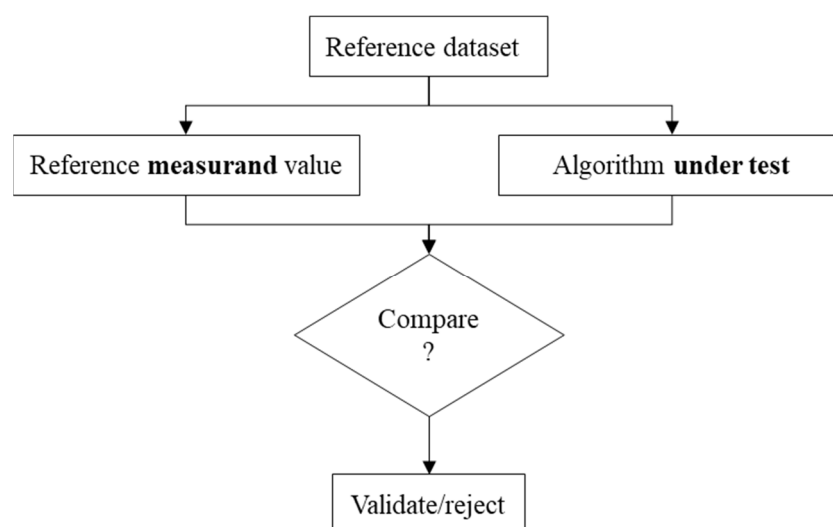
Figure 2. Illustration of the type F1 algorithm measurement standard.

A second methodology to evaluate metrology algorithms is based on the use of reference algorithms defined in ISO 5436-2 2012 as *"Type F2 algorithm measurement standards"* [11]. There are traceable metrology algorithms against which the tested algorithm will be compared. A common set of data points is submitted to both metrology algorithms (reference algorithm and algorithm under test) and the two results are then compared in order to take an accept/reject decision (Figure 3). Reference algorithms do not exist for a wide range of applications in metrology. Their development is not always straightforward, especially for applications such as MZ fitting.
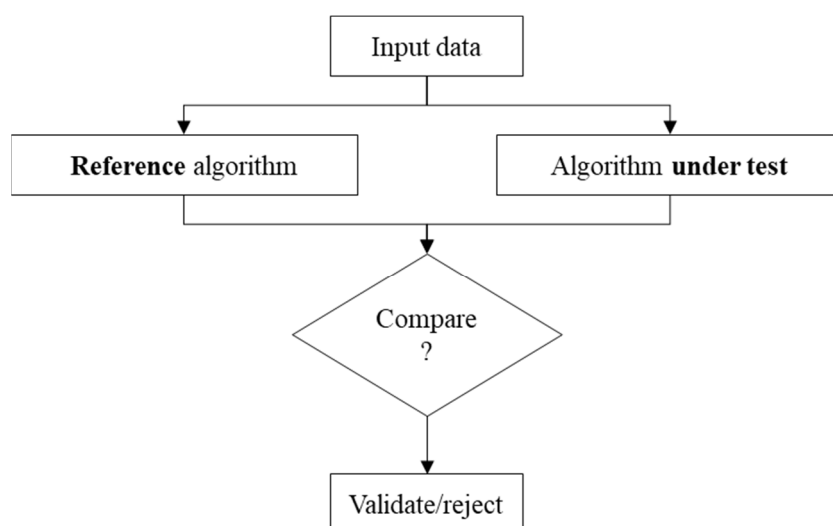

Figure 3. Illustration of the type F2 algorithm measurement standard.

For MZ fitting, Forbes et al. [1] developed a method for the generation of reference softgauges with vertex solution. Similarly to LS fitting, the Karush–Kuhn–Tucker (KKT) optimality conditions for the MZ fitting problem are indispensable and the data points are generated such as to meet the number of conditions reported by Boyd et al. [25]. A set of points are selected on the nominal shape, such that the linear independence constraint qualification (LICQ) of the problem holds [26]. Then, equations resulting from KKT conditions are solved in order to determine the Lagrangian multiplier [25]. If the resulting Lagrange multipliers are not positive (dual feasibility does not hold), a new set of data points are chosen and the procedure is repeated. Once the dual feasibility condition is

satisfied, the contacting points are constructed and additional random points are generated on the surface. A flowchart describing the proposed approach is illustrated in Figure 4.
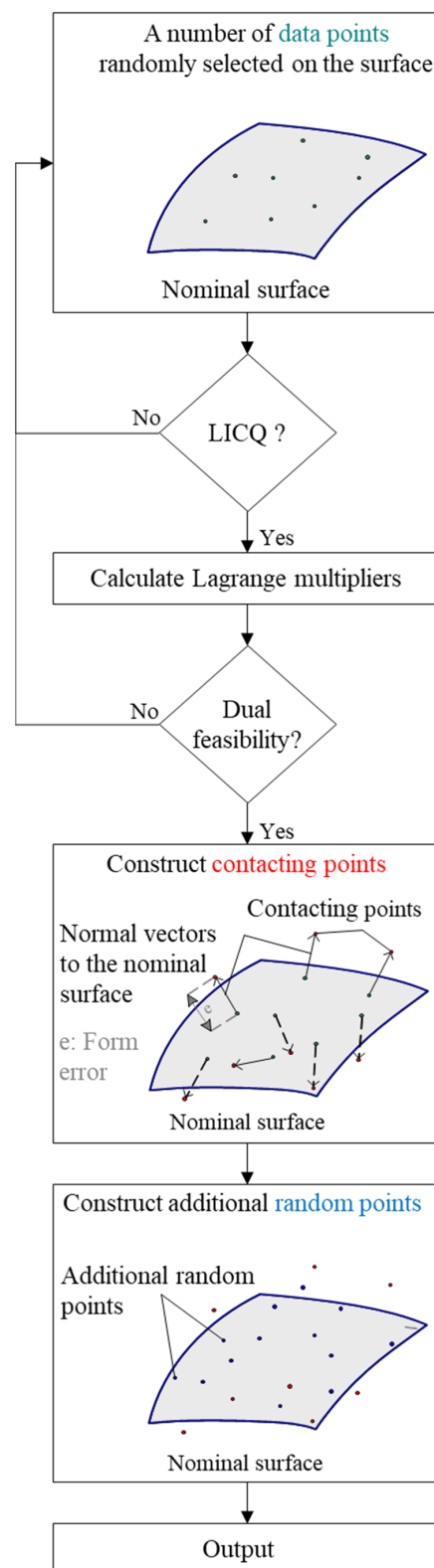

Figure 4. Flowchart describing the generation of reference softgauges for MZ fitting.

4. Mathematical Formulation of the Reference Softgauges Generation for MZ Fitting

Based on Figure 5, MZ fitting is formulated as a nonlinear programming (3).

$$\min_{x,e} e \quad \text{such that} : -e \leq d_i(x) \leq e \; \forall i \in \{1, \ldots, N\} \tag{3}$$

where $e \geq 0$ is the form error, N the number of measured points and $d_i = \|P_i - Q_i\|$.
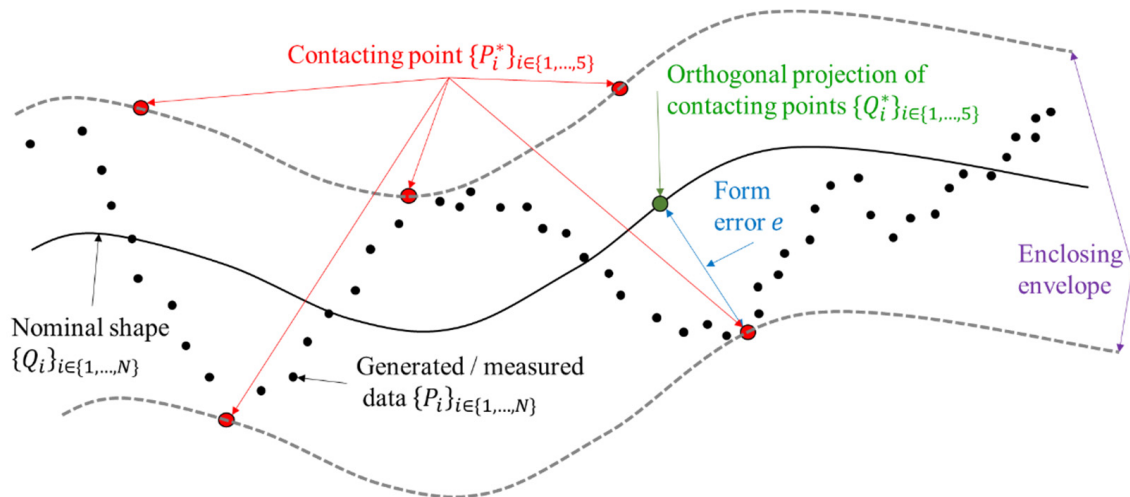

Figure 5. Illustration of five contacting points to the enclosing envelope. Three located on the upper bound and two on the lower bound.

The problem (3) could be formulated in the standard form given in (4).

$$\min_{y} f(y) \quad \text{subject to } C_i(y) = \left\{ \begin{array}{l} C_i^+(y) \geq 0 \\ C_i^-(y) \geq 0 \end{array} \right. \; \forall i \in \{1, \ldots, N\} \tag{4}$$

where $y = (e, x) \in \mathbb{R}^{n+1}$, $C_i^+(y) = e - d_i(x)$, $C_i^-(y) = e + d_i(x)$ and $f(y) = e$.

Let y^* be a local minimum of the problem (4). Hence, y^* could be feasible when it satisfies the condition $C_i(y^*) \geq 0$, $\forall i \in \{1, \ldots, N\}$.

At a feasible point y^*,

$$\left\{ \begin{array}{ll} C_i(y^*) = 0 & i^{th} \text{ constraint is active} \\ C_j(y^*) > 0 & j^{th} \text{ constraint is inactive} \end{array} \right. \tag{5}$$

An active constraint could be interpreted as a point belonging to the measured data for which the distance to the reference surface is equal to the form error ("*a contacting point to the enclosing envelope*") as illustrated in Figure 5. Whereas, this distance is lower than the form error for inactive constraint.

4.1. Vertex vs. Non-Vertex Solution

A solution y^* is said to be vertex if the number of active constraints is greater or equal to $n + 1$ (n is the number of unknowns parameters). If the number of active constraints is strictly less than $n + 1$, y^* is said to be a non-vertex solution. By definition, a constraint $C(y) \geq 0$ is active at y^* only if $C(y^*) = 0$.

For canonical surfaces, the MZ fitting is almost formulated as a linear programming [26] and the solution could be at some vertex of the feasible domain (Figure 6). For complex surfaces, the solution could be either at some vertex of the feasible domain, or at a face or an edge (non-vertex solution (Figure 7)).
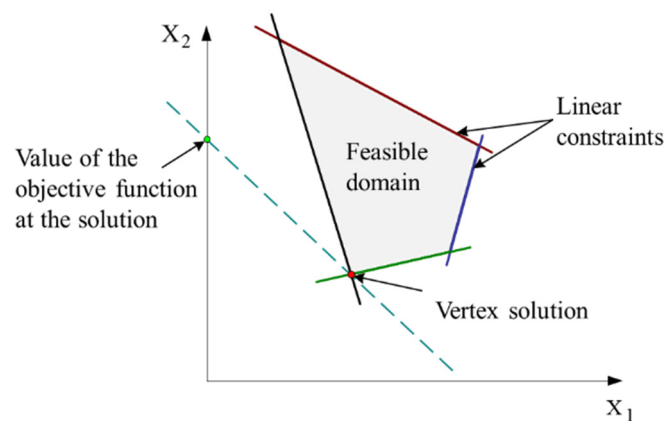
Figure 6. Representation of the vertex solution for the case of a linear programming involving the two variables X_1 and X_2.
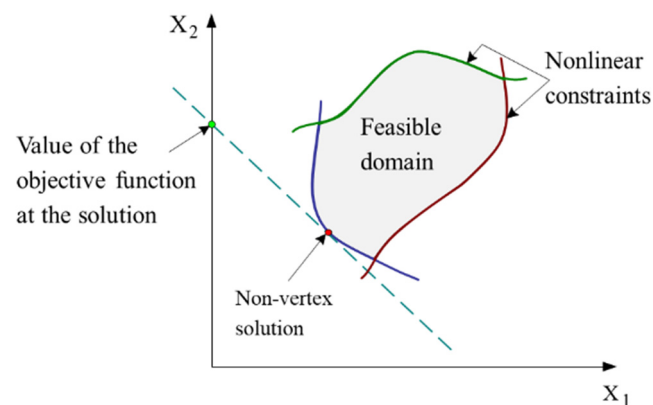

Figure 7. Representation of the non-vertex solution for the case of a nonlinear programming involving the two variables X_1 and X_2.

If at P^* there can be no binding feasible descent direction (Figure 5), then P^* solves the equality constrained sub-problem in (6).

$$\min_{y} f(y) \text{ subject to } C_i(y) = 0, \ i \ \epsilon \ I^* \tag{6}$$

where I^* is the set of active constraint.

Therefore, P^* is considered as a local solution of the equality constrained problem (6), in particular when there exist Lagrangian multipliers λ^* for which (P^*, λ^*) is a stationary point of the Lagrangian $L(x, \lambda)$ defined in (7).

$$L(x, \lambda) = f(x) - \Sigma_{i \in I^*} \lambda_i C_i(x) \tag{7}$$

Considering the first order derivative of Lagrangian $L(x, \lambda)$, both λ^* and P^* should satisfy the KKT Equation (8).

$$\begin{cases} \nabla_x f(P^*) = \sum_{i \in I^*} \lambda_i^* \nabla_x C_i(P^*) \\ C_i(P^*) = 0, \quad i \in I^* \end{cases} \tag{8}$$

4.2. Optimality Conditions

If the number p (belongs to I^*) of active constraints is less than n (number of unknowns parameters), let $Z = z(a)$ be the $n \times (n - p)$ orthogonal complement to C^* the matrix of gradients $\nabla_a C_i$, $i \in I^*$, so that $Z_t C^* = 0$. Hence, the second order optimality conditions are required.

- The necessary conditions for P^* to be a local minimiser are:
  - ○ feasibility: $C_i(p_i^*) \geq 0 \; \forall i \in \{1, \ldots, N\}$
  - ○ first order: if I^* is the set of active constraints at p^*, there exist a Lagrangian multiplier λ^* for which:

$$\begin{cases} \nabla_x f(P^*) = \sum_{i \in I^*} \lambda_i^* \nabla_x C_i(P^*) \\ \lambda_i^* \geq 0; \; i \in I^* \end{cases} \tag{9}$$

  - ○ second order: if $p < n$, the matrix $Z(P^*)^t W(P^*, \lambda^*) Z(P^*)$ is positive semi-definite *(W is the Hessian matrix of the Lagrangian (7))*.

- The sufficient conditions for P^* to be a local minimizer under constraint qualification are:
  - ○ feasibility: $C_i(P^*) \geq 0 \; \forall i \in \{1, \ldots, N\}$
  - ○ first order: if I^* is the set of active constraints at P^*, there exist a Lagrange multiplier λ^* for which:

$$\begin{cases} \nabla_x f(P^*) = \sum_{i \in I^*} \lambda_i^* \nabla_x C_i(P^*) \\ \lambda_i^* > 0; \; i \in I^* \end{cases} \tag{10}$$

  - ○ Second order: if $p < n$, the matrix $Z(P^*)^t W(P^*, \lambda^*) Z(P^*)$ is positive definite.

4.3. Reference Softgauges Generation for Complex Surfaces: Aspherical Shapes

A method to generate reference softgauges is introduced for the case of complex surfaces when motion parameters are sought. Since the aspherical lenses are rotationally symmetric surfaces, only five motion parameters are unknown $x = \{T_X, T_Y, T_Z, \theta_X, \theta_Y\}$ (translations in X, Y and Z directions as well as rotations around X- and Y-axis), then a non-vertex solution consists of, at most, five contacting points (active constraints). The proposed algorithm includes 7 steps as follows:

Step 1: five points $\{Q_i^*\}_{i=1,\ldots,5}$ are randomly selected on the nominal aspherical surface (Figure 5). These points represent the orthogonal projections of the contacting points $\{P_i^* = (x_i^*, y_i^*, z_i^*)\}_{i=1,\ldots,5}$ onto the nominal shape.

Step 2: an index $\alpha_i \in \{0, 1\}$ is associated to each point in $\{Q_i^*\}_{i=1,\ldots,5}$. A point with $\alpha_i = 1$ represents the orthogonal projection of a contacting point assigned to the lower surface $(\mathcal{S}^-)$, i.e., for which $d_i(Q_i^*, P_i^*) = -e$. Those with $\alpha_i = 0$ are assigned to the upper $(\mathcal{S}^+)$ i.e., for which $d_i(Q_i^*, P_i^*) = e$ (e is the desired form error).

Step 3: verify that the gradient vectors of distances with respect to motion parameters $\nabla d_i(Q_i^*, P_i^*)$ are linearly independent. Otherwise, go to step 1. The gradient vectors are calculated using equations given from (11) to (15).

$$\frac{\partial d_i}{\partial T_X} = (-1)^{\alpha_i} n_{i,X} \tag{11}$$

$$\frac{\partial d_i}{\partial T_Y} = (-1)^{\alpha_i} n_{i,Y} \tag{12}$$

$$\frac{\partial d_i}{\partial T_Z} = (-1)^{\alpha_i} n_{i,Z} \tag{13}$$

$$\frac{\partial d_i}{\partial \theta_X} = (-1)^{\alpha_i} (Q_i^* \times n_i).e_X \tag{14}$$

$$\frac{\partial d_i}{\partial \theta_Y} = (-1)^{\alpha_i} (Q_i^* \times n_i).e_Y \tag{15}$$

where $n_i = (n_{i,X}, n_{i,Y}, n_{i,Z})$ is the normal vector to the nominal shape at the point $\{Q_i^*\}_{i=1,...,5}$, $a \times b$ defines the cross product of the two vectors a and b. e_X (resp.e_Y) is the unit vector with respect to the X-axis (resp. Y-axis).

Step 4: determine Lagrangian multipliers $\lambda^* = (\lambda_1^*, \ldots, \lambda_5^*)$ by solving the quadratic programming given in (16).

$$\min_{\lambda} ||G\lambda - b|| \; st. \; \lambda \geq 0 \tag{16}$$

where G is a 6×5 matrix, such that $G = (g_1, \ldots, g_5)$, $g_i^T = ((-1)^{\alpha_i} \nabla d_i^T, 1)$ and $b^T = (0,0,0,0,0,1)$. If $||G\lambda - b|| > \epsilon$, where ϵ is a predefined parameter, go to step 1. This step represents the resolution of the equation resulting from KKT conditions of the problem given in (3) cited in [14].

Step 5: determine a nonzero null space vector p such that $G^T p = 0$.

Step 6: verify that $p_b^T H p_b > 0$ where p_b is the vector composed of the first five elements of p and H is the Hessian of the Lagrangian given in (17).

$$H = \sum_{i=1}^{5} (-1)^{\alpha_i} \lambda_i^* \nabla^2 d_i \tag{17}$$

If this condition is not satisfied, then go to step 1. Otherwise, calculate contacting points coordinates by setting $P_i^* = Q_i^* + (-1)^{\alpha_i} e n_{i \; (i=6,..,N)}$.

Step 7: generate additional random points $\{P_i\}_{i=6,..,N}$ such that $P_i = Q_i + \theta_i n_i$ with $\{\theta_i\}_{i=6,...,N}$ is a set of randomly selected numbers in the domain $[-e, e]$.

It is worth mentioning that the proposed robust reference softgauges algorithm could work on all complex surfaces, as the only required knowledge is the normal vector with respect to the X-, Y- and Z-axis. Furthermore, the developed approach could be applied to continuous nominal shape described by a mathematical model or to discrete high accurate measured dataset.

5. Hybrid Trust Region vs. Exponential Penalty Function for Minimax Fitting

5.1. Hybrid Trust Region (HTR) Algorithm

The hybrid trust region algorithm consists of performing either the trust region step, line search step or curve search step, according to the specific situation faced at each iteration [8,27,28]. It enables one to avoid having to solve the trust region problem many times. Each iteration relies on obtaining a trust region trial step d_k by solving the following quadratic problem (QP) given in (18).

$$\min_{(d \epsilon \mathbb{R}^{n+1})} \frac{1}{2} < d, B_k d > + z = M_k(d, z),$$
$$S.t < \nabla f_i(x_k, d) > - z \leq \phi(x_k) - f_i(x_k), \; i = 1, \ldots, m \tag{18}$$
$$d_{k\infty} \leq \Delta_k$$

B_k is $(n \times n)$ symmetric definite matrix, Δ_k is user-defined coefficient of the domain of the trust region, z is a parameter that depends on the first derivative of φ, meanwhile $< .,. >$ is the dot product. The trust region domain is defined using MZ instead of LS; thus, QP becomes easily solved. The QP in (18) always has a solution, since (0,0) lies inside the feasible domain. This QP could be solved using adapted methods, such as interior point method [25]. If the resulting trust region trial step d_k could not be accepted, a corrected step $d_k + \tilde{d}_k$ is determined by solving the QP in (19).

$$\begin{cases} \min_{(\tilde{d} \epsilon \mathbb{R}^{n+1})} \frac{1}{2} < d_k + \tilde{d}, B_k \left(d_k + \tilde{d} \right) > + \tilde{z} = \widetilde{M}_k \left(\tilde{d}, \tilde{z} \right), \\ S.t < \nabla f_i \left(x_k, \tilde{d} \right) > - \tilde{z} \leq \phi(x_k + d_k) - f_i(x_k + d_k), \; i = 1, \ldots, m \\ d_k + \tilde{d}_{\infty} = \Delta_k \end{cases} \tag{19}$$

If the corrected $d_k + \widetilde{d}_k$ is not accepted in the trust region scheme, then a line search or curve search along d_k is performed when d_k is a descent direction $r_k > 0$. If $(r_k \leq 0)$, a step length is sought by performing a curve search that verifies (20).

$$\begin{cases} \phi\left(x_k + t_k d_k + t_k^2 \widetilde{d}_k\right) \leq \phi(x_k) - \alpha t_k < d_k, B_k d_k > \\ \alpha \in \left[0, \frac{1}{2}\right] \end{cases} \qquad (20)$$

d_k is the solution of (18) and $\widetilde{d}_k$ is the solution of (19). In this case, $\|d_k\| \leq \|\widetilde{d}_k\|$, $\widetilde{d}_k$ should be taken to be 0. Additional details are discussed in [8,28] and the following flowchart (Figure 8) illustrates how the HTR algorithm performs.
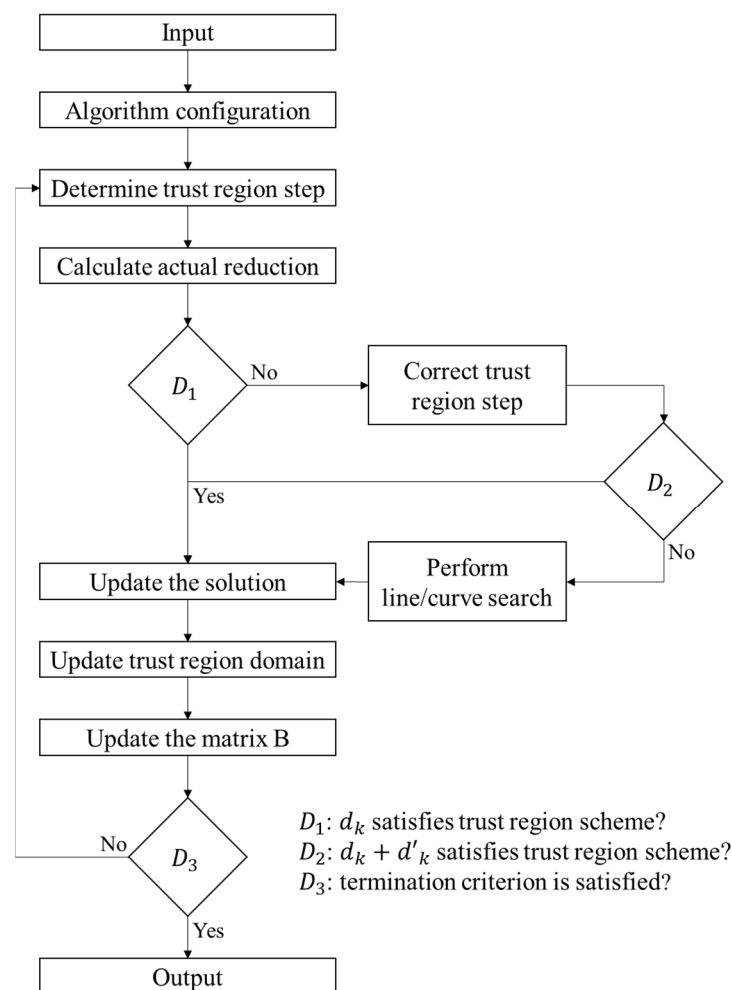

Figure 8. Flowchart of the hybrid trust region (HTR) algorithm.

5.2. Exponential Penalty Function (EPF) Algorithm

Exponential penalty function is a technique that aims at approximating the non-smooth objective function in the MZ fitting by a parameterized smooth function. The resulting smooth function is optimized using Newton-based methods [23,27,29]. Let (21) be the approximation function, $\{d_i(x)\}$ are the set of distances between the measured data and the reference surface.

$$min \Phi_{MZ} = \max_{1 \leq i \leq N} d_i(x) \qquad (21)$$

Let F_p be the continuously differentiable approximation function and $(p > 0)$ the smoothing parameter, as in (22). One can demonstrate the inequality given in (23).

$$F_p = \frac{1}{p} Log \left(\sum_{i=1}^{N} \exp(pd_i(x)) \right) \tag{22}$$

$$\forall x \in \mathbb{R}^n, \ F(x) \leq F_p(x) \leq F(x) + \frac{LogN}{p} \tag{23}$$

When the value of p goes to infinity, F_p converges to F. For a given value of the smoothing parameter p, minimisation of F_p is carried out through derivative-based methods. Once the optimum of F_p is found, p is multiplied by a user-defined coefficient, and a new approximation function is formulated. This process is repeated until the resulting approximation function F_p becomes sufficiently close to the original objective function F.

5.3. Numerical Validation on Aspherical Shapes and Discussion

The developed method for reference softgauges generation based on a non-vertex solution is applied for the case of aspherical shapes. An asphere could be defined as a rotationally symmetric surface with a radius of curvature that varies gradually from the centre of the lens (Figure 9). Optical aspherical surfaces are very popular in the domain of optical design regarding their superiority over classical spherical lenses, as well as advancements made in manufacturing techniques [30]. Several formulations could be employed for the description of asphere; however, the most widely used is the one given in ISO 10110-12:2007 [30] called monomial formulation, given in (24). This formulation depends on the sag of the surface parallel to the radial symmetric axis z, radius r, radius of curvature R, conic constant κ and monomial coefficients a_{2m+4}.

$$z(r) = \frac{r^2}{R \left(1 + \sqrt{1 - (1+\kappa)\frac{r^2}{R^2}} \right)} + \sum_{m=0}^{M} a_{2m+4} \ r^{2m+4} \tag{24}$$
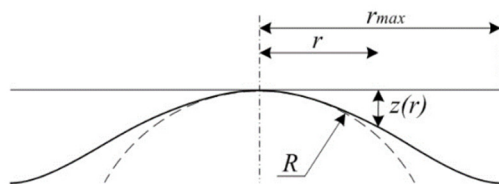

Figure 9. Description of aspherical shapes.

Three configurations are selected, as given in Table 1. For each configuration, a dataset with predefined number of points $N = \{10{,}404, \ 50{,}024, \ 100{,}489\}$ and $MZ_{ref} = 10^{-4}$ mm are generated to evaluate the performance of the two implemented MZ fitting algorithms (HTR and EPF).

Table 1. Nominal values of the aspheric shapes parameters.

Configuration	R (mm)	k	a_4 (mm^{-3})	a_6 (mm^{-5})	a_8 (mm^{-7})	a_{10} (mm^{-9})
I	19.79	−0.9	-1.5×10^{-17}	-7.55×10^{-18}	-3.77×10^{-18}	-1.88×10^{-19}
II	8.88	−0.8	-1.9×10^{-12}	-9.72×10^{-13}	-4.86×10^{-13}	-2.43×10^{-13}
III	4.14	−0.9	-4.1×10^{-12}	-2.08×10^{-12}	-1.04×10^{-12}	-5.21×10^{-13}

The corresponding PV values respectively PV_{EPF} (or MZ_{EPF}) and PV_{HTR} (or MZ_{HTR}), as well as execution time, respectively T_{EPF} and T_{HTR}, are compared with respect to the

methodology presented in Figure 10. Initial data is rotated by angle $\pi/20$ around X-axis and $\pi/15$ around Y-axis, as well as translated by -1 mm in X-axis, 1 mm in Y-axis and -1 mm in Z-axis. Both HTR and EPF algorithms were implemented on a computer based on Intel Core i7/x64 platform with 16 GB of RAM and a 2.30 GHz processor.
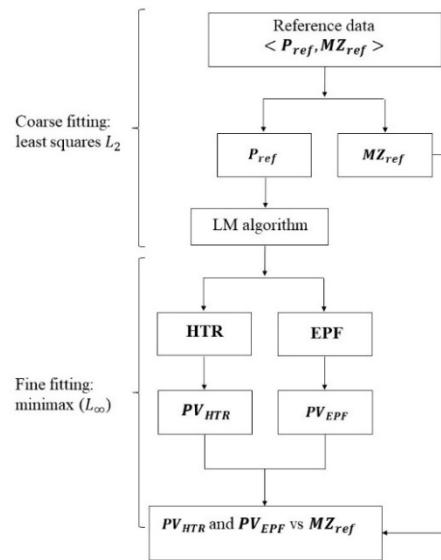

Figure 10. Comparison methodology (LM algorithm: Levenberg–Marquardt algorithm).

Nine reference aspherical softgauges were generated and Figures 11 and 12 illustrate only one generated softgauges with five contacting points. Each generated softgauge was submitted to both HTR and EPF fitting algorithms, and Tables 2–4 summarize the returned results of MZ-MZ_{ref} and the execution time for configurations I, II and III.
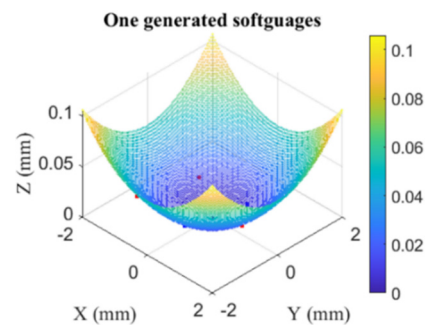

Figure 11. One generated reference softgauges with a non-vertex solution (five contacting points).
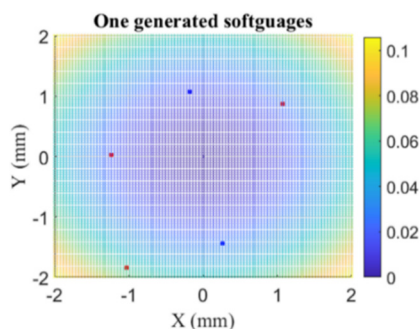

Figure 12. Illustration of the five contacting points (red and blue). Red indicates contacting points for which the form error is equal to e. Blue indicates form error equals to $-e$.

Table 2. Values of MZ-MZ$_{\text{ref}}$ and execution time (s) for HTR and EPF (configuration I).

N	$MZ_{HTR} - MZ_{Ref}$ (mm)	$MZ_{EPF} - MZ_{Ref}$ (mm)	T_{HTR} (s)	T_{EPF} (s)
10,404	1.78×10^{-16}	4.51×10^{-16}	12.1	61.91
50,024	1.71×10^{-16}	4.78×10^{-16}	24.5	135.71
100,489	1.64×10^{-16}	4.92×10^{-16}	41.51	226.96

Table 3. Values of MZ-MZ$_{\text{ref}}$ and execution time (s) for HTR and EPF (configuration II).

N	$MZ_{HTR} - MZ_{Ref}$ (mm)	$MZ_{EPF} - MZ_{Ref}$ (mm)	T_{HTR} (s)	T_{EPF} (s)
10,404	1.41×10^{-15}	3.08×10^{-15}	4.27	36.50
50,024	3.78×10^{-16}	4.78×10^{-16}	10.78	157.71
100,489	$6.73 \times 1\,0^{-15}$	6.78×10^{-15}	20.52	255.06

Table 4. Values of MZ-MZ$_{\text{ref}}$ and execution time (s) for HTR and EPF (configuration III).

N	$MZ_{HTR} - MZ_{Ref}$ (mm)	$MZ_{EPF} - MZ_{Ref}$ (mm)	T_{HTR} (s)	T_{EPF} (s)
10,404	1.15×10^{-16}	2.14×10^{-15}	35.88	200.16
50,024	4.78×10^{-16}	1.78×10^{-16}	42.35	240.45
100,489	9.86×10^{-15}	1.05×10^{-14}	50.03	274.75

According to Tables 2–4, it seems that HTR and EPF fitting algorithms return quite similar results of MZ, accurate at the sub-nanometre level, with a noticeable upper hand of HTR when the execution time is regarded. This is due to EPF approximation of the non-smooth objective function in the MZ fitting by one parametrized smooth function. Then, the applied Newton method requires the computation of the Hessian matrix, which is proportional to the number of points in the softgauges. Moreover, the accuracy of the obtained descent direction is not always guaranteed. Hence, corrections must be brought to the Hessian matrix whenever needed.

Regarding the HTR algorithm, the matrix B_k is chosen to be symmetric positive definite while setting up the QP, through Powell modification BFGS (Broyden–Fletcher–Goldfarb–Shanno) formula. Therefore, there is no need to compute the second order derivation terms, which considerably reduces the execution time.

6. Difficulty Degree and Performance Measure

Measuring an algorithm quality could be established by setting a framework, including: (1) the preconditions for quality measurement, (2) the analysis strategy of the meaningfulness of quality measures, as well as (3) the interpretation and use of the measured values [31].

The degree of difficulty was discussed by Cox and Harris in [32]. It aims at defining a quantity associated to each dataset that indicates at which level the generated reference softgauges challenge the algorithm under test. Thus, the methodology of testing metrology MZ fitting algorithms consists of generating datasets with increasing difficulty number. The output of the algorithm under test is assessed using a performance measure on each set. Therefore, the degree of difficulty could be sought as the difficulty to converge to a global optimum. Usually, the selected feature could help in predicting the difficulty of the MZ fitting problem. It regroups: (1) the nature of the solution, (2) the number of points affecting tremendously both execution time as well as the accuracy, (3) the initial position of the measured data compared to the final position.

The suggested degree of difficulty denoted by λ is given in (25).

$$\lambda = \beta_1 V + \beta_2 \frac{N}{N_0} + \beta_3 \frac{\Theta}{\Theta_0} \tag{25}$$

where β_1, β_2 and β_3 are user-defined parameters, such that: $\sum_{i=1}^{3} \beta_i = 1$, V is a binary variable that indicates whether the set of data is vertex or non-vertex, N is the number of the data points in the set and N_0 is the maximum number of points that could be handled by the algorithm under test. Θ is the measure of the initial position. Θ_0 is the estimation of the maximum initial alignment, calculated by taking the norm of the vector of maximum permissible translation and rotation.

The performance measure For MZ fitting could be considered as the difference between the minimum zone value returned by the algorithm under test and the reference value. When execution time is involved, the performance measure could combine both result accuracy and execution time. In addition, the heuristic/deterministic aspect of the algorithm under test should be considered, because deterministic algorithms are more appreciated for metrology software. The latter requirement could be assembled in a sample of performance measure denoted η given in (26).

$$\eta = \alpha_1 f_1(E, \lambda) + \alpha_2 f_2(T, \lambda) + \alpha_3 f_3(\Delta, \lambda) \tag{26}$$

where α_1, α_2 and α_3 are user-defined parameters, such that: $\sum_{i=1}^{3} \alpha_i = 1$. The expression of the functions f_1, f_2 and f_3 are given in (27)–(29), respectively.

$$f_1(E, \lambda) = \begin{cases} \frac{E_0(\lambda)}{E}, & if\ E > E_0(\lambda) \\ 1 \ , & \text{otherwise} \end{cases} \tag{27}$$

$$f_2(T, \lambda) = \begin{cases} \frac{T_0(\lambda)}{T}, & if\ T > T_0(\lambda) \\ 1 \ , & \text{otherwise} \end{cases} \tag{28}$$

$$f_3(\Delta, \lambda) = \begin{cases} 0 \ , & \text{if heuristic} \\ 1 \ , & \text{if deterministic} \end{cases} \tag{29}$$

E_0 and T_0 are the recommended error and execution time of each value of the number of difficulty, while T and E are the actual execution time and the *MZ* value.

Figure 13 illustrates the performance of a given *MZ* fitting algorithm under test, as well as the acceptance domain for reference algorithms and operating ones. Since an algorithm of reference and an operating one do not have the same operational requirements, these two types of algorithms present two different characteristics in the degree of difficulty and performance measure domain.
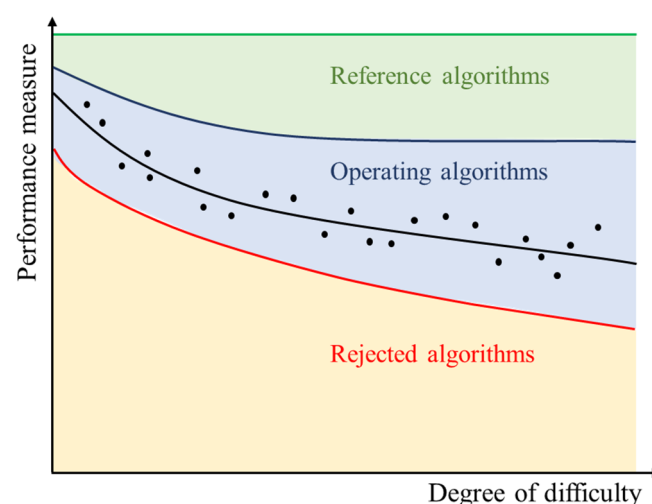

Figure 13. Performance measure vs. of degree of difficulty for a given algorithm under test.

7. Analysis of the Results and Discussion

With the aim of defining the weights associated to the samples of performance measure and the degree of difficulty, a survey was conducted involving 45 engineers and researchers from industrial and academic fields, NMIs (National Metrology Institutes) and others.

The survey is based on questions regarding how important it is that an algorithm gives accurate results, runs in short time and returns deterministic or heuristic results. Then, the weights α_1, α_2 and α_3 were estimated based on the survey's answers (Table 5).

Table 5. Estimation of the weights based on the survey answers.

	Associated Weights	
	Reference algorithm	Operating algorithm
α_1	0.41	0.4
α_2	0.24	0.28
α_3	0.35	0.32

EPF is considered as an operating algorithm and the determination of β_1, β_2 and β_3 is given as follow: $\beta_1 = 0.5$, $\beta_2 = 0.5$ and $\beta_3 = 0$ (since the coarse fitting has taken place before proceeding to fine fitting). The recommended execution time and the form error were determined through L_2- fitting (LS).

The function that determines the accepted limits could be formulated as a convex combination of the form $tb + (1 - t)a$, where a is the performance measure of the algorithm under test for the first set, $b = a - \epsilon$, ϵ is the user-defined coefficient and $t \in [0, 1]$.

The obtained performance measure results are presented in Figure 14. Therefore, HTR is as stable as an algorithm of reference could be. This is due to its accurate MZ_{HTR} values and the execution time that was not affected by the number of points in the dataset. EPF is unstable with an increased degree of difficulty, as it is sensitive to the number of points in the dataset.
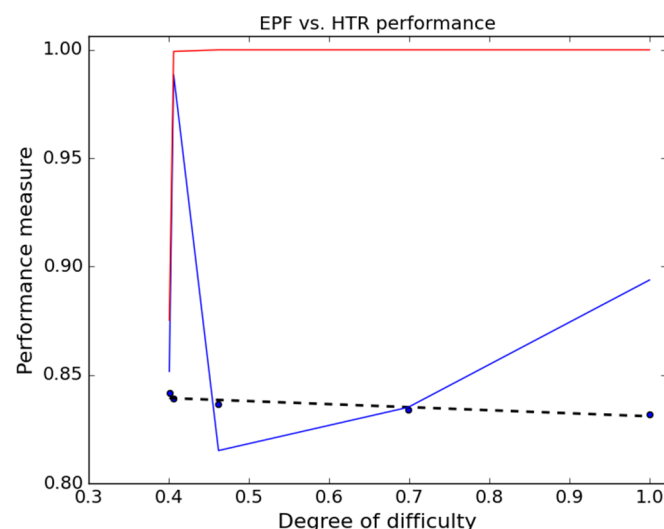

Figure 14. EPF vs. HTR performance in blue and red (respectively). In black: the fitted line defines the accepted limits of the performance measure.

Nevertheless, vertex or non-vertex solution cannot be the only concern when constructing reference softgauges for MZ fitting. There are still some requirements needed before proceeding to softgauges generation; in particular, the scope and characteristics of the algorithm under test must be clearly identified. Then the abilities claimed by the algorithm should be determined so that task-specific data points are generated and the algorithm to test is not "disfavoured".

These characteristics may include:

- **Uniqueness of the solution:** generated reference softgauges must have a unique associated reference solution [33]. For fitting problems, reference softgauges with two different "substitute geometry" will be problematic, especially if these geometries will be taken as datum. These situations must be avoided when generating reference softgauges.

- **Number of points:** the number of points contained in the reference softgauges must not exceed the maximum number of points that could be handled by the algorithm.

- **Initial alignment:** initial position of measured data highly affects the performance of MZ fitting algorithms. Some algorithms could only perform if the input data are close to the solution position.

- **Error-free:** validation of metrology MZ fitting algorithms must be performed in the perfect operator approximation. This means that measuring errors (or errors from any origin) must not be embedded in constructed data. Adding measuring errors to the reference softgauges might induce some difficulties, since we cannot tell whether inaccuracy results from measurement or processing.

- **Stability:** a small perturbation in the designed data must not affect the reference value of the measurand. The analysis of an MZ fitting algorithm stability was related to the values of Lagrangian multipliers [11,34].

- **Uncertainty:** it should be associated to reference softgauges. Considering the data coordinates $\left(x_i^*, y_i^*, z_i^*\right)$ and targeted form error e_t expressed in double precision as well as the LICQ and dual feasibility conditions satisfied, the actual value of form error e is given in (30).

$$e = \sqrt{\left(x_i - x_i^*\right)^2 + \left(y_i - y_i^*\right)^2 + \left(z_i - z_i^*\right)^2} \tag{30}$$

with $x_i = x_i^* + e_t n_{i,X}$, $y_i = y_i^* + e_t n_{i,Y}$ and $z_i = z_i^* + e_t n_{i,Z}$

Uncertainties of x_i^*, y_i^* and z_i^* and e_t depend on the accuracy of the machine architecture, while uncertainties of x_i, y_i, z_i, $n_{i,X}$, $n_{i,Y}$ and $n_{i,z}$ could also be calculated analytically using propagation rules. Thus, the uncertainty of the actual form error e denoted by $u(e)$ could be estimated.

8. Conclusions

In this paper, the generation of reference softgauges dedicated to the assessment of MZ fitting algorithms of complex shapes is provided. The developed robust approach is based on satisfying KKT first and second order optimality conditions before inferring reference softgauges with non-vertex solutions. It could be applied to continuous nominal shape described by a mathematical model or to discrete high accurate measured dataset.

The implemented approach based on a non-vertex solution was adopted and investigated for the case of aspherical shapes. Nine reference softgauges were generated with predefined number of points and $MZ_{ref} = 10^{-4}$ mm. The reference softgauges were submitted to two implemented MZ fitting algorithms (HTR and EPF). Results show the conformance of the returned values of MZ_{HTR} and MZ_{EPF} with the reference measurand ones MZ_{ref}. However, the performance of the HTR fitting algorithm in terms of execution time is noticeable in comparison to the EPF one.

Two metrics were introduced to define the performance measure in the function of the degree of difficulty to measure the performance of an MZ fitting algorithm. As these two metrics contains some arbitrary variables, a survey based on a number of industrial and academic professionals has allowed us to determine these weights. An application on HTR and EPF fitting algorithms was carried out such as to measure their performances. The given results reveal that EPF performs poorly when the number of points is higher, especially when the execution time is taken into consideration.

Requirements of the generated reference softgauges were also discussed in this article. In fact, generated softgauges must be aligned with characteristics of the MZ fitting algorithm under test. Data points' extraction, number of points, stability of the solution,

etc. must be taken into account when generating softgauges. Most of the considered requirements need further research. Therefore, future work will mainly cover original methods for assessing the stability of generated softgauges, in particular for the case of complex shapes.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Forbes, A.B.; Minh, H.D. *Form Assessment in Coordinate Metrology*; Springer: Singapore, 2011; pp. 69–90.
2. Anthony, G.T.; Bittner, B.; Butler, B.P.; Cox, M.G.; Drieschner, R.; Elligsen, R.; Forbes, A.B.; Gross, H.; Hannaby, S.A.; Harris, P.M.; et al. *Chebyshev Reference Software for the Evaluation of Coordinate Measuring Machine Data, October 1993, Report/Guide (Technical Report)*; Commission of the European Communities: Brussels, Belgium, 1993.
3. Cheung, C.; Ren, M.; Kong, L.; Whitehouse, D. Modelling and analysis of uncertainty in the form characterization of ultra-precision freeform surfaces on coordinate measuring machines. *CIRP Ann.* **2014**, *63*, 481–484. [CrossRef]
4. Hopp, T.; Levenson, M. Performance-Measures for Geometric Fitting in the NIST Algorithm Testing and Evaluation Program for Coordinate Measurement Systems. *J. Res. Natl. Inst. Stand. Technol.* **1995**, *100*, 563–574. [CrossRef] [PubMed]
5. Monahan, J.F. *Numerical Methods of Statistics*; Cambridge University Press: Cambridge, UK, 2001.
6. Hutzschenreuter, D.; Härtig, F.; Wendt, K.; Lunze, U.; Löwe, H. Online Validation of Chebyshev Geometric Element Algorithms Using the TraCIM-System. *J. Mech. Eng. Autom.* **2015**, *5*, 94–111.
7. Al-Subaihi, I.; Watson, G.A. The Use of the l1 and l∞ Norms in Fitting Parametric Curves and Surfaces to Data. *Appl. Numer. Anal. Comput. Math.* **2004**, *1*, 363–376. [CrossRef]
8. Arezki, Y.; Nouira, H.; Anwer, N.; Mehdi-Souzani, C. A novel hybrid trust region minimax fitting algorithm for accurate dimensional metrology of aspherical shapes. *Measurement* **2018**, *127*, 134–140. [CrossRef]
9. Bui, S.H.; Vorburger, T.V. Surface metrology algorithm testing system. *Precis. Eng.* **2007**, *31*, 218–225. [CrossRef]
10. Linares, J.; Goch, G.; Forbes, A.; Sprauel, J.; Clément, A.; Haertig, F.; Gao, W. Modelling and traceability for computationally-intensive precision engineering and metrology. *CIRP Ann.* **2018**, *67*, 815–838. [CrossRef]
11. ISO. *ISO 5436-2:2012—Geometrical Product Specifications (GPS) Surface Texture: Profile Method*; Measurement Standards—Part 2: Software Measurement Standards; ISO: Geneva, Switzerland, 2012.
12. Forbes, A.B.; Minh, H.D. Generation of numerical artefacts for geometric form and tolerance assessment. *Int. J. Metrol. Qual. Eng.* **2012**, *3*, 145–150. [CrossRef]
13. Arezki, Y.; Mehdi-Souzani, C.; Anwer, N.; Nouira, H. Reference data simulation for L∞ fitting of aspheres. *Procedia CIRP* **2018**, *75*, 331–336. [CrossRef]
14. Murthy, T.; Rao, S.; Peters, J. A Simple Approach for Evaluation of Cylindrical Surfaces. *CIRP Ann.* **1981**, *30*, 441–444. [CrossRef]
15. Shunmugam, M. Comparison of linear and normal deviations of forms of engineering surfaces. *Precis. Eng.* **1987**, *9*, 96–102. [CrossRef]
16. Murthy, T.S.R. A comparison of different algorithms for cylindricity evaluation. *Int. J. Mach. Tool Des. Res.* **1982**, *22*, 283–292. [CrossRef]
17. Nouira, H.; Bourdet, P. Evaluation of roundness error using a new method based on a small displacement screw. *Meas. Sci. Technol.* **2014**, *25*, 044012. [CrossRef]
18. El-Hayek, N.; Nouira, H.; Anwer, N.; Gibaru, O.; Damak, M. A new method for aspherical surface fitting with large-volume datasets. *Precis. Eng.* **2014**, *38*, 935–947. [CrossRef]

19. Gosavi, A.; Cudney, E. Form Errors in Precision Metrology: A Survey of Measurement Techniques. *Qual. Eng.* **2012**, *24*, 369–380. [CrossRef]

20. Murthy, T.; Abdin, S. Minimum zone evaluation of surfaces. *Int. J. Mach. Tool Des. Res.* **1980**, *20*, 123–136. [CrossRef]

21. Shunmugam, M.S. New approach for evaluating form errors of engineering surfaces. *Comput. Aided Des.* **1987**, *19*, 368–374. [CrossRef]

22. Zhang, X.; Jiang, X.; Scott, P.J. A minimax fitting algorithm for ultra-precision aspheric surfaces. *J. Phys. Conf. Ser.* **2011**, *311*, 012031. [CrossRef]

23. Zhang, X.; Zhang, H.; He, X.; Xu, M.; Jiang, X. Chebyshev fitting of complex surfaces for precision metrology. *Measurements* **2013**, *46*, 3720–3724. [CrossRef]

24. Zhang, X.; Jiang, X.; Forbes, A.B.; Minh, H.D.; Scott, P.J. Evaluating the form errors of spheres, cylinders and cones using the primal–dual interior point method. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2013**, *227*, 720–725. [CrossRef]

25. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004; ISBN 9780521833783.

26. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2000.

27. Arezki, Y.; Zhang, X.; Mehdi-Souzani, C.; Anwer, N.; Nouira, H. Investigation of minimum zone assessment methods for aspheric shapes. *Precis. Eng.* **2018**, *52*, 300–307. [CrossRef]

28. Wang, F.W.A.C.; Wang, C. A New Trust-Region Algorithm for Finite Minimax Problem. *J. Comput. Math.* **2012**, *30*, 262–278. [CrossRef]

29. Jørgensen, M. Software quality measurement. *Adv. Eng. Softw.* **1999**, *30*, 907–912. [CrossRef]

30. ISO. *ISO 10110-12:2007—Optics and Photonics—Preparation of Drawings for Optical Elements and Systems—Part 12: Aspheric Surfaces*; ISO: Geneva, Switzerland, 2007.

31. Butler, B.; Cox, M.; Forbes, A.; Hannaby, S.; Harris, P. A methodology for testing classes of approximation and optimization software. In *Quality of Numerical Software*; Springer: Boston, MA, USA; pp. 138–151.

32. Cox, M.; Harris, P. Design and use of reference data sets for testing scientific software. *Anal. Chim. Acta* **1999**, *380*, 339–351. [CrossRef]

33. Bertsekas, D.P. Nonlinear Programming. *Athena Sci.* **1995**, *48*, 334.

34. Xiao, Y.; Yu, B. A truncated aggregate smoothing Newton method for minimax problems. *Appl. Math. Comput.* **2010**, *216*, 1868–1879. [CrossRef]