Contribution à la reconstruction de surfaces complexes à partir d'un grand flot de données non organisées pour la métrologie 3D.

Nadim El-Hayek

2014-ENAM-0055

École doctorale n° 432 : Sciences des Métiers de l'Ingénieur

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers

Spécialité " Mécanique et Matériaux "

présentée et soutenue publiquement par

Nadim EL HAYEK

le 18 Décembre 2014

Contribution à la reconstruction de surfaces complexes à partir d'un grand flot de données non organisées pour la métrologie 3D.

(Contribution to complex surfaces reconstruction from large and unorganized datasets for 3D metrology)

Directeur de thèse : **Olivier GIBARU**

Co-encadrement de la thèse : **Nabil ANWER, Mohamed DAMAK, Hichem NOUIRA**

Jury

M. Marc HIMBERT, Professeur, CNAM, directeur du Laboratoire Commun de Métrologie (LCM)	Président
M. Jean-Marc LINARES, Professeur, ISM, Aix-Marseille Université	Rapporteur
M. Jean-François FONTAINE, Professeur, Le2i, Université de Bourgogne	Rapporteur
M. Marc DANIEL, Professeur, LSIS, Ecole PolyTech Marseille	Examinateur
M. Olivier GIBARU, Professeur, LSIS, Arts et Métiers ParisTech	Examinateur
M. Hichem NOUIRA, Chercheur, LNE Paris, Laboratoire Commun de Métrologie (LCM)	Examinateur
M. Nabil ANWER, Maître de Conférences, LURPA, ENS de Cachan	Examinateur
M. Mohamed DAMAK, PAST, LSIS, Arts et Métiers ParisTech / Président de Geomnia	Examinateur
M. Nacim IHADDADENE, Enseignant chercheur, ISEN, Université Catholique de Lille	Invité

T H È S E

Arts et Métiers ParisTech - Centre de Lille
Laboratoire des Sciences de l'Information et des Systèmes (LSIS)

Remerciements

A la fin de cette thèse, qui marque une étape importante de ma vie et qui a permis d'enrichir mon expérience aux niveaux professionnel et personnel, je ne peux que remercier tous ceux qui ont participé et contribué à la réussite de ce travail.

Je tiens à exprimer tout d'abord mes remerciements aux membres du jury : M. ***Marc Himbert****, Professeur au CNAM et directeur du Laboratoire Commun de Métrologie, pour avoir accepté de présider le jury, M.* ***Jean-François Fontaine****, Professeur au Le2i de l'Université de Bourgogne ainsi que M.* ***Jean-Marc Linares****, Professeur à l'ISM d'Aix-Marseille Université, pour avoir accepté d'être les rapporteurs de ce manuscrit, M.* ***Marc Daniel****, Professeur au LSIS de l'Ecole PolyTech Marseille, pour avoir accepté d'évaluer le travail réalisé, et finalement, M.* ***Nacim Ihaddadene****, Maître de Conférences à l'ISEN de l'Université Catholique à Lille, pour avoir accepté l'invitation au jury et pour m'avoir accordé un temps précieux afin de me transmettre de son expertise en informatique et de ses qualités scientifiques.*

Je souhaite également remercier mon directeur de thèse ***Olivier Gibaru****, professeur au LSIS d'Arts et Métiers ParisTech pour le temps qu'il a pu m'accordé ainsi que pour ses qualités pédagogiques et scientifiques. J'adresse mes remerciements sincères à mes encadrants* ***Nabil Anwer****, Maître de Conférences au LURPA de l'ENS de Cachan,* ***Mohamed Damak****, Président de la société Geomnia et* ***Hichem Nouira****, chercheur au LNE Paris. Leur support scientifique, leur encouragement mais surtout leur motivation durant toute la thèse ont vu leur fruit dans un travail abouti. Je ne peux manquer de mentionner aussi leur investissement dans la valorisation du travail réalisé.*

Je n'oublierai jamais les moments chaleureux passés avec mes chers collègues de l'ENSAM, Laurent Gajny, Franck Hernoux, Juliette Morin, Alain Vissière, Ky Nguyen, Sijun Liu, Laure Arbenz, Adel Olabi, Karama Sriti Guimbal, Pierre Rault, Pierre Besset, Hussein Zahr, Tiago Moraes, Bassel Aslan, Paul Sandulescu, Fabien Minguet, Marta Berardengo, Marouene Oueslati et Mac Duy Hung.

Et comment oublierais-je mon frère et mes parents, qui étaient toujours là pour m'encourager. Leur solide soutien au fil des années a été plus qu'apprécié et nécessaire.

Contents

Nomenclature

BIPM: Bureau International des Poids et Mesures

BS: Base Surface

CAD: Computer Aided Design

CMI: Czech Metrology Institute

CMM: Coordinate Measuring Machines

DBACD: Discrete B-Spline Active Contour Deformation algorithm

IBSPE: IBS Precision Engineering

ICP: Iterative Closest Point

JRP: Joint Research Project

L-BFGS: Limited memory Broyden-Fletcher-Goldfarb-Shanno

LM: Levenberg-Marquardt

LNE: Laboratoire National d'Essais (or French National Metrology Institute)

METAS: Federal Institute of Metrology

MKEH: Hungarian Metrology Institute

MRF: Magnetorheological Finishing

NMI: National Metrology Institute

PCA: Principal Components Analysis

PTB: Physikalish-Technische Byndesanstalt (or German National Metrology Institute)

PV: Peak-to-Valley

QR: Q-R Decomposition

RMS : Root Mean Square

SI: International System of Units

SMD: Belgian National Metrology Institute

SPDT: Single Point Diamond Turning

SPM: Scanning Probe Measurement

SVD: Singular Value Decomposition

TNO: Netherlands Organization for Applied Scientific Research

TWI: Tilted-Wave Interferometer

UHPMM: Ultra High-Precision Measuring Machines

VIM: International Vocabulary of Metrology

VSL: Dutch Metrology Institute

Chapter 1

PSI: Phase-Shifting Interferometer

LPIB: Local Polishing Ion Beam

LPPB: Local Polishing Plasma Beam

RH: Relative Humidity

α: Machine table axes perpendicularity angles

k: Coverage factor

Δ: Spatial sampling step

d: Tactile probe tip diameter

$f_{spatial}$: Spatial frequency

F: Implicit classical asphere function

$P^{(\alpha,\beta)}(x)$: Jacobi polynomial basis

Chapter 2

P: Data points

$\boldsymbol{P_i^{Car}}$: Data point in Cartesian coordinates (X_i, Y_i, Z_i)

$\boldsymbol{P_i^{Cyl}}$: Data point in Cylindrical coordinates (R_i, Θ_i, Z_i)

$\boldsymbol{p}$: Projection of a data point on the theoretical model

$\boldsymbol{p^o}$: Orthogonal projection of a data point on the theoretical model

$\boldsymbol{p^v}$: Vertical projection of a data point on the theoretical model

d^o: Orthogonal distance between a data point and its orthogonal projection point

d^v: Vertical distance between a data point and its vertical projection point

d_{pm}: Point-to-mesh distance

R: Rotation matrix representing the rotational transformation parameters

$\boldsymbol{R_\beta}$: Rotation matrix about the x-axis

$\boldsymbol{R_\gamma}$: Rotation matrix about the y-axis

T: Translation vector

t_x, t_y, t_z: Components of the translation vector

F: Aspherical model implicit function

r: Cylindrical r-coordinate of the asphere equation

z: Cylindrical z-coordinate of the asphere equation

θ: Cylindrical θ-coordinate of the asphere equation (since asphere is axis-symmetric, this coordinate is a dummy variable)

$\boldsymbol{\alpha}$: classical asphere model parameters

$\mathbf{b}$: Forbes asphere model parameters

c: Curvature at the asphere apex

κ: Conic constant of the asphere

Q^{con}: Forbes *strong* asphere polynomials

Q^{bfs}: Forbes *mild* asphere polynomials

$\vec{t}$: Tangent vector to the asphere surface at a certain point and in a given rz-plane

H: Hessian matrix

J: Jacobian matrix

ϕ: Objective function for L-BFGS, LM and ICP

S: Newton-Raphson's footpoint computation objective function

g: Gradient

x: Solution of the objective function minimization problem

d: Descent direction

α: Step length in the descent direction

λ: The Levenberg-Marquardt parameter

$\varepsilon_1, \varepsilon_2, \varepsilon_3$: Stop criteria for optimization algorithms

Chapter 3

d: Degree of the B-Spline curve

n_p: Number of data points

m: Size of knot vector

n_C: Number of control points

L: Level of subdivision

n_S: Number of subdivided points

n_d: Number of distances data

P: Data points

Seg(**P**): Segmentation of **P**

C: Control points

Q: Subdivided polygon points

Seg($\mathbf{Q}$): Segments of the subdivided polygon

$\mathbf{M}$: Subdivision matrix from $\mathbf{C}$ to $\mathbf{Q}$

$\mathbf{M_u}$: Subdivision matrix in the u-direction

$\mathbf{M_v}$: Subdivision matrix in the v-direction

$\mathbf{N}$: Normal orientations of control points $\mathbf{C}$

$\boldsymbol{\varepsilon}$: Error vector from $\mathbf{Q}$ to $\mathbf{P}$

$\boldsymbol{\delta}$: Distance vector in the direction of $\boldsymbol{\varepsilon}$

$\mathbf{t}$: Translation vector of control points

ϵ: Algorithm's stop criterion (mean of residual errors)

$\mathbf{A}$: Optimization matrix without fairing

λ: Fairing parameter

τ: Knot sequence

$\mathcal{C}''(u)$: Smooth B-Spline curve's second derivative

$N_{j,\tau}^{d''}(u)$: B-Spline basis functions' second derivatives

$\boldsymbol{\mathcal{N}}_\tau$: Diagonal matrix of second derivative B-Spline basis functions

$\mathbf{B}$: Optimization matrix with fairing

$\boldsymbol{\zeta}$: Right-hand side of the linear system to solve with fairing

General Introduction

With the advances of technology, the trend in the design of parts leans more and more towards the use of complex shapes. We use the term complex shapes when we want to refer to surfaces exhibiting variations in shape and geometry and which reveal hurdles in their manufacturing and measurement. Another term to refer to complex surfaces is freeform surfaces. Among the wide variety of freeform shapes families which are classified with respect to their size, shape complexity and relative tolerance in regard to manufacturing and measurement, optics and aerofoil parts are of particular interest to us in this thesis.

Optics have seen remarkable changes implying better performance and optimized optical systems in various fields such as metrology, photonics, energy, medicine, ophthalmology… [1]. The shapes that characterize the family of optics today are freeform optical surfaces and aspheres. Aspherical optics are a subset of freeform optics with the particularity of being described by a specific mathematical formula. Although aspherical surfaces can have a revolute invariance degree, they are of complex geometry, i.e. a composition of two geometries such as, for example, a conic part and a polynomial part. Advances in manufacturing as well as measurement capabilities have allowed designers to conceive complex optical surfaces with improved functionalities and performances. Their shape/form is controlled with a nanometric level of accuracy thanks to the development of dedicated Ultra High-Precision Measuring Machines (UHPMM).

Freeform surfaces have also seen outstanding advances and find an application in a multitude of fields like the automotive, aircraft and energy (wind turbines) industries, etc… Freeform parts have come to enhance the products and systems they are integrated in, thanks to the development of new geometries. They have allowed improvements in the designs and performances of aircrafts and car bodies while reducing fuel consumption and ecological footprint. Freeform and more particularly aerofoil parts, have completely different manufacturing processes and different metrology systems, and by that they are at a different level of complexity. The sought precision for freeform surfaces is rather sub-micrometric. Freeform surfaces are mainly measured using scanning contact probe based Coordinate Measuring Machines (CMM) which are a physical representation of Cartesian coordinates in space.

Due to the difference in precision sought and geometry, we want it to be clear in this work that we will deal with aspherical optics and turbine blades in disconnected manners. In fact, the thesis has come up to be about complex surfaces reconstruction and metrology in

general because there are two motives: a European joint research project and an industrial project.

In order to promote leadership in high-end optics, the European Metrology Research Programme (EMRP) has launched a project (IND10: FORM) entitled "*Optical and tactile metrology for absolute form characterization*", with the aim of developing methods for the measurement and form characterization of aspherical surfaces with a nanometric level of uncertainty. The project is coordinated by the German national metrology institute, PTB (Dr. Michael Schulz), and involves many National Metrology Institutes (NMIs) such as LNE (France), VSL (Netherlands), METAS (Switzerland), SMD (Belgium), CMI (Czech Republic), MKEH (Hungary) as well as academic (ITO from Stuttgart University (Germany), Fraunhofer-Institut für Produktionstechnologie IPT (Germany), Technical University of Ilmenau (Germany), XPRESS Precision Engineering (Netherlands)) and industrial partners (TNO (Netherlands), IBSPE (Netherlands)).

Nowadays, there are no known methods capable of achieving very low levels of uncertainty in the case of aspherical surfaces. The French National Metrology Institute (LNE), in collaboration with Arts et Métiers ParisTech (ENSAM), Ecole Normale Supérieure de Cachan (ENS Cachan) and Geomnia company, is a major actor on a work package of this project dealing with the comparison of optical and tactile measurements of aspherical surfaces. It is also a leader of another work package focusing on data analysis and the development of software for form characterization of aspheres. The work of this thesis is mainly focused on the second work package. While inspection on simple shapes is mastered and well understood, the objective of project IND10 is to assess the measurement capabilities regarding complex shapes and to evaluate aspherical form. Generally speaking, form refers to the shape at the macroscopic scale and corresponds to the spatial wavelengths that are larger than 1 mm for aspherical and freeform optics [2].

Geomnia, a company with core skills in 3D metrology participates in this project for its expertise and know-how in metrology systems engineering and software solutions and because one of its recent interests is the characterization of gas turbine blades. Turbine blades characterization methods are not standardized and not robust but are only based on common practices.

In both cases, the characterization of complex surfaces involves data processing and brings up the problem of surface reconstruction. Surface reconstruction is an extensive field

of research long studied by different communities such as Computer Graphics, Reverse Engineering and Metrology. Each of these disciplines defines different tools and methodologies to solve the inverse problem. The process of surface reconstruction starts with a cloud of points and aims at retrieving the shape of the underlying object represented by the points. The Computer Graphics community seeks to build a piecewise-linear interpolation of the points by constructing a mesh. The Reverse Engineering community performs fitting of CAD models as an approximation of the points. The metrologists look at the problem completely differently and fit known mathematical models to the data points. So, a metrology application holds if and only if a nominal model of the measured surface is known.

With the technological advances of instruments and systems, some optical measurements can generate very large volumes of data exceeding a million points within a short time (typically few seconds). The resulting cloud of points is a set of unorganized points. The connectivity between points is not inherent to the cloud of points; therefore, the latter does not infer any knowledge pertaining to the geometry and the topology of the underlying surface. Moreover, the cloud of points is **noisy**, might contain **outliers**, and can present regions of **overlap** after aligning multiple raw scans, making the process of surface reconstruction even more elaborate. Ideally, a measurement would lead to data that fall exactly on the surface. However, in the real world, one observes the presence of parasite data and noise which add to the relevant data. Noise can either be internal, related to the electronic apparatus, or external, related to the surrounding environment. As for an outlier, it is a point situated remarkably far from the rest of the data points. Its manifestation might be a result of a measurement error or a false measurement due to an obstacle that blocks the intended region to be measured.

Moreover, due to the complexity of the intended complex shapes, measurement and processing times are quite long. What we seek here is a fast processing algorithm that would be at least equivalent in time to the measurement time. The algorithm must also be evaluated and validated for its intended purpose.

How to get a surface reconstruction algorithm from large amounts of data to be - for metrology purposes - automatic in favor of unorganized noisy data, robust against outliers and fast relative to measurement time?

This research work tries to bring solutions to this issue and addresses the problems related to the measurement systems and data processing as well. The measurement systems cover a wide range of different techniques varying from contact to contactless techniques. In

this project, we deal with both tactile (stylus) and optical single point and line measurements (confocal and laser triangulation). These measurement probes can be mounted on UHPMM, CMM, robotic arms or can even be held in hand. Each measurement system has its own specificities. Some of these might be, for example, the ability to generate structured lines of scans, XY scans in grids, spiral-like scans, or even completely unorganized 3D scans. Hence, any knowledge about the measurement system, or the measurand, has an influence on the reconstruction process as well as on the choice of the reconstruction algorithm.

Metrology reveals the hidden structures and geometrical aspects (form and dimensions) of an object, and by that, verifies that it is conform to its design specifications. Measurement techniques using CMMs are copiously applied today because they are accurate, reliable and traceable (Fig. 1 and Fig. 2). However, the uncertainty budget of a measurement can merely be established for regular geometry parts.

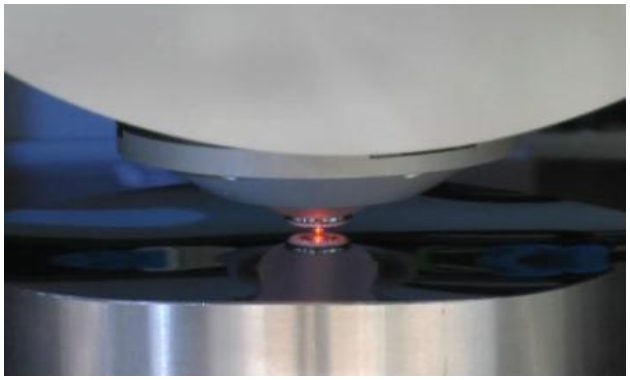


Fig. 1 Asphere measurement on the Nanomefos ultra-high precision machine (TNO) [2].

Fig. 2 Turbine blade measurement by laser triangulation scanning (Nikon metrology labs).

Standards replicating regular geometries such as spheres and cylinders exist. They are manufactured with fine quality, calibrated by means of primary instruments and then employed as reference standards for CMM calibration. The feature of a standard is that it has completely known dimensions and form. When a machine is calibrated using a standard, the machine's measurement becomes traceable according to the International System of Units (SI) meter definition given by the Bureau International des Poids et Mesures (BIPM) (Fig. 3). The task of the BIPM is to ensure world-wide uniformity of measurements and their traceability to the SI. The latest adopted meter definition suggests that the meter is the length of the path travelled by light in vacuum during a time interval of $1/299{,}792{,}458$ of a second [3].
Nevertheless, when dealing with freeform shapes, the Guide to the expression of Uncertainty in Measurement is not easily applicable to complex measurement processes on CMMs due to

the absence of universal calibrated workpieces (standards) [4,5]. Since a freeform surface has no such parameters like diameter or well-known features, the creation of calibrated workpieces out of freeform shapes represents a major challenge [6]. Some task-specific artifacts have been proposed in [7].

When ultra high-precision is demanded, CMMs are not helpful anymore and are replaced by UHPMMs. The main asset of UHPMMs is that they do not require calibration based on standards because measurement here occurs with direct comparison to primary instruments mounted in-situ. Primary instruments, such as laser interferometers are at a higher level on the traceability pyramid (Fig. 3).

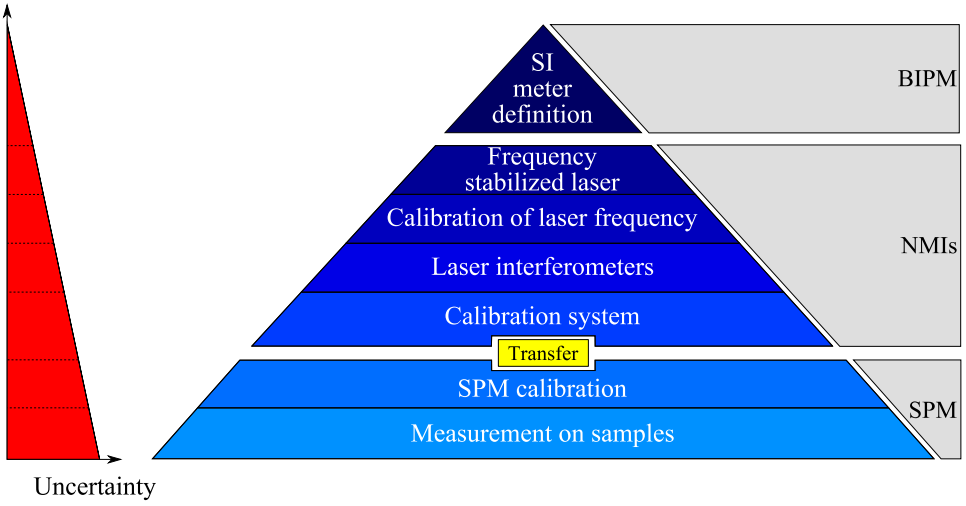


Fig. 3 Metrology traceability pyramid (SPM: Scanning Probe Measurement).

The measuring machines (CMM and UHPMM) main issue is that they constitute a time consuming two-fold process. The first part of the process is a set-up procedure where the object is aligned with respect to the machine's reference frame. Alignment is key to achieve high precision in measurement because if the part reference and the machine measuring head reference coordinate systems are not well aligned, the measurement contains errors. We distinguish three different alignment procedures. A visual alignment operation that uses feelers and a light source, a physical operation where the part is mounted on pre-installed accurate fixtures (Fig. 4), and a mathematical operation that performs registration of the part with respect to its CAD model. The visual alignment is an old method that is still used nowadays but only in applications where very high-precision is not required. It is also still used in some applications involving the measurement of turbine blades [8].

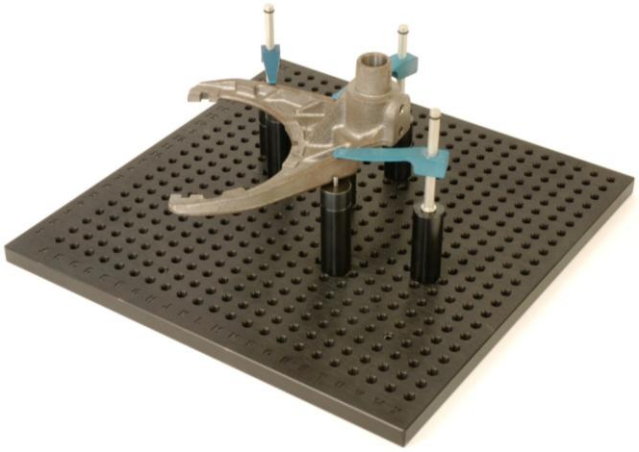


Fig. 4 Example of an alignment of a part (aluminum material) using accurate fixtures (blue).

The physical alignment is a method used for the repeated measurement of similar parts and especially for parts which geometry is perfectly known. The fixtures are usually designed and pre-installed on the measuring machine in a way to hold the part and have its reference coordinate system well aligned with the reference coordinate system of the machine [9]. However, the work of Zhu *et al* confirms that precision in inspection of freeform surfaces is difficult to achieve when dealing with accurate fixtures [10]. The mathematical alignment, also known as the indirect comparison process [8], is the most accurate process among all and is generally done using the Iterative Closest Point (ICP) method or some adapted variants of it [8,10–13]. Here again the computer aided processing is effective and more precise.

The second part of the process is the measurement itself and it is usually done point-by-point or by sets of 2D scans, not always representative of a 3D object's complete geometry. The corresponding maximum data acquisition rate is around 50 to 60 points per minute [6]. Today's metrology involves more advanced measurement devices that lean towards being independent of alignment, acquiring data at very fast rates of 20,000 points/s, and are in most cases, non-contact optical systems. They offer the possibility to convey very large datasets containing more than 100,000 points but they drop in precision. Although it is not our concern here, we bring to mind this trend in metrology instruments to say that in case high-precision is not sought, technologies generating unorganized and large datasets exist and build up the difficulty regarding surface reconstruction. 3D scanners make up a recent measurement capability with increased effectiveness when working with complex shapes. However, with the non stopping emergence of a multitude of sensors and technologies, 3D scanning can hardly have a unified and well-determined calibration process as well as a traceable measurement. In some specific cases, measurement is a fusion of a multitude of scans,

eventually coming out from different instruments, and this is indeed another issue with 3D scanning.

Two types of complex surfaces serve as a support for this research.

I. Aspherical surfaces:
    A. the AO775 aspherical lens model which is manufactured by Anteryon® company and measured at LNE on the high-precision profilometer using a tactile probe [14]. The resulting dataset contains around 1,500,000 points.
    B. the same lens is measured at LNE on the same machine with a confocal probe. The resulting dataset contains around 1,000,000 points. This measurement aims at comparing optical and tactile measurements in regard to measurement uncertainty and to robustness of surface reconstruction.
    C. the same lens is measured by other IND10 partners (VSL, METAS, TNO and IBSPE).

Aspherical models are known and defined as axis-symmetric surfaces in ISO 10110-12 [15]. The challenge in regard to surface reconstruction of aspheres is that they contain a great number of parameters, and with a large number of points, optimization algorithms are slow.

II. Freeform surfaces, where a dataset of a turbine blade measured by laser triangulation is described (Fig. 2). For this dataset, four scans have been merged together to obtain a cloud of points containing 127,000 points distributed in lines of scan. The difficulties regarding surface reconstruction here are the abundance of noise, the regions of overlap due to the fusion of data and the disparity in the spatial frequency of the points.

Freeform surfaces do not usually have a mathematical model. In fact, they can sometimes be defined by a CAD model but in the scope of this work, we are interested in working on freeform surfaces without any given model. For the specific example of blades and the problem encountered by Geomnia [16], where profile characterization at a certain elevation is sought, a continuous representation of the underlying surface is of primary importance so that any cross section can be deduced by extracting the intersection between the reconstructed surface and a predefined cutting plane (Fig. 5). If the reconstructed surface is a mesh, the intersection between the cutting plane and the mesh will make room for a piecewise linear

curve that would allow extracting the dimensions needed (Fig. 6). That same curve can be smoothened if the latter operation guarantees extracting the same dimensions more precisely. It is only by performing the above procedure that a metrologist can estimate the dimensions sought by the designer at any cross-section of the blade (Fig. 5 and Fig. 6) [8].

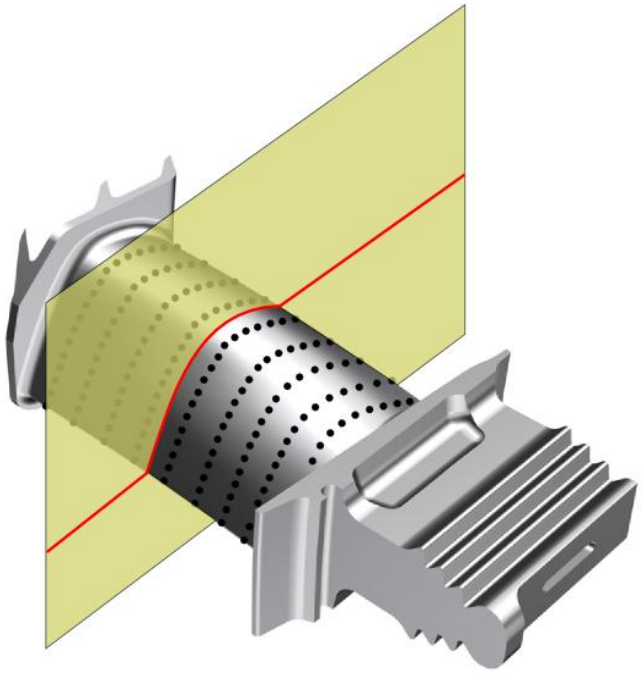


Fig. 5 Cross-sectional characterization of turbine blades.

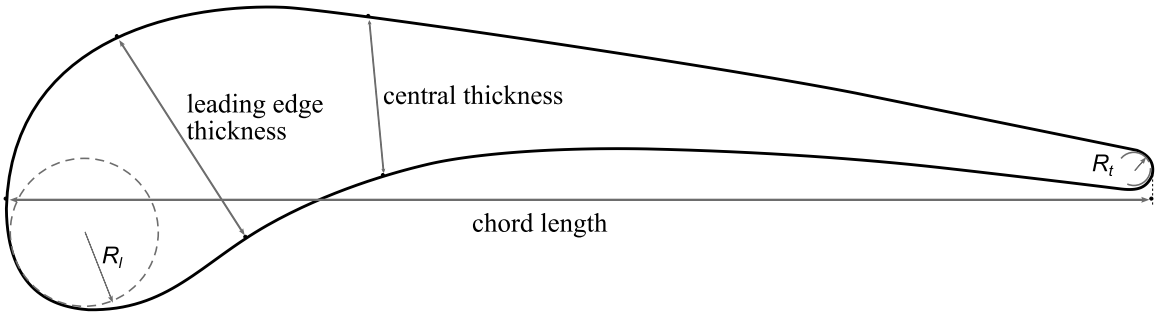


Fig. 6 Some geometrical dimensions defined on a turbine blade profile (R_l: leading edge radius of curvature; R_t: trailing edge radius of curvature).

Regardless of the measurement technique used, the process of characterizing a surface urges the need to reconstruct one and have a continuous representation of it. With an unorganized 3D point set, the determination of nearest points is hard. Surface reconstruction in computer graphics solves this issue by creating a piecewise linear surface known as a mesh. The mesh is a data structure that has several uses. On the one hand, it is used to partition the

points space more clearly and then to associate parametric patches to each partition so that a smooth representation of the surface is created [17–23]. These algorithms transform triangular meshes into quadrilateral meshes which are more regular and susceptible to patching. Then a parametric base function is fitted to each mesh patch. The issues encountered here are the generation of a regular and consistent quadrangular mesh and the establishment of the connectivity among patches. On the other hand, a mesh can be used to calculate differential metrics on the point set, such as normals and curvatures. By computing such information, filtration of undesired and irrelevant points can be done. A first order approximation is not sufficient to determine intrinsic smooth surface metrics. When smooth surfaces are necessary, the approach cited above is one way of performing association of smooth surfaces. Otherwise, a new active contour deformation approach based on B-Splines will be detailed later and applied directly on the unorganized points.

The aim of this thesis is to be able to associate/fit a surface to the data in order to be able to proceed to the deviation analysis, the estimation of form defects and the inspection of optical and mechanical freeform surfaces. The metrology of aspheres is done through the association of a mathematical model to the points. Depending on the fitting criterion used, or what is widely known as the norm of minimization, form metrology is assessed differently since it is achieved by computing the minimal zone that would contain all the points in the dataset. According to the standards, form metrology makes use of the infinite norm L_∞ in order to calculate the envelope enclosing the points in the dataset by minimizing the difference between the maximum deviation and the minimum deviation between the surface and the dataset. Whilst it is not deterministic for complex surfaces [24,25], the fitting problem is still solved by the Least-Squares norm (Fig. 7). Although it is not a rigorous parameter to depict form defects, current research practices calculate the Peak-to-Valley (PV) error from the difference between the maximum and minimum residual errors. In fact, the commonly used L_2 method is prone to over-estimation so it remains a safe approach.

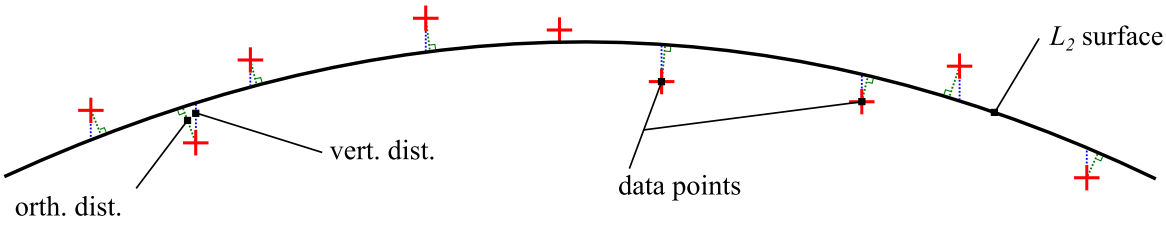


Fig. 7 Least-Squares fitting (orth. dist.: orthogonal distance, vert. dist.: vertical distance).

In a work developed by METAS, the Swiss metrology institute, aspheric models are fit to the dataset based on a Least-Squares distance minimization. The distances are calculated in the z-direction assuming that both the model and the dataset are aligned along the z-axis which is coincident with the axis of symmetry [26]. Concerning project IND10: FORM, no constraint is set with regard to the methodology used for the analysis of form deviation. Nonetheless, the fitting problem is complex because of the large size of the datasets available from optical measurements and the fact that they are generally unorganized. For this matter, we propose to study the mostly used association techniques and try to select the methods that are the most suitable.

The global approach to complex surface reconstruction follows the scheme in Fig. 8. The preprocessing phase is important since it reduces noise and removes outliers from the raw dataset. Eventually, preprocessing might include point-set orientation that may be done by Principal Component Analysis methods, using Voronoi Diagram approaches [27,28], or other approaches [29–31]. This part of the process is not our main concern here. On the contrary, reconstruction and association are our main focus in this manuscript. We firstly study the need for a mesh reconstruction and then, with the analysis and comparison of many surface reconstruction methods, we come up with some assessment tools and criteria to validate a method and use it to characterize freeform surfaces. Meshing is a step that we include in our study but the reader must be advised that it may not be necessary for the rest of the process. We then perform the association of implicit models when they exist and B-Spline models when they do not, to the points. Those points are either the outcome of the pre-processing phase or the ordered points resulting from the meshing phase in case data structure was indispensable. The last phase consists of both the analysis of the residual errors that remain after the association phase and the evaluation of the form and/or some required geometric dimensions of the measured surface.

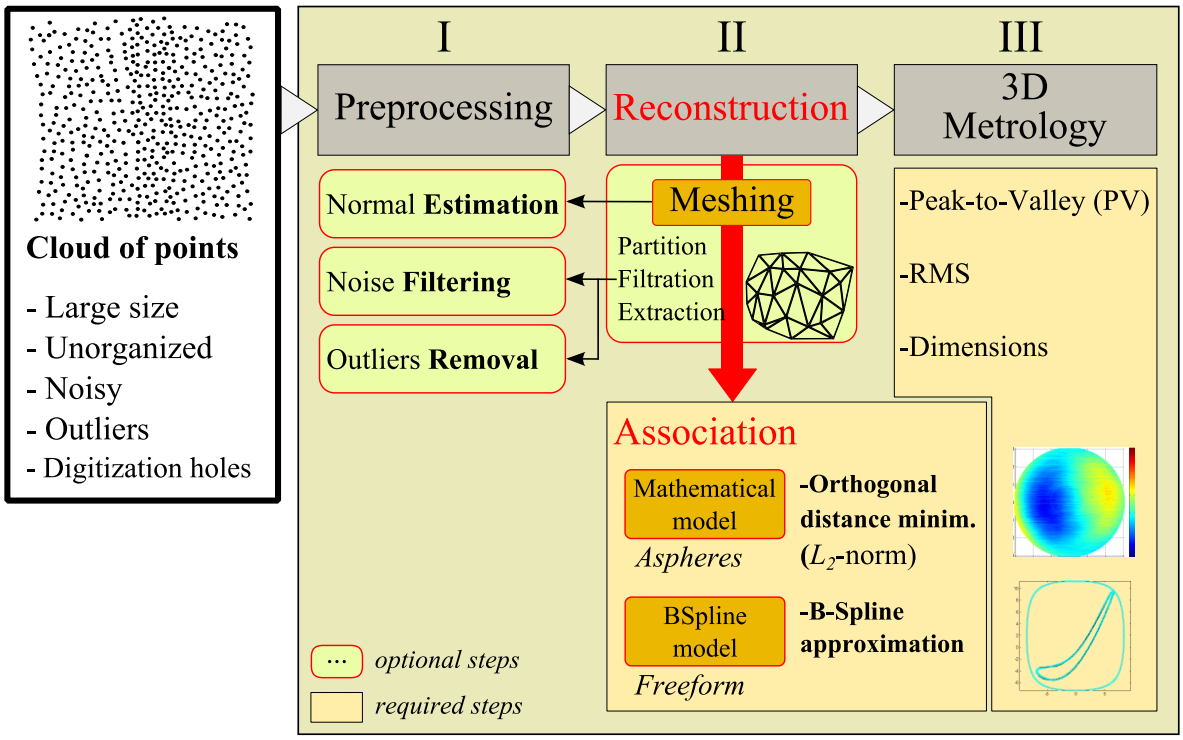


Fig. 8 Data processing scheme.

The manuscript is organized in three chapters:

- In chapter 1 we start by presenting a classification of complex surfaces according to criteria like invariance class, shape complexity and tolerance. The classification leads to the distinction between aspherical surfaces and turbine blades which we will deal with separately for the rest of the manuscript. We then focus on the design and specification of such surfaces, describe some of their advanced manufacturing technologies and lastly center the attention on the measurement methods of aspherical surfaces and turbine blades. We give a review of ultra-high precision machines and probe technologies for the measurement of aspherical surfaces used by the project partners and then expose, in details, the principles of the LNE's high-precision profilometer and the probing technologies used along with details about their calibration. Finally, we describe the principle of the usual 3D coordinate measuring machines for the measurement of turbine blades.
- In chapter 2, we concentrate the process of surface reconstruction onto aspherical surfaces in particular, knowing that any mathematically defined surface can be characterized using the same scheme. We give a brief review of aspherical fitting

by presenting the principles of two classical optimization approaches, the Levenberg-Marquardt (LM) and the Iterative Closest Point (ICP) and then point out a newly used Limited memory-Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm for aspherical surface fitting. All above algorithms are based on Least-Squares minimization. We define a procedure for the validation of the algorithms for aspherical surface fitting by experimenting on simulated data. We compare the algorithms and show the efficiency and performance of L-BFGS as compared to LM and ICP. Finally, we apply those algorithms to measured data of an aspherical lens and show again the effectiveness of L-BFGS in regard to running time and precision.

- In chapter 3, we focus on the general freeform surfaces reconstruction which are not defined by any mathematical formulation. We even assume that a CAD model is not available and perform the reconstruction using B-Spline parametric models. We start by an exhaustive literature survey of surface reconstruction techniques, concentrate on the state-of-the-art of B-Spline fitting techniques and then propose a contribution to this problem by developing a new algorithm that solves major issues involved in B-Spline fitting. Our algorithm does not need any particular close curve/surface initialization or location parameters calculations. This algorithm applies local knot insertion for the improvement of the fitting precision and considers fairing for overall curve smoothness. The algorithm works well on closed curves such as turbine blade profiles and converges relatively fast to the pre-set tolerance in most cases. Some robustness issues need to be addressed in later works as well as the extension to surfaces.

Chapter 1 - Design, manufacturing and measurement of complex surfaces

I. Introduction

With the advances of technology, the trend in the design of parts leans more and more towards the use of complex shapes. We use the term complex shapes when we want to refer to surfaces exhibiting variations in shape and geometry and which reveal hurdles in their manufacturing and measurement. An artifact for complex shapes is the NPL freeform artifact which is a new freeform reference standard.

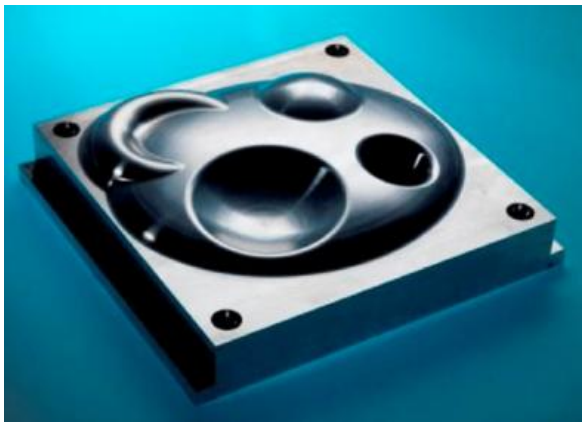


Fig. 9 The NPL freeform artifact.

Manufacturing and measurement are two key stages of a product's life cycle following design. At the very first stage, a designer may have an outstanding concept of a given product, however, if the blueprint cannot be manufactured according to design specifications, or if it cannot be verified by the existing/available measurement systems, this product cannot be guaranteed to fulfill its functionalities [32]. Today, technology has led to overcome the hurdles of all what is related to high precision measurement and accurate manufacturing [33], [34], [35]. Not only high precision processes, such as rectification and polishing techniques, have reduced manufacturing defects, but also, measurement techniques have shifted to a whole new scale, a scale of high precision, high repeatability and low uncertainty [36,37], [38], [39], [40], [41]. Ultra-high precision measuring machines (UHPMM) show distinguished performances since they are equipped with ultra-high precision mechanical guiding systems and are motion-controlled by laser interferometers [2,14,42–45], [46,47], [48]. Interferometers can achieve 0.1 nm accuracy and, by that, are the most accurate among optical measurement systems that exist today [49,50].

So it has become promising to go further in design specifications and ask for more complex product designs. From aspherical surfaces to freeform shapes, we describe in this

chapter how these complex surfaces are classified. The classification is based on both the ISO standard on Geometrical Product Specification [51] as well as the keynote paper of Savio *et al* [8] and helps distinguish between aspherical surfaces and turbine blades. We then portray how each of these two types of surfaces are defined and specified according to standards. We give an overview of their design, make a brief listing of the state-of-the-art manufacturing processes and focus on the state-of-the-art measurement technologies and machines dedicated for complex surfaces metrology. Mainly, we review the UHPMM used for aspherical surfaces measurement and then thoroughly describe the LNE High-precision profilometer, also for the measurement of aspherical surfaces. We finally recall the measuring principle of CMMs for the measurement of freeform shapes in general and turbine blades in particular.

II. Classification of complex surfaces

Complex shapes are defined to be surfaces which exhibit shape variations and compound geometrical features. The Geometrical Product Specifications (GPS) standards define shapes according to their invariance class [51] and complex shapes are the ones having no unconstrained degrees of freedom (Table 1). The invariance class is a "*group of ideal features defined by the same displacement(s) of the ideal feature for which the feature is kept identical in the space*" [51]. Another definition of complex shapes is given by Scott *et al* [52]. They identify complex shapes as surfaces exhibiting variations in shape consisting of partial geometries that can be decomposed into features of interest. The NPL artifact is a concrete example and is composed of both convex and concave forms as well as spherical, cylindrical and flat features (Fig. 9) [53].

Based on these definitions of complex shapes, we come up with our own definition as a mix of the previous definitions. We refer to complex surfaces, by the surfaces that are composed of different geometries and which generally, but not necessarily, have no invariance degree. From this point on, aspherical surfaces which are of the revolute invariance class will be considered as complex if their mathematical formulation is based on composite geometrical terms.

Invariance class	Unconstrained degrees of freedom
Complex	none
Prismatic	1 translation along a straight line
Revolute	1 rotation around a straight line
Helical	1 translation along and 1 rotation combined around a straight line
Cylindrical	1 translation along and 1 rotation around a straight line
Planar	1 rotation around a straight line and 2 translations in a plane perpendicular to the straight line
Spherical	3 rotations around a point

Table 1 Invariance classes of shapes [51].

According to Savio *et al* [8], there are different types of freeform surfaces and these are classified according to criteria related to their geometry and their specification. The aim of the classification is to distinguish between freeform surfaces in order to know which precision and care to take when it comes to their manufacturing and measurement. The main types of freeform surfaces are airplane wings and fuselage, automotive body parts, turbine blades and blisks, optical parts and haptic sensor surfaces.

Airplane wings/fuselage

In the objective of reducing fuel consumption while increasing passenger capacity, the aircraft wing and fuselage designs are of primary significance. The improvements in the designs do not only come from using lighter and new types of materials, but also from the development of new freeform geometries which require specific measurement techniques and specific precision.

Fig. 10 Airbus A380: today's largest airplane.

Automotive body parts

Reducing fuel consumption and improving performances are not the only criteria to specify the geometry of car body parts as, nowadays, the emotional and ergonomic aspects are increasingly affecting this industry. Car body parts are manufactured by a stamping process that requires precision forming tools which should be precisely measured and before being processed.

Fig. 11 Automotive B-pillar.

Haptic interfaces

Haptic surfaces are interfaces that transmit, in forward and backward modes, mechanical forces or excitations to or from a user through the sense of touch. The need for such surfaces today covers many fields in engineering, such as robotics, bio-medicine, communications and many other disciplines [54,55] (Fig. 12). First generation haptic interfaces are kinds of vibratory motors integrated into cell phones that would create an alert to phone calls. Second and third generation haptic interfaces include audio and electrostatic haptic technologies. Fourth generation haptic interfaces can deliver pressure sensitivity but are still under research. In biomedical devices like prosthetic limbs, research is conducted to make the haptic interfaces deliver the essential feedback to the wearer (Fig. 12b) and improve its manufacturing quality [56].

(a) (b)

Fig. 12 Haptic interfaces: (a) in robotics applications; (b) in biomedical applications.

Turbine blades and blisks

Turbine blades and blisks (bladed disks) are found in a wide variety of industrial products such as cooling fans, turbochargers, jet engines, etc… (Fig. 13). The fabrication of jet engines, for example, is the most challenging regarding material selection and the blade shape design.

(a) (b)

Fig. 13 (a) Turbine blades in a jet engine; (b) Axial flow blisk.

In turbomachinery, freeform geometries are essential for optimizing performance and minimizing energy losses (Fig. 14).

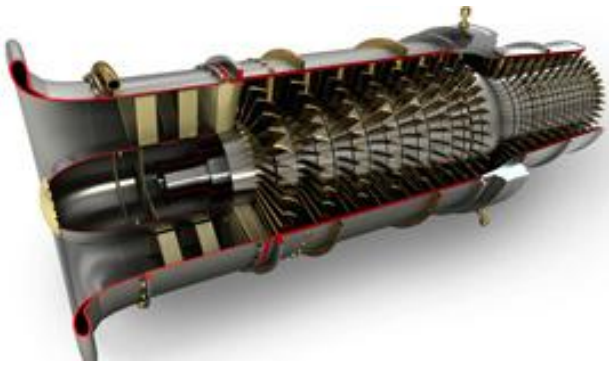


Fig. 14 Set of blades in a turbo-machine.

Optical parts

Aspheric and freeform optics have seen enlarged applications because of their unmatched optical performances. They can be found in bar code scanners, laser diode collimation systems, cameras, satellite surveillance systems and even medical products [57], [58]. Their designs vary depending on the application: spiral mirrors used in laser scanning, discontinuous or step-like lenses (Fresnel lenses) used in lighting, structured/functional surfaces used in retro-reflective applications, etc… [59]. They have widely replaced spherical lenses because they have the property to reduce wavefront error, eliminate spherical aberrations and focus all incident light into one point (Fig. 15) [60]. Freeform surfaces can offer even better optical performances than aspherical surfaces as they additionally allow for increasing the depth of field and expanding the field of view [59].

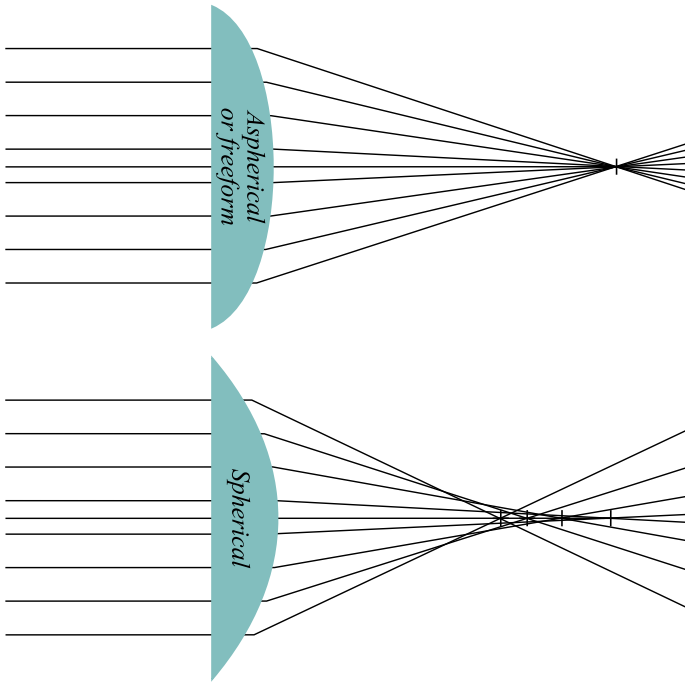


Fig. 15 Aspherical and freeform optics reduce spherical aberrations [57].

For an equivalent optical performance, one aspherical or freeform component can replace several spherical components in a given optical system (Fig. 16), reducing both size and cost of the system.

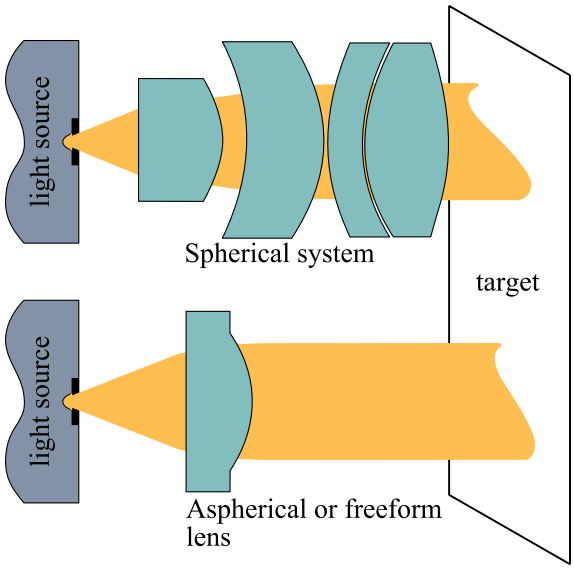


Fig. 16 Aspherical lens based systems replacing spherical systems.

Video projectors for example have seen quite a remarkable transformation thanks to the advances in asphere design, manufacturing and measurement technologies. A hand-sized

projector is available in the market today and can achieve the same resolutions as a standard-sized projector (Fig. 17).

(a) (b)

Fig. 17 Micro video projector: (a) for Smartphone; (b) for standard use.

Each of the complex surfaces that we have evoked above are actually classified based on both their geometry (shape complexity classification) and the precision sought regarding their manufacturing and inspection (relative tolerance classification). Savio *et al* [8] propose a shape complexity classification which is based on three categories: (a) the low class for surfaces that are nearly flat, aspheric or limited in curvature change. (b) the medium class for multi-facetted surfaces or surfaces exhibiting moderate or large curvature changes. (c) the high class for surfaces exhibiting undercuts, internal features or limited access/visibility features. Freeform surfaces are also classified with respect to their specification as it varies considerably with the precision sought (Fig. 18). This classification is based on relative profile tolerance defined by the ratio between *tolerance* and *main part dimension*. Savio *et al* [8] summarize the classifications for the freeform surface types listed previously (Table 2).

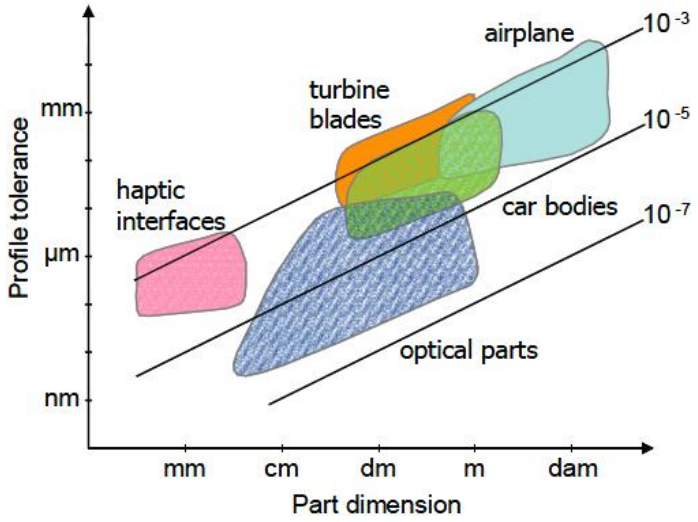


Fig. 18 Tolerance versus part dimension of freeform surfaces [8].

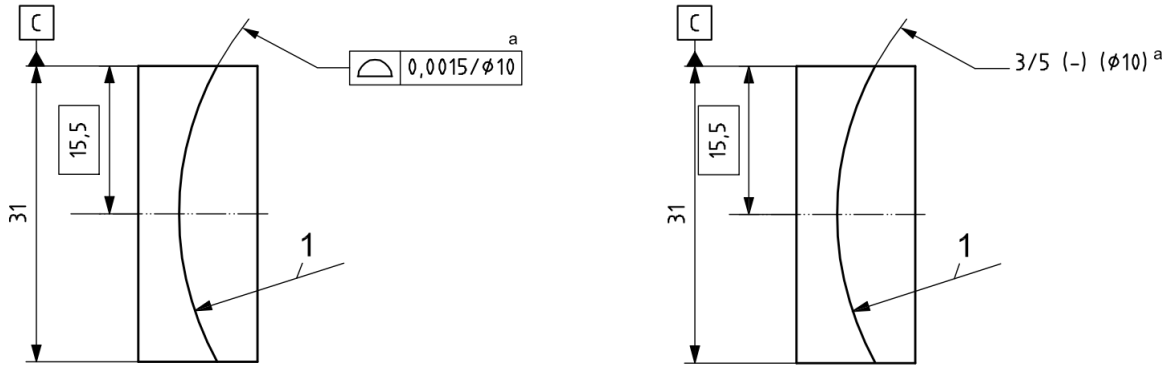
	Airplane parts	**Auto body parts**	**Haptic interfaces**	**Turbine blades/blisks**	**Optical parts**
***Part dimensions* (m)**					
large ($10^0 - 10^2$)	××			×	×
medium ($10^{-1} - 10^0$)	×	××		××	××
small ($10^{-3} - 10^{-2}$)			××		××
micro (10^{-3})			×		×
Shape complexity					
low	××				××
medium	×	××	××	×	×
high			×	××	
Relative tolerance					
medium ($10^{-4} - 10^{-3}$)	××	××		××	
fine ($10^{-5} - 10^{-4}$)			××	×	××
ultra-fine ($< 10^{-5}$)					××

Table 2 Classification of shapes with respect to shape complexity and tolerance according to Savio *et al* [8]. (legend: ××: typical case; ×: less frequent case).

Regarding our applications, aspherical surfaces, a particular case of freeform optics, are classified as having a low shape complexity but fine/ultra-fine relative tolerance and therefore necessitate ultra-high precision manufacturing and measurement (Fig. 18 and Table 2). Turbine blades, on the contrary, have a high shape complexity and require medium precision for manufacturing and inspection. For the remainder of the chapter, the focus will be on those two types of surfaces.

III. Definition and design specification of aspherical surfaces and turbine blades

Geometrical and functional requirements that are set by the designer of a part are mapped into geometrical specifications either according to tolerancing practices of the industrial companies or in reference to existing standards on geometrical dimensioning and tolerancing [61]. Geometrical tolerances are specified in conformance with functional requirements of the part and can be influenced also by the manufacturing and inspection operations. For aspherical surfaces, form tolerances follow the procedures indicated in ISO 1101 [61] and ISO 10110-Part 12 [15]. Form tolerance appears with the symbols ⌒ and ⌓ on a specification drawing according to [61] (Fig. 19a), but can also be indicated as in Fig. 19b with the "slash" symbol. For turbine blades, the specifications are either according to cross-sectional profiles or surface tolerances but in this case, the surface is decomposed into separate features [51], [62]. The geometry of the leading and trailing edges cannot be controlled by specifying a tolerance zone alone and thus further mathematical specification is needed [52].



(a) (b)

Fig. 19 Geometrical tolerance specification of an asphere using two different practices: (a) ISO 1101; and ISO 10110-Part 5 [63] (taken from [15]).

1. Definition and specification of aspherical surfaces

1.a Classical definition of aspherical surfaces

Aspherical surfaces have seen enlarged applications because of their unmatched performances and because it is now possible to achieve manufacturability and metrology up to

a nanometric level of precision [6,64,65]. Their size is determined by two dimensions, the clear aperture (CA) which is the effective diameter of the lens and the sag (S) which is the maximum height of the lens (Fig. 20).

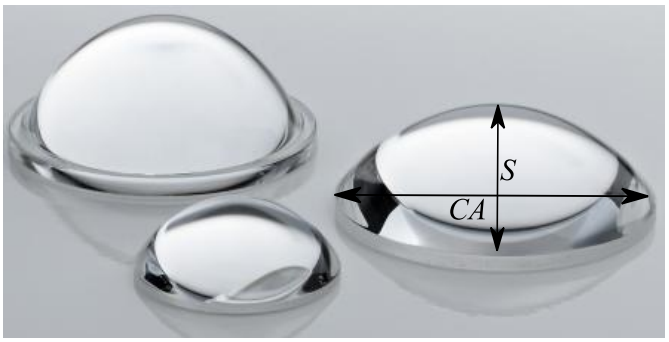


Fig. 20 Aspherical lenses: illustrating the clear aperture (CA) and sag (S).

The mathematical formulation of axis-symmetric aspherical surfaces is detailed and standardized in ISO 10110-Part 12 [15]. According to this standard, aspherical surfaces are defined as being imbedded in an orthogonal coordinates frame in which the z-axis is the optical axis and the origin is the apex of the asphere (Fig. 21). For axis-symmetric aspheres, the optical axis coincides with the axis of symmetry. Unless specified otherwise, when drawing aspheres, the z-axis is always found in the drawing plane and oriented from left to right [15]. This specification is important in order to give a conventional meaning to the sign of the aspherical model parameters. For instance, the radius of curvature at a given point on the surface of the asphere is affected the positive sign if the curvature center is located to the right side of the apex of the surface. It is affected the negative sign otherwise.

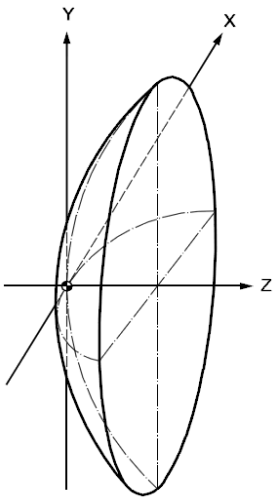


Fig. 21 Coordinate system of aspheres according to ISO 10110-Part 12 [15].

Aspherical surfaces are of many types. The two main types that are most abundantly used are the generalized surfaces of order 2 and the surfaces of higher order. Among the generalized surfaces of order 2, we can distinguish the conic surfaces, the quadratic surfaces and the parabolic surfaces. Among the high order surfaces we differentiate between polynomial and toric surfaces. An asphere can also be a combination of any of these types of surfaces. A summary of all possible types of aspheric surfaces is available in appendix A of ISO 10110-Part 12 [15]. The general axis-symmetric aspherical form of revolute invariance class, $F(r, z; c, \kappa, \boldsymbol{a})$ is defined in implicit form such as in (1):

$$F(r, z; c, \kappa, \boldsymbol{a}) = z - f(r) = 0, \tag{1}$$

where $r = \sqrt{x^2 + y^2}$ 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, $\boldsymbol{a} = (a_4, a_6, \dots, a_m)$ is the vector of the m^{th}-order aspherical deviation parameters. For a given r, the height in z is independent of the angle $\theta = \tan^{-1}\left({}^{y}/_{x}\right)$, and $f(r)$ can be written as in (2):

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

For any $(r, z) \in \mathbb{R}^2$, $F(r, z; c, \kappa, \boldsymbol{a})$ is twice differentiable.

1.b Forbes definition of aspherical surfaces

The new paradigm in representing asphere surfaces is leaning towards a representation derived by Forbes [66,67]. Forbes models are an equivalent representation of aspherical surfaces in which, however, the polynomials are orthogonal meaning that the asphere parameters are independent [68]. In this case, design becomes easier as each parameter controls one specific aspect of the shape of the asphere [69]. There are two proposed models to define aspheres, the *mild* aspheres and the *strong* aspheres. The *mild* aspheres are the lenses with soft departure from a spherical shape, whereas the *strong* aspheres are aspheres that are strongly non spherical and are closer to conical shapes (3). Tests based on the *strong* asphere definition will be performed in chapter 2 and compared to the tests based on the classical

asphere definition as no work has been performed on the characterization of asphere models defined by any of the Forbes models.

$$f(r) = -\left(\frac{cr^2}{1+\sqrt{1-(1+\kappa)c^2r^2}} + v^4 \sum_{j=0}^{P} a_j Q_j^{con}(v^2) \right), \tag{3}$$

where c and κ and the a_j's have different values than in the classical model. $a_j Q_j^{con}$ are the terms of a set of orthogonal polynomials that represent the departure from the conical shape and $v = r/R_{\max}$ with $R_{\max}$ being the aperture of the asphere. The polynomials are a set of Jacobi polynomials derived from:

$$Q_j^{con}(x) \coloneqq P_j^{(\alpha,\beta)}(2x-1). \tag{4}$$

where,

$$P_j^{(\alpha,\beta)}(x) = \sum_i \binom{j+\alpha}{i}\binom{j+\beta}{j-i}\left(\frac{x-1}{2}\right)^{j-i}\left(\frac{x+1}{2}\right)^{i}, \quad j \geq i \geq 0,$$

and for integer j,

$$\binom{z}{j} = \begin{cases} \dfrac{\Gamma(z+1)}{\Gamma(j+1)\Gamma(z-j+1)} & j \geq 0 \\ 0 & j < 0 \end{cases},$$

with $\Gamma(z)$ being the Gamma function.

There exists a conversion software called the QED surface conversion tool [70] developed by Forbes in order to convert classical models into Forbes models and conversely.

2. Definition and specification of turbine blades

Unlike simple shapes and rotationally symmetric aspheres, freeform surfaces are non-rotationally symmetric and can therefore be of any shape. They are characterized by having no invariance degree [8]. In some attempts to define such surfaces, Campbell and Flynn propose a rather informal definition to freeform surfaces by stating that they are surfaces which are composed of one or more non-planar and non-quadric surface patches [71]. Besl [72] equivalently states that "a free-form surface has a well defined surface normal that is continuous almost everywhere except at vertices, edges and cusps". Freeform shapes are

interesting because their surfaces are designed in a way to improve the functional specifications of an object and meet its aesthetic requirements [8], [73], [74].

For turbine blades in particular, some general tolerancing specifications are done on cross-sectional profiles according to ISO 1101 and following common practices within a given company (Fig. 22). The specification of each profile is independent of the others as the thickness of one cross-section can be larger or smaller and the length of trailing and leading edges longer or shorter from one profile to the other. Depending on the complexity of the profile and the environment in which turbine blades are used, more constraints and specifications than the general tolerancing can be required. This might be a constraint related to the flow of the air stream for example [75]. Another common practice in turbine blade profile specification is profile splitting. According to Petitcuenot *et al* [75], Scnema® specifies aerodynamic constraints by splitting the profile into features like leading edge, trailing edge, pressure curve and suction curve. Then each portion is appended a tolerance specification alone.

According to Makem *et al* [62], the geometric design parameters of a turbine blade are the blade dimensions, the profile tolerance and the blade displacement and orientation. The commonly inspected blade dimensions are the chord length, the length from the leading/trailing edge to the stacking axis and the blade's thickness (Fig. 22). Profile tolerancing is a measure of form error and is defined, as previously mentioned, on a specific region of the blade's profile (pressure or suction surfaces, etc…) (Fig. 22). Blade displacement and orientation are evaluated based on the stacking points and the stacking axis [62].

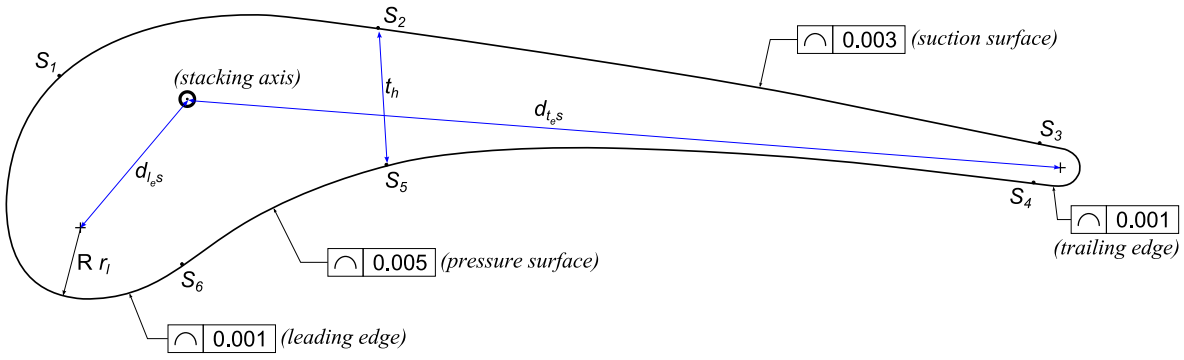


Fig. 22 Geometrical tolerancing of a turbine blade profile according to ISO 1101 (d_{l_es}: leading edge to stacking axis distance; d_{t_es}: trailing edge to stacking axis distance; S_i: stacking points; r_l: leading edge radius of curvature; t_h: central thickness).

Freeform surfaces are rarely described by a complete explicit or implicit mathematical form and are rather defined in parametric form. Parametric forms facilitate design because objects can be locally controlled and modified, and are easily sampled [71]. The general parametric form is given in (5):

$$S(u,v) = \begin{bmatrix} f(u,v) \\ g(u,v) \\ h(u,v) \end{bmatrix} \tag{5}$$

where $f(u,v)$, $g(u,v)$ and $h(u,v)$ are functions of two parametric variables u and v. Without loss of generality, u and v can be restricted to the square domain $[0,1] \times [0,1]$. The mostly used parametric form in CAD software is the Non-Uniform Rational B-Spline (NURBS). However, we are interested in a simpler form called the Uniform B-Spline such as in (6):

$$S(u,v) = \sum_i \sum_j N_i^d(u) N_j^l(v) C_{i,j} \, , \tag{6}$$

where $C_{i,j}$ are the bidirectional control points of the surface, $N_i^d(u)$ and $N_j^l(v)$ are the $d^{th} -$ and $l^{th} -$ degree B-Spline basis functions in u and v directions, respectively.
Aspherical surfaces can also be defined in parametric form but this is not of our interest for project IND10: FORM.

IV. Manufacturing of aspherical surfaces and turbine blades

1. Manufacturing of aspherical surfaces

In the field of aspherical and freeform optics manufacturing, different processing and manufacturing techniques are exhaustively summarized in Fang *et al* in [59]. For instance, one of the major manufacturing processes for rotationally symmetric surfaces is Single Point Diamond Turning (SPDT) shown in Fig. 23 [76]. This process is applied for the fabrication of surfaces that are made of non-ferrous metals or ceramics. It can also be used with freeform surfaces provided that an adaptation is applied, like precision grinding or Fast Tool Servo (FTS) [2,77]. Other manufacturing methods for freeform optics and freeform optical molds

are diamond fly-cutting and ball-end milling, respectively [78,79]. Some recent computer aided polishing methods, also referred to as deterministic polishing methods, enable achieving high precision and high repeatability in the manufacturing of complex surfaces (Fig. 24). Among these methods we cite the ion and plasma beam machining [36], [80] (Fig. 25), the Magnetorheological Finishing (MRF®) [38] (Fig. 27), the Magnetorheological Control Servo polishing [37], and the Precession Process [81] (Fig. 24b). Computer assisted tool path generation techniques have also been improved for accurate manufacturing of freeform optical surfaces. Brecher *et al* [82] developed a NURBS-based tool path generation scheme that allows an off-axis FTS manufacturing achieving a surface roughness below 10 nm. According to Walker *et al* [81], the Precession Process allows to control the form error of aspherical lenses.

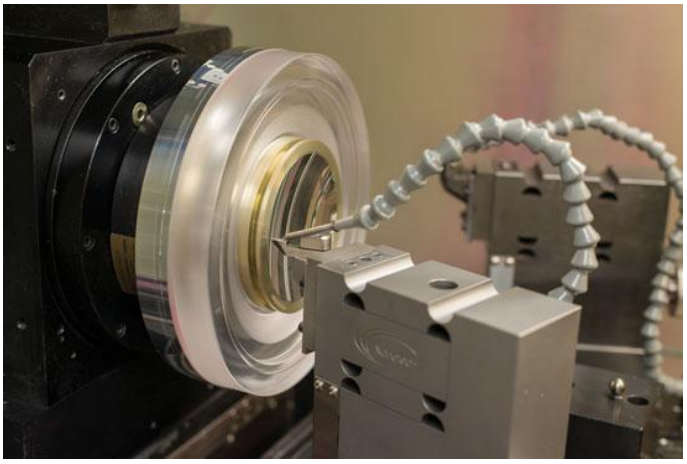


Fig. 23 Single Point Diamond Turning from Empire Precision.

(a) (b)

Fig. 24 (a) High precision polishing machine by Zeeko LTD; (b) Precession process polishing [81], [83].

Fig. 25 Ion and plasma beam machining from TNO.

Henselmans [2] proposes an auto-correction manufacturing process consisting of a closed loop control system in which the NANOMEFOS ultra high-precision measuring machine recursively sends feedback after a manufacturing operation is done until the desired form and roughness errors are below a pre-defined tolerance (Fig. 26).

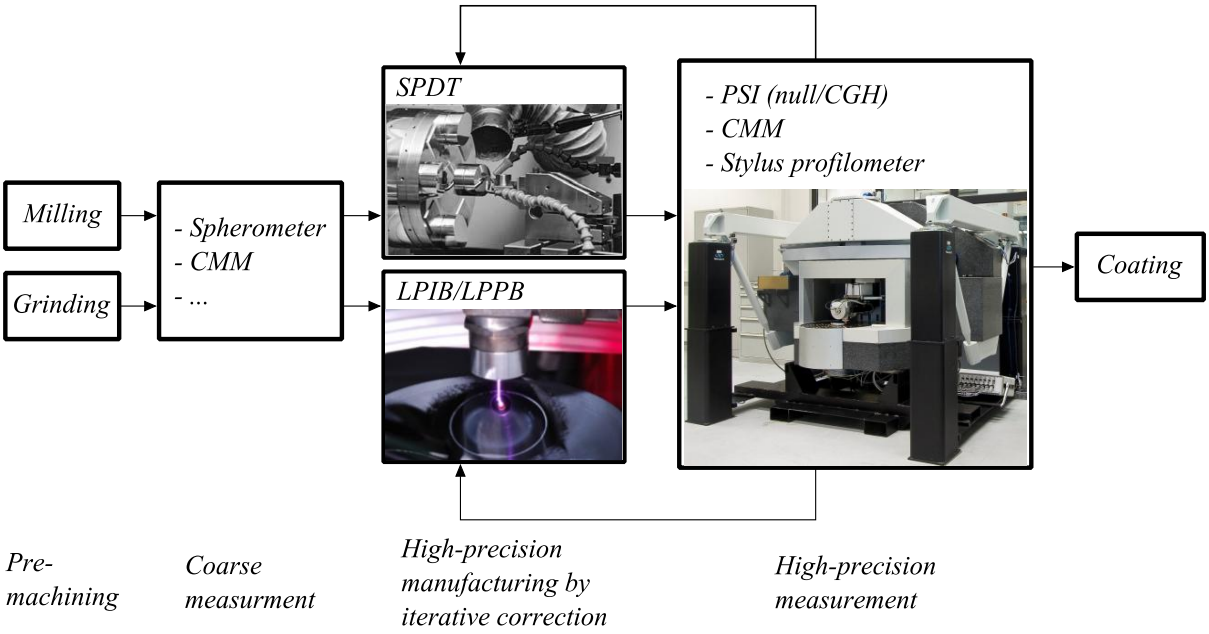


Fig. 26 General manufacturing process chain for aspherical and freeform optics (*PSI*: Phase-shifting interferometer, *LPIB*/*LPPB*: Local polishing ion beam/plasma beam) [2].

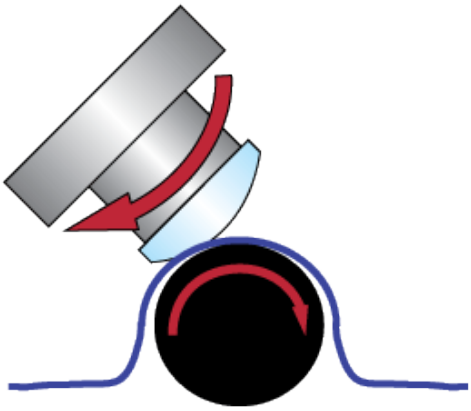


Fig. 27 Magnetorheological Finishing polishing process.

Another manufacturing process for aspherical parts which is also abundantly used is the molding process (Fig. 28). But here yet again, the molds need to be manufactured by some turning techniques. For instance, Konika Minolta®'s lenses are produced using pre-manufactured molds using a SPDT technique (Fig. 23).

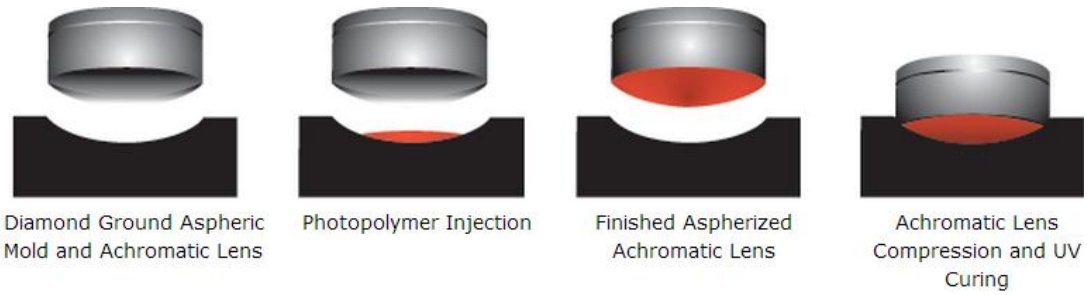


Fig. 28 Aspheres molding [84].

2. Manufacturing of turbine blades

Freeform shapes are primarily manufactured on 5-axis Computer Numerical Control (CNC) milling machines [79], [82]. It is the case for turbine blades manufacturing in most of their applications (Fig. 29), nevertheless these parts can also be manufactured using closed die hot-forging techniques [85] (Fig. 30). In the latter case, numerical simulation methods using a Finite Elements Method model have been developed in order to optimize the forging process. Moreover, virtual inspection frameworks have been introduced to improve the manufacturing precision [62].

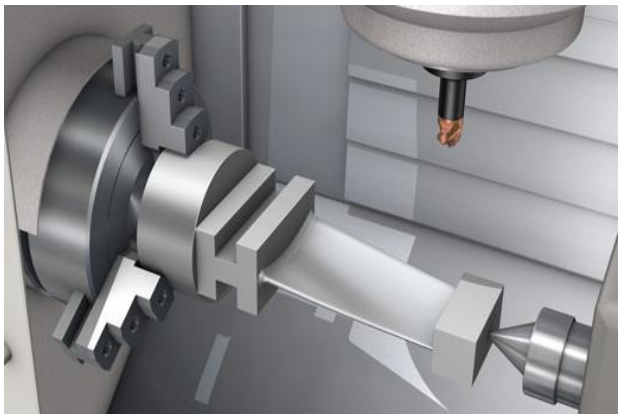


Fig. 29 Turbine blade milling, retrieved from Sandvik Coromant's website [86].

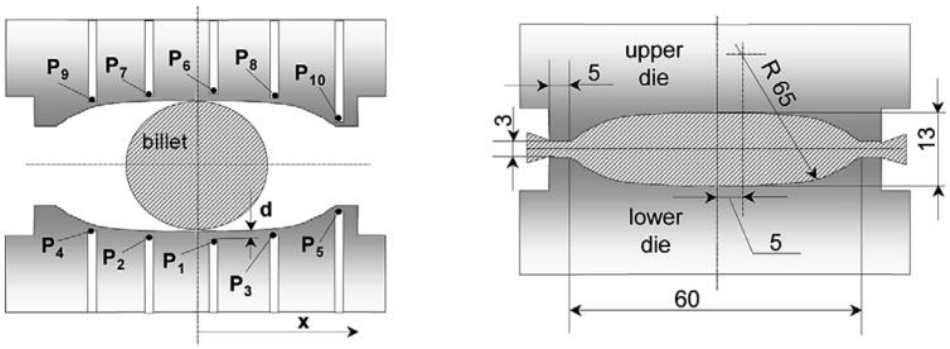


Fig. 30 Closed die hot-forging of stainless steel turbine blades [85].

Turbine blades are fabricated using three dominant categories of materials, Prepreg (pre-impregnated composite fibers), Infusion materials such as resin and high strength stainless steel. Other high-performance materials like ultra-high temperature titanium and nickel alloys or titanium aluminides (TiAl) are also used but they are difficult to shape and require special processes, precision casting and/or isothermal forging followed by precise finishing (Electro-chemical machining [87]). The material used influences the weight of the blade and the efficiency of the system in which it is integrated. For wind turbine blades in particular, materials such as fiberglass and carbon fibers are used, and, according to Veer *et al* [88], techniques like open-mold wet process, vacuum-assisted resin transfer molding (VARTM) and single-shot infusion are predominant in their manufacturing.

For gas turbine blades, alloy improvement and directional crystal solidification as well as the use of coating systems have allowed for improved thermodynamic efficiency, for increased system strength and for higher gas temperatures of more than 1400 °C [8].

V. Measurement of aspherical surfaces and turbine blades

While current techniques allow for manufacturing arbitrary optical surfaces [2], high precision measurement of optical surfaces (aspherical and freeform optics) as well as large data processing are still a challenge in industry [33], [35]. A detailed benchmark of the measurement techniques that are used in today's freeform surfaces metrology is reported by Savio *et al* [8] and improved here in the graph of Fig. 31. For aspherical surfaces, nanometric level of uncertainty is sought and we will give a thorough description of the UHPMM that were developed for this purpose as well as the corresponding measuring instruments [89]. For turbine blades, a rather sub-micrometric uncertainty is sought and the description of the CMM as well as the related measuring technologies will be given.

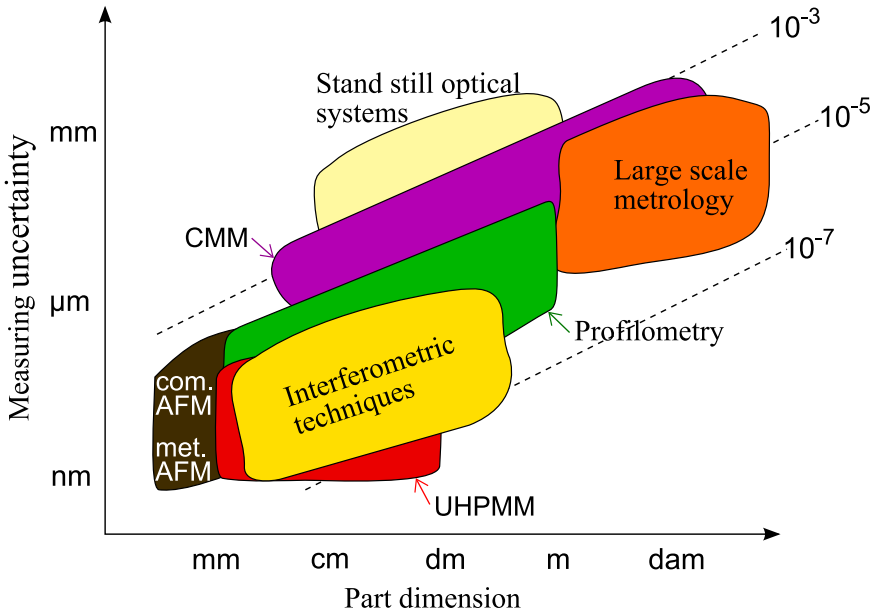


Fig. 31 Associated measurement uncertainty of different measurement systems (com.: commercial, met.: metrological; AFM: Atomic Force Microscopy).

1. Review of existing ultra-high precision measuring machines for the measurement of aspherical surfaces

The aim of project IND10: FORM is to improve the measurement of high quality optical surfaces such as aspherical lenses. The project gathers a number of NMIs, such as, LNE, PTB, VSL, METAS, SMD, CMI and MKEH, industrial partners such as IBSPE and TNO and academic partners. In the field of ultra-precision 3D metrology, various dedicated UHPMMs have been developed and calibrated at the cited institutes and laboratories [89]. These

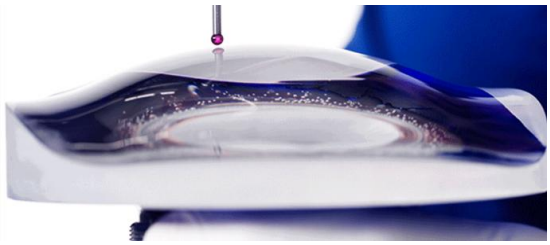
machines typically feature 3D measuring ranges less than $100 \times 100 \times 100$ mm^3, and usually embody the same set of fundamental principles. The main one consists in achieving high positioning and measuring accuracy with perfect respect of the Abbe principle [90]. The measuring instruments can be stylus-based or optical-based [91][92][93]. The traceability of these measuring apparatus is performed using laser interferometers which are in-turn traceable to the SI meter definition through a frequency calibration by comparison with an I2-stabilized primary He-Ne laser source [94], [95].

The mostly known UHPMMs dedicated for the measurement of ultra-high precision optical surfaces, the Zeiss F25 (VSL, SMD), the METAS µCMM apparatus, the ISARA 400 apparatus (IBSPE), The NANOMEFOS apparatus (TNO), The Tilted-Wave Interferometer (PTB) apparatus and the LNE high-precision profilometer are detailed here.

1.a The Zeiss F25 apparatus

The Zeiss F25 UHPMM has been developed by the Zeiss company in collaboration with the Dutch Metrology Institute (VSL) and the Eindhoven University of Technology (Fig. 32a) [45]. The ultra-precise performance of this machine not only relies on the above described principles, but also on the high-tech components and systems of the machine. The latter uses highly accurate air-bearing kinematics structure with linear drives and an active air damping base. The three axes length measurements are controlled by ultra-precise glass-ceramic scales which have a resolution below 8 nm for the older model of F25 and below 1 nm for the latest F25 generation. The measuring volume of F25 is about $130 \times 130 \times 100$ mm^3. In order to reach very high accuracy, the Abbe principle is applied and realized using additional intermediate bodies in x- and y- directions as compared to conventional CMM designs [45]. The intermediate bodies A and B are supported on orthogonal beams I and II (Fig. 33) which are connected to the probe by a moving platform PL and fixed to a base. This architecture makes the first order Abbe error in the x- and y- directions equal to 0. The z-Abbe error is not similarly controlled and is thus not null. However, it is considerably reduced by mounting the z- linear drive as close as possible to the center of mass of the platform center of mass.

(a)



(b)

Fig. 32 The F25 µCMM. (a) complete apparatus; (b) freeform optical surface measurement using a tactile probe [45].

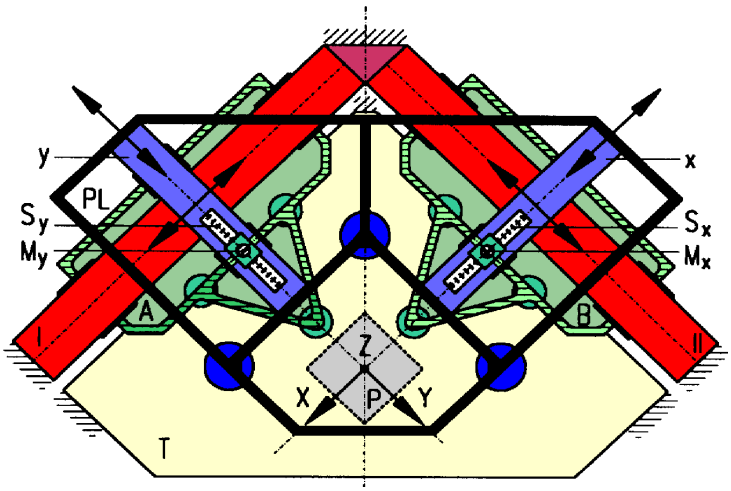


Fig. 33 Schematic of the F25 XY platform (Top View). PL: moving platform (thick black); x and y: x- and y- linear drives; S_x and S_y: Ultra-precise glass ceramic length measuring systems with nanometric scales [45].

The total sources of error that contribute to the uncertainty of the measurement are reduced to thirteen. They result from the sum of the individual errors related to the moving bodies, the platform motion, the probing entity and the imperfect perpendicularity between the x-, y- and

z- guiding systems. The error model is described in more details in the thesis of Marcus Vermeulen [45].

Finally, this machine accepts two types of probing systems, a tactile micro-probe as well as a tilt-robust confocal probe (Fig. 34). The tactile probe consists of a very thin shaft with a contact ball at its end which can have a diameter as low as 120 µm. The shaft length of several millimeters makes depth measurements possible. The measuring forces with this probe are as low as 0.5 mN. The confocal probing system is developed at the Belgian metrology institute (SMD) and is a tilt-robust probe that incorporates an aperture monitoring system to compensate for errors of tilt. It uses the same principle as the one of the NANOMEFOS machine, however without the interferometer system that allows to extend the measurement range. The probe delivers measurement data with nanometer level of uncertainty.

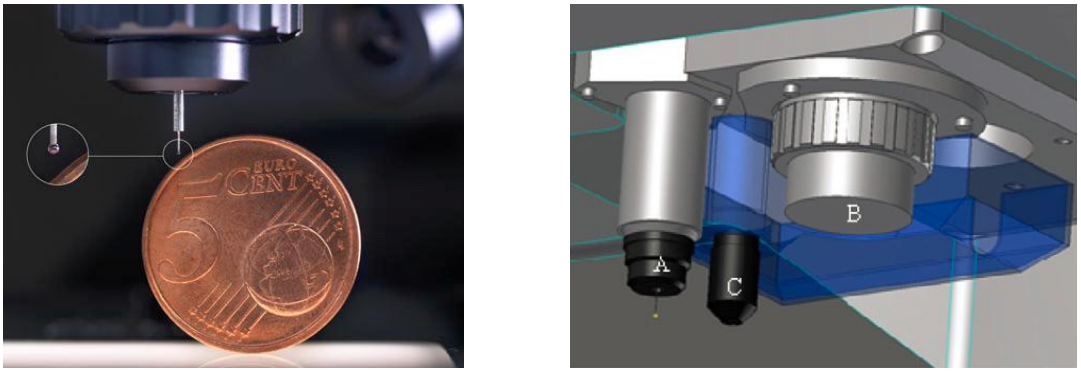
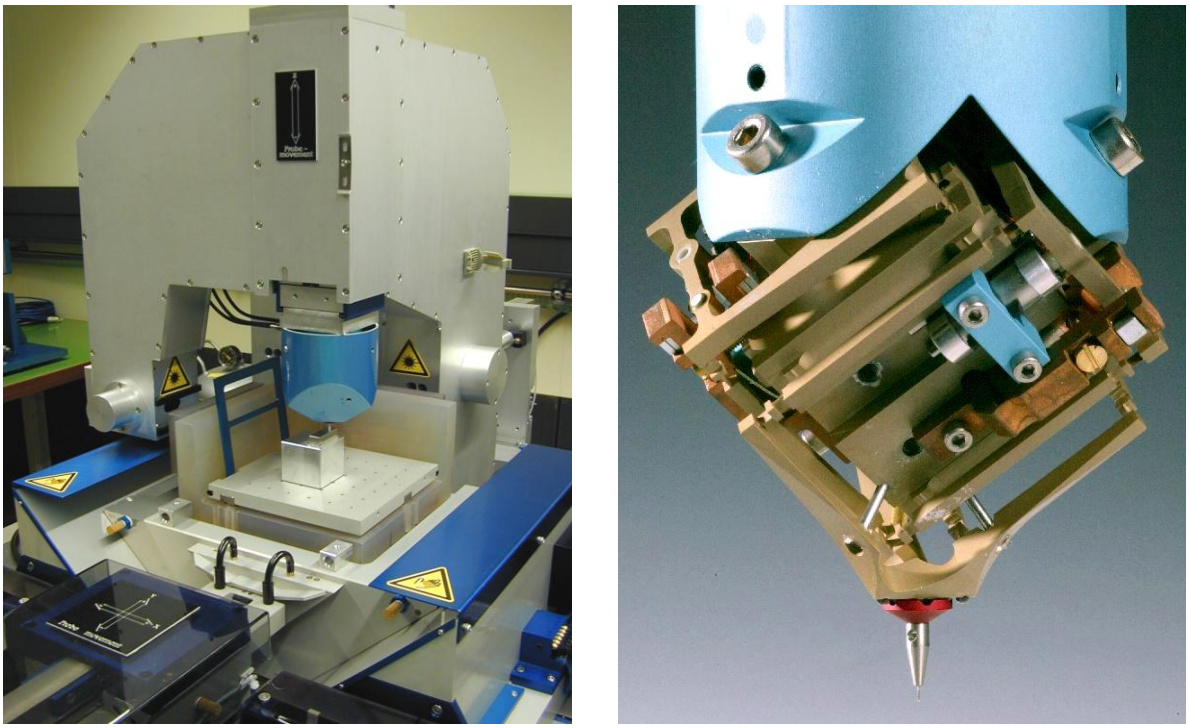


Fig. 34 (a) Tactile probe of the F25 machine; (b) conceptual design and layout of the tactile (A) and confocal (C) probes mounted on F25 [89].

1.b The METAS µCMM apparatus

The METAS µCMM has been developed and especially designed for the measurement of micro-parts with an ultra-high precision [42]. It is composed of a tactile probe and a motion table (Fig. 35a). The motion table is constituted of vacuum preloaded air bearings driven by Lorenz actuators and is controlled by position measurement interferometers. A special slide configuration with two wedges makes the table movement very compact and stiff. A pneumatic weight compensation system minimizes the heat generation for lifting up the heavy stage in the *z*-direction. The working volume of the machine is about $90 \times 90 \times 38$ mm^3.

During measurement, the probe is fixed and only the motion table moves the workpiece around and about the probe in all directions. The workpiece is posed on a Zerodur corner cube on which three flat mirrors are mounted in all three directions to form the reference coordinate system of the measurement. The displacement in each direction is monitored by a laser interferometer that points to the center of the tactile probing sphere. Thus the Abbe principle is satisfied along the three axes for the entire volumetric working range. The metrology frame of the METAS µCMM consists of the interferometers, the motion table and the probe. Since the frame has to be large to go around the table and include the *z*-axis interferometer from below, it becomes very sensitive to temperature variations. For this machine, Aluminum was found to be optimal as in regard to its predictable thermal behavior. To limit thermal impact, the machine is shielded inside a thick Aluminum shell and is placed inside a temperature controlled clean room where temperature of the frame varies by not more than 1 mK during one measurement.



(a) (b)

Fig. 35 (a) METAS µCMM apparatus and (b) tactile microprobe [42].

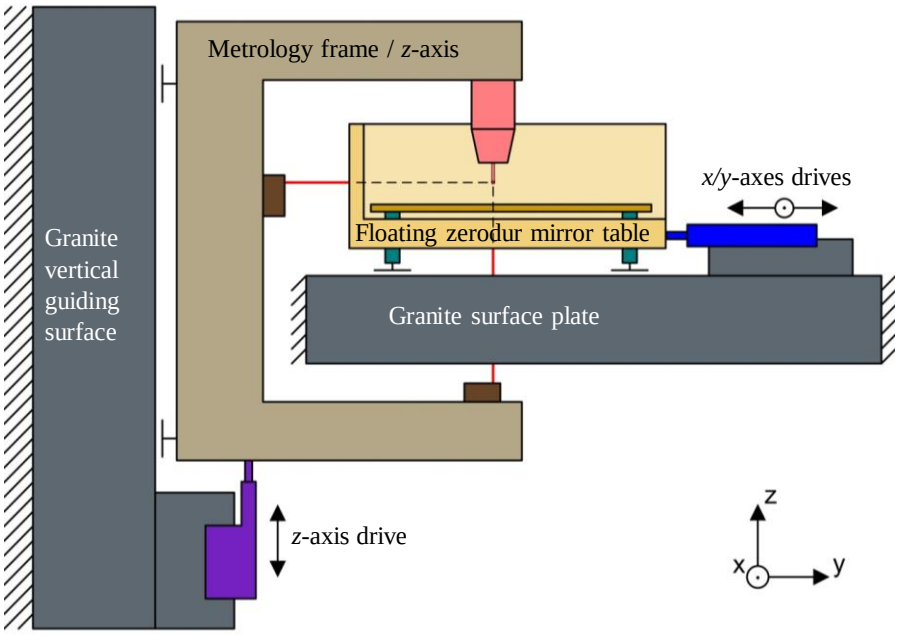
The 3D touch probe (Fig. 35b) head has a particular parallel kinematic structure that exhibits very week and isotropic probing forces, and minimizes the moving mass within the probing

system. Having a very low stiffness, deformations caused by the force of gravity are compensated by an adjustable system of permanent magnets.

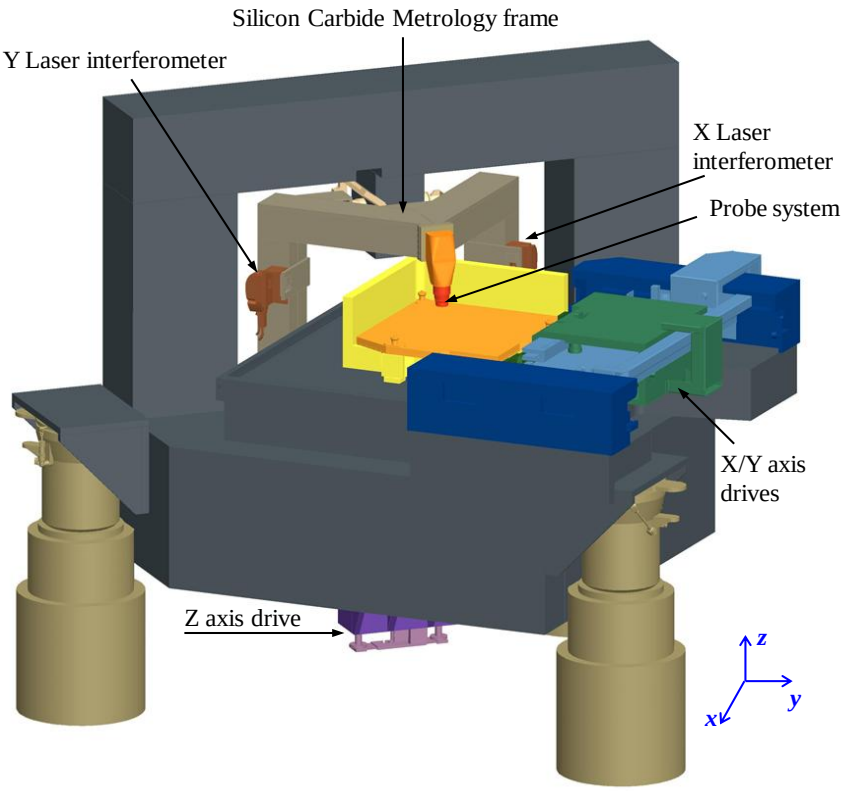
1.c The ISARA 400 apparatus

The ISARA 400 has been developed by the IBSPE company and is a UHPMM featuring the largest measurement volume of about $400 \times 400 \times 100$ mm^3 [43,96] (Fig. 36). Such as for the previous machines, the axes motions are controlled by laser interferometers monitoring each direction independently. The moving stage that holds the workpiece has three flat mirrors that are used as reflectors to the laser interferometers in x-, y- and z- directions. These interferometers are mounted in one metrology frame which also holds the measuring probe system (Fig. 37). The laser interferometer beams all point towards the center of the probe tip making the measurement satisfy the Abbe principle. Since this property remains true within the complete measuring volume, straightness errors as well as rotations of the three translation stages will have zero first order influence on the measurement result. On the ISARA 400, the flatness and squareness errors of the three mirrors are reduced by means of a series of in-machine calibration measurements. The moving stage can translate over a granite base plate in x- and y- directions and is guided by air bearings in a "*floating table*"-like configuration. The entire metrology frame moves in the z-direction with a guide provided by air-bearings against a vertical granite surface (Fig. 37a). The main function of the metrology frame is to maintain the respective position and alignment of the probe with the laser interferometers with high stability. This frame is made of hollow beams of Silicon Carbide (SiC) making it both stiff and light-weight, while also enabling a good thermal stability. The stage is a table made of Zerodur material and has three reflective sides serving as the flat reference mirrors.

Fig. 36 Photograph of the ISARA 400 measuring machine [89].



(a)



(b)

Fig. 37 Design of ISARA 400 measuring machine; (a) 2D design concept; (b) 3D design concept [96].

The ISARA 400 tactile micro-probe system is the *Triskelion* described in detail in [43] and shown in Fig. 38. The design of this micro-probe features an elastically suspended stylus which can deflect at its tip in any of the x-, y- or z- directions. The small size of the tip enables extremely high spatial resolution. The elasticity of the probe further provides the ability to reduce probing force and damage of the workpiece surface. Overall, for a probe tip deflection of less than 5 µm, measurement errors are below 10 nm per axis and less than 15 nm in 3D.

Fig. 38 IBSPE Triskelion B-500 miniature probe [89].

1.d The NANOMEFOS apparatus

The NANOMEFOS machine [2] was originally designed specifically for non-contact measurement of aspherical and freeform optics (Fig. 39). As these surfaces are in general rotationally symmetric, the machine has a cylindrical setup, therefore has less moving axes and higher measurement speed as compared to orthogonal setup machines. The machine is capable of measuring slopes up to 45 ° in both concave and convex configurations. The measurement volume is relatively large and is about Ø500 mm in diameter and 100 mm in height. About 4 million points sampled with a density of 0.2 mm can be measured within 15 minutes.

Fig. 39 NANOMEFOS non-contact measurement machine for freeform optics [2].

On this machine the workpiece is mounted on an air bearing spindle that rotates at constant speed. The optical probe is mounted on a rotating axis (Ψ-axis) which can make sure that the

probe is always oriented in the orthogonal direction with respect to the surface at the point being measured. Moreover, the probe can translate in the radial and vertical directions by the R (radial) and Z (vertical) stages, respectively. Then the measurement trajectory can either be spiral or annular.

When a freeform surface is measured, the measurement distance can vary by few millimeters due to the shape of the surface. For optical probes, the focal depth is of a few micrometers when nanometer resolution is required. To keep the probe in focus, the R and Z stages need to be actuated with large accelerations, resulting in undesirable dynamics of some hundreds of kilograms of mass. In order to avoid this issue, a specific optical probe was designed with a 5 mm range and nanometric resolution in which only the 50 g objective lens translates [97].

The measurement uncertainty is mainly determined by the metrology loop between the probe and the workpiece and is about 10 nm and for the largest surfaces. By applying the dissociated metrology structure principle [98], [99], the metrology loop becomes much shorter and independent from the structural loop (Fig. 40). According to Vissiere *et al* [100] the cited principle considers that the structural loop involves all the elements which are required for maintaining the relative position of the probe with respect to the target. Conversely, the metrology loop is a virtual chain involving all elements such as supports, probes and linkages which are required to determine the position between the probe and the target. Both loops are linked together using isostatic links to avoid any influence from the structure on the measurement, such as deformations due to unpredictable loads.

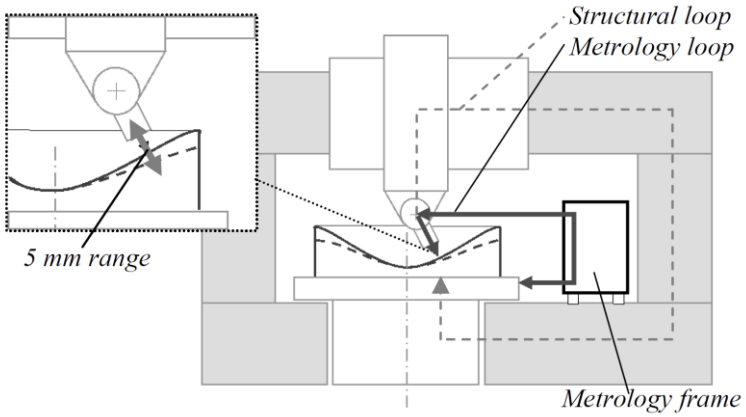


Fig. 40 Machine concept with long range optical probe and separate metrology frame [2].

With the probe almost constantly orthogonal to the surface, the error sensitivity is different compared to the error sensitivity in vertical machine setups. As the surfaces are smoothly

curved, the tangential error sensitivity is negligible (Fig. 41), reducing the metrology problem to a 2D-problem.

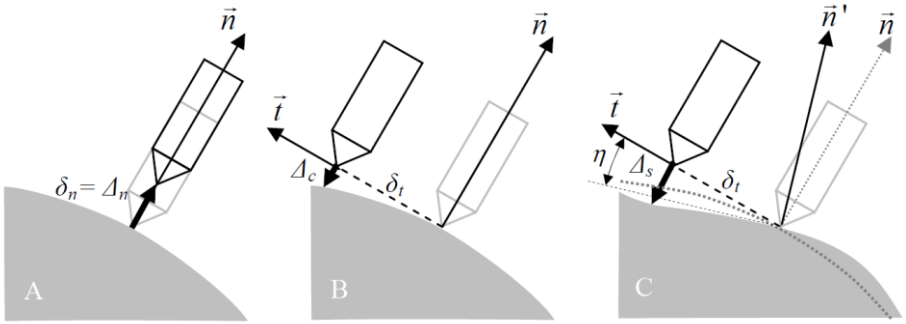


Fig. 41 Error sensitivity with a probe orthogonal to the surface [2].

Hence, a metrology system (Fig. 42) consisting of interferometers (3,6), capacitive probes (9) and a Silicon Carbide metrology frame (2) provide a short and stable loop in the measurement plane. A high-stability parallel air-bearing stage setup with split position and preload frames provides sub-micrometer and micro-radian repeatability in the other directions (Fig. 43).

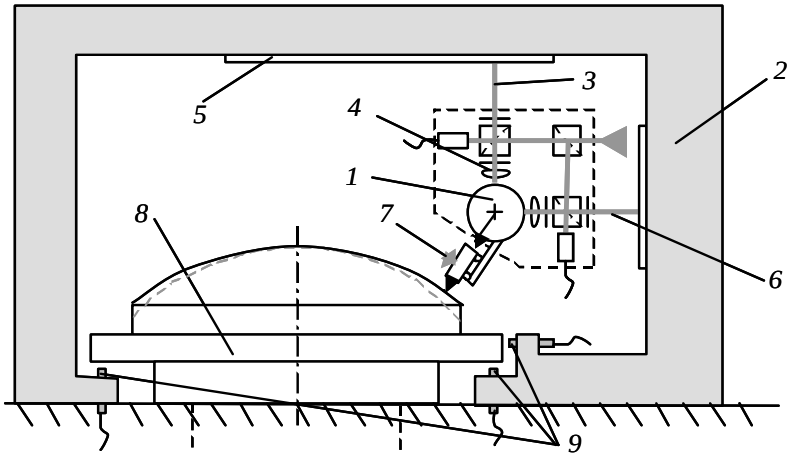


Fig. 42 Short metrology loop [2].

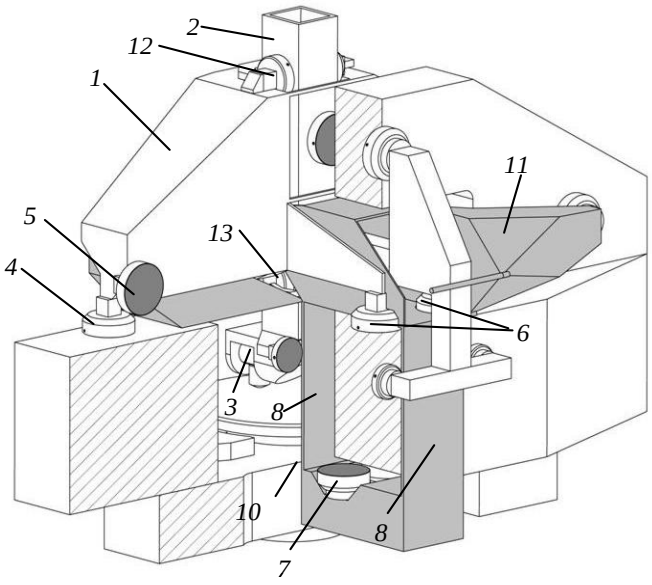


Fig. 43 Parallel air-bearing stage setup with separate preload and position frames [2].

When measuring an aspherical surface, only the probe focusing mechanism is moving dynamically while measuring a circular track. The static and dynamic displacements of probe and workpiece that occur during this measurement are recorded by the metrology system and can be compensated for in the (off-line) data-processing. The measurement uncertainty is of 10 nm when the probe is perpendicular to the surface under test and 35 nm when the surface is at 5 °. The non-contact probe system used was developed by Cacace et al [97]. It consists of the combination of a differential confocal method with a dual pass interferometer that allows to extend the measurement range. The system has a measuring range of 5 mm. A motion controller detects the surface and maintains a good focus of the objective of the probe onto the surface with some tens of nanometers servo error.

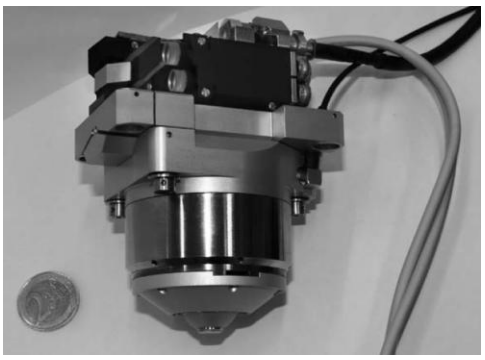


Fig. 44 NANOMEFOS probe assembly [2].

1.e The Tilted-Wave Interferometer

The Tilted-Wave Interferometer (TWI) apparatus was recently developed at the Institute for Applied Optics (ITO) in collaboration with Mahr® company [39,93], [101,102] ().This measuring machine is especially designed for the measurement of aspheres and freeform optics surfaces [103] and is based on an adapted interferometric measurement (Fig. 46).

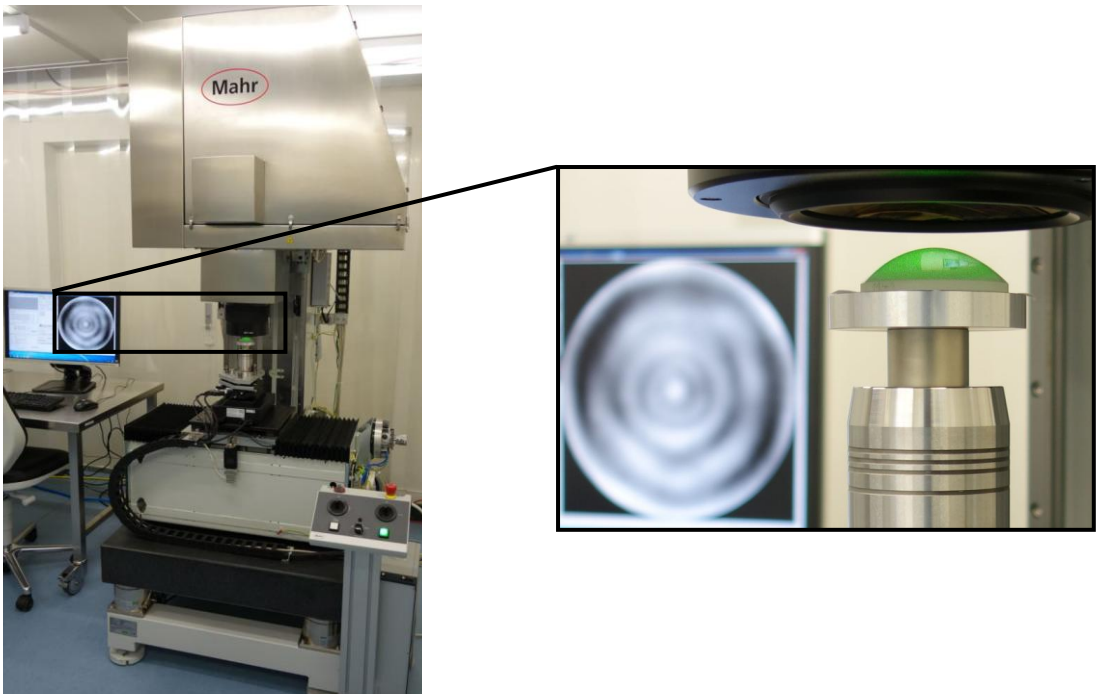


Fig. 45 The TWI apparatus.

The main challenge of measuring an aspherical or freeform surface with interferometry is that the optical rays no longer impinge perpendicularly on the surface. This phenomenon, known as the interferometric null-test condition, requires that the interferometer's axis is perfectly aligned with the target at all times. Usually, the targets of interferometers are flat reflectors, but with the TWI, the target is an asphere or even a freeform. The violation of the null-test induces retrace errors in the measurement because incoming and outgoing rays to and from the surface take uncalibrated paths through the interferometer [104]. The TWI's solution to overcome the problem of standard interferometric null-test is through the usage of a set of tilted waves which will locally compensate for the deviation of the surface under test from the spherical shape [102]. Pruss *et al* [103] propose a technique to overcome null-test violation based on the usage of optical elements which adapts the wavefront to the design shape of the surface. The optical elements evoked are some state-of-the-art technology referred to as

Computer Generated Holograms (CGH). As compared to other technologies that take into consideration retrace errors, the CGH technology is the most precise but is expensive and time consuming since CGH must be produced for each different surface.

The basic setup of the TWI is that a coherent laser source L is split into two waves, a reference wave and a test wave using a polarizing beam splitter (Fig. 46). The test wave travels through a micro lens array followed by a pinhole array (PSA). The exiting rays are a set of beams that will go through a beam splitter. Then a collimator C1 transforms the incoming spherical wavefronts into a set of plane wavefronts with different amounts of tilt. Those wavefronts are transformed back into spherical wavefronts thanks to a transmission sphere TS in order to compensate the basic spherical form of the surface under test (SUT). All wavefronts are reflected back onto the beam splitter. Then, they propagate and cross an aperture **A** that filters fringes having a density larger than the Nyquist criterion. An imaging optics L1 projects the rays onto a camera arm C [96]. The main contributions of this interferometer as compared to scanning-type interferometers are that the TWI can acquire data in parallel and without the need to move the SUT during measurement.

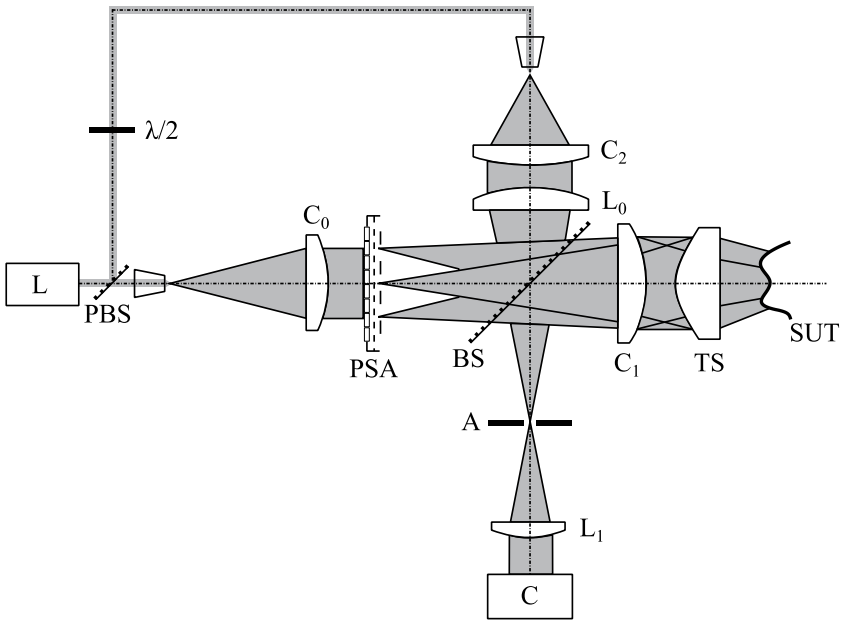


Fig. 46 The Tilted-Wave Interferometer setup. (L: coherent laser source, PBS: polarizing beam splitter, BS: beam splitter, $C_{0,1,2}$: collimators, PSA: micro-lens array followed by a pinhole array, $L_{0,1}$: imaging optical lens, TS: transmission sphere, SUT: Surface Under Test) [96].

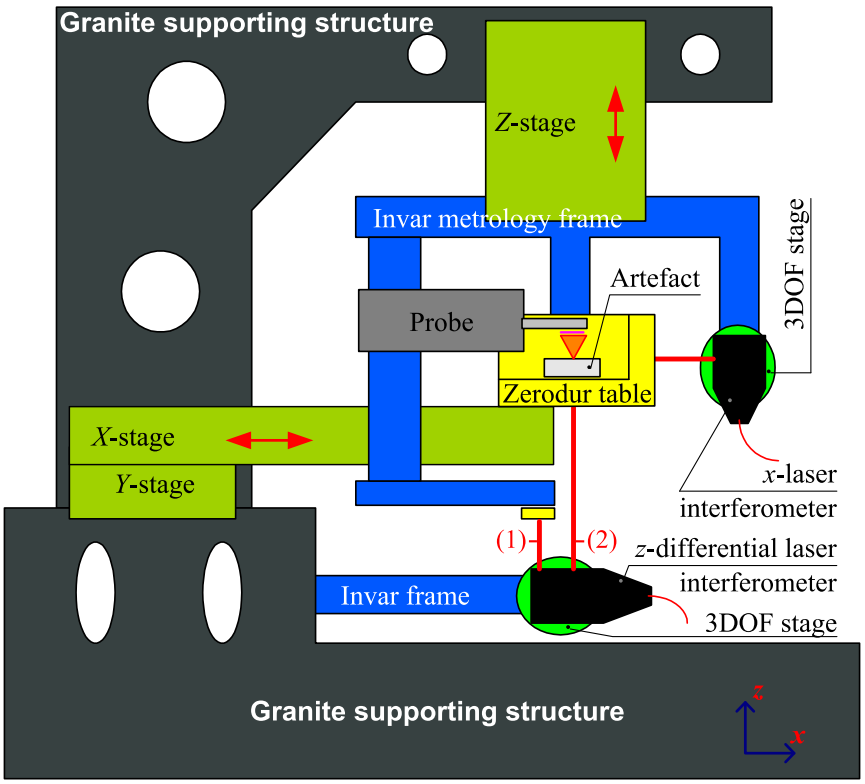
The TWI measuring apparatus is capable of performing a measurement over an asphere within few seconds, provided that the angle deviation from the best fit sphere does not exceed

10 °. This interferometer also allows for automatic alignment, high flexibility and high lateral resolution. The current aim is to integrate this measurement technique in the process chain of asphere and freeform manufacturing.

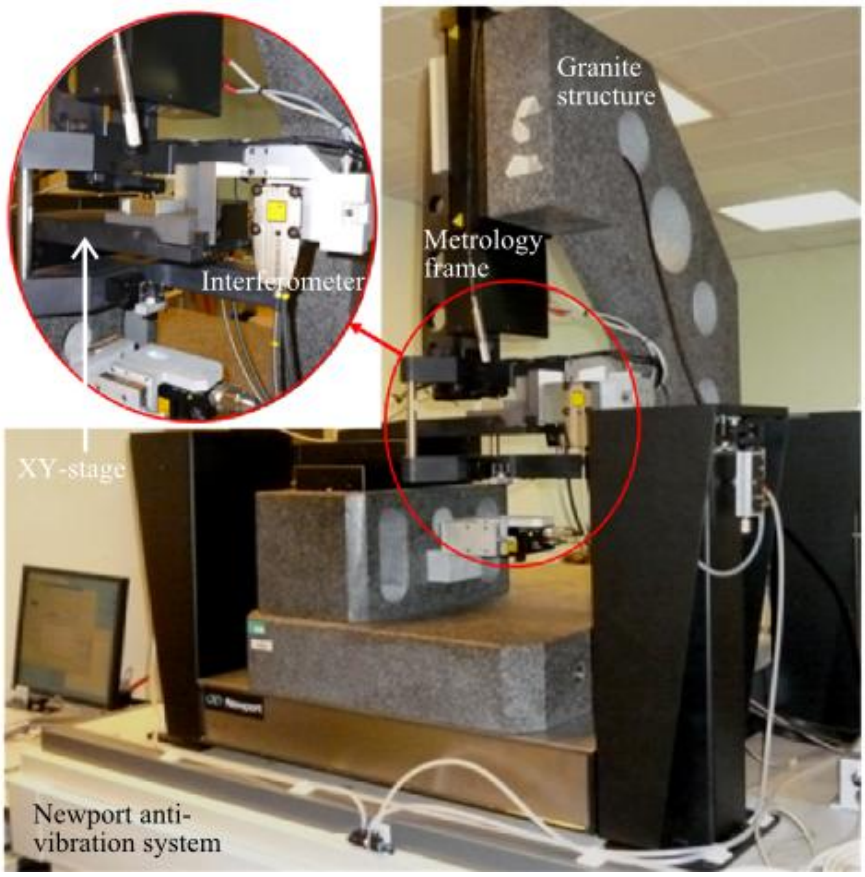
2. The LNE high-precision profilometer

In the same context, the LNE developed its own machine and validated the usage of this machine for the measurement of planar surfaces roughness with nanometric accuracy. Just like the previously described machines, the architecture of the LNE's high-precision profilometer perfectly complies with the Abbe principle. The metrology loop is optimized to be as short as possible. The design of the profilometer allows for both tactile and chromatic confocal probing. On this apparatus, three high precision guiding axes equipped with encoders insure three independent translational degrees of freedom, in x-, y- and z- directions (Fig. 47). A Zerodur table on which the measured object is posed travels along x- and y- directions and its movement is controlled by two independent Renishaw laser interferometers to a nanometric level of accuracy. The working range in the xy- plane is 50×50 mm². The fixture of the Zerodur table on the top side of the x-mechanical guiding system is carried-out via three balls with a diameter less than 10 mm, to insure isostatic linkages. The probe and its supporting structure are mounted on the vertical guiding system in the z- direction along which the measurement is done (Fig. 47 and Fig. 48). The working range of the mechanical guiding system in z- direction is about 100 mm but the practical working range strongly depends on the travel range of the probe used. A third Renishaw differential laser interferometer controls the movement in z with a nanometric level of accuracy and its use allows reducing the metrology loop (Fig. 48). A differential laser interferometer directly measures the phase difference between two parallel emitted beams separated by a small spatial offset, thereby providing a more direct determination of local displacement. The metrology frame involves parts and components made of Invar which makes it less sensitive to thermal expansion and other environmental fluctuations. The thermal expansion coefficient of Invar is about 1 µm/m/°C. The thermal behavior of the metrology frame made of Invar with the dimensions of $200 \times 200 \times 200$ mm^3 is estimated by varying the surrounding temperature by 0.1 °C. It generates a temperature change in the Invar structure of less than 0.01 °C, especially when the environment temperature varies smoothly. For this case, the thermal expansion of the metrology frame is estimated to 2 nm which can be considered small. For

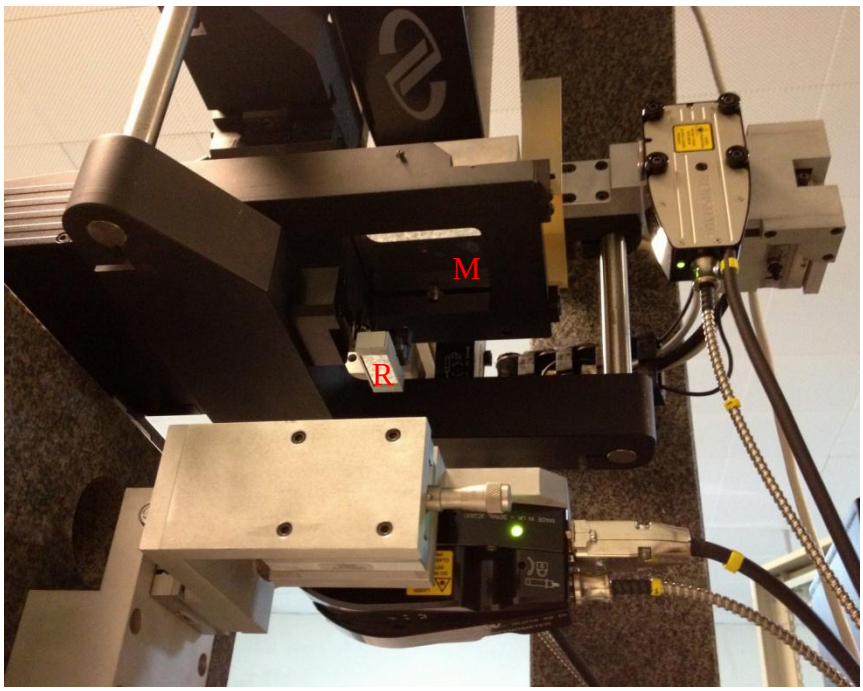
Zerodur, the thermal expansion coefficient is about 0.05 µm/m/°C and the dimensions of the table are $200 \times 200 \times 50$ mm^3. For the same temperature variation, the thermal expansion of the table is even smaller and is estimated to 0.5 nm. The mechanical guiding systems, the probe and the metrology frame are all supported by a structure made of massive granite. Any vertical expansion or deformation of the supporting frame does not influence the metrology frame since the vertical motion is controlled by the differential laser interferometer. The vertical thermal expansion of the granite structure induces an identical variation of the first and second laser beams of the differential interferometer and is therefore directly compensated. In differential laser interferometry, only the variations of the distance between the external reference (R) mirror and the external moving mirror (M) are taken into account (Fig. 47c). The advantage of differential interferometry is that it reduces the metrological chain (Fig. 48). The high-precision profilometer applies the dissociated metrology frame principle which means that the metrology frame is dissociated from the supporting frame. The metrology frame is fixed on the supporting frame using isostatic links (flexible blades) to avoid any transmission of eventual mechanical strain induced by the supporting frame. As a consequence, the metrology frame supports its own mass and only performs the function of measurement [44,98].



(a)



(b)



(c)

Fig. 47 The LNE's high precision profilometer. (a) architecture of the apparatus. (b) Picture of the apparatus. (c) The differential laser interferometer system (R is the reference mirror and M is the moving mirror).

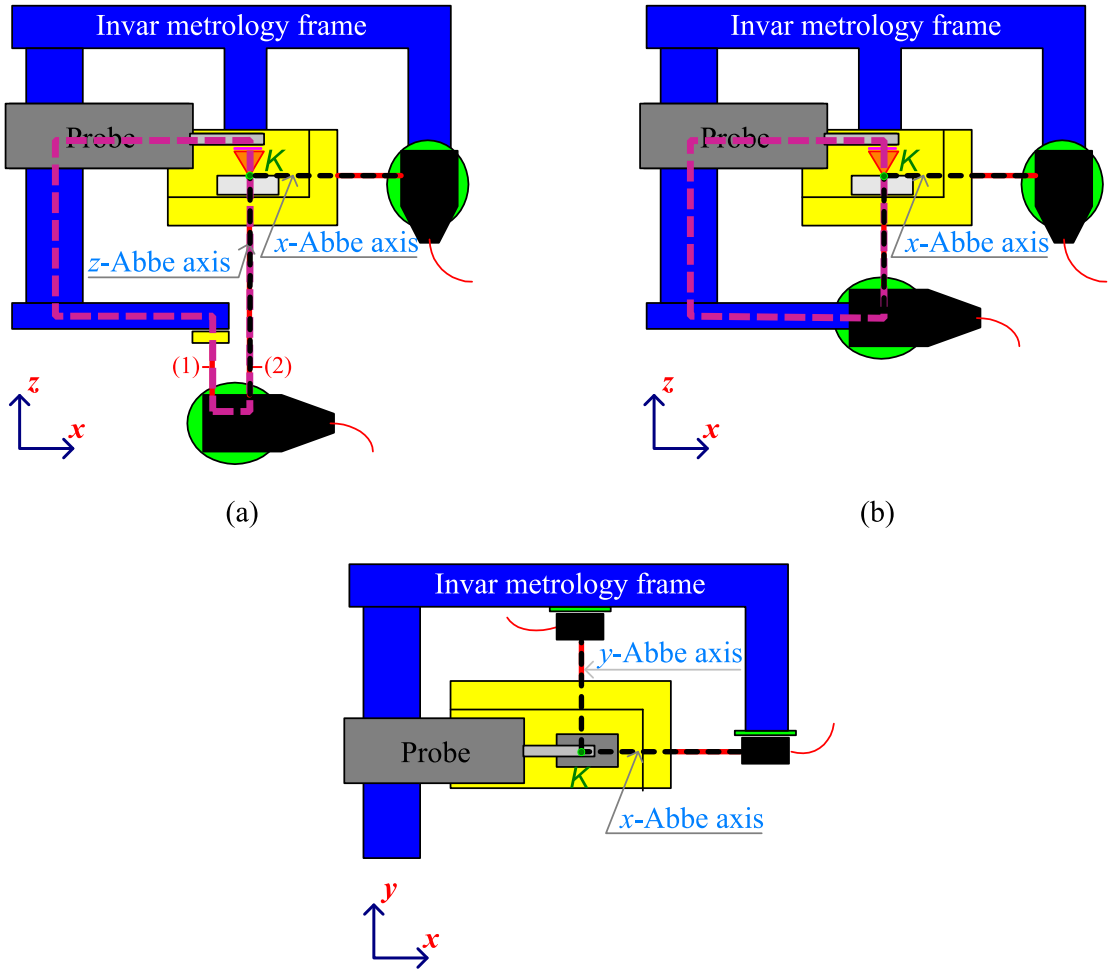


Fig. 48 A differential interferometer reduces the metrology loop. (a) Differential interferometer; (b) Standard interferometer; (c) Top view of the Abbe axes.

The machine respects the Abbe principle in all directions [90]: the measuring probe's axis and the differential laser interferometer's beam are collinear during the measurement operation. However during in-situ calibration, the Zerodur table remains fixed. This means that the reference mirror facing beam (1) in Fig. 48a becomes the moving reflector and the underside of the Zerodur table becomes the reference reflector. The touching element of the contact probe in the case of tactile measurement, or the focus point of the optical single point probe in the case of an optical measurement, are coplanar with the x- and y- laser interferometer beams. Since the x- and y- laser interferometers and the probe are all on the same metrology frame, any displacement of the frame induces a displacement of all these elements. The machine is configured to hold both tactile and optical single point scanning probes that can be calibrated in-situ. The Zerodur table is controlled by the three laser

interferometers as mentioned above and shown in Fig. 47a and b, so the reflecting elements and the interferometers should be well aligned. Each interferometer beam must be perpendicular to its target reflecting mirror and collinear with the respective direction of motion within the acceptable angle of 25 arc-seconds. A four quadrant photodiode fixed on the moving table is used for the alignment of each laser beam with the direction of motion. The laser beam must theoretically remain focused at the center of the photodiode over the entire travel range. The misalignment is measured and the average value found on this machine for x- and y- motions is about 50 µrad per 50 mm range. The alignment error is estimated to 6.25×10^{-8} mm and considered negligible. Since the x-, y- and z- motions are independent, the mirrors facing the laser interferometer beams should be orthogonal among themselves. The evaluation of orthogonality is performed using the LNE's coordinate measuring machine ("CMM5") which is accurate to 0.5 µm over a working volume range of 0.5 m^3. To guarantee such a volumetric uncertainty, the translation errors (two straightness and one positioning) for each mechanical guiding system, and the rotational errors (pitch, yaw and roll) are calibrated using a ball-bar (alternatively hole-bar) system. Many other instruments can be used for the calibration of measuring machines such as step gauges, gauge blocks, ball plates, the Zeiss check artifact, hole plates, ball-ended bars, laser interferometers, tracking interferometers and tracer interferometers. The perpendicularity between each two axes is calibrated twice: first, using the ball-bar and then using an angle gauge block. For the perpendicularities between the x-, y- and z-axes, the uncertainty is estimated to $0.8''$. More details about the calibration of CMM5 are widely presented in [105–108].

The perpendicularities between the different sides of the Zerodur table are measured by the CMM5 machine (Fig. 49). At least 10 points are measured on each side and the Least-Squares plane is fitted. The angles between normal directions to each of the planes are α_1, α_2 and α_3 and are equal to $90° 00' 10''^{\pm 0.8''}$, $89° 59' 34''^{\pm 0.8''}$ and $89° 59' 18.1''^{\pm 0.8''}$, respectively. These misalignments are tolerated since they are identified and compensated in software [14].

The motion errors of the guiding elements induce inclination of the Zerodur table and must also be corrected in the software. These errors are characterized using the long-term extremely stable and accurate probe (0.001 mm/m), Leica Nivel20 shown in Fig. 50. For the 50 mm working range of the apparatus, the motion induced inclination errors are below 1 nm.

The high precision profilometer is placed in the LNE's cleanroom where environmental conditions are optimal. The temperature is controlled to $20^{\pm 0.3}$ °C and humidity to $50^{\pm 5}$

%RH. The variation in temperature is very slow and smooth in the bandwidth ± 0.3 which leads to a very low temperature variation in the parts of the machine.

The Newport anti-vibration system as shown in Fig. 47b attenuates all low-frequency vibrations generated by the surrounding environment. Furthermore, all the above system is mounted on a concrete anti low-frequency vibration massif that isolates it from the room floor.

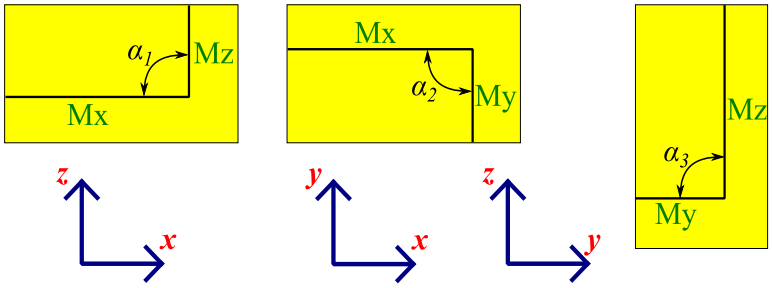


Fig. 49 The moveable Zerodur table angles check by CMM5.

The uncertainty budget established for the measurement according to the GUM [4] takes into consideration all of the aforementioned error sources such as: the error motions of the mechanical guide systems, the Abbe and cosine errors, the dynamics of the machine, the geometry of the Zerodur table, thermal drift and the tactile probe and laser interferometer errors. For the case of a flat artifact, uncertainty budget for a tactile measurement is established considering all sources of error. It results in an expanded uncertainty of $15 + 10^{-6}\,L$ nm, using a coverage factor k of 2. This uncertainty is mainly affected by the performance and the behavior of the probe which will be detailed later. The stated value is only valid for a flat artifact measured by tactile probing. When using chromatic confocal probing on aspherical artifacts, the uncertainty budget should be re-evaluated.

Fig. 50 The Nivel20 inclination sensor of Leica industry.

From contact measurement techniques, which were firstly introduced by McMurtry (Renishaw) in 1972 and then implemented by Zeiss in 1973, to the wide explosion of non-contact measurement techniques, metrology has seen a remarkable evolution and a great diversification of technologies [34,35].

Contact measurement on the LNE's high-precision profilometer

In the field of dimensional metrology, the most widely used measurement technique is stylus profilometry because this process is very well understood. A tactile probe, or stylus probe, is a sensor that measures the profile of a surface through contact with the surface. The tip of the probe is a high precision sphere of a given diameter and is located at one end of a beam which pivots around its center (Fig. 51). When the probe tip touches the surface, it induces a rotation Φ of the beam which at the other end consists of an armature that displaces between two coils. The variation of the position of the armature changes the relative inductance between both coils which are connected in an AC bridge circuit. This means that when the armature is centrally positioned, the AC bridge is balanced and no output is generated. Whereas, when the armature moves, the AC bridge is unbalanced and the relative inductance between the coils varies, generating a proportional output to its displacement. The direction of displacement is identified by the phase of the signal. Contact probes present lower repeatability than some contactless probes due to their mechanical nature. In fact, coils do not always return to their original positions after each reading. However, progress in technology has considerably improved the quality of mechanical linkages reducing the bias errors due to hysteresis down to few nanometers [92].

On the LNE's high-precision profilometer, the used tactile single point scanning probe has a stylus tip angle of 90 °, a tip radius of 2 µm and a static measuring force below 0.7 mN (Fig. 51). Its measuring range can be selected among three possible ranges, the smaller range $0 - 100$ µm, the medium range $0 - 500$ µm and the larger range $0 - 1000$ µm, depending on the maximum depth to be measured.

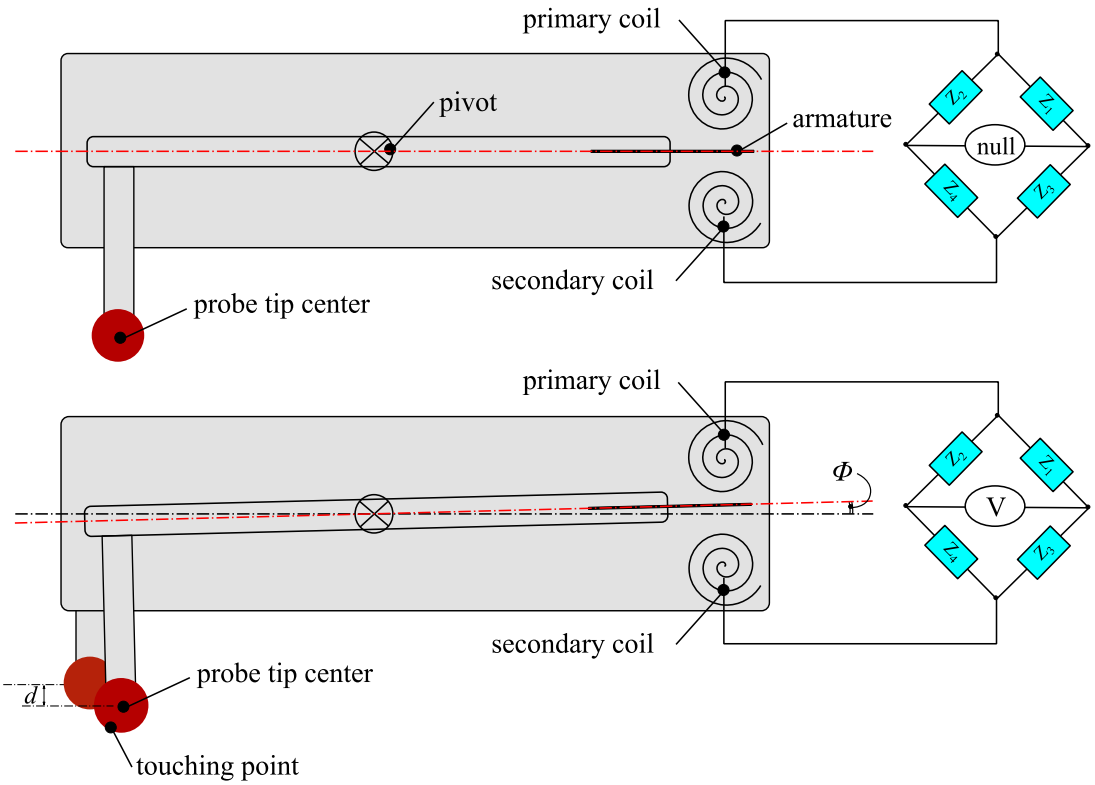
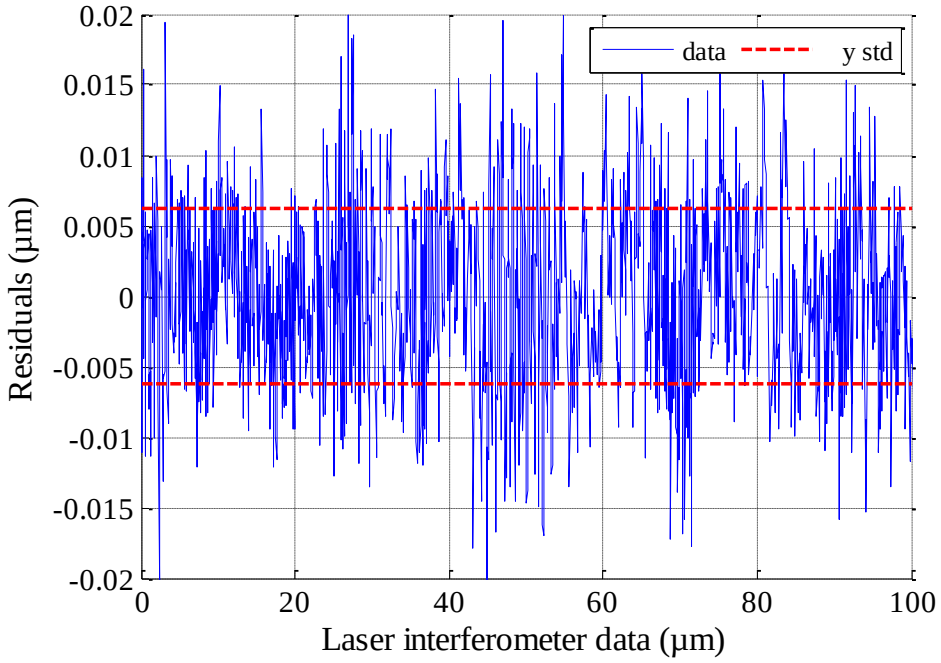
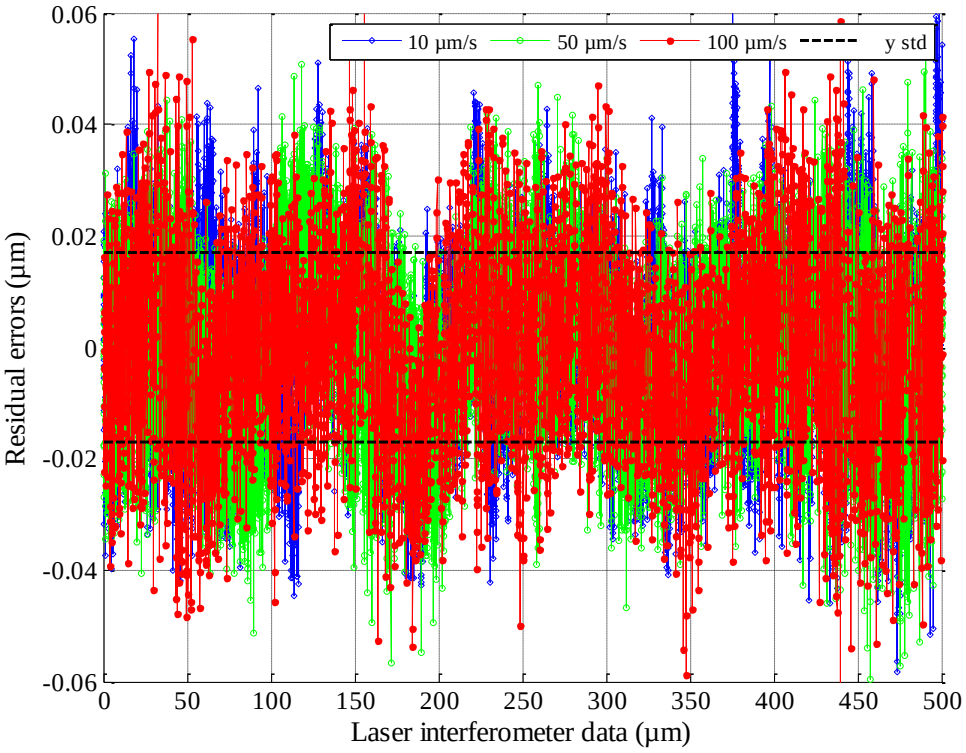


Fig. 51 The stylus probe principle.

Before real measurement and as part of the preprocessing phase (Fig. 8), the probe is first calibrated in-situ since the LNE high precision profilometer offers this possibility. During in-situ calibration facing a flat artifact, the Zerodur table is kept fixed and the metrology frame is put into motion over the entire travel range with perfect respect of the Abbe principle. The tests are repeated 60 times and data are recorded and fitted by a 9^{th}-order polynomial model. For the smaller range, the residual errors vary between ± 20 nm with a standard deviation of ± 6.5 nm (Fig. 52a) and for the medium range, the residual errors vary between ± 60 nm with a standard deviation of ± 18 nm (Fig. 52b). The repeatability of the measurements with the tactile probe has been tested and approved to be less than 3 nm.



(a)



(b)

Fig. 52 In-situ calibration of the tactile probe: Evolution of the residual errors versus the displacement measured by the z-differential laser interferometer with a 9^{th} order polynomial approximation of the data: (a) over its smaller range of 100 µm at a fixed scanning speed of 10 µm/s giving a standard deviation for residual errors (y-std) less than 6 nm (red). (b) over its medium working range of 500 µm for three different scanning speeds (10, 50, 100 µm/s) giving a standard deviation for residual errors (y-std) less than 18 nm (black dotted line).

The main limitation of contact metrology is inherent to its nature, more precisely, to the fact that a contact is established between the probe and the measured object. For non rigid objects, contact might cause surface deformations and alter the shape of the object. Although it has been taken care of recently by error compensation, the contact point between the probe tip and the surface changes constantly and this also adds to the uncertainty of the measurement. Moreover, spatial resolution is limited by the diameter of the spherical tip (7).

$$\frac{1}{N.\Delta} \leq f_{spatial} \leq \frac{1}{2.\Delta} \tag{7}$$

where, N is the number of points on the profile and Δ is the spatial sampling step itself dependent of the diameter d of the sphere ($\Delta_{min} = d$).

Contactless measurement on the LNE's high-precision profilometer

It is true that stylus profilometry has become more accurate, but it is not free of limitations [109] and this is why non-contact technologies have become the major focus in today's metrology research fields. As mentioned above, the LNE's high-precision profilometer accepts confocal probing too (Fig. 53). The principle of such a system based on white-light chromatic confocal technology is illustrated in Fig. 54.

Fig. 53 Chromatic confocal probing system: (a) micro-epsilon [110]; (b) STIL [111].

A LED sends a white light beam through a lens that diffracts emerging light into spectral waves. These spectral waves are directed towards the surface being measured, then reflected back and analyzed in a spectrometer. Only the wavelength (λ) that is best focused on the surface (point M) of the object being measured passes through a mechanical filter called the

pinhole [91,112]. This wavelength is in reality a small range of wavelengths which will be diffracted onto a CCD camera and analyzed. Hereafter, the system searches for a peak in the signal and finds the location of this peak on the CCD. Equation (8) shows how the location of the peak is related to the magnitude of the focalized point M, therefore to the distance d separating the chromatic confocal probe's head from the surface being measured.

$$d = f(\lambda) \tag{8}$$

where d is the sought distance and λ is the peak wavelength.

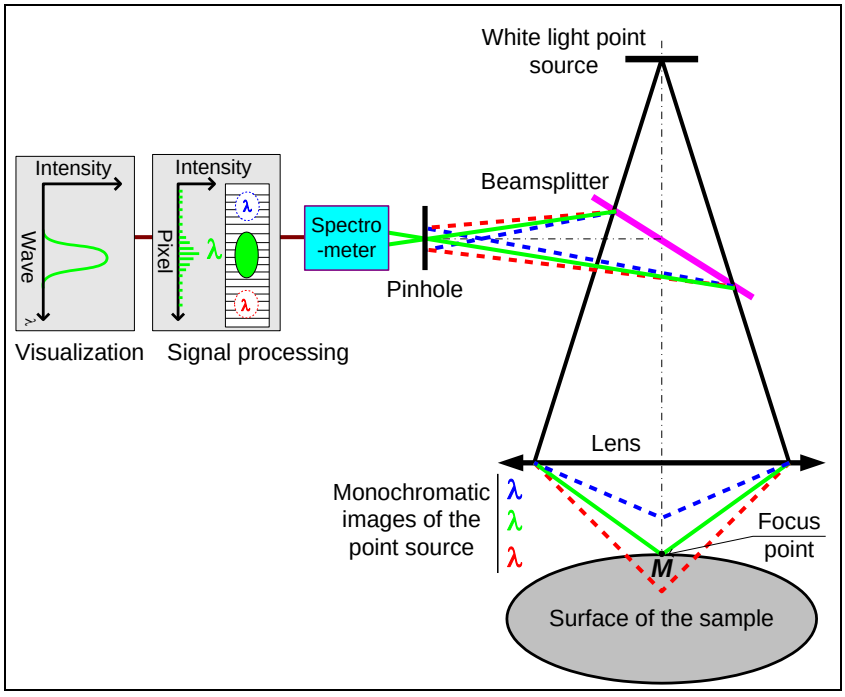


Fig. 54 The working principle of the chromatic confocal probe [113].

The chromatic confocal probe is firstly calibrated on a separate test bench specifically designed for the identification of all sources of errors involved in the confocal probe measurement. The sources of errors are listed in Fig. 55 as detailed by Nouira *et al*. in [113].

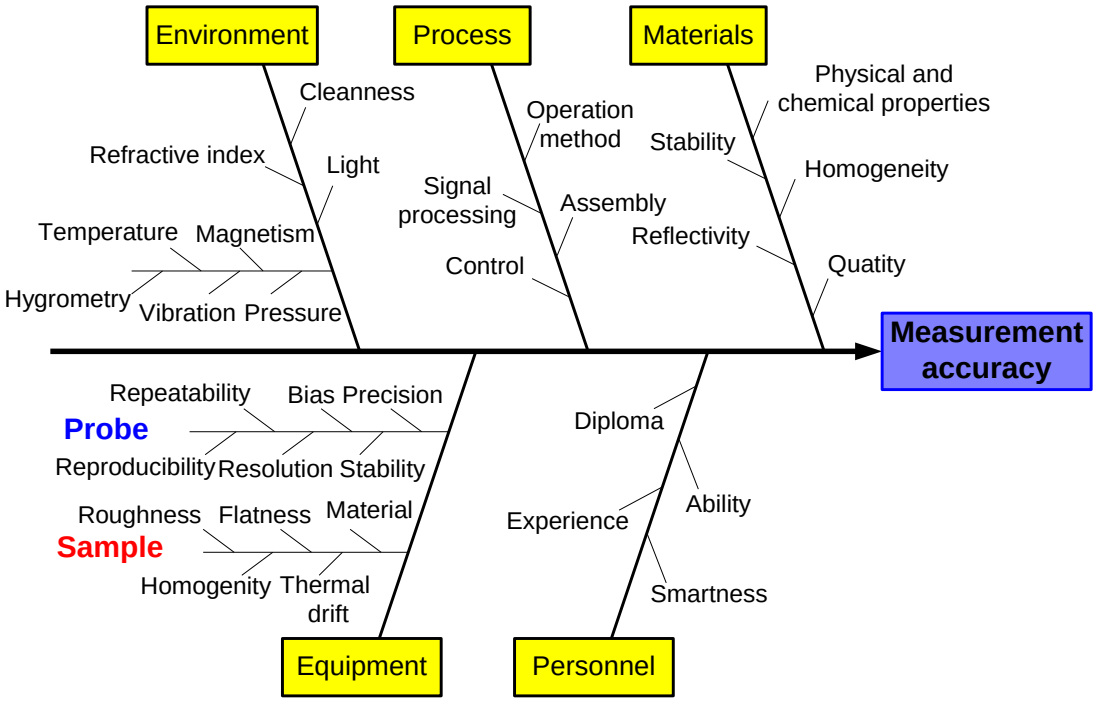


Fig. 55 Ishikawa diagram: measurement error sources affecting uncertainty. Sample refers to the measured workpiece (sample: workpiece).

For this calibration on a flat standard, most of the cited systematic errors are identified and assessed separately through different calibration tests described in detail by Nouira *et al.* [113]. The nano-scale confocal probe used for the measurement is calibrated on a separate calibration bench which respects the Abbe principle and includes two laser interferometers as reference sensing elements (Fig. 56). The intrinsic properties of the flat standard (material, color/reflectivity and roughness), the inclination/tilt of the probe or the workpiece, the intensity/power of the light source, the acquisition frequency and the scanning speed may influence the maximum measurable local slope and the behavior of the probe ([91] and ISO 25178-602:2010 [114]). The surface roughness creates a diffuse reflection of light. The small part that is reflected should be enough to detect the peak position in the intensity curve recorded in the spectrometer. The variation of the percentage of reflected light may affect the quality of the spectrum (intensity curve) [91].

The calibration of the optical confocal probe on a flat target after identification of the sources of error and their integration into a piece-wise linear model constituted of 2048 linear parts, show that the residual errors are considerably reduced (Fig. 57).

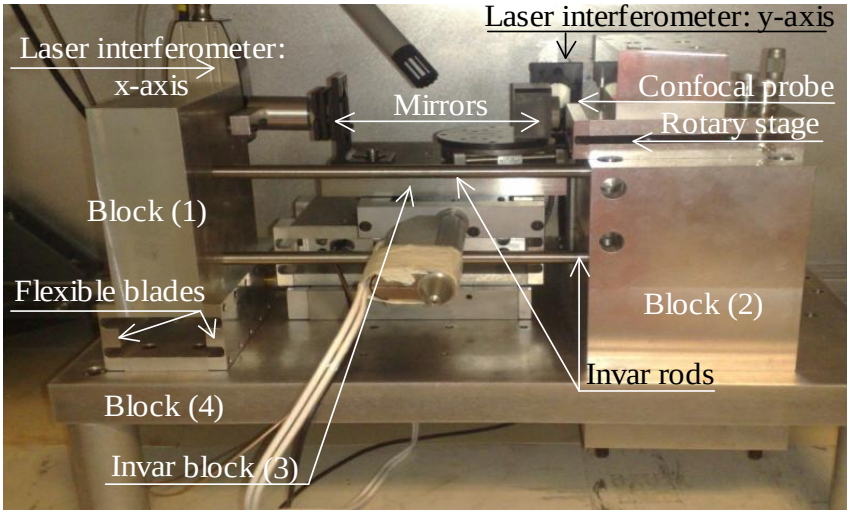


Fig. 56 The calibration bench designed for the characterization of the chromatic confocal probe used in the measurement of aspherical lenses [115].

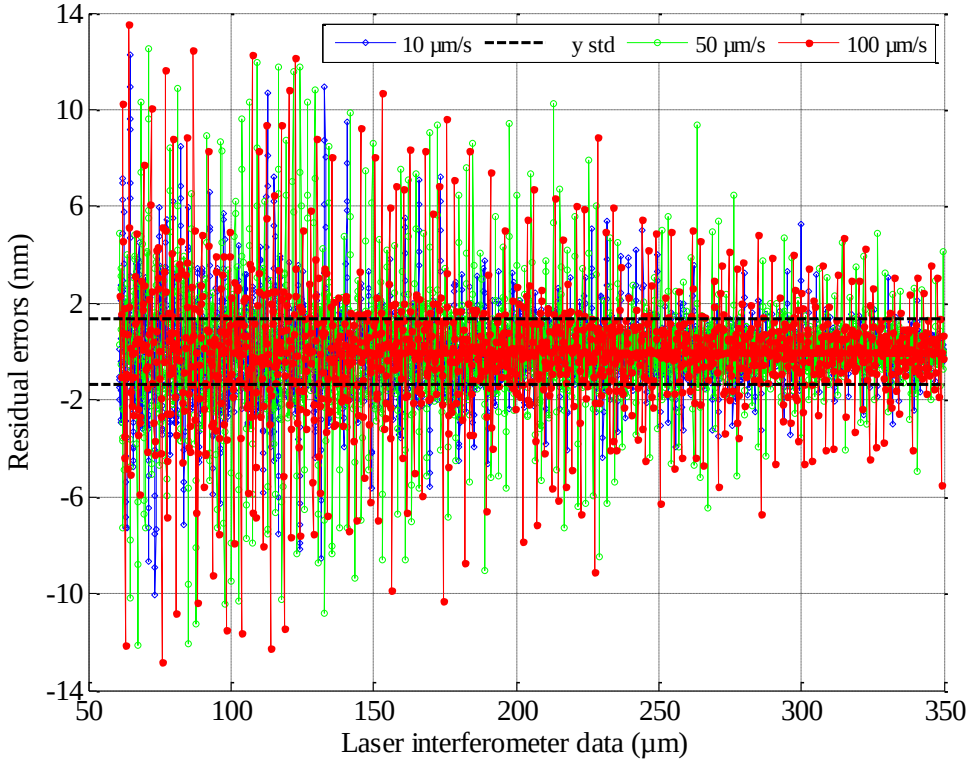


Fig. 57 In-situ calibration of the chromatic confocal probe over the entire working range of 350 µm. Evolution of the residuals versus the displacement measured by the *z*-differential laser interferometer for three values of speed: 10, 50 and 100 µm/s (blue, green, red). Modeling of the data with a piecewise linear model of 2048 models giving a standard deviation of residual errors below 2 nm (black).

Interferometry as a reference for the measurement

Since the LNE high-precision profilometer is equipped with laser interferometers, the principle of interferometry is worth the description. Interferometry is the most precise optical measuring device and when calibrated, it serves as a reference for the actual measurement in conformance with the traceability pyramid (Fig. 3). Interferometry lies on coherence, which is related to the ability of light to undergo interference. The interference in interferometry is the result of merging two coherent wavelengths, each coming from a distinct path. The Michelson interferometer is the most basic interferometer and easily illustrates the principle of interferometry (Fig. 58).

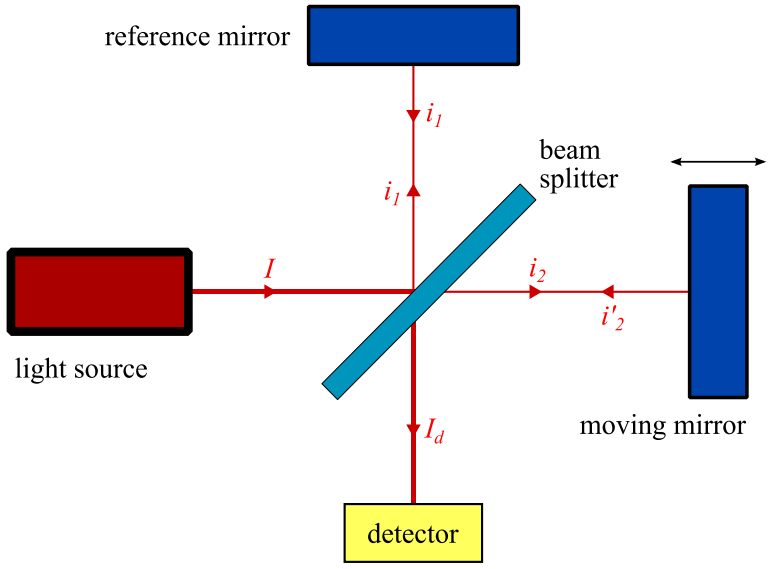


Fig. 58 The Michelson interferometer.

An initial light beam (laser or white light), also called the emitted beam I, is projected on a beam splitter that splits I into two separate beams, i_1 and i_2. i_1 travels towards a reference mirror that does not move and is reflected back as is. i_2 is directed towards a moving mirror which is normally the target of the interferometer (the measured sample) and is reflected back with an altered signal i'_2. The difference between i_2 and i'_2 is a small phase shift that is due to the displacement of the moving mirror. When i_1 and i'_2 are merged, the signal I_d in the receiver is phase-modulated (9).

$$I_d = i_1 + i'_2 + 2\sqrt{i_1 . i'_2}.\cos(\varphi) \quad (9)$$

where the phase shift φ is directly proportional to the displacement δ of the target and the laser source wavelength λ:

$$\varphi = 4\pi \frac{\delta}{\lambda} \quad (10)$$

Differential laser interferometers apply the same principle with the difference that they emit two equal and parallel signals having different frequencies [116].

Measurement of the AO775 aspherical lens on the LNE's high-precision profilometer

The aspherical lens AO775 that is measured is illustrated in Fig. 59. This lens was manufactured by Anteryon® company using a Single Point Diamond Turning (SPDT) process and finished with a high precision polishing process and glass coating. It has a rectangular base of dimensions 11.1×19.2 mm² and a height of 2.2 mm.

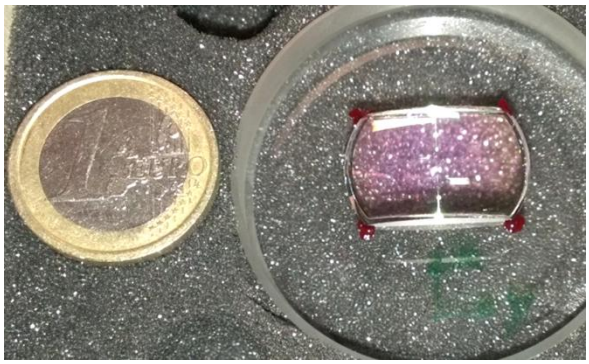


Fig. 59 The AO775 aspherical lens model.

The lens is mounted on the LNE high-precision profilometer for the measurement process. The asphere is posed on the Zerodur table (Fig. 47a) and a manual alignment process is performed. On this machine, it is not possible to exactly align the asphere's axis of symmetry with the z-axis of the measurement (Fig. 60), 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.

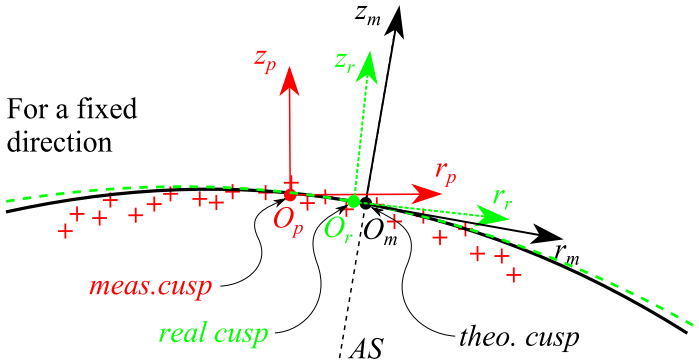


Fig. 60 Illustration of the problem with the alignment of the measurement coordinate system. Indices m, p and r stand for model, measured point-set and real surface, respectively; *AS*: theoretical axis of symmetry.

Once the software-aided alignment is done, the data are recorded and reported in Cartesian coordinates (X, Y, Z). The surface is scanned over a rectangular area using either a tactile probe or a confocal probe (Fig. 61). The measurement area depends on the measuring range of the probe used, the shape of the measured surface and the level of uncertainty sought. For the tactile probing, the measured area is 6×6 mm² because we restrained the measuring range to $[0 - 100]$ µm which has an associated uncertainty of ± 6.5 nm. Beyond this measuring range, the uncertainty deteriorates. For the chromatic confocal probe, OP350 µm by STIL, the measurement area is 5×5 mm² and this is totally limited by the probe's uncertainty. The comparison and effect of using either of these probes is detailed in the works of El-Hayek *et al* [117], [118].

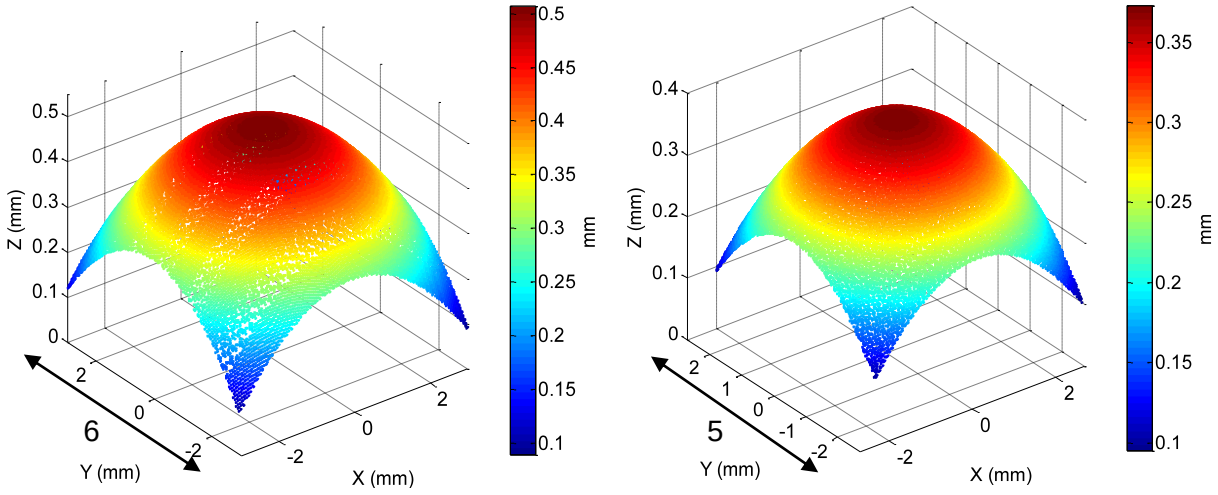


Fig. 61 The aspherical lens AO775 measurements. (a) tactile measurement; (b) confocal measurement.

Measurement of the AO775 aspherical lens performed by the project partners

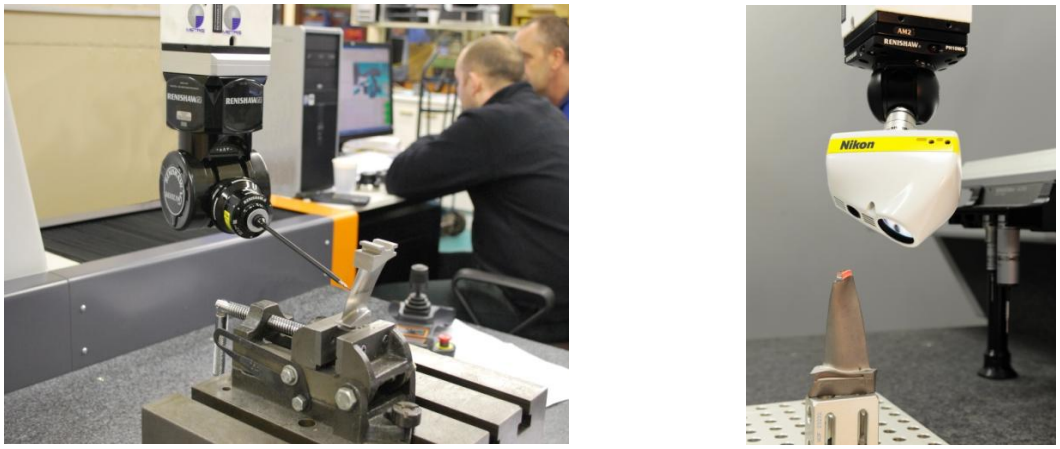
The partners of IND10: FORM project have all measured the same AO775 asphere on their respective UHPMMs in order to assess the measurement capabilities of each of them and compare the results in an attempt to come up with a standard procedure for characterizing aspherical surfaces. The comparison results given by the partners will be given in chapter 2.

3. Coordinate Measuring Machines for the measurement of turbine blades

The measurement of turbine blades, considered as highly freeform, presents real challenges. In some cases, visual inspection is applied where the blade is posed against a master template and the inspection of the gaps between the blade and the master is performed using either a light source or microscopy [8]. However, the mostly used metrology system is the CMM, an accurate physical realization of a 3D rectilinear Cartesian coordinate system [119]. CMMs allow for measurement with relatively high accuracy because they are assisted with numeric control [120]. Generally, every measured point is only represented by its Cartesian coordinates, except for advanced systems which can also indicate the normal direction at a measured point [119]. A measurement on CMM is accompanied by an alignment process which consists of aligning the measured part to a reference model so that the coordinate systems are identical. As depicted by Savio *et al* [8], alignment is a major difficulty and then the choice of the measuring instrument is also variable.

Depending on the application, CMMs can have different configurations. Pereira *et al* [119], enumerate a few configurations, out of which the horizontal arm configuration is specific to car bodies measurement and the Gantry configuration is specific to aerospace structures measurement.

CMMs can be equipped with either contact or non-contact probing systems (Fig. 62a and Fig. 62b). According to a recent keynote paper, Weckenmann *et al* [46] assert that CMMs equipped with contact probes can measure up to 200 points per second at speeds as high as 150 mm/s. Non-contact probes such as laser scanners can perform much faster measurements of some thousands of points per second. Since turbine blades have particular specifications, faster measurement strategies using contact probing can be employed. For instance, cross-sectional or spiral-like scanning can be used and generate less noise. For helix scanning the part rotates around an axis and the measuring machine performs a motion along the same axis. In freeform metrology in general, it is however fundamental to measure points covering the entire surface area of the part so that a precise inspection is insured.

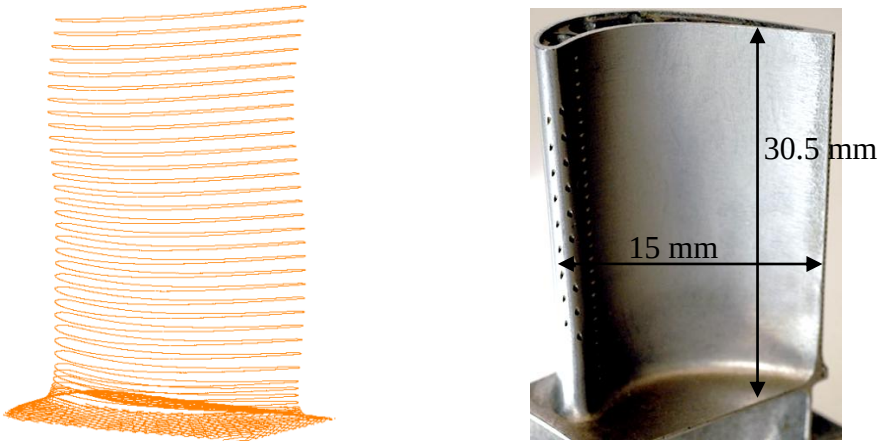


(a) (b)

Fig. 62 Measurement of a blade: (a) Contact point-by-point CMM measurement (METRIS machine) [121]; (b) Non-contact laser scanning CMM measurement (Nikon metrology labs).

Due to the advances of technology, today's CMM metrology involves also optical measurement devices that lean towards being fast and independent of referencing. Here we cite the most common optical principles involved: autofocus and triangulation. The turbine blade presented at the beginning of the report is measured using a laser triangulation instrument mounted on CMM (Fig. 62b). The measurement strategy is in the form of parallel cross-sectional lines scanned along the length of the blade but in 4 separate raw scans: one

raw scan has been done facing the pressure surface of the blade, another facing the suction surface and two separate scans facing the leading and the trailing edges, respectively. The actual cloud of points of the blade provided by Geomnia [16] is the result of a registration operation performed on the 4 raw scans (Fig. 63).



(a) (b)

Fig. 63 The turbine blade provided by Geomnia: (a) the measured dataset; (b) the real part.

Measurement using laser triangulation is among the mostly used techniques in freeform metrological applications, although laser scanners are not the most precise instruments (Fig. 64). But as specified in the beginning of the chapter, turbine blades are classified in a cluster that does not need ultra-fine precision measurement.

Fig. 64 OptoNCDT single-point laser scanner by micro-epsilon [110].

The principle of triangulation states that a distance can be measured by knowing the length of any side and two angles in a triangle [112] (Fig. 65). The knowledge of the relative position of

both the detector (CCD or CEMOS camera or Position Sensing Device (PSD)) and the laser source with respect to the target being measured makes the calculation of the coordinates of the measured datum straightforward by simple trigonometric relations.

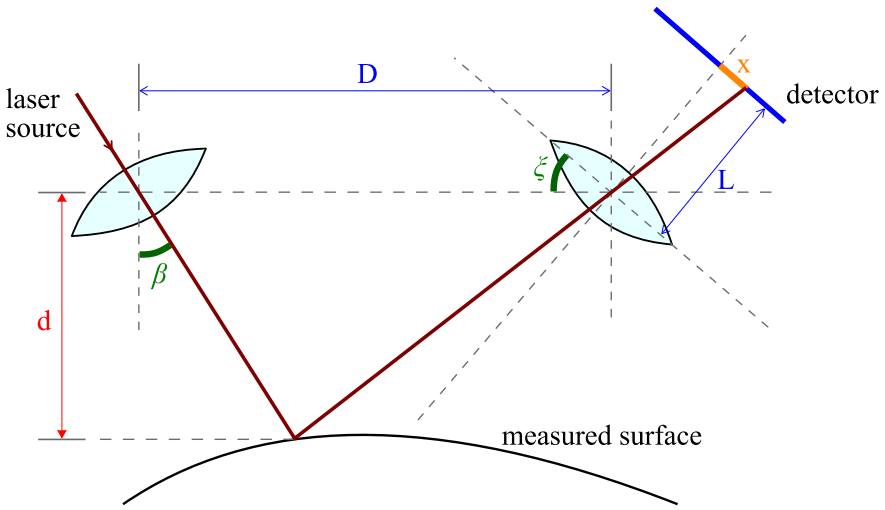


Fig. 65 Laser triangulation principle. The distance d is the unknown that is computed during measurement.

The major drawback of using optical instruments in the measurement of turbine blades is that the leading and trailing edges can offer quite a challenge in data acquisition [8].

VI. Conclusion

Aspherical and freeform surfaces have complex geometries and have been classified according to their shape complexity, relative tolerance and invariance class. Aspherical surfaces can be completely freeform but we will be interested in just the revolute class invariant aspheres. Turbine blades are classified as highly freeform surfaces and their specification is done according to cross-sectional profiles or surface partial features.

Both aspherical and freeform shapes need to be precisely manufactured in order to meet their intended performances. Aspherical optics require a nanometric precision while turbine blades do not generally go below a sub-micrometric precision. Aspherical optics which have found many applications because of their unmatched designs are mainly produced with high precision manufacturing processes, the main ones being SPDT and molding, whilst molding inherently involves SPDT for the fabrication of the mold. Turbine blades are mainly

manufactured either using closed die hot-forging techniques or on 5-axis milling machines but other methods such as electromechanical machining and molding exist.

Although high-precision manufacturing technologies exist and are applied today, measurement remains inevitable. Measurement is what gives an insight of the dimensional characteristics of a shape as well as of its form. High-precision measuring instruments are available in the market, the mostly used being the stylus that can achieve measurements with nanometric level of uncertainty. Despite the advances of stylus profilometry, this technique still shows the evident limitation of being in contact with the target. Contactless techniques have invaded the market and the most commonly used are point and line laser triangulation, chromatic confocal single point probing and interferometry. Interferometry is the most precise having a level of uncertainty below the nanometer, whereas laser triangulation is the least precise achieving micrometric uncertainties.

The AO775 aspherical lens model is measured by the different project partners using UHPMMs. The LNE performed two measurements of AO775 in order to compare tactile and confocal single-point measurement probes and to characterize its form errors (c.f. chapter 2). The tactile probe has been calibrated on a flat artifact and the resulting uncertainty with systematic error compensation amounts to 6 nm. The confocal probe has been calibrated also on a flat artifact and the resulting uncertainty amounts to 2 nm. The laser scanner used to measure the turbine blade has an announced uncertainty of some tenths of micrometers. We also introduced two different definitions of an aspherical surface, the classical and the Forbes definitions, which we will compare in the next chapter also. The Forbes definition consists of a polynomial part with independent polynomial coefficients.

The turbine blade provided by Geomnia was measured using a laser triangulation technique. Due to the fact that the resulting cloud of points is the outcome of separate raw scans, neither order nor structure are inherent to the data points. Therefore, we choose to tackle the turbine blade surface reconstruction problem in its most generic aspect by disregarding all assumptions about measurement strategy. Since the actual specifications of turbine blades are on cross-sectional profiles, we will introduce an algorithm for the reconstruction of curves in the plane and present all the elements for its extension to surfaces in 3D (c.f. Chapter 3).

Chapter 2 - Form metrology of aspheres: characterization and evaluation of fitting algorithms

I. Introduction

The term reconstruction is very broad in the sense that it refers to solving an inverse problem. An inverse problem consists of extracting information from an actual measurement or observation and converting it into parameters which characterize the observed object. While the forward problem has a deterministic solution, the inverse problem does not [122]. Metrologists would rather refer to the term reconstruction as association or fitting. In metrology applications particularly, the mathematical description of the surface is known and is either explicit or implicit and the characterization of simple shapes is very well understood and mastered [123], [124]. Literature concerning the characterization methods of aspherical surfaces is not very extensive and only a few works have been published [13,125] and [126–130]. The European project IND10: FORM has been launched in order to come up with reliable solutions to this matter [131].

When given data to fit, the issues encountered are mainly related to understanding the fitting problem, choosing an appropriate optimization algorithm, identifying the variables that must be optimized, setting the minimization criterion, as well as defining the process by means of which the algorithm is validated. Then, in the context of metrology applications involving large and noisy data, the algorithm must also meet the requirements related to automatism, robustness and time complexity.

The problem of aspherical form evaluation is unconstrained and non-linear so the optimization algorithms studied here are chosen accordingly. We look into classical methods such as the Levenberg-Marquardt (LM) and the Iterative Closest Point (ICP) algorithm and compare them to the newly introduced Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm which will be shown to be more efficient [132]. LM is abundantly used in fitting applications and 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 [133]. This optimization algorithm may sometimes be slower than the simple Gauss-Newton algorithm but at the advantage of guaranteeing convergence when the initial solution is chosen appropriately [134]. The ICP is very popular when it comes to registration or alignment which is nothing but minimizing the relative position and orientation difference between two discrete datasets. It consists of applying a rigid transformation of one point-set towards the other and optimizing for transformation parameters (rotation and translation parameters) only. This operation either serves for the analysis of the resulting deviations between both datasets, or can serve for the fusion of separate raw scans of the same

object [135]. Since ICP does not optimize for shape parameters, we will set the rotation and translation parameters as the parameters to identify for all three optimization algorithms so that these can be compared.

The process of fitting surfaces to data implies that an objective function must be iteratively minimized while optimizing for some variables and evaluating the output at each iteration, such as:

$$\Phi = \min_{\boldsymbol{x}}\big(\boldsymbol{\phi}(\boldsymbol{x})\big). \tag{11}$$

where $\boldsymbol{x}$ is the vector of variables to optimize, $\boldsymbol{\phi}(\boldsymbol{x})$ is the objective function to minimize and Φ is the function value at each iteration.

Fitting surfaces to data also implies that there are two optimization routines: one routine for the calculation of the projections of datapoints, also referred to as footpoints, and a routine for minimizing distances according to a criterion (norm). Ahn [136] gives a thorough description of the paradigm to solve Least-Squares fitting problems for curves and surfaces in space. Ahn underlines the existence of two approaches to the problem, being the *total method* and the *variable-separation method*. In the first approach, the footpoints and the variables are computed simultaneously while in the second approach, footpoints and variables are computed sequentially in a nested scheme. We implement the studied algorithms following a sequential minimization scheme that includes an outer iterative loop for the global minimization of distances and an inner loop for the approximation of footpoints.

We use the Least-Squares criterion, L_2 norm, because of its robustness to outliers and noise and its simplicity in regard to implementation and problem solving. Moreover, it is the common criterion chosen for comparing results among the partners of the IND10: FORM project. Despite the fact that the MinMax criterion, or infinite norm, is normally used for form metrology applications, it is very sensitive to outliers and its implementation is not straightforward. It becomes even more complicated with aspherical shapes which are considered complex.

The evaluation of the fitting algorithms is done in conformance to ISO 10360-Part 6 [107] which gives directives for the generation of simulated data on simple shapes. The standard does not give any directives for aspherical surfaces, so we propose a framework. Then, in order to assess the fitting results, we evaluate the residual errors and quantify them by means of two commonly used values, the Root-Mean-Square (RMS) and the Peak-to-Valley (PV) [137].

In this chapter, we start by giving a thorough review of the existing fitting methods of aspherical surfaces and describe the commonly used optimization algorithms. We describe the two classically used algorithms for the fitting of aspherical surfaces, LM and ICP and then introduce our proposed L-BFGS algorithm. The evaluation and comparison of the three algorithms based on simulated datasets as well as the assessment of the L-BFGS optimization parameters are performed. After that the three algorithms are confronted to real measured data. The simulations and experimental results show the superior effectiveness of L-BFGS. There are several participating partners in the project which have all measured the same aspherical part. Therefore, finding a unified framework for the comparison of measurements is a serious issue that we address in this chapter. Finally, an extension to the *strong* Forbes model definition of asphere is presented and compared with the classical model definition in regard to the problem of fitting. Forbes models have the particularity to be constituted of independent asphere parameters, thus a study is performed to show the usefulness of such an alternate description of aspheres.

II. Methods for form evaluation of aspheres in metrology

The form evaluation of aspheres can be done by performing the association of a known aspherical model to the measured data. This process, defined in ISO 17450-Part 1 [51], is the operation used to fit an ideal feature (the model) to a non-ideal feature (the data points) according to an association criterion (such as Least-Squares or MinMax). To make things simple, we refer to the ideal feature here as being the reference model which can be expressed in different forms. Either in discrete form such as a point-set model or a mesh model, or in continuous form such as a CAD model or an implicit mathematical model. 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-triangle basis and on a point-to-point basis, respectively. Otherwise, the reference model is a mathematical equation and distances are calculated on a point projection basis. In both cases, the algorithm incorporates a footpoint calculation step followed by the distances minimization step. In discrete form, the distance is an Euclidean distance because as shown in Fig. 66a, the point-to-point distance d_{pp} is not orthogonal (or in other words minimal) and d_{pm} might also happen to be not orthogonal. Whereas in continuous form, the distance is optimized to be orthogonal to the surface at all times.

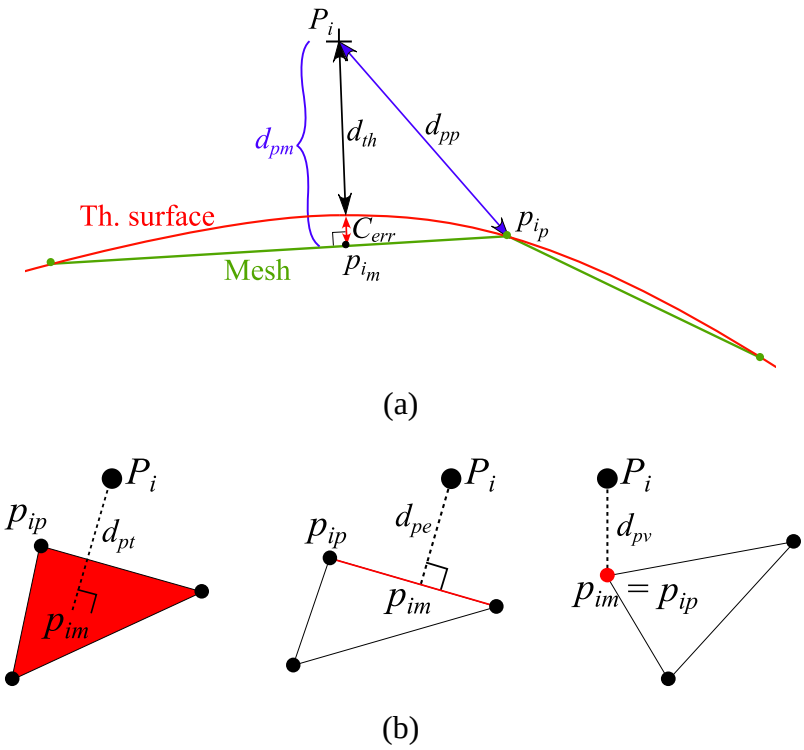


Fig. 66 Distance calculation in case of a discrete (mesh) model. (a) Difference between a point-to point distance (d_{pp}) and a point-to-mesh distance (d_{pm}); (b) Point-to-mesh distance configurations with d_{pv}: point-to-vertex distance, d_{pe}: point-to-edge distance and d_{pt}: point-to-triangle distance; (th.: theoretical, C_{err}: chord error).

The fitting process goes by iteratively optimizing for six transformation parameters consisting of both rotation and translation, $\boldsymbol{R}$ and $\boldsymbol{T}$:

$$\phi(\boldsymbol{R},\boldsymbol{T}) = \sum_{i=1}^{N} \|\boldsymbol{p_i} - (\boldsymbol{R}.\boldsymbol{P_i} + \boldsymbol{T})\|^2, \tag{12}$$

N being the number of points in the dataset, $\boldsymbol{p_i}$ the footpoints, $\boldsymbol{P_i}$ the datapoints, $\boldsymbol{T}$ the translation vector and $\boldsymbol{R}$ the combined rotation matrix about x, y and z. In the case of a discrete model fitting, the footpoints $\boldsymbol{p_i}$ might either be points in the second set $\boldsymbol{p_{i_p}}$ or the projections $\boldsymbol{p_{i_m}}$ of $\boldsymbol{P_i}$ onto a mesh model (Fig. 66a).

Many fitting algorithms exist but only a few were actually applied to aspherical surfaces since these are of complex shapes. In general, they are axis-symmetric but can also be freeform so with very large dataset sizes, optimization algorithms become computationally expensive.

Among the known families of numerical algorithms, we cite, the non-linear orthogonal Least-Squares minimization algorithms, the Iterative Closest Point (ICP) algorithm and its variants [138–140], the simplex algorithm [141] and non deterministic algorithms such as the differential evolution, the genetic algorithm, etc... In this manuscript, we focus on the algorithms that are mostly used in metrology applications for curve/surface fitting and point out the Newton Raphson method, the Gauss-Newton method, the gradient descent method, the Levenberg-Marquardt algorithm, the ICP algorithm and call attention to the newly used Limited memory- Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm.

The Newton-Raphson method [142] is used in many optimization problems that are not highly non-linear or complex (Fig. 67). So we make use of it here for the computation of footpoints which are required for the orthogonal distance minimization problem.

```
1   Init  x = x_0 , ε_1 ,  I_max
2   i = 0
3   Repeat
4     δ = S(x) / S'(x)
5     x = x - δ
6     i = i + 1
7   Until  |δ| < ε_1 or i = I_max
8   If  |δ| ≥ ε_1 then do
9     print  'Did not reach convergence'
10  end if
```

Fig. 67 Pseudo-code of the Newton-Raphson algorithm (I_{max}: maximum number of iterations and ε_1 is the stop criterion).

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 x_0 (the relative position of the data and the model should be close to the optimal solution) [142,143]. 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 to the optimal orthogonal projection point p_i^o (Fig. 68). Newton-Raphson iterates until p_i^o is accurately approximated with a stop criterion: $x_{i+1} - x_i \leq \varepsilon_1 = 10^{-16}$.

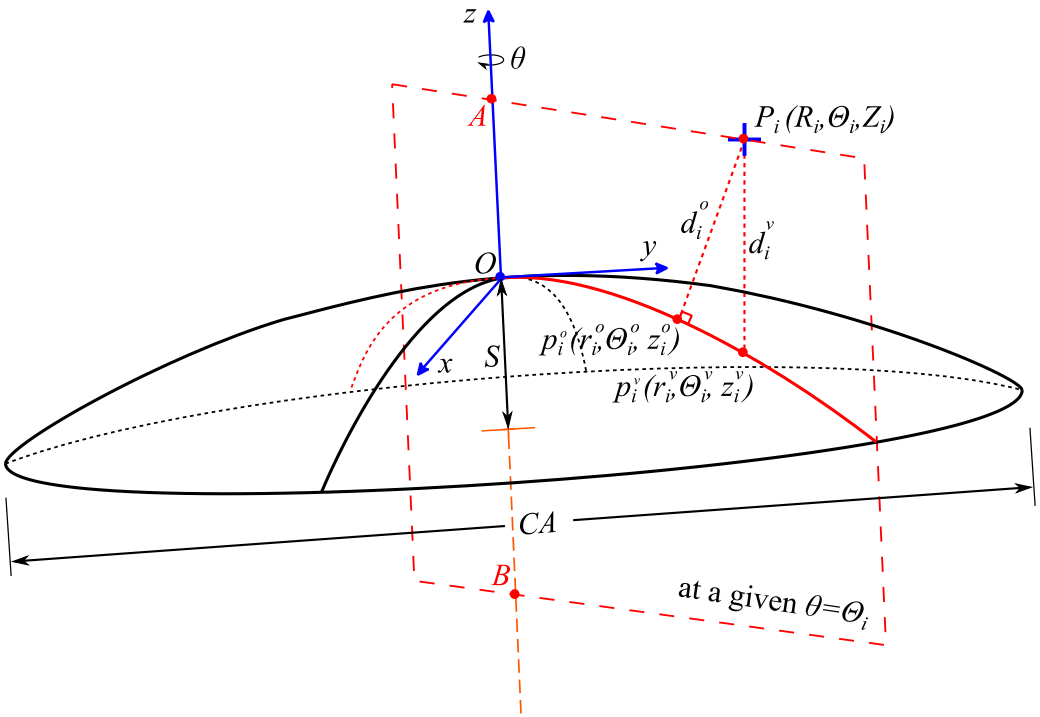


Fig. 68 Orthogonal/vertical distance vector from a data point P_i to its footpoint p_i^o/p_i^v on an asphere is contained in the rz-plane (red) at an angle Θ_i passing through the axis of symmetry (points O, A, B, p_i^o and p_i^v are coplanar).

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 LM and other Newtonian or quasi Newtonian methods.

1. Classical methods

LM is a well-known optimization algorithm that is based on an interpolation between a Gauss-Newton approach and the gradient descent as shown in (13) [144,145]. The pseudo-code for the three algorithms, Gauss-Newton, gradient descent and LM are given in Fig. 69, Fig. 70 and Fig. 71, respectively. LM 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 [133].

1 **Init** $x = x_0$, ε_0 , j_{max}
2 $j = 0$
3 **While** $\Delta\phi > \varepsilon_0$ **and** $j < j_{max}$ **do**
4 $\delta_j = \arg\min_{\delta} \| \phi(x_j) + J\phi_j \delta \|^2$
5 $x_{j+1} = x_j + \delta_j$
6 $j = j + 1$
7 **end while**
8 **return** x_j

Fig. 69 Pseudo-code of the Gauss-Newton algorithm (j_{max}: maximum number of iterations, ε_0 is the stop criterion and J is the Jacobian matrix).

1 **Init** $x = x_0$, ε_0 , j_{max}
2 $j = 0$
3 **While** $\Delta\phi > \varepsilon_0$ **and** $j < j_{max}$ **do**
4 $\delta_j = \arg\min_{\delta} \phi(x_j - \delta \nabla\phi)$
5 $x_{j+1} = x_j - \delta_j \nabla\phi$
6 $j = j + 1$
7 **end while**
8 **return** x_j

Fig. 70 Pseudo-code of the gradient descent algorithm (j_{max}: maximum number of iterations and ε_0 is the stop criterion).

1 **Init** $x = x_0$, λ , ε_3 , υ , j_{max}
2 $j = 0$
3 **While** $\Delta\phi > \varepsilon_3$ **and** $j < j_{max}$ **do**
4 Find δ_j s.t. $(H(x_j) + \lambda \mathrm{diag}[H(x_j)])\delta_j = J^T\phi_j$
5 $x_{j+1} = x_j + \delta_j$
6 **If** $\phi(x_{j+1}) < \phi(x_j)$ **then**
7 $x_j = x_{j+1}$
8 $\lambda = \lambda/\upsilon$
9 **else**
10 $\lambda = \upsilon\lambda$
11 **end if**
12 $j = j + 1$
13 **end while**
14 **return** x_j

Fig. 71 Pseudo-code of the Levenberg-Marquardt algorithm (j_{max}: maximum number of iterations).

The solution update for the LM algorithm goes as in the following equation:

$$x_{j+1} = x_j - \left(\mathbf{H}(x_j) + \lambda_j . diag[\mathbf{H}(x_j)]\right)^{-1} . \nabla\phi(x_j), \quad (13)$$

where x_j is the solution vector at iteration j, $\mathbf{H}(x_j)$ is the Hessian matrix at iteration j, $diag[\mathbf{H}(x_j)]$ is the diagonal elements of the Hessian and $\nabla\phi$ is the gradient of the objective function. λ is the parameter that can be assimilated to the coefficient of the gradient descent. λ changes at each iteration such as when it is large, the gradient descent predominates the optimization process, otherwise Gauss-Newton predominates. According to Shakarji [133], the Levenberg-Marquardt parameter λ is suggested to start with the value of 0.0001. This value is updated at each iteration according to the objective function's evolution. If ϕ turns out to be decreasing, it means that the solution is approaching the minimum and that more effect has to be given to the Gauss-Newton component. In this case, λ is reduced by a factor of 0.04, otherwise, λ is increased by a factor of 10. In an application to freeform surfaces, Jiang *et al* [146] propose another way of determining the value of λ based on the smallest singular value of the Hessian matrix. Since the Hessian is positive semi-definite, a decomposition of the form given by (14) is equivalent to a Singular Value Decomposition (SVD) according to the spectral theorem [147].

$$\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathbf{T}} \quad (14)$$

with $\mathbf{D} = diag\{\sigma_1, \sigma_2, \dots, \sigma_n\}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ being the singular values of $\mathbf{H}$.
By these means, λ is chosen in accordance with the smallest singular value σ_n of $\mathbf{H}$ found in the $\mathbf{D}$ matrix. If $\sigma_n > \eta$, where η is a user specified threshold, then $\lambda = 0$, otherwise, $\lambda = \eta - \sigma_n$. Generally, this algorithm converges reasonably quickly and accurately for a wide range of initial guesses that are close to the optimal solution [145].
In [146] 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 needs to be calculated. The fitting of B-Spline curves using the LM algorithm also requires the calculation of a large Jacobian matrix $\mathbf{J}$ and the storage of a considerable system of linear equations, as described by Speer *et al* [148]. The LM algorithm further necessitates the computation of the Hessian matrix by linear approximation $\mathbf{H} = \mathbf{J}^{\mathbf{T}}\mathbf{J}$ 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 ranges between $O(n^{2.37})$ (Williams algorithm) and $O(n^3)$ (Gauss-Jordan elimination) with n being the number of variables in the system [149]. For a large n, the cost of inverting **H** is considerable.

2. The ICP method

The ICP is a registration algorithm which finds a spatial transformation to align two point-sets. It is the mostly used algorithm in registration and/or alignment because it is designed for discrete data, making it a relatively fast algorithm with very low memory storage. ICP is based on two main operations, point identification and point matching which are usually computationally expensive. In this context, some ICP variants have been developed to reduce the point matching phase time by applying a k-D tree subdivision to the space of points [150,151]. The general ICP framework is such that an iterative loop identifies pairs of points and matches them according to a distance criterion (Euclidean, Hausdorff, …). The matching phase triggers a Least-Squares distance minimization which results in a transformation matrix that brings one point-set closer to the other with residual errors quantified by the mean square error $\bar{\phi}$ – also called the error metric and shown in Fig. 72 – with ϕ being the objective function such as defined in (12). All points are involved in the point matching phase but only a subset of those is selected for the computation of the transformation matrices, $\boldsymbol{R}$ and T. Different approaches to calculating $\boldsymbol{R}$ and T are available: the Unit Quaternion method [140], the Dual Quaternions method [152], the Singular Value Decomposition (SVD) method [153], the LM method [154], the Neural Network Modeling method [155], etc …

```
1  Init (R_0, T_0), ε_3 , τ , n , j_max
2  j = 0
3  While φ̄ > τ and j < j_max do
4    While Δφ̄ > ε_3 do
5      point matching w.r.t closest point, (n)
6      calculate and apply R_j + T_j
7    end while
8      n = n + Δn
9      j = j + 1
10 end while
```

Fig. 72 Pseudo-code of the ICP algorithm ($\bar{\phi}$: mean square error (average of the objective function), j_{max}: maximum number of iterations, ε_3: stop criterion, τ: mean square error tolerance).

We can see from Fig. 72 that if the variation of mean square error $\Delta\bar{\phi}$ is larger than the threshold value ε_3, point identification and matching restarts until the error cannot be improved anymore ($\Delta\bar{\phi} \leq \varepsilon_3$). Further, if the mean square error is still larger than a predefined tolerance τ, the entire process restarts with different parameters (n) in point identification and matching. To have fine precision on the results, it is preferable that the size of both point-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 replaced by a mesh representation of it. A mesh model offers the advantage of obtaining a more accurate distance calculation than a point model does [156]. Fig. 66a shows that the point-to-mesh distance d_{pm} is smaller than the point-to-point distance d_{pp} and that there are three possible configurations for a point-to-mesh distance (Fig. 66b). A mesh is a first order approximation of the surface so it is a better representation of the theoretical surface than points only. It is a regular triangular mesh built from theoretical points simulated on the asphere's surface and reconstructed using a Delaunay triangulation technique [157]. To guarantee accurate residual errors, equivalent to those found when fitting smooth models, 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. 66a). 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.

III. L-BFGS, a newly proposed method in metrology

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 [158]. Like any minimization algorithm, BFGS preferably requires a twice differentiable objective function 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 [159,160]. Subsequently, Nocedal describes an improved method called L-BFGS which keeps updating the Hessian matrix using a limited amount of storage [161]. 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* [159] 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 location parameters simultaneously if the initial curve is sufficiently close to the underlying shape of the dataset. Additionally, neither formulating nor solving linear equations is needed, making the algorithm very efficient and faster than other methods such as the one in [148]. Zheng *et al* [159] 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 follows: consider the problem in which the objective function ϕ has gradient $\nabla\phi$ and Hessian **H**.

a) Initialization

The initialization consists of four 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 $\mathbf{H}_0^{-1}$ such as the identity matrix.

b) Iterations

Perform a *line search* routine to compute the quasi-Newton direction. By being positive definite, the Hessian ensures that d_j is a descent direction (15).

$$d_j = -\mathbf{H}_{j-1}^{-1}\,\nabla\phi\left(x_{j-1}\right). \tag{15}$$

Determine the *step size* α (for example by backtracking line search). α_j should satisfy the Wolfe conditions (16) at each iteration. β' and β are parameters to control the accuracy of the line search routine. β' defines a tolerance on the function ϕ and β a tolerance on its gradient.

$$\begin{aligned} \phi\left(x_j + \alpha_j . d_j\right) \;&\leq\; \phi\left(x_j\right) + \beta^{'}\,\alpha_j . \nabla\phi\left(x_j\right)^T . d_j\,, \\ \nabla\phi\left(x_j + \alpha_j . d_j\right)^T d_j \;&\geq\; \beta\,\nabla\phi\left(x_j\right)^T . d_j\,. \end{aligned} \tag{16}$$

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 ϕ and making a reasonable progress in minimization. Then the update rule for the solution x is done as (17) indicates.

$$x_{j+1} = x_j + \alpha_j d_j. \tag{17}$$

c) Update $\mathbf{H}_j^{-1}$

The inverse Hessian at step $j+1$ is the inverse Hessian at step j plus a certain variation Δh:

$$\mathbf{H}_{j+1}^{-1} = \mathbf{H}_j^{-1} + \Delta h\left(\Delta x_j, \Delta(\nabla\phi)_j, \mathbf{H}_j^{-1}\right). \tag{18}$$

Δh depends on the change in the variables, Δx_j, as well as in the change of the gradient, $\Delta(\nabla\phi)_j$ expressed as:

$$\Delta x_j = X_j = x_{j+1} - x_j \quad \text{and} \quad \Delta(\nabla\phi)_j = G_j = (\nabla\phi)_{j+1} - (\nabla\phi)_j. \tag{19}$$

Therefore the inverse Hessian becomes:

$$\mathbf{H}_{j+1}^{-1} = \left(I - \frac{X_j G_j^T}{G_j^T X_j}\right) \mathbf{H}_j^{-1} \left(I - \frac{G_j \left(X_j\right)^T}{G_j^T X_j}\right) + \frac{X_j \left(X_j\right)^T}{G_j^T X_j}. \tag{20}$$

Due to the iterative process, an inverse Hessian matrix at an iteration m can be written in function of $\mathbf{H}_0^{-1}$ and the vectors Δx_j and $\Delta(\nabla\phi)_j$ of all iterations up to m only. This automatically means that the Hessian matrices shall not be stored and this reduces memory storage by much. The update goes as follows:

$$\mathbf{H}_1^{-1} = \mathbf{H}_0^{-1} + \Delta h(\Delta x_0, \Delta(\nabla\phi)_0, \mathbf{H}_0^{-1})$$

$$\mathbf{H}_2^{-1} = \mathbf{H}_0^{-1} + \Delta h(\Delta x_0, \Delta(\nabla\phi)_0, \mathbf{H}_0^{-1}) + \Delta h(\Delta x_1, \Delta(\nabla\phi)_1, \mathbf{H}_1^{-1})$$

$$\vdots \qquad (21)$$

$$\mathbf{H}_m^{-1} = \mathbf{H}_0^{-1} + \Delta h(\Delta x_0, \Delta(\nabla\phi)_0, \mathbf{H}_0^{-1}) + \Delta h(\Delta x_1, \Delta(\nabla\phi)_1, \mathbf{H}_1^{-1}) + \cdots + \Delta h(\Delta x_{m-1}, \Delta(\nabla\phi)_{m-1}, \mathbf{H}_{m-1}^{-1})$$

At iteration $m+1$, in order to limit storage in L-BFGS, the first term $\Delta h(\Delta x_0, \Delta(\nabla\phi)_0, \mathbf{H}_0^{-1})$ is deleted. Nonetheless, $\Delta h(\Delta x_1, \Delta(\nabla\phi)_1, \mathbf{H}_1^{-1})$, $\Delta h(\Delta x_1, \Delta(\nabla\phi)_1, \mathbf{H}_2^{-1})$, … , $\Delta h(\Delta x_m, \Delta(\nabla\phi)_m, \mathbf{H}_m^{-1})$ all depend on $\Delta h(\Delta x_0, \Delta(\nabla\phi)_0, \mathbf{H}_0^{-1})$, hence, only the terms that are enough to calculate $\Delta h\left(\Delta x_j, \Delta(\nabla\phi)_j, \mathbf{H}_j^{-1}\right)$ for $0 < j \leq m$ are kept.

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

IV. Implementation of the algorithms for aspherical fitting

The previously described optimization algorithms are going to be tested for aspherical surface fitting, knowing that we are putting forward the L-BFGS algorithm for this purpose [132]. The algorithms, are evaluated in conformance with standards defined in ISO 10360-Part 6 [107] and the works provided by Lin *et al* [162]. We propose a procedure inspired from the ISO standards in order to evaluate L-BFGS and LM and conclude on their robustness to input dataset size, initial alignment and computational time.

The Orthogonal non-linear Least-Squares algorithms as well as ICP follow the same structure and sequentially compute for footpoints and transformation parameters ($\boldsymbol{R}$: rotation matrix, $\boldsymbol{T}$: translation vector) as shown in Fig. 73. After being evaluated on simulated datasets, the algorithms are tested on measured surfaces with large volumes of data and with different initial relative positions of the data with respect to the reference model (initial rough alignment). The fitting process, illustrated in Fig. 73, is the process in which the model parameters (c, κ and $\boldsymbol{a}$) are fixed and the only variables to optimize are the transformation parameters often called motion parameters so that comparison with ICP is possible. The AO775 aspherical lens is axis-symmetric, therefore, the rotation about z is a redundant degree

of freedom and is inhibited from the minimization except for ICP. Hence, only 5 transformation parameters out of 6 are required to determine a good fitting of the dataset with respect to the model: a rotation matrix $\boldsymbol{R_\beta}$ about the x-axis, a rotation matrix $\boldsymbol{R_\gamma}$ about the y-axis and three translations t_x, t_y and t_z in x-, y- and z- directions, respectively.

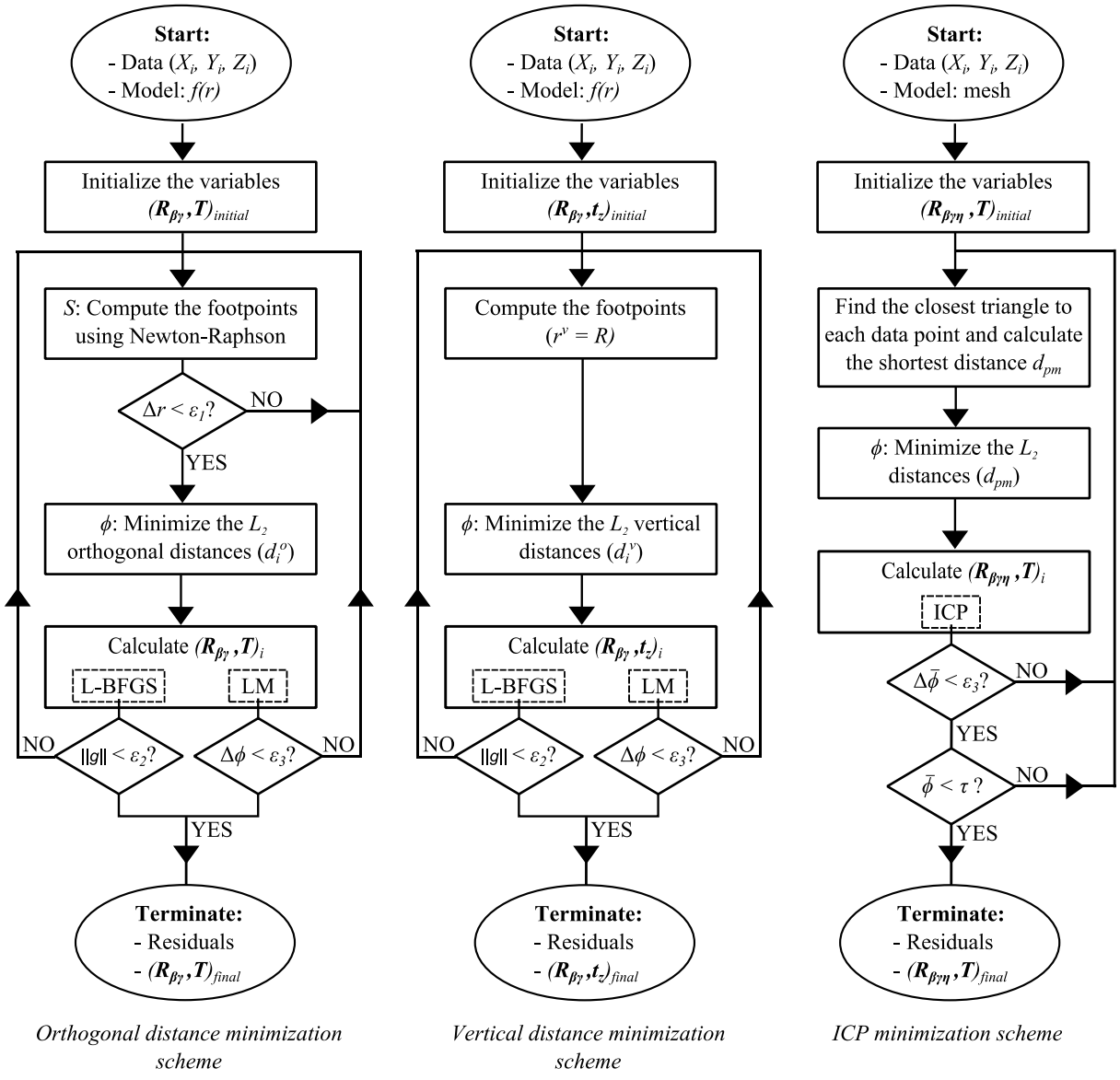


Fig. 73 Sequential algorithm for the fitting problem. L_2: Least-Squares norm; $\boldsymbol{R_{\beta\gamma}}$ or $\boldsymbol{R_{\beta\gamma\eta}}$: combined rotation matrix and $\boldsymbol{T}$: translation vector; $\Delta\phi$: change in the objective function value; $\|g\|$: norm of the gradient of ϕ; $\varepsilon_1, \varepsilon_2, \varepsilon_3$: the algorithms stop criteria; τ: ICP convergence tolerance.

The objective function ϕ to minimize is the sum of the squared distances between the transformed data points $\boldsymbol{P_i}(X_i, Y_i, Z_i)$ and their respective footpoints $\boldsymbol{p_i}(x_i, y_i, z_i)$ on the

model. Equation (22) is an expression of this functional in which $\boldsymbol{T} = (t_x, t_y, t_z)$ denotes the translation vector and β and γ the rotation angles about x and y, respectively:

$$\Phi(\beta, \gamma, \boldsymbol{T}) = \min_{\beta,\gamma,t_x,t_y,t_z} \sum_{i=1}^{N} \|\boldsymbol{p_i} - (\boldsymbol{R_{\beta,\gamma}}.\boldsymbol{P_i} + \boldsymbol{T})\|^2, \quad (22)$$

where $\boldsymbol{R_{\beta,\gamma}} = \boldsymbol{R_\beta R_\gamma}$ is the combined rotation matrix about x and y.

L-BFGS, LM and ICP optimization algorithms are compared based on the RMS and PV of residual errors since those quantities are commonly used in fitting applications [137]. Orthogonal distance-based fitting and vertical distance-based fitting are also compared because, as described by Sun et al [128], the aspherical surface fitting onto simulated small-sized datasets can be achieved by applying a Gauss-Newton optimization with vertical distance minimization. We will show that the usage of vertical distance is limited to cases where both entities (dataset and model) are defined in the same reference frame, whereas orthogonal distance calculation guarantees the freedom to have the measured dataset and the model in different coordinate systems [136].

1. Implementation of L-BFGS and LM

The L-BFGS and LM algorithms are implemented in sequential computation of footpoints and transformation parameters. Vertical footpoints are simply the projection of each data point onto the surface along z and by that we do not necessarily assume that the z-axes of both sets are in the vertical direction. We will only make this assumption when we perform an optimization involving vertical distances minimization. When we deal with orthogonal distances minimization, orthogonal footpoints need to be computed. This is done by starting with the vertical footpoint projections and then solving for orthogonal projections in a nested minimization loop within the main transformation parameters minimization loop using the Newton-Raphson method (Fig. 74).

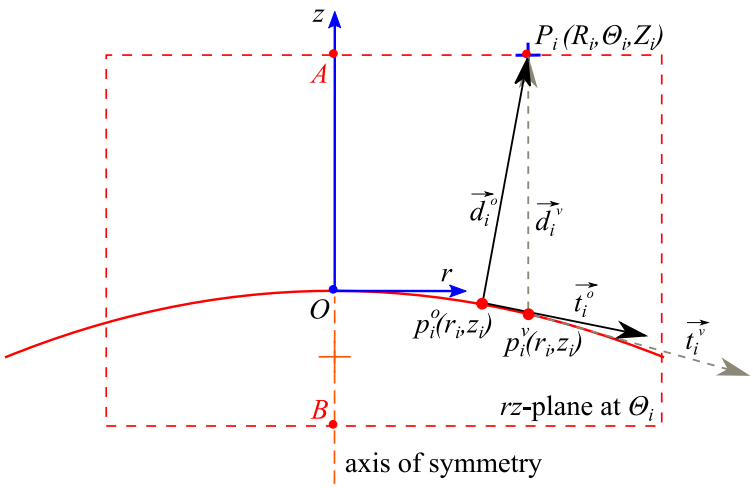


Fig. 74 Orthogonal/vertical distance vector from a data point P_i to its footpoint p_i^o/p_i^v on an asphere is contained in the rz-plane at an angle Θ_i passing through the axis of symmetry (red points O, A, B, P_i, p_i^o and p_i^v are coplanar): planar view in the rz-plane.

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. 74). Usually, the projection of a point $\boldsymbol{P_i}$ on a surface requires the optimization of two parameters, here r_i and θ_i. But, since the model of the asphere (2) 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 $\boldsymbol{P_i^{Car}}(X_i, Y_i, Z_i)$ is written in cylindrical coordinates $\boldsymbol{P_i^{Cyl}}(R_i, \Theta_i, Z_i)$ (23). Then, as shown in Fig. 74, the footpoint $\boldsymbol{p_i^v}$ (vertical) or $\boldsymbol{p_i^o}$ (orthogonal) is computed in the selected rz-plane at $\theta_i = \Theta_i$.

$$\boldsymbol{P_i^{Cyl}} = \begin{vmatrix} R_i = \sqrt{X_i^2 + Y_i^2} \\ \Theta_i = \tan^{-1}\left({}^{Y_i}/_{X_i}\right) \\ Z_i = Z_i \end{vmatrix} \tag{23}$$

For the vertical distance minimization problem, the r-coordinate of the vertical footpoint $\boldsymbol{p_i^v}$ (Fig. 74) 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 equation (24):

$$z_i = f(r_i) = -\left(\frac{c r_i^2}{1 + \sqrt{1 - (1+\kappa) c^2 r_i^2}} + \sum_{j=1}^{5} a_{2j} r_i^{2j} \right) \tag{24}$$

For the orthogonal distance projection problem, the *r*-coordinate of the footpoint is initially taken to be that of the data point, $r_{i,ini}^{o} = r_i^{v} = R_i$. Then, the Newton-Raphson method iteratively optimizes for r_i^{o}, and $z_i^{o} = f(r_i^{o})$ is calculated following (24). 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 $S_i = \vec{d}_i . \vec{t}_i$ to 0. In this formulation, $\vec{d}_i$ is the distance vector from $\boldsymbol{P_i^{Cyl}}$ to $\boldsymbol{p_i}(r_i, z_i)$ in the corresponding θ_i-oriented rz-plane (Fig. 74) and $\vec{t}_i$ is the tangent vector at $\boldsymbol{p_i}$ and is expressed as in (25), by differentiating the implicit function F of the surface at $\boldsymbol{p_i}$.

$$\vec{t}_i = \begin{pmatrix} \left. \dfrac{\partial F(r,z)}{\partial z} \right|_{r=r_i, z=f(r_i)} \\ \left. \dfrac{-\partial F(r,z)}{\partial r} \right|_{r=r_i, z=f(r_i)} \end{pmatrix} \tag{25}$$

The initial distance vector input to (26) is the vertical distance vector $\vec{d}_i^{\,v} = (r_i^{v} - R_i\,, z_i^{v} - Z_i)$. The orthogonal footpoints function S_i is iteratively solved until the dot product between $\vec{d}_i$ and $\vec{t}_i$ falls below a certain predefined threshold $\varepsilon_1 = 10^{-16}$.

$$S_i = (r_i - R_i) \frac{\partial F(r_i, z_i)}{\partial z} + (z_i - Z_i) \frac{-\partial F(r_i, z_i)}{\partial r} = 0, \tag{26}$$

The derivative S_i' is needed in order to update the Newton-Raphson solution given in (27) at each iteration k:

$$r_i^{(k+1)} = r_i^{(k)} - \frac{S_i\left(r_i^{(k)}\right)}{S_i'\left(r_i^{(k)}\right)}. \tag{27}$$

The orthogonal footpoint calculated in cylindrical coordinates is then transformed back to 3D Cartesian coordinates following (28).

$$\boldsymbol{p_i^o} = \begin{vmatrix} r_i^o . \cos\ \Theta_i \\ r_i^o . \sin\ \Theta_i \\ f(r_i^o) \end{vmatrix} \tag{28}$$

With the coordinates of each footpoint, vertical $\boldsymbol{p_i^v}(x_i^v, y_i^v, z_i^v)$ or orthogonal $\boldsymbol{p_i^o}(x_i^o, y_i^o, z_i^o)$, 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 and measures the relative error desired in the sum of squared distances (Fig. 73).

2. Implementation of a variant of ICP

With the proposed discrete fitting approach on a mesh model, the objective function to minimize in the case of the ICP is the sum of the squared distances d_{pm} (29). d_{pm} is the shortest distance separating a data point P_i from its footpoint p_{im} situated on the closest triangle in the mesh, i.e., $d_{pm} = \min(d_{pv}, d_{pe}, d_{pt})$ with respect to the closest triangle (Fig. 75). Normally, it is not possible to claim rotational symmetry for point-sets therefore in the case of ICP, the variables cannot disregard the angle about z.

$$\Phi(\beta, \gamma, \eta, \boldsymbol{T})_{ICP} = \min_{\beta,\gamma,\eta,t_x,t_y,t_z} \sum_{i=1}^{N} \left\| \boldsymbol{p_{im}} - \left(\boldsymbol{R_{\beta,\gamma,\eta}} . \boldsymbol{P_i} + \boldsymbol{T} \right) \right\|^2 , \tag{29}$$

with $\boldsymbol{p_{im}} \in \{\boldsymbol{p_{i_v}}, \boldsymbol{p_{i_e}}, \boldsymbol{p_{i_t}}\}$.

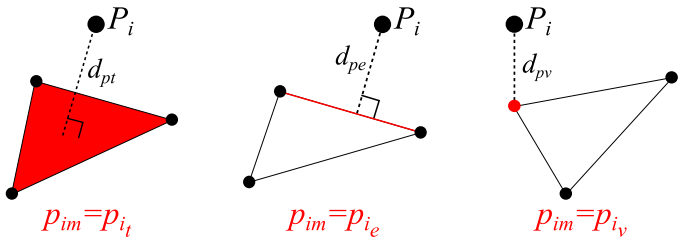


Fig. 75 Distance calculation in the case of ICP and choice of projection point according to minimal point-to-triangle distance.

V. Evaluation of the fitting algorithms on simulated data

In this project, we deal with a 10^{th}-order aspherical surface that is not exactly expressed as in ISO standard 10110 - Part 12 [15] due to the presence of the a_2 term (here j starts at 1) which was originally mingled with the curvature parameter c in the ISO (30).

$$f(r) = -\left(\frac{cr^2}{1+\sqrt{1-(1+\kappa)c^2r^2}} + \sum_{j=1}^{5} a_{2j} r^{2j} \right). \tag{30}$$

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 lens model AO775, the curvature at the apex is positive and equal to $c = 10^{-20}$ mm^{-1}, the conic constant $\kappa = -1$ and the asphere parameters are $a_2 = 0.0223, a_4 = 7.293 \times 10^{-6}, a_6 = 4.52 \times 10^{-9}, a_8 = -1.061 \times 10^{-11}, a_{10} = 9.887 \times 10^{-15}$. The manufacturer of AO775, Anteryon® states that the asphere has a sag $S = 3.217$ mm of and a clear aperture $CA = 11.74$ mm (Fig. 20).

For the evaluation of the optimization algorithms, the aspherical model is simulated based on equation (30) by generating points around the asphere's axis on a square grid (Fig. 76). This is because the Matlab function ("*surf*") used to plot the simulated data in 3D requires having a z matrix over a square XY grid. Note, however, that this issue only concerns the graphical representation and does not apply to the process of fitting which is based on taking points within the clear aperture of the lens. To evaluate the robustness and correctness of the developed fitting algorithms in conformance with ISO 10360: Part 6 [1], we propose a new procedure. Two simulations each containing 10^7 points are performed: one with added orthogonal random errors to characterize surface roughness and another with orthogonal 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, whereas measurement random errors manifest in the vertical direction.

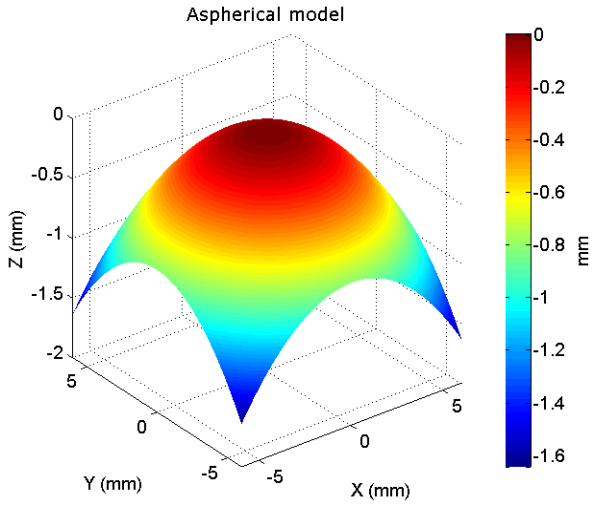
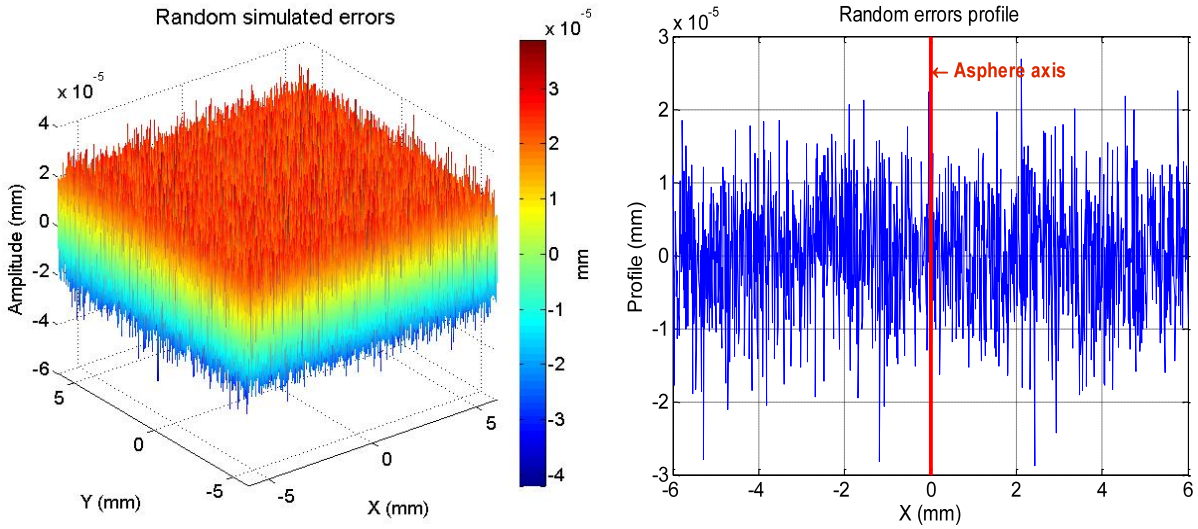


Fig. 76 Simulated asphere model AO775.

1. Random errors simulated data

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 [163]. A Matlab function ("*randn*") is used to generate the noise (Fig. 77) which is added to the theoretical data of Fig. 76 in the orthogonal direction at each data point.



(a) (b)

Fig. 77 Simulated Gaussian noise ($\mu = 0$, $\sigma = 8$ nm). (a) 3D plot and (b) 2D distribution at the middle section in the xz-plane.

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 theoretically 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 [25]. The machine used for the tests is an Intel core i7/x64 platform with 8 Gb of RAM and a 2.0 GHz processor.

The obtained residual PV values, reported in Table 3, 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 7 repeated tests which prove that the algorithm is repeatable.

Gen. err. (nm)		$L-BFGS\ (nm)$			$LM\ (nm)$			$ICP\ (nm)$		
RMS	*PV*	*RMS*	*PV*	*ΔPV* (%)	*RMS*	*PV*	*ΔPV* (%)	*RMS*	*PV*	*ΔPV* (%)
8.003	77.325	8.003	77.334	0.0117	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
7.997	82.265	7.997	82.196	−0.084	7.997	82.203	−0.075	7.999	82.301	0.044
8.002	83.252	8.002	83.299	0.056	8.002	83.301	0.059	8.003	83.387	0.162
8.003	85.256	8.003	85.241	−0.018	8.003	85.230	−0.030	8.004	85.276	0.023
8.008	88.871	8.008	88.888	0.019	8.008	88.889	0.020	8.009	88.379	−0.554

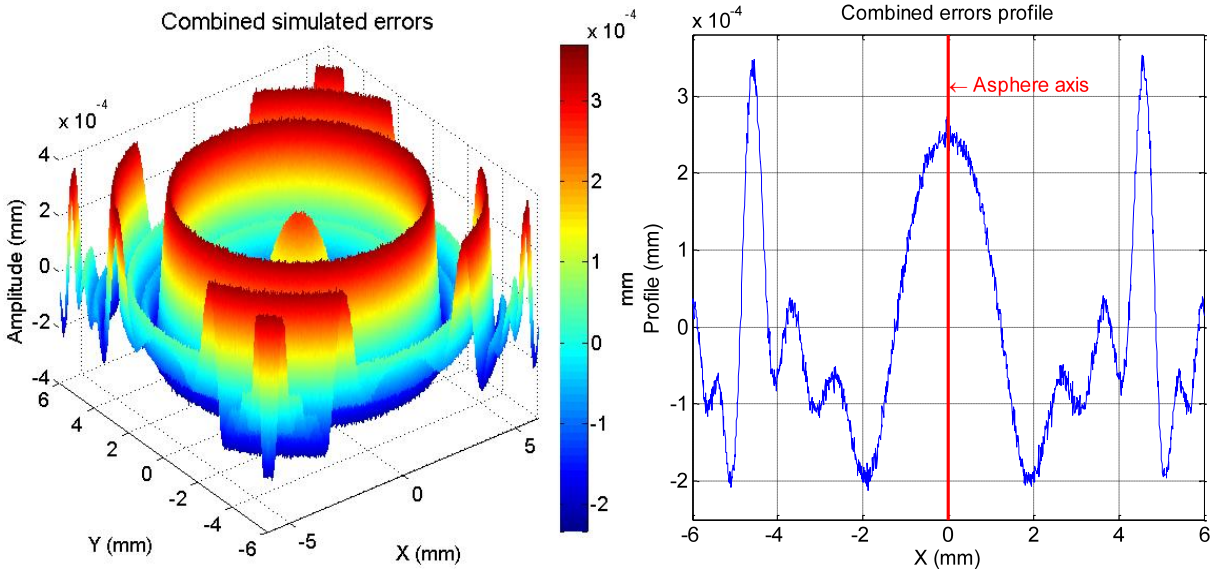
Table 3 Fitting of 7 random error datasets with Gaussian noise of 0 mean and 8.0 standard deviation; 1,500,000 points are used. (Gen. err.: Generated errors).

2. Combined random and systematic errors simulated data

The test involves superposing systematic errors onto the previous random errors in the normal direction by projecting the error value onto the normal vector to the surface. Systematic errors F_h are referred to as form deviations and are generated using Fourier harmonics according to (31) and ISO10360-Part 6 standard for simple shapes. Fig. 78 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 guiding systems [2,163]. The generated PV value of the Fourier harmonics ($\approx$ 609 nm) corresponds to a realistic form error on aspherical lenses.

$$F_h(\tau) = \sum_{k=1}^{n} A_k \sin(2\pi k w_0 Z) + B_k \cos(2\pi k w_0 Z), \tag{31}$$

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 simulation is obtained using the following parameter values: $w_0 = 2.0$ Hz, $n = 3$, $A_{1,2,3} = \{0.0001, 0.00007, 0.000075\}$ and $B_{1,2,3} = \{0.0001, 0.000092, 0.000055\}$.



(a) (b)

Fig. 78 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.

The systematic errors are taken as axis-symmetric (Fig. 78) because one of the major manufacturing processes of aspheres is a turning process that combines a rotation of the substrate and a functional motion of the tool [164]. They occur in the orthogonal direction to the surface and that is why an orthogonal distance fitting is applied. For this simulation, 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. Fig. 79 is an example showing how sampling one every 3 points is done.

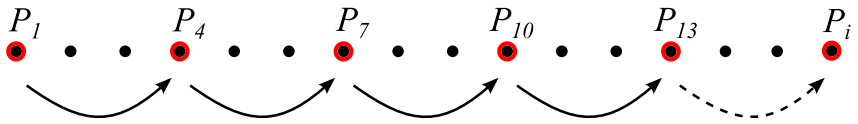


Fig. 79 Sampling strategy: reading one every 3 points.

All three fitting methods return the same RMS and PV value as the simulated ones (Table 4). The RMS and PV values remain quasi-unchanged whichever the number of points N in the dataset is as long as it is not too small (< 1000 points). When the number of considered points is below 100, a RMS variation of about 2 nm is observed.

N	$Gen.err.(nm)$		$L-BFGS\ (nm)$		$LM\ (nm)$		$ICP\ (nm)$	
	RMS	PV	RMS	PV	RMS	PV	RMS	PV
100	136.482	609.593	137.437	642.155	137.436	642.142	138.621	642.875
1,000			136.490	616.270	136.490	616.204	137.246	616.894
10,000			136.456	609.031	136.456	609.022	137.250	609.623
100,000			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

Table 4 Fitting using Least-Squares orthogonal distance minimization for the combined systematic and random errors dataset. N is the number of generated/simulated points. (Gen. err.: Generated errors).

In order to further examine the correctness of the algorithms, the estimated motion parameters resulting from the fitting of the simulated dataset with combined errors are reported in Table 5. We impose two theoretical initial alignments and check whether the optimizations return the same values. The first initial alignment IA_1 is generated with an offset of 1 mm in all three directions x, y and z with respect to the model while keeping rotations at zero and IA_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 ($< 0.00035\%$ variation for the angles and $< 0.00277\%$ variation for the translations) good estimates. ICP fails to return estimations for the case of IA_2 as the algorithm diverges when the angle is greater than 15 °. It returns inaccurate transformation parameters in the case of IA_1 with more than 45% variation in the translation parameters.

		Gen. alignment	*L-BFGS*	*LM*	*ICP*
IA_1	β	0 °	0.00000094 °	0.0000056 °	0.0000046 °
	γ	0 °	0.000033 °	0.000097 °	0.000029 °
	η	0 °	×	×	0.00402 °
	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
IA_2	β	30 °	−30.000034 °	−30.000082 °	×
	γ	30 °	−30.000066 °	−30.000106 °	
	η	0 °	×	×	
	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	

Table 5 Fitting using orthogonal distance minimization for the simulated combined errors dataset with two theoretical initial alignments.

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. In Table 6 we show 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. Recall that for the vertical distance case, the input datasets are perfectly aligned in x and y directions with respect to the model.

N	*Gen. RMS err.* (nm)	*Orthogonal* (nm)		*Vertical* (nm)	
		$L-BFGS$	LM	$L-BFGS$	LM
1,000	136.482	136.490	136.490	139.991	139.964
10,000		136.456	136.456	139.832	139.832
100,000		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

Table 6 Comparison of vertical and orthogonal distance minimization for different dataset sizes and for combined simulated errors: reporting RMS values. N is the number of generated/simulated points. (Gen. RMS err.: Generated RMS error).

The results show that orthogonal distance minimization is more accurate than vertical distance minimization. This is clearly due to the fact that noise was added to the data in the orthogonal

direction. It is to note, however, that vertical distance minimization fitting time is extremely fast (< 3 seconds for 10^6 points). If it were legitimate, we could have considered both the noise generated by the probing system and the systematic errors generated by the machine motions to be in the vertical direction and the optimization would have been extremely fast.

Another simulation is 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 and orthogonal distance minimization are performed and compared residual errors are compared (Table 7). It is shown that, in this particular case where errors manifest in the vertical direction, 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. Nonetheless, the combination of vertical and orthogonal effects in the same optimization problem is the most exact solution, and this topic should be addressed in the follow-up project EMPIR.

Algorithm	Gen. err. (nm)		*Orthogonal* (nm)		*Vertical* (nm)		Δ (%)	
	RMS	PV	RMS	PV	RMS	PV	ΔRMS	ΔPV
$L-BFGS$	3.996	39.134	3.902	38.886	3.996	39.148	-2.35	-0.63
LM	3.996	39.134	3.902	38.901	3.996	39.163	-2.35	-0.60
ICP	3.996	39.134	3.903	39.361	3.997	39.964	-2.33	0.58

Table 7 Comparison of vertical and orthogonal distance minimization for a simulated random errors dataset in the vertical direction using L-BFGS, LM and ICP.

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. 80 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.

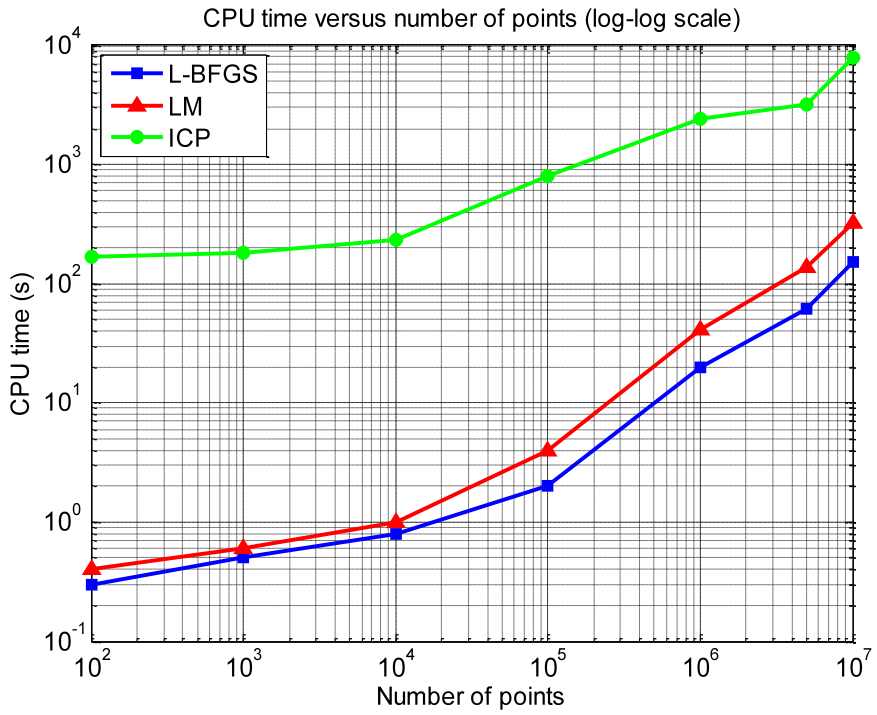


Fig. 80 Time performances in seconds of the algorithms versus the number of points for a simulated dataset with added Gaussian noise of 0 mean and 8.0 standard deviation.

Regarding memory storage, all algorithms use less than 1 Gb in general which mainly consists of the size of the data. 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 grows, 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. 81) and confirms that the time complexity of the L-BFGS and LM algorithms is linear in the number of data points.

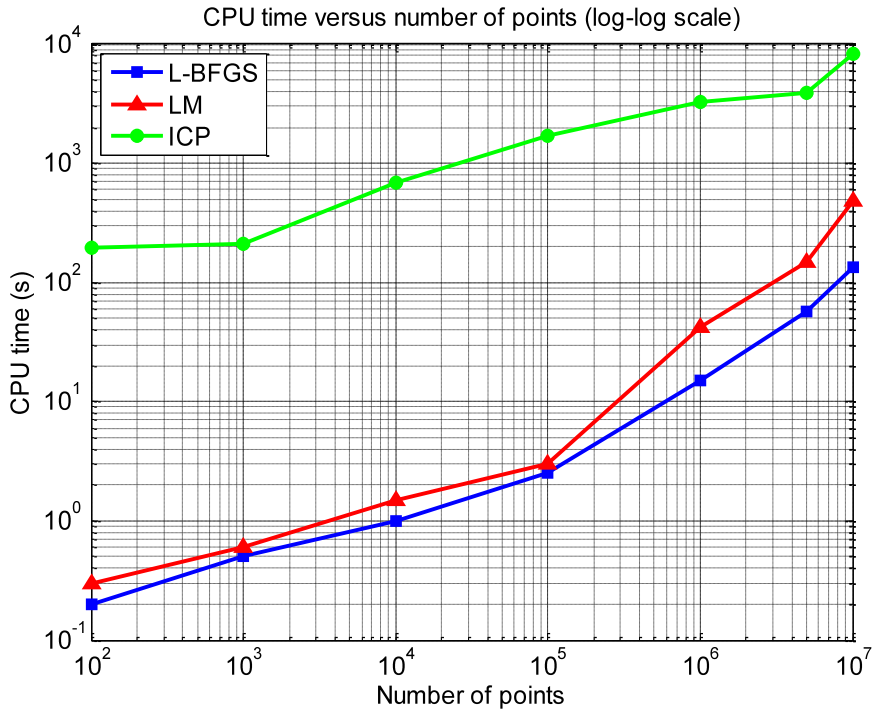


Fig. 81 Time performances in seconds of the algorithms versus the number of points for a simulated dataset with added random and systematic errors.

5. Sensitivity of L-BFGS intrinsic parameters

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 parameters because a similar analysis has been done for LM parameters [133].

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. 82. The time complexity of L-BFGS is not affected by the value of β' as only a fluctuation of 2 seconds is observed. Fig. 83 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.

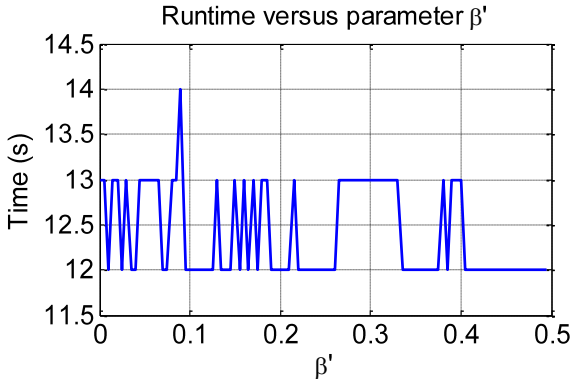
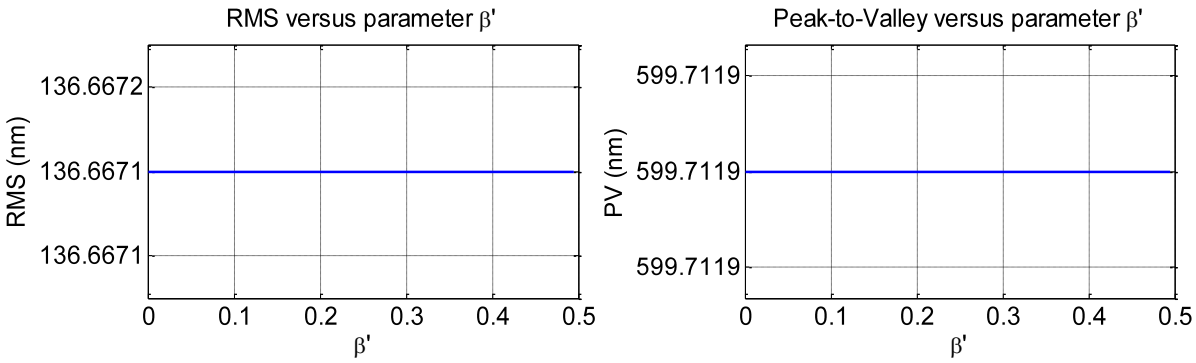


Fig. 82 The influence of β' on L-BFGS computational time for the simulated combined errors dataset.



(a) (b)

Fig. 83 The influence of β' on the residual errors for the simulated combined errors dataset. (a) RMS of residual errors; (b) PV of residual errors.

Fig. 84 and Fig. 85 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 %.

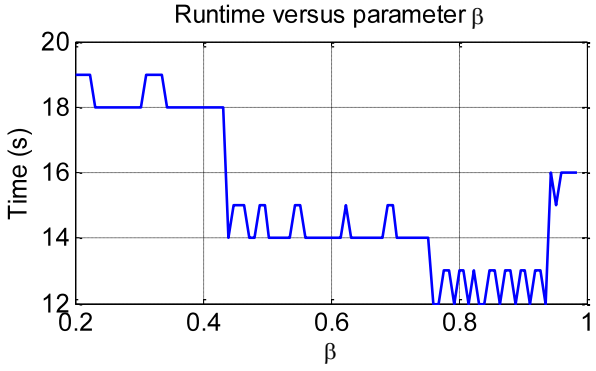
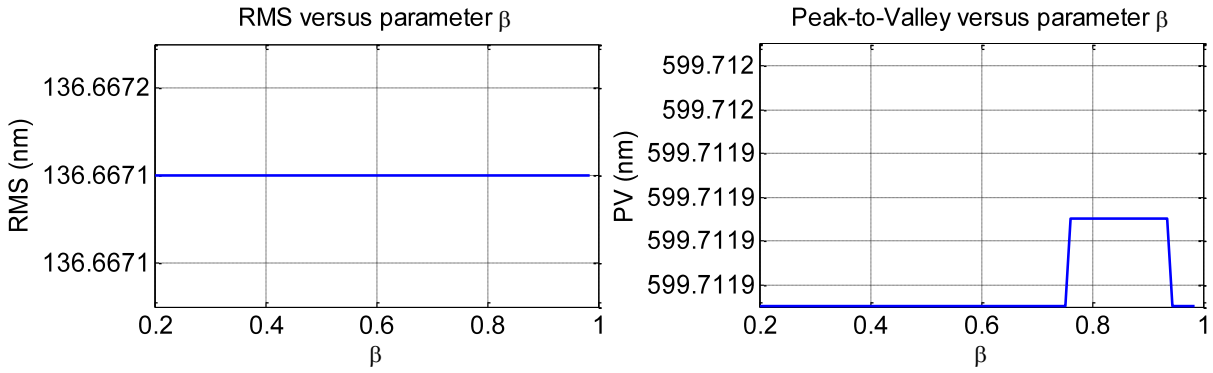


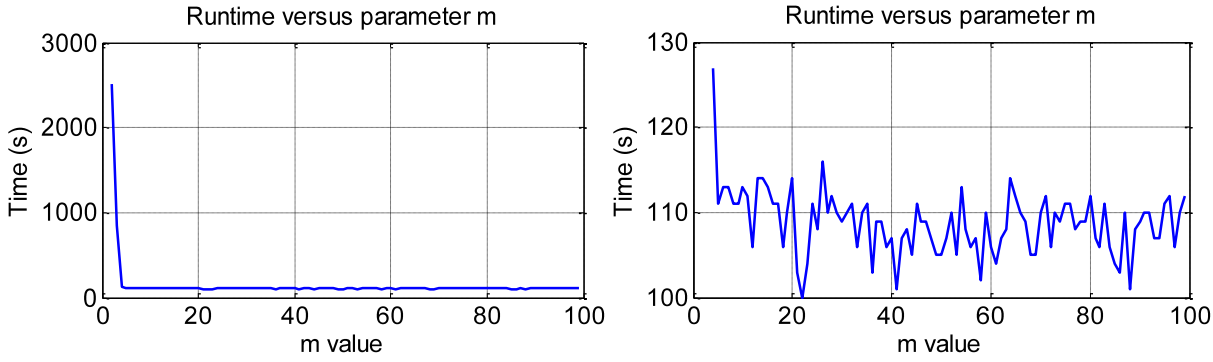
Fig. 84 The influence of β on L-BFGS computational time for the simulated combined errors dataset.



(a) (b)

Fig. 85 The influence of β on the residuals for the simulated combined errors dataset. (a) RMS residuals; (b) PV residuals.

Fig. 86 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.



(a) (b)

Fig. 86 The influence of m on the time performance. (a) $m > 1$; (b) $m > 3$.

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 give the reader an idea about convergence rate.

Fig. 87 is a plot of the evolution of the objective function's value over the iterations of the fitting algorithm. Starting from around the 3^{rd} 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 23^{rd} iteration. As compared to the LM function evolution, it takes 7 iterations to stabilize and 43 iterations to reach the stop criterion. The sum of squared distances evolution depicts the rate of convergence of the L-BFGS algorithm and confirms that it is relatively fast compared to LM. Fig. 88 shows the evolution of the transformation (motion) variables over the iterations. An offset of 0.1 mm along the z-direction is added on purpose while all other variables are set to zero. It is therefore clear from Fig. 88e why parameter t_z decreases fast by 0.1 mm. The objective function's value decreases constantly and follows the decay of variable t_z.

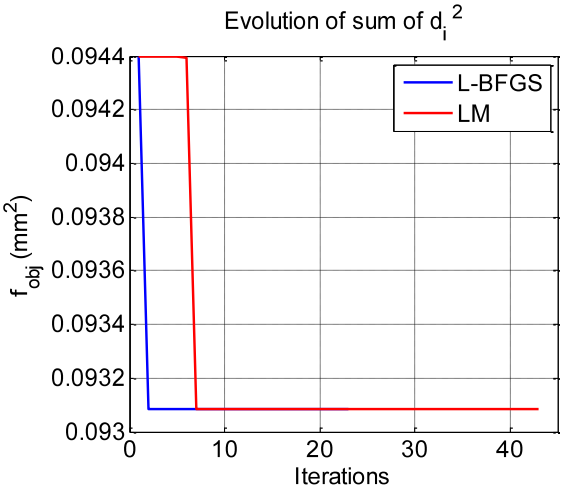
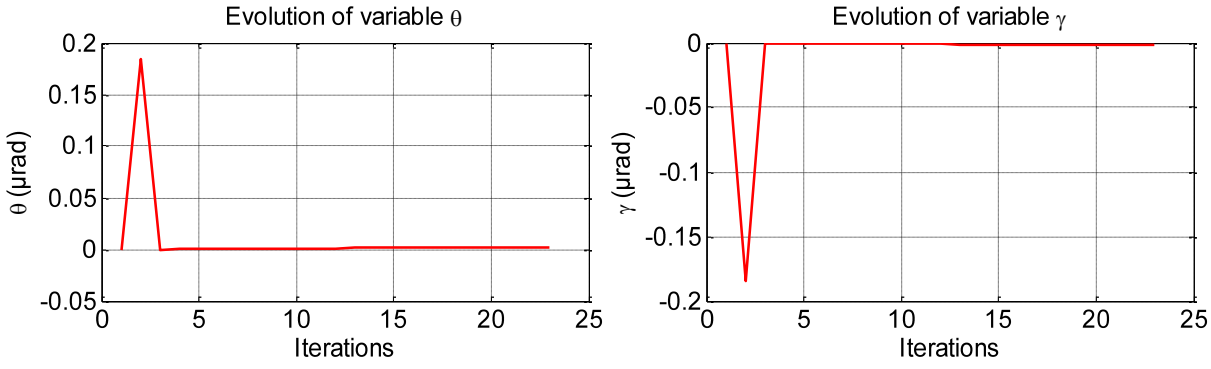


Fig. 87 The evolution of the objective function's value over the iterations for both L-BFGS (blue) and LM (red).



(a) (b)

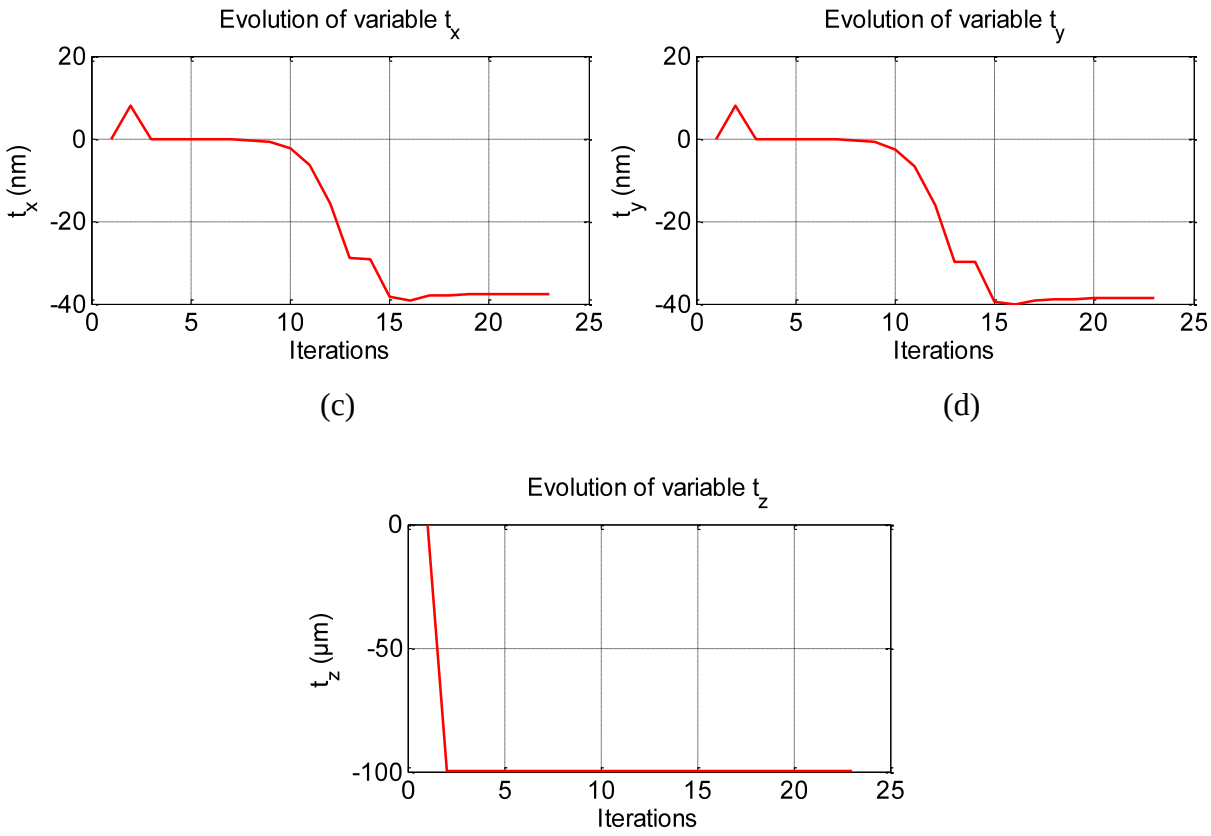


Fig. 88 Evolution of each of the transformation (motion) parameters over the iterations of L-BFGS: (a) θ; (b) γ; (c) t_x; (d) t_y; (e) t_z.

6. Impact of variable sampling density

In this paragraph, we simulate variants of the combined errors dataset by subdividing it into two separate regions having different and various sampling densities. For a given aperture value, we distinguish the inner points from the outer points and select, in each region, a percentage of the existing points. Table 8 shows the residual errors that result from different choices of aperture and percentage points.

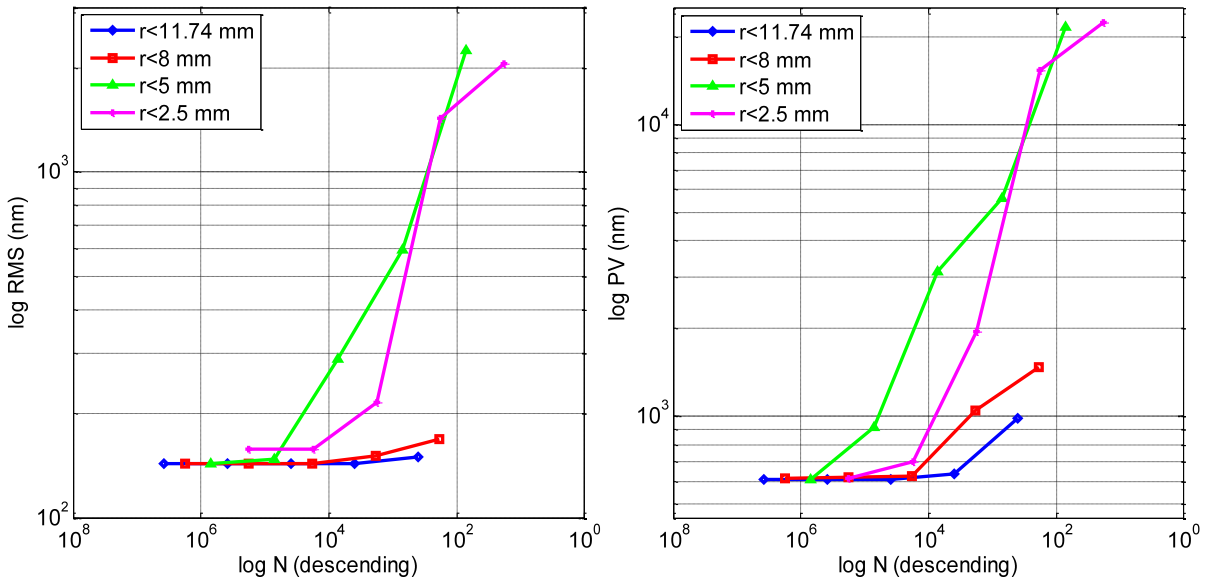
The main conclusion to expose from these tests is that sampling density does not play a major role on the PV of the residual errors, however, it does affect the results of RMS by an amount not exceeding 4 % and equivalent to a few nanometers. Furthermore, the less points considered in the inner region, the higher the value of RMS.

r	Inn. percentage	Out. percentage	RMS (nm)	PV (nm)
11.74	100 %	×	136.455	609.364
2	20 %	10 %	136.461	609.361
	5 %	50 %	136.462	609.364
	1 %	50 %	136.465	609.360
4	50 %	10 %	138.752	609.484
	5 %	50 %	137.976	609.294
	1 %	50 %	138.420	609.283
6	100 %	1 %	136.799	609.441
	1 %	50 %	140.811	609.252
	1 %	100 %	141.229	609.221
8	10 %	1 %	136.452	609.363
	100 %	1 %	136.453	609.399
	5 %	50 %	136.657	609.305

Table 8 Impact of sampling density based on separate aperture regions on the residual errors for the combined errors dataset (r: aperture radius; Inn.: inner; Out.: outer).

7. Impact of the region

Since it seems that the considered region has impact on the residual errors, we perform some last simulations to study the impact of the region alone. We consider the same dataset with combined random and systematic errors and fit the aspherical model to the dataset by accounting for a sub-region of the points delimited by an aperture value. The original simulated dataset covers an area of 6×6 mm² but here we will consider a larger dataset that would cover the entire clear aperture ($CA = 23.48$ mm $\Leftrightarrow r = 11.74$ mm) of the measured asphere. In these tests we also vary the ratio of data to be read so that we study the effect of reducing the number of points N too. Fig. 89 illustrates the effect of fitting data by changing the value of r which delimits the region within which data are read. The blue curve represents the full aperture where all the points have been accounted for.



(a) (b)

Fig. 89 Log-log plot of the evolution of the residual errors with aperture and number of points for the simulated combined errors dataset: (a) evolution of RMS; (b) evolution of PV.

The major conclusion is that even if $N < 1000$ points, the residual errors do not change by much for the full aperture. Conversely, for larger values of N but smaller apertures, the residual errors vary considerably (green and magenta curves).

VI. Application to the measured aspherical lens AO775

Once the algorithms are validated on simulated datasets, the aim is to tackle real measured data with the three proposed algorithms in order to further compare them and show the efficiency of L-BFGS. With the data in-hand, we expose a comparison involving the other project partners who have measured the same aspherical lens. The evaluation of the asphere form errors must be done over the confinement of its clear aperture.

The aspherical lens described earlier is mounted on the LNE high precision profilometer for the measurement process. The asphere is posed on the Zerodur table as described in Fig. 60 and a manual alignment is performed. Since the alignment of the measurement coordinate system with the model coordinate system is approximated, vertical distance minimization is not valid because it inhibits horizontal degrees of freedom t_x and t_y. 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, therefore they have no impact in the vertical direction. The asynchronous vertical motion errors are

supposed negligible compared to form errors which act in the orthogonal direction, hence, it is wiser to apply orthogonal distance fitting from here on.

The aspherical lens is measured on the LNE high-precision profilometer twice: once with the tactile stylus probe and once with the optical chromatic confocal probe. The measurement takes place in the LNE's cleanroom where temperature and humidity are monitored and controlled (c.f. Chapter 1 - V. 2.). Moreover, the aspherical lens is carefully cleaned with special products before each measurement.

1. Comparison of the fitting algorithms

The measurement process is automatic and the data are recorded and reported in Cartesian (X, Y, Z) coordinates. The lens is firstly scanned using the tactile probe over an area of 6×6 mm², giving a grid of 1225×1225 points denoted as TScan_1. Then the lens is scanned using the chromatic confocal probe over an area of 5×5 mm², giving a grid of 1002×1002 points denoted as OScan_1.

For practical purposes, portions of each scan are extracted and are denoted as TScan_2 and Oscan_2. These extracted portions are about 3.5×3.5 mm² in area and are non-centered grids of about $500{,}000$ points each. The results of the L-BFGS, LM based on Least-Squares orthogonal distance minimization are compared to ICP for all the experimental datasets: the complete measurements and the extracted portions. Table 9, Table 10, Table 11 and Table 12 report the residual errors (RMS and PV) for different dataset sizes for TScan_1, TScan_2, OScan_1 and Oscan_2, respectively.

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

Table 9 Fitting using orthogonal distance minimization for TScan_1. N denotes the number of points.

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

Table 10 Fitting using orthogonal distance minimization for $TScan_2$. N denotes the number of points.

	$L-BFGS$		LM		ICP	
N	$RMS\ (nm)$	$PV\ (\mu m)$	$RMS\ (nm)$	$PV\ (\mu m)$	$RMS\ (nm)$	$PV\ (\mu m)$
50,000	335.95	6.1580	335.95	6.1584	336.42	6.1597
200,000	335.95	6.1564	335.95	6.1567	336.43	6.1604
500,000	335.94	6.1571	335.94	6.1576	336.43	6.1594
1,000,000	335.94	6.1570	335.94	6.1572	336.43	6.1598

Table 11 Fitting using orthogonal distance minimization for $OScan_1$. N denotes the number of points.

	$L-BFGS$		LM		ICP	
N	$RMS\ (nm)$	$PV\ (\mu m)$	$RMS\ (nm)$	$PV\ (\mu m)$	$RMS\ (nm)$	$PV\ (\mu m)$
75,000	176.69	1.0493	176.69	1.0494	176.85	1.0511
160,000	176.69	1.0583	176.69	1.0585	176.85	1.0644
500,000	176.69	1.0582	176.69	1.0583	176.85	1.0651

Table 12 Fitting using orthogonal distance minimization for $OScan_2$. N denotes the number of points.

The residual errors are almost identical for all three algorithms, especially L-BFGS and LM. For the tactile measurement, $TScan_1$ returns a RMS value of about 217.2 nm and $TScan_2$ returns a RMS of about 44.1 nm. This result is coherent since $TScan_1$ covers a wider measured area for which the measurement distance of the probe becomes larger at farther regions from the apex. Fig. 90 illustrates the computational time in CPU seconds for the fitting of the measured points of $TScan_1$ for different dataset sizes. It shows a linear complexity for L-BFGS with respect to the number of points. Fig. 91 plots the evolution of the sum of squared distances over the iterations of the L-BFGS and LM algorithms for the case of $TScan_1$. 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. As for the optical measurement, $OScan_1$ returns a RMS value of

about 336 nm and $OScan_2$ returns a RMS of about 176.7 nm. Again, this result is reasonable since the measurement distance in the case of $OScan_1$ is larger than the distance in the case of $OScan_2$. As mentioned previously in Chapter 1, the uncertainty of confocal measurement is very sensitive to the measurement distance.

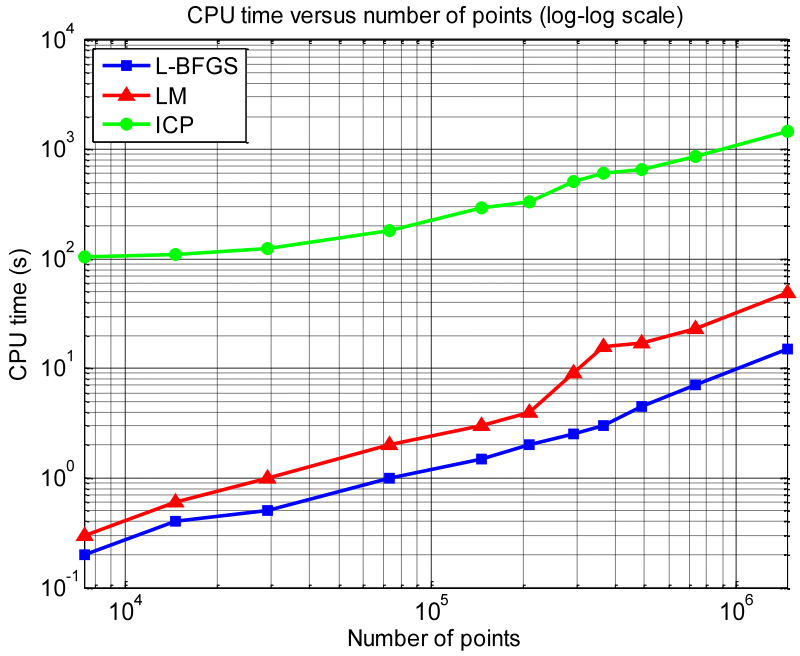
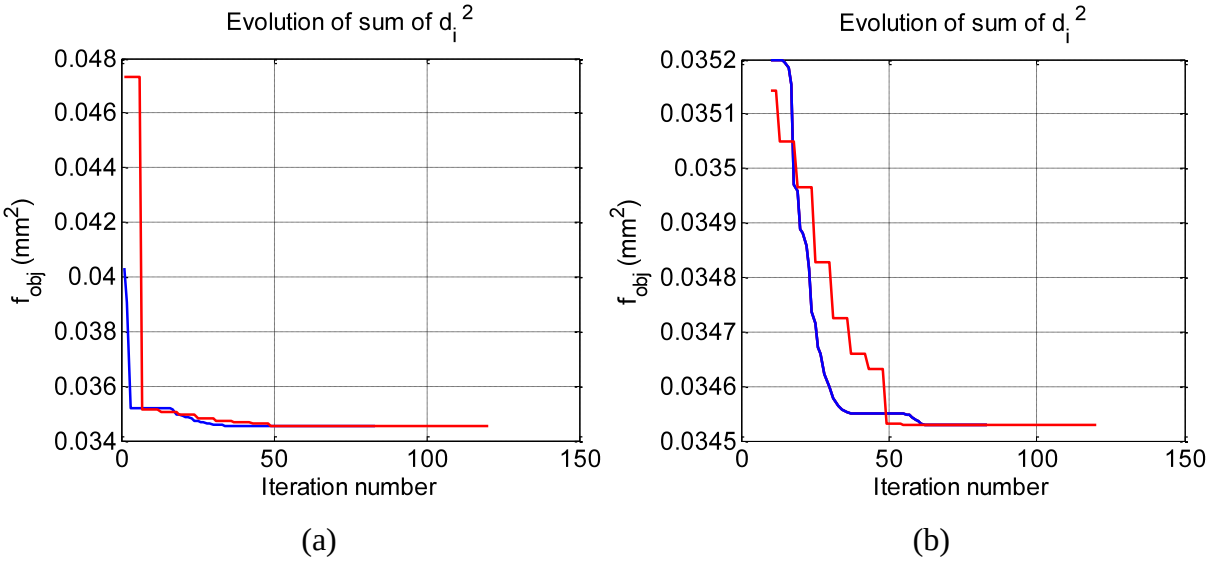


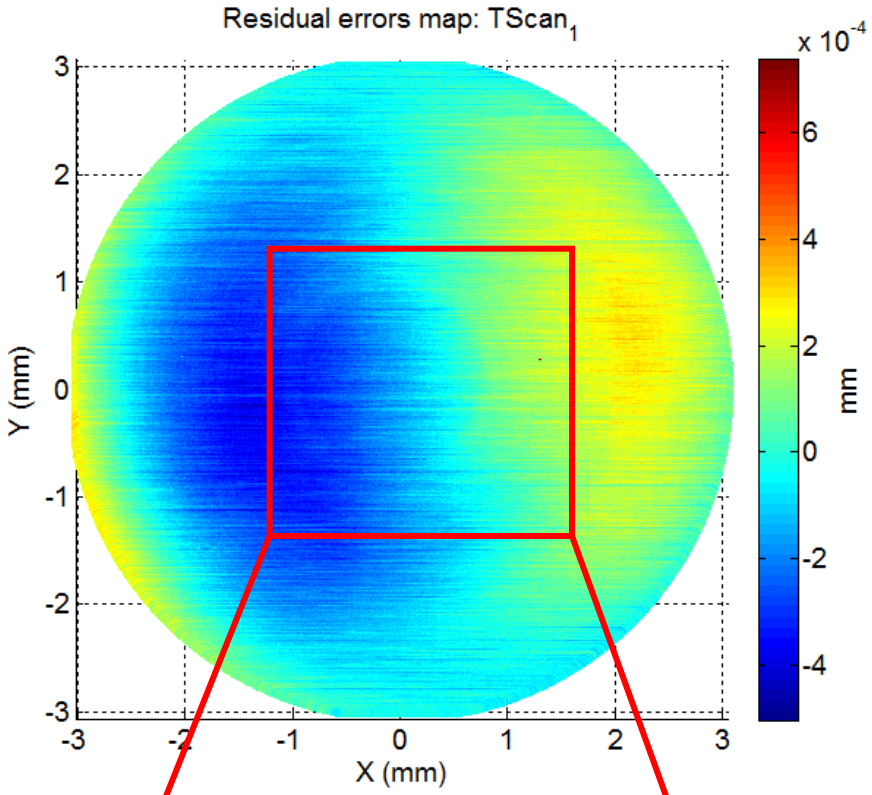
Fig. 90 Computational time of the algorithms versus dataset size.



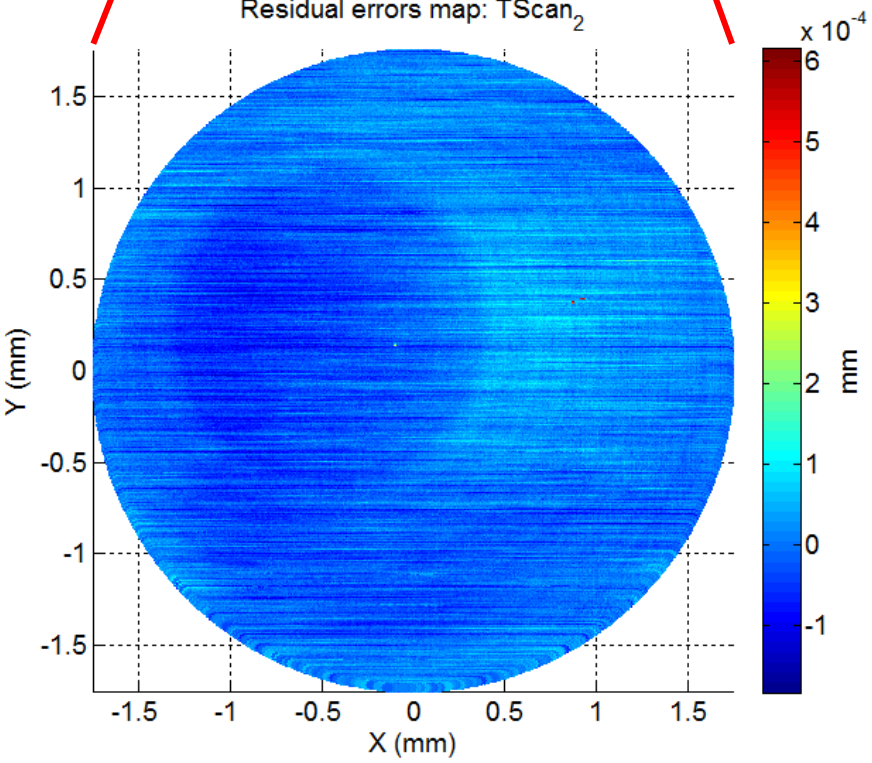
(a) (b)

Fig. 91 Evolution of the objective function value through the iterations for the case of $TScan_1$ (legend: L-BFGS: blue, LM: red, $f_{obj}=\phi$). (a) All iterations; (b) Starting from iteration 10.

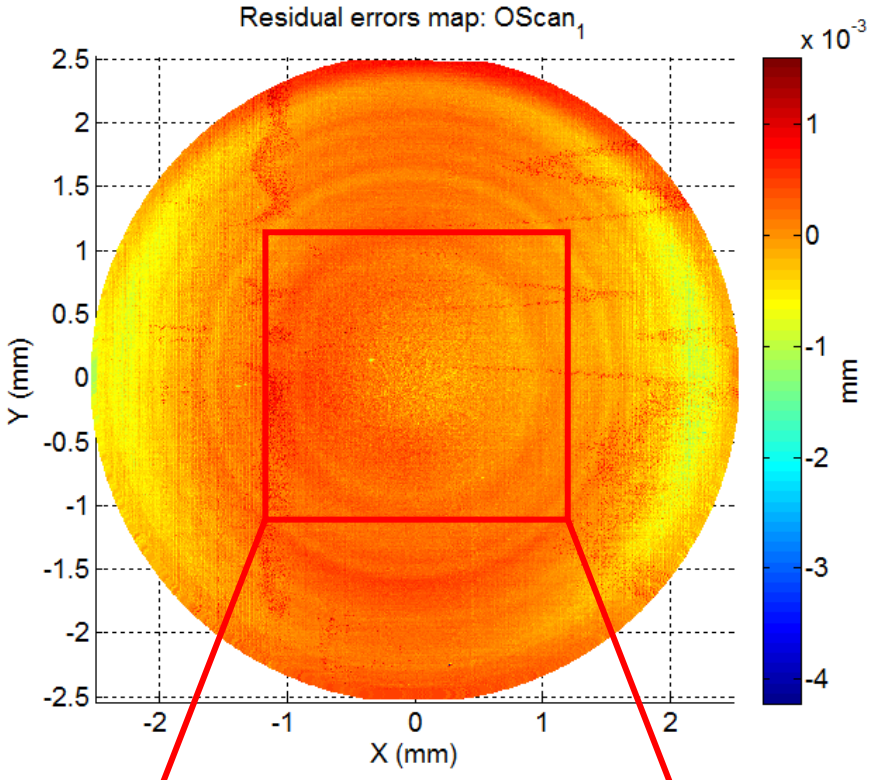
The error maps of the residual errors resulting from the fitting of each of the tactile measurement datasets using L-BFGS are plotted in Fig. 92a and Fig. 92b, and show that the residual errors distribution depend on the measured zone. The plots are shown only for the clear apertures of 6 mm (complete measurement) and 3.5 mm (extracted portion), respectively. The error maps of the residual errors resulting from the fitting of each of the optical measurements are represented in Fig. 92c and Fig. 92d for the clear apertures of 5 mm (complete measurement) and 3.5 mm (extracted portion), respectively.


Residual errors map: TScan1
x 10-4
6
4
2
0
-2
-4
mm
Y (mm)
X (mm)
3
2
1
0
-1
-2
-3

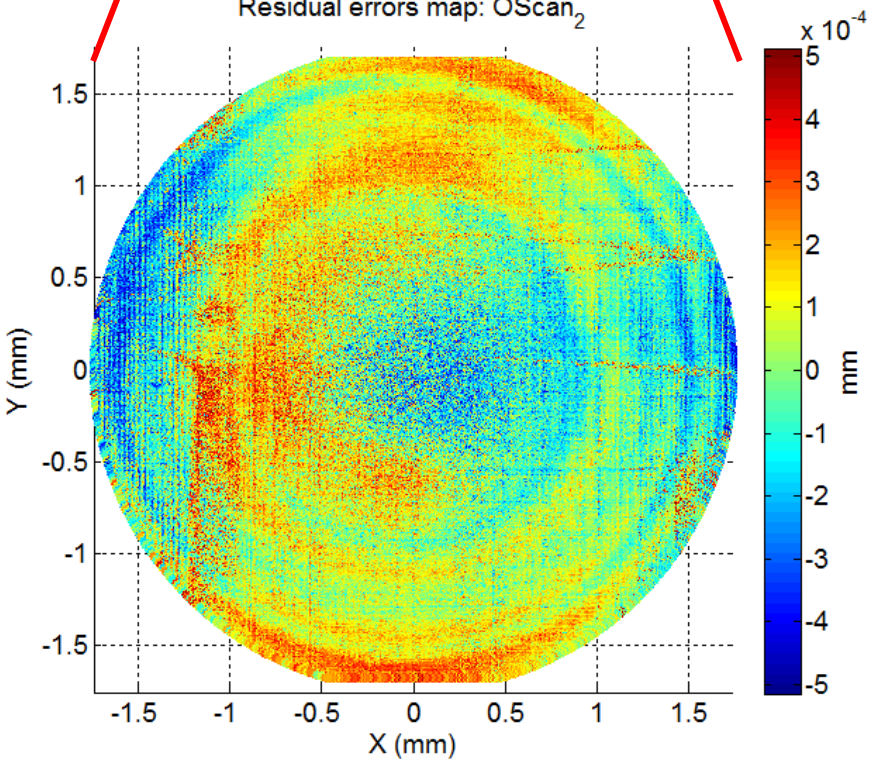
(a)


Residual errors map: TScan2
x 10-4
6
5
4
3
2
1
0
-1
mm
Y (mm)
X (mm)
1.5
1
0.5
0
-0.5
-1
-1.5

(b)



(c)



(d)

Fig. 92 3D residual errors map characterizing form defects. (a) $TScan_1$, (b) $TScan_2$, (c) $OScan_1$, (d) $OScan_2$.

2. Comparison with the IND10 partners measurements

As mentioned previously, the same aspherical lens AO775 is measured at different metrology labs and industries by means of different UHPMMs and measurement techniques. Each partner used its own machine with the aim of assessing its measuring capabilities over aspherical parts and employed its own software for data analysis. However, in order to have a truthful comparison of the results, the measurement and the data processing techniques must be founded on a common basis. A round-robin schedule was set for this purpose after which the outcomes have been gathered and exposed here. The major differences between the recorded datasets is that they have different numbers of recorded data points and constitute dissimilar measurement areas (Fig. 93). VSL, METAS and IBSPE (Measurement 1: IBSPE_1) have measured the entire surface of the lens surface on their respective UHPMMs, F25, µCMM and ISARA 400. TNO on NANOMEFOS and IBSPE on F25 (Measurements 2 and 3: IBSPE_2 and IBSPE_3) have measured the lens over a 10 mm clear aperture. Finally the LNE has measured the lens over smaller square grids around the apex (Fig. 93).

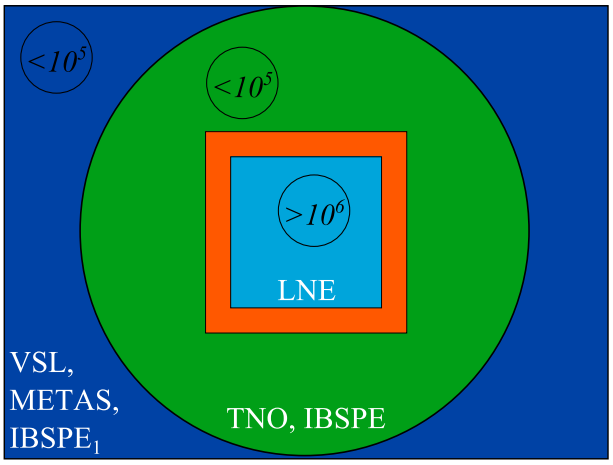
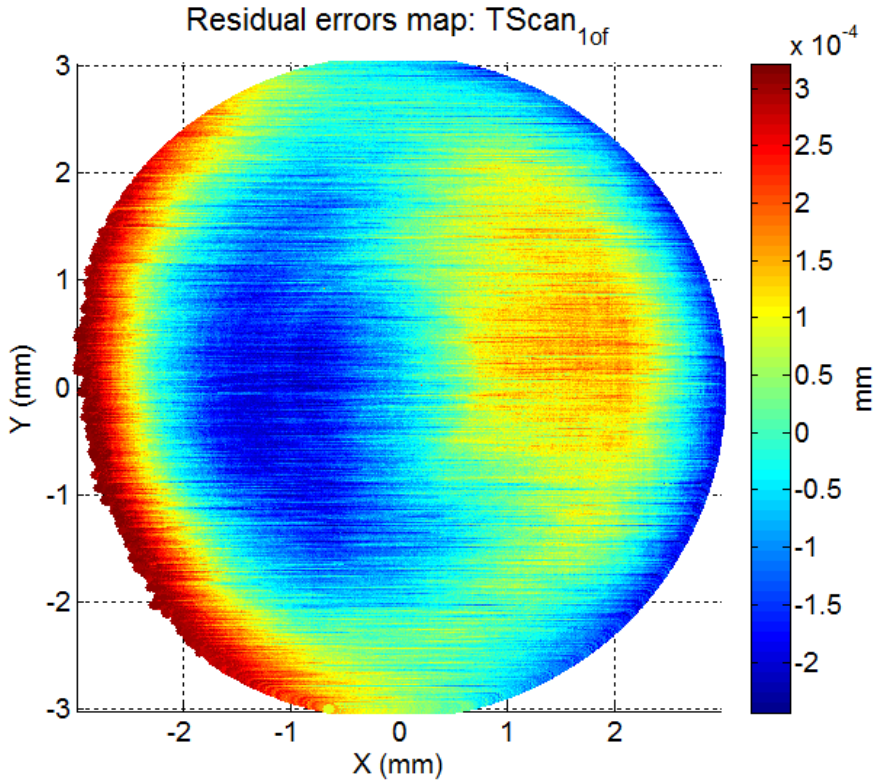
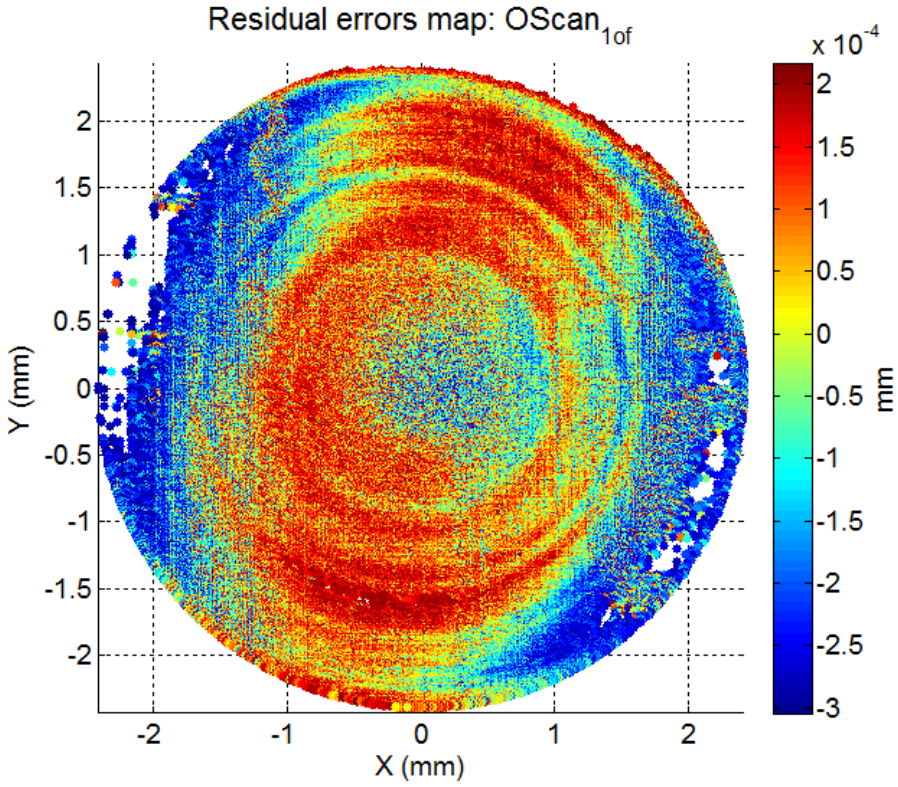


Fig. 93 Different measurement regions and number of points for the same aspherical surface.

The meetings held with the JRP partners revealed that the data they provide result from a pre-processing stage which might have included filtration, outlier removal, tilt and coma corrections, etc... Our first fitting of the LNE data was applied to the raw data directly with no filtration or special treatment what-so-ever ($TScan_1$ and $Oscan_1$). However, by removing outliers, we are able to improve our results and obtain more coherent residual errors ($TScan_{1\_of}$ and $Oscan_{1\_of}$). The partners have agreed to discard points which have residual errors greater than 250 nm because beyond this value, data are considered to be dust particles.



(a)



(b)

Fig. 94 Residual errors map: (a) tactile and (b) optical measurements after removing outliers from the data (holes indicate positions of removed outliers).

Usually outlier removal happens during a pre-processing stage and one way of proceeding is by fitting the raw data to the model a first time and then detecting residuals larger than 250 nm. Fitting is applied a second time and the outliers-free residual errors are re-evaluated (Table 13 and Fig. 94). In view of comparing the measured data of all project partners properly, we firstly need to bring back all datasets to the same reference frame. To do that, we perform a first fitting operation to all the datasets and expose the results (Mean, RMS and PV of residual errors) in Table 13. The available residual errors of the measurements (Mean and/or RMS) originally found by each partner are displayed in the last column. As compared to the RMS results we find with our proposed L-BFGS software, we observe that the partners announce similar values (Fig. 95). Without having the values of PV from the partners, we are not able to compare them to the values we find with our software.

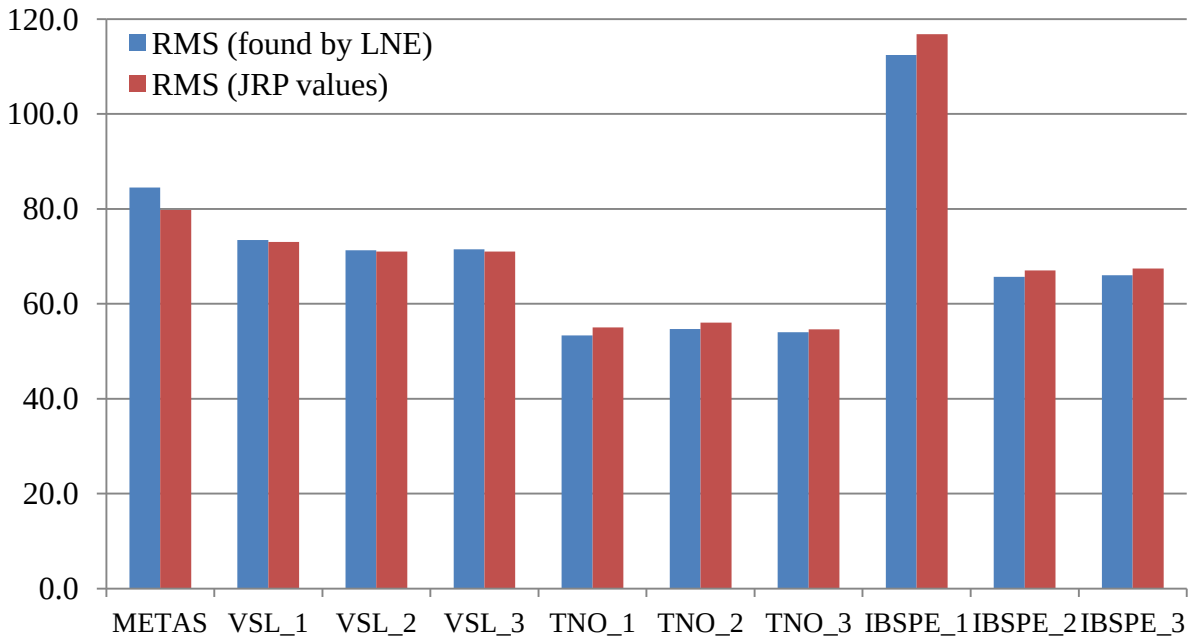


Fig. 95 Comparison of RMS values found by JRP partners and RMS values found by LNE with the proposed L-BFGS software for the datasets measured by JRP partners.

Data	Nbr. of Points	Stats (nm)	L-BFGS	LM	ICP	Partners available results
$TScan_1$	1,500,000	*Mean*	163.5	163.5	163.6	×
		RMS	217.2	217.2	217.3	
		PV	2172.6	2173.2	2198.3	
$TScan_{1\_of}$	485,000	*Mean*	38.9	38.9	×	×
		RMS	106.3	106.3	×	
		PV	565.5	565.4	×	
$OScan_1$	1,000,000	*Mean*	261.5	261.5	261.7	×
		RMS	336.4	336.4	336.5	
		PV	6157.2	6157.2	6157.9	
$OScan_{1\_of}$	460,000	*Mean*	59.4	59.4	×	×
		RMS	128.4	128.4	×	
		PV	521.5	521.6	×	
METAS	88,000	*Mean*	62.1	62.1	62.1	79.8 nm *RMS*
		RMS	84.5	84.5	84.5	
		PV	1 607.0	1 607.0	1 607.0	
VSL_1	1,100	*Mean*	57.8	57.6	57.8	73 nm *RMS*
		RMS	73.4	73.0	73.1	
		PV	645.0	645.0	645.1	
VSL_2	1,100	*Mean*	56.4	56.4	56.5	71 nm *RMS*
		RMS	71.3	71.3	71.3	
		PV	617.0	617.0	617.1	
VSL_3	1,100	*Mean*	56.6	56.6	56.7	71 nm *RMS*
		RMS	71.5	71.5	71..6	
		PV	545.0	545.0	545.1	
TNO_1	31,000	*Mean*	40.4	40.6	40.4	55 nm *RMS*
		RMS	53.3	54.8	53.3	
		PV	519	536	519	
TNO_2	31,000	*Mean*	41.9	41.6	41.9	56 nm *RMS*
		RMS	54.7	54.8	54.7	
		PV	490	490	490	
TNO_3	31,000	*Mean*	41.2	41	41.2	54.6 nm *RMS*
		RMS	54	54.1	54	
		PV	462	462	462	
IBSPE_1	3,200	*Mean*	90.7	91	90.7	94.1 nm *Mean* 116.8 nm *RMS*
		RMS	112.4	112.5	112.4	
		PV	765	765	765	
IBSPE_2	2,600	*Mean*	52.4	52.7	52.4	53.7 nm *Mean* 67 nm *RMS*
		RMS	65.7	65.7	65.7	
		PV	354	354	354	
IBSPE_3	2,600	*Mean*	52.8	53	52.8	54 nm *Mean* 67.4 nm *RMS*
		RMS	66	66	66	
		PV	355	355	355	

Table 13 Measurement results (residual errors) for AO775 aspherical lens of IND10 project partners (Nbr. of Points: Number of Points; Stats: Statistics).

In order to make the comparison more accurate, we now need to bring all the previous datasets (Table 13) down to the same area. The least common area among all datasets corresponds to the area of the dataset ($OScan_1$) having the smallest aperture ($r = 2.5$ mm). Recall that, in section V. 7. , the simulations have shown that taking a smaller region might alter the results. The new results which are suitable for comparison are exposed in Table 14 and plotted in Fig. 96. For a matter of avoiding repetitive results, we only run the tests with the L-BFGS software.

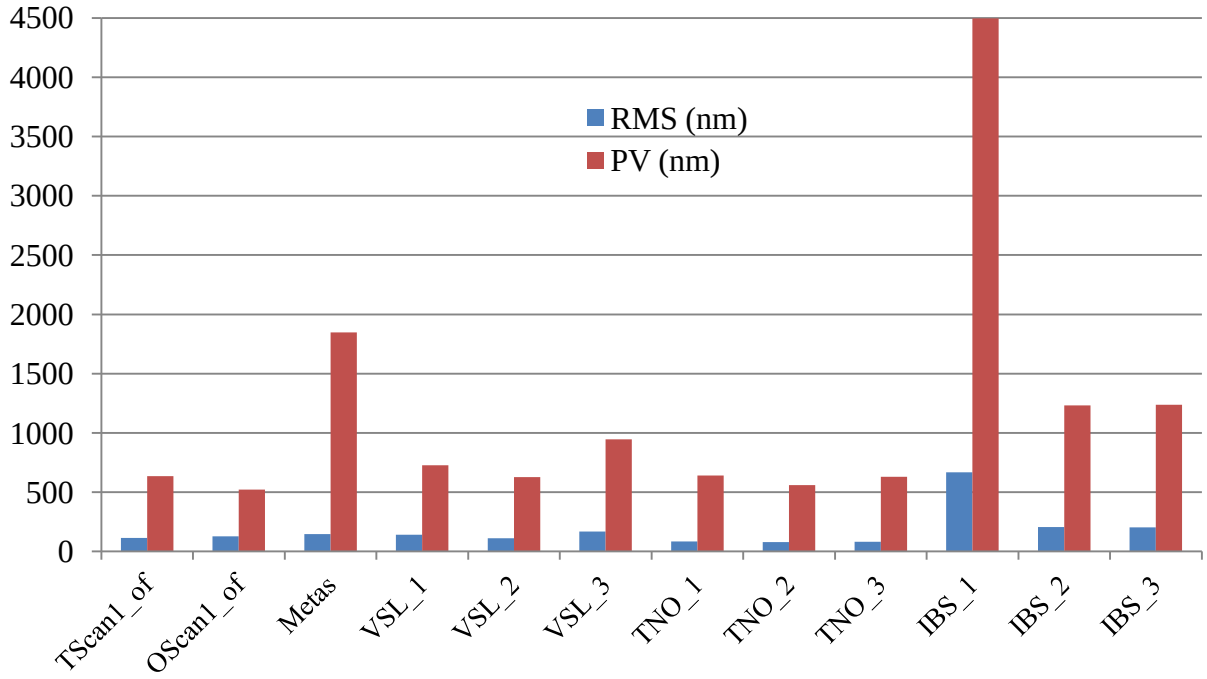


Fig. 96 Comparison of the RMS and PV of the residual errors of the measured data by the project partners.

The results show that within the lower bound on the number of points which is about a few hundreds of points, the residual errors found still have dissimilarities. These are the consequence of different pre-processing approaches and different measurement uncertainties as shown in the kernel smoothing density estimate plot of Fig. 97 using function "*ksdensity*" in Matlab. The Kernel Density Estimation, also known as the Parzen–Rosenblatt window method, is a method that can be used for estimating the underlying probability density function of multivariate distributions.

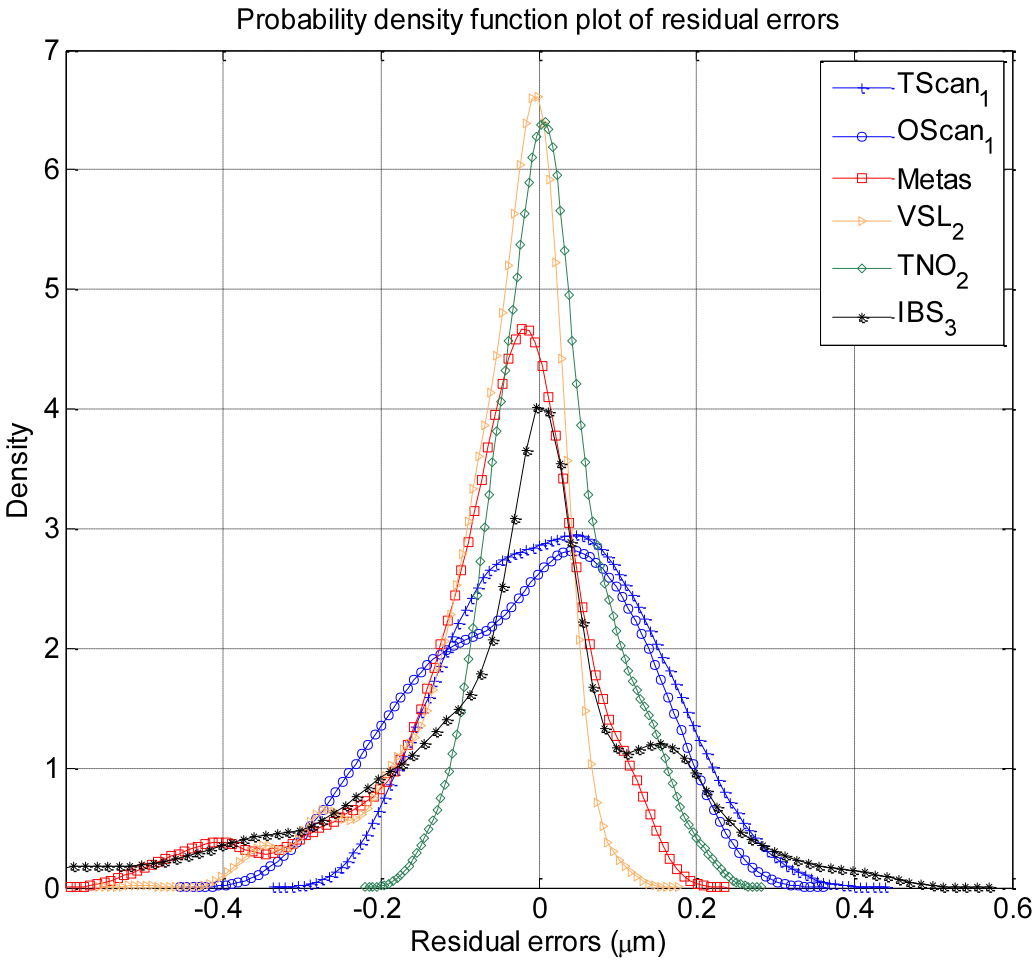


Fig. 97 Probability density function plot of the residual errors of the project partners data fitting.

For VSL, TNO and IBS measurements we choose to consider the data which gave the smallest residual errors (Fig. 97). We also consider the same number of sample points. From this plot, we can see that each fitting gives a different residual errors distribution. Although the mean of residual errors is small for the fitting of the METAS data, we can see that there are residual errors that are considerably high.

DATA	Original measured aperture (mm)	Nbr. of Points	Stats (nm) with L-BFGS	
$TScan_{1\_of}$	3	380 000	$Mean$	15.5
			RMS	112.7
			PV	635.2
$OScan_{1\_of}$	2.5	460 000	$Mean$	-0.05
			RMS	128.4
			PV	522.5
METAS	10	19 074	$Mean$	-64.08
			RMS	145.3
			PV	1848.8
VSL_1	10	196	$Mean$	-56.6
			RMS	141.1
			PV	726.7
VSL_2	10	196	$Mean$	-59.4
			RMS	110.4
			PV	626.7
VSL_3	10	196	$Mean$	-58.3
			RMS	168.3
			PV	945.6
TNO_1	10	7 858	$Mean$	19.1
			RMS	85.0
			PV	641.8
TNO_2	10	7 855	$Mean$	22.9
			RMS	79.9
			PV	558.8
TNO_3	10	7 858	$Mean$	21.6
			RMS	81.4
			PV	628.9
IBS_1	10	317	$Mean$	-102.2
			RMS	668.7
			PV	4496.6
IBS_2	10	484	$Mean$	-42.2
			RMS	205.6
			PV	1233.0
IBS_3	10	484	$Mean$	-43.6
			RMS	203.5
			PV	1237.4

Table 14 Measurement results (residual errors) for AO775 of IND10 project partners taken on the least common aperture of 2.5 mm (Nbr. of Points: Number of Points of reduced datasets; Stats: Statistics).

VII. Extension to Forbes models (*strong* aspheres)

The classical expression of aspheres lacks of simple and one-to-one interpretation, meaning that the model parameters are dependent among themselves and do not describe shape clearly. The parameters do not act independently in the definition of the shape of the surface and they can reach very low values for higher-order asphericity. Although not embodied in any ISO standard, two new expressions of aspherical surfaces have been introduced by Forbes *et al* [66,67]. According to Forbes, these expressions are more "efficient" than the classical one across two aspects. On one hand, the polynomial coefficients (a_js) can be expressed with less decimal digits and increase numerical precision. On the other hand, the independency of the model parameters makes the processes of optical design, tolerancing and human interpretation much easier. Forbes gives detailed clarifications on this topic [66] and proposes a first equation based on conics and a second one based on a departure from a sphere. Each formula is more or less appropriate according to whether the aspherical shape is closer to a conical shape with slight departures from a conic or closer to a sphere with slight departures from the best-fit sphere.

Conical shapes are known to have special optical properties and the equation of what Forbes calls them *strong* aspheres is given in (32).

$$z(u) = \frac{cu^2}{1+\sqrt{1-\kappa c^2 u^2}} + v^4 \sum_{j=0}^{M} b_j Q_j^{con}(v^2) \tag{32}$$

where, c, κ and the b_js have different values than in the classical expression with less decimal digits. $b_j Q_j^{con}$ are the terms of a set of orthogonal polynomials that represent the departure from the conical shape. The polynomials are written in terms of the normalized variable $v = {}^{u}/_{u_{max}}$ where u_{max} is the aperture of the asphere. They are a particular case of the Jacobi polynomials (4). The first few polynomials are a set of Jacobi polynomials calculated with $\alpha = 0$ and $\beta = 4$:

$$\begin{cases} Q_0^{con}(x) = 1 \\ Q_1^{con}(x) = -6x - 5 \\ Q_2^{con}(x) = 28x^2 - 42x + 15 \\ Q_3^{con}(x) = 120x^3 - 252x^2 + 168x - 35 \\ \vdots \end{cases} \tag{33}$$

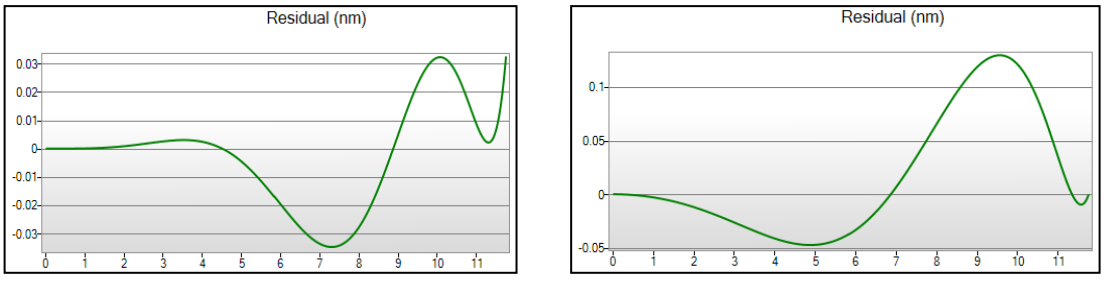
Not only the manufacturing of aspheres is more cost effective if the deviation from a sphere is constrained, but also metrology is more appropriate. *mild* aspheres, evoked in Chapter 1, are another expression for aspherical shapes which also consist of polynomials with independent model parameters (34).

$$z(u) = \frac{c_{bfs}u^2}{1+\sqrt{1-c_{bfs}^2u^2}} + \frac{v^2(1-v^2)}{\sqrt{1-c_{bfs}^2\,u^2}}\sum_{j=0}^{M} b_j Q_j^{bfs}(v^2) \qquad (34)$$

where, c_{bfs} is the curvature of the best-fit sphere, $b_j Q_j^{bfs}$ are the terms of a set of orthogonal polynomials that represent the departure from the best-fit sphere.

Being part of QED Technologies company, Forbes has developed a software called QED surface conversion tool capable of converting aspherical surface expressions from one model to another [70] (Fig. 99). The measured aspherical lens model involved in this thesis has been converted into both of the Forbes asphere definitions. The conversion residual errors have shown that the aspherical model we are dealing with is closer to a *strong* asphere than a *mild* one. The residual errors due to the conversion to a *strong* asphere are negligible as they range between ± 0.03 nm; whereas the residual errors after the conversion to a *mild* asphere range between -0.05 and $+0.14$ nm (Fig. 98).

The *strong* asphere model has the following model parameters: $c = 0.0445434248$, $\kappa = -0.361929$, $b_0 = 0.0013172$, $b_1 = -0.0011596$, $b_2 = -0.0001597$, $b_3 = 2.2 \times 10^{-6}$, and $b_4 = -1.0 \times 10^{-7}$. Unlike the classical model parameters, the *strong* asphere parameters values turn out to be limited in the decimal digits they have and are not too small, i.e. not very close to 0.



(a) (b)

Fig. 98 Residual errors after conversion from the classical model to: (a) the *strong* asphere model; (b) the *mild* asphere model [70].

Our objective here is to try to depict whether there are advantages in using Forbes equations in the characterization of aspherical surfaces. Besides describing the shape better and facilitating its interpretation, we want to investigate the possibility of improving algorithmic complexity. For the converted model, fitting is applied on the same simulated datasets as well as the measured ones [165].

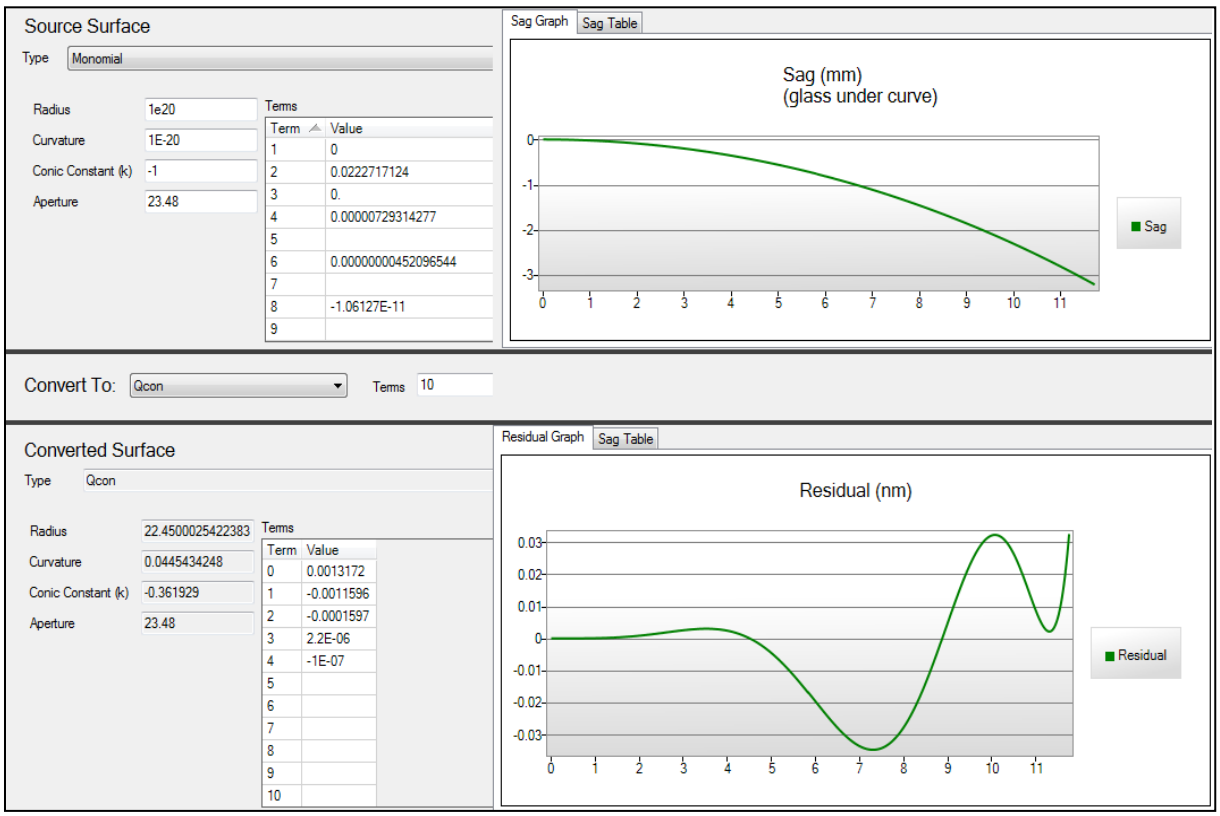


Fig. 99 Screenshot of QED surface conversion tool software [70]. Upper left: input of classical model parameters. Lower left: output of converted parameters. Lower right: plot of the residual errors between both models.

1. Tests on simulated data

For this study, we are not only interested in residual errors (RMS and PV), but also in model parameters estimation. The correctness of the estimation as well as the time to perform the fitting for a different number of parameters are all taken into consideration. The parameters are initialized to their design values and we use L-BFGS for the optimization. Table 15 shows the computational time and the residual errors output by fitting the combined errors simulated dataset with both the classical and the Forbes model and solving for the motion parameters only.

	Forbes *strong* asphere model			Classical model		
	$\boldsymbol{RMS}\ (\boldsymbol{nm})$	$\boldsymbol{PV}\ (\boldsymbol{nm})$	$\boldsymbol{Time}\ (\boldsymbol{s})$	$\boldsymbol{RMS}\ (\boldsymbol{nm})$	$\boldsymbol{PV}\ (\boldsymbol{nm})$	$\boldsymbol{Time}\ (\boldsymbol{s})$
10,000	136.480	609.032	1	136.456	609.031	0.5
100,000	136.455	609.322	4	136.455	609.321	2
1,000,000	136.455	609.404	48	136.455	609.403	24
10,000,000	136.455	609.366	309	136.455	609.364	165

Table 15 Residual errors and computational time for the fitting of both the classical and the conical Forbes models with the combined errors simulated dataset.

The results reveal that the complexity is not reduced when using the Forbes equation. For instance, the computational time measured in seconds is almost half for the classical model fitting with respect to the Forbes model fitting. Another test is done by increasing the number of variables in the minimization problem in order to see the effect of adding more variables on the behavior of the algorithm with each aspherical model. For this test, we optimize for the respective model parameters as well (c, κ and a_js for classical and c_F, κ_F and b_js for Forbes strong asphere, respectively). Table 16 shows the optimization results in which we solve for the motion and the model parameters. The estimation of model parameters is more accurate with the usage of the Forbes equation than with the usage of the classical asphere equation.

Forbes *strong* asphere model		Classical model	
Design values	**Estimated values**	**Design values**	**Estimated values**
$c_F = 0.0445434$	$\hat{c}_F = 0.0446471$	$c = 1\mathrm{E}^{-20}$	$\hat{c} = 1.5904\mathrm{E}^{-15}$
$\kappa_F = -0.361929$	$\hat{\kappa}_F = -0.372094$	$\kappa = -1$	$\hat{\kappa} = -1$
$b_0 = 0.0013172$	$\hat{b}_0 = 0.0012998$	$a_2 = 0.0222717$	$\hat{a}_2 = 0.0234798$
$b_1 = -0.0011596$	$\hat{b}_1 = -0.0011417$	$a_4 = 7.29314\mathrm{E}^{-6}$	$\hat{a}_4 = 9.45617\mathrm{E}^{-5}$
$b_2 = -0.0001597$	$\hat{b}_2 = -0.0001583$	$a_6 = 4.52097\mathrm{E}^{-9}$	$\hat{a}_6 = 2.45704\mathrm{E}^{-7}$
$b_3 = 2.2\mathrm{E}^{-6}$	$\hat{b}_3 = 2.2311\mathrm{E}^{-6}$	$a_8 = -1.061\mathrm{E}^{-11}$	$\hat{a}_8 = -2.5894\mathrm{E}^{-8}$
$b_4 = 1.0\mathrm{E}^{-7}$	$\hat{b}_4 = -1.033\mathrm{E}^{-7}$	$a_{10} = 9.887\mathrm{E}^{-15}$	$\hat{a}_{10} = 6.4874\mathrm{E}^{-12}$
$\boldsymbol{RMS}\ (nm)$	129.735	$\boldsymbol{RMS}\ (nm)$	133.998
$\boldsymbol{PV}\ (nm)$	594.829	$\boldsymbol{PV}\ (nm)$	601.463

Table 16 Model parameters estimation for the fitting of both the classical and the conical Forbes models with the combined errors simulated dataset.

Moreover, the deviations between design parameters and estimated parameters are partly due to the fact that only a portion of the asphere is simulated with data as it is shown in Fig. 100. The dotted line delimits the region $0 \leq r \leq 3$ mm which contains the data. The result of the fitting is illustrated in Fig. 100a for the Forbes model and in Fig. 100b for the classical model. The fitting using the Forbes equation gives slightly more accurate RMS and PV values than the fitting with the classical equation (Table 16). Although the shapes considerably diverge when $r > 6$, the region $r \in [0, 3]$ is fitted well. However, from the results reported in Table 16, we deduce that for the classical equation, the model parameters may vary a lot but converge to a shape that fits the data very well, while this phenomenon does not occur in the second case. With the Forbes equation, the parameters are independent therefore they do not change by much. Parameter b_4 has changed randomly but has not affected the fitting accuracy in the interest region $r \in [0, 3]$ and this could be explained by the fact that b_4 only influences the shape at higher values of r. The sign difference for parameter b_4 explains why the estimated curve diverges in the opposite direction.

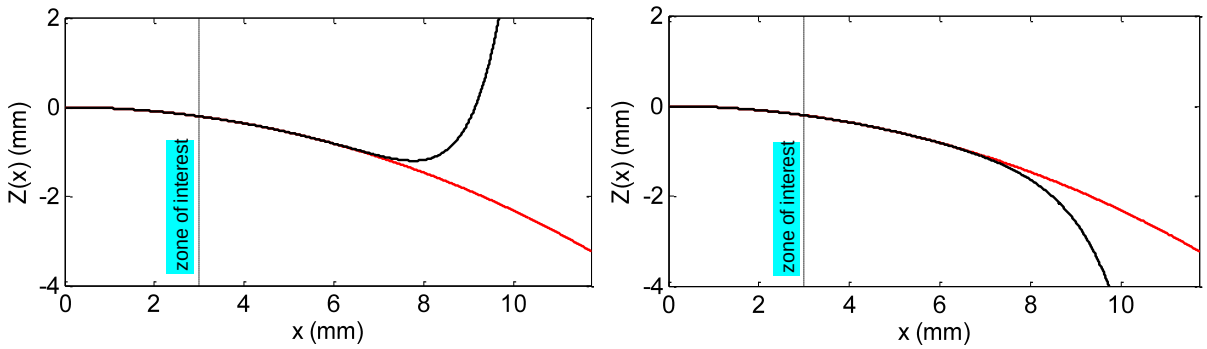


Fig. 100 The estimated aspherical form after fitting the simulated data with: (a) the estimated Forbes model (black) and (b) the estimated classical model (black). The design shape is represented in red. The dotted line delimits the region of available data at a radius of 3 mm.

The above analyses have been also performed using the LM algorithm and output similar results.

2. Tests on measured data

The first tests involve fitting the data of $TScan_1$, $TScan_{1\_of}$, $OScan_1$ and $OScan_{1\_of}$ with both the classical and *strong* Forbes models without model parameters estimation (Table 17).

On measured data, the same conclusions can be extracted, the fitting is identical whether with the classical or the *strong* Forbes model without model parameters estimation and the time complexity is almost double with the Forbes equation.

	Forbes *strong* asphere model			Classical model		
	$\boldsymbol{RMS}\ (\boldsymbol{nm})$	$\boldsymbol{PV}\ (\boldsymbol{nm})$	$\boldsymbol{Time}\ (\boldsymbol{s})$	$\boldsymbol{RMS}\ (\boldsymbol{nm})$	$\boldsymbol{PV}\ (\boldsymbol{nm})$	$\boldsymbol{Time}\ (\boldsymbol{s})$
$TScan_1$	217.19	2198.93	137	217.19	2198.93	69
$TScan_{1\_of}$	106.27	565.46	282	106.27	565.46	136
$OScan_1$	336.40	6157.19	243	336.40	6157.19	130
$OScan_{1\_of}$	128.44	522.45	195	128.44	522.45	102

Table 17 Residual errors and computational time for the fitting of both the classical and the conical Forbes models with some measured datasets.

We repeat the same tests, however this time, we include the estimation of model parameters and compare the outcomes of both the classical and the *strong* Forbes models (Table 18). We observe similar outcomes as compared to the simulations in Table 16.

Forbes *strong* asphere model		Classical model	
Design values	**Estimated values**	**Design values**	**Estimated values**
$c_F = 0.0445434$	$\hat{c}_F = 0.0447046$	$c = 1\mathrm{E}^{-20}$	$\hat{c} = 4.75143\mathrm{E}^{-5}$
$\kappa_F = -0.361929$	$\hat{\kappa}_F = -0.35493$	$\kappa = -1$	$\hat{\kappa} = -1$
$b_0 = 0.0013172$	$\hat{b}_0 = 0.0012914$	$a_2 = 0.0222717$	$\hat{a}_2 = 0.022367$
$b_1 = -0.0011596$	$\hat{b}_1 = -0.0011842$	$a_4 = 7.29314\mathrm{E}^{-6}$	$\hat{a}_4 = 2.6229\mathrm{E}^{-6}$
$b_2 = -0.0001597$	$\hat{b}_2 = -0.0001711$	$a_6 = 4.52097\mathrm{E}^{-9}$	$\hat{a}_6 = 7.3321\mathrm{E}^{-7}$
$b_3 = 2.2\mathrm{E}^{-6}$	$\hat{b}_3 = 2.3175\mathrm{E}^{-6}$	$a_8 = -1.061\mathrm{E}^{-11}$	$\hat{a}_8 = -3.3597\mathrm{E}^{-8}$
$b_4 = 1.0\mathrm{E}^{-7}$	$\hat{b}_4 = -1.029\mathrm{E}^{-7}$	$a_{10} = 9.887\mathrm{E}^{-15}$	$\hat{a}_{10} = 5.410\mathrm{E}^{-10}$
$\boldsymbol{RMS}\ (nm)$	133.42	$\boldsymbol{RMS}\ (nm)$	139.31
$\boldsymbol{PV}\ (nm)$	1451.27	$\boldsymbol{PV}\ (nm)$	1482.01

Table 18 Model parameters estimation for the fitting of both the classical and the conical Forbes models with the measured dataset $TScan_1$.

VIII. Conclusion

This chapter detailed the fitting of aspherical surfaces in the aim of characterizing their form errors. The L-BFGS method was proposed and compared to the classically used LM and ICP algorithms on both simulated and measured data. Two cases were proposed for the simulations in conformance with ISO10360-Part 6: 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 (RMS and PV) in the proposed way of selecting points. Nonetheless, L-BFGS shows linear time complexity with respect to the number of points and runs faster than LM and largely faster than ICP.
It was also shown that vertical distance minimization is more suitable than orthogonal distance minimization when errors manifest in the vertical direction. Indeed, random errors that are purely due to noise act in the measurement direction and in the case of the LNE high-precision profilometer, this direction is vertical. It is only in particular contexts where the measurement direction is constantly normal to the surface (NANOMEFOS machine), that orthogonal distance minimization is the most appropriate.

In an attempt to compare data processing techniques and measurement capabilities, we have listed the results of the project partners against our results. The partners, with their own fitting methods, find similar RMS values to those we find using the L-BFGS method and our version of LM. This means that the optimization algorithms are similar with the exception that the newly proposed L-BFGS runs faster. L-BFGS can handle very large data of several millions of points and converge relatively fast.

Also in this chapter, we outlined a comparison between the usage of the classical model and the usage of one of the new aspherical models proposed by Forbes. It has been observed that performing the fitting with the Forbes *strong* aspherical model gives a more accurate fit that consists of smaller residuals and a more accurate estimation of model parameters. Nonetheless, fitting using the *strong* aspherical model is much slower than fitting with the classical model.

The major conclusion that we stress is that the detailed data processing scheme for aspherical form evaluation based on L-BFGS does not exclusively hold for aspheres. In fact, any surface can be processed as long as it is defined by an analytical formulation. The

mathematical expression of the surface needs to be input to the program and the L-BFGS fitting algorithm can be used for form characterization.

According to the standards, form metrology makes use of the infinite norm L_∞ in order to calculate the envelope enclosing the points in the dataset by minimizing the difference between the maximum deviation and the minimum deviation between the surface and the dataset (Fig. 101). This method grows in complexity as the number of points in the dataset increases especially that the involved algorithms are unstable and non-deterministic.

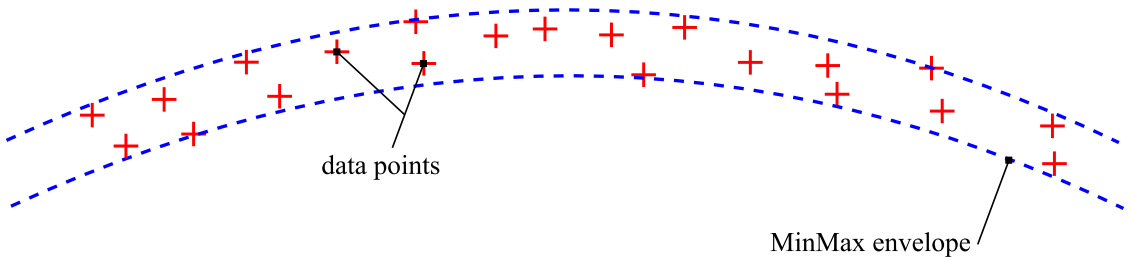


Fig. 101 MinMax fitting (minimal zone containing the maximum distance).

Despite the fact that this method works for simple geometries, it is still a major challenge when used on complex geometries such as aspheres. Only the recent works of Zhang *et al* [24,25] disclose approaches to a Chebyshev fitting of aspherical surfaces, however using non-deterministic algorithms such as the Differential Evolution (DE) algorithm and the Exponential Penalty Functions (EPF) algorithm. We intend to address this issue of MinMax fitting of aspherical surfaces in future works and that would be the core subject of the next European project EMPIR.

Chapter 3 - Reconstruction of freeform curves and surfaces

I. Introduction

Freeform surfaces exhibit a real challenge because in numerous applications, they might only be defined using a CAD model instead of an implicit or an explicit mathematical formula besides the parametric one [166]. As discussed at the beginning of the manuscript, blades or other freeform surfaces lack of a mathematical definition and are described using parametric models with polynomial or rational basis functions, such as B-Spline or Non-Uniform Rational B-Spline (NURBS) models [167]. Consequently, metrological characterization becomes more complicated. Classically, freeform surface reconstruction follows the general framework illustrated in Fig. 102 which consists of building a triangular mesh, transforming it into a quadrangular mesh and finally associating B-Spline or NURBS patches to identified regions of the mesh [168], [17–23].

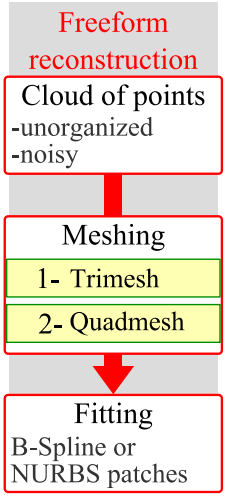


Fig. 102 Classical freeform surface reconstruction framework (Trimesh: triangular mesh; Quadmesh: quadrangular mesh).

Since a cloud of points does not infer too much information about the measured surface, a triangular mesh is constructed in order to organize the points. The mesh is a connectivity graph whose vertices are the given points and edges their connections. Triangular meshing techniques are classified into two categories: the category of methods based on combinatorial structures, such as Voronoi and Delaunay structures, and the category of methods, based on the calculation of implicit functions. The first section of this chapter is dedicated to the description of those methods and their assessment. Mesh reconstruction algorithms based on combinatorial structures were developed in the field of Computer Graphics and are founded on the assumption that data are simulated points on a C^2-smooth surface. Mesh reconstruction

algorithms based on implicit functions are founded on the assumption that an initial surface can be built using the implicit functions.

Nevertheless, these algorithms are studied here in the aim of applying them to measured data of real parts. For our purpose of reconstructing freeform surfaces in an automated fashion, we report that meshing techniques do not always give satisfying results and tend to make the process much longer. Moreover, the described framework indicates the fitting of parametric surface patches rather than a single surface. This involves problems related to the continuity between the patches which builds up in terms of complexity. We therefore come to propose an algorithm that avoids going through a meshing phase but that associates B-Spline curves and surfaces directly to the raw data without any parameterization requirement (Fig. 103).

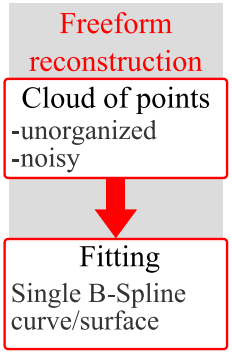


Fig. 103 Our proposed surface reconstruction framework that excludes the meshing phase.

To describe our algorithm and expose its outcomes and limitations we give a demonstration on closed curves. The algorithm will be referred to as the Discrete B-Spline Active Contour Deformation algorithm (DBACD). In section II, we thoroughly study and compare mesh reconstruction methods. In section III, we describe the proposed DBACD algorithm on closed curves and then give the basis for extending the matter to surfaces.

II. Meshing techniques

The surface reconstruction techniques which build a mesh from unstructured datasets are detailed and compared here. We proceed by describing the algorithms and then comparing their performances according to criteria that are good indicators of both robustness and reconstruction quality [169–174]. We distinguish the family of algorithms based on combinatorial structures such as the Voronoi diagram and its dual representation, the Delaunay triangulation, from the family of algorithms based on implicit techniques via the

calculation of implicit functions [175]. The algorithms that are based on combinatorial structures have all the same objective of extracting the restricted triangles that belong to the underlying surface. Nevertheless, each of the algorithms provides a different means of building the mesh. The algorithms that are based on implicit functions seek to output the same kind of mesh, however, not through a geometrical and topological approach, but through the calculation of a function that would be zero at the underlying surface's level.

The goal of this section is to expose the limits of mesh reconstruction techniques and demonstrate that they are not sufficient in general to our purpose of having an automated reconstruction from large and noisy clouds of points. We establish the evaluation on the following three assessment criteria: guarantees, robustness and complexity.

1- Guarantees. The surface reconstruction algorithm should output a mesh with equivalent topology and close geometry to the underlying surface which is generally unknown.

a) The topological equivalence is based on homeomorphisms. According to Moore [176], a homeomorphism between two topological spaces X and Y is a one-to-one function h from X onto Y such that h is continuous and the inverse of h is also continuous. A mesh is homeomorphic to a 2-manifold (surface) if the *star* (Fig. 104) of each mesh vertex is homeomorphic to a topological 2-disc [177]. This can be verified once the mesh is built through a simple algorithm that would traverse all mesh vertices and their incident triangles. In general, these triangles must be of sizes that are relatively small and proportional to the sampling density. Large triangles are considered to be topologically incorrect as they badly link points among themselves.

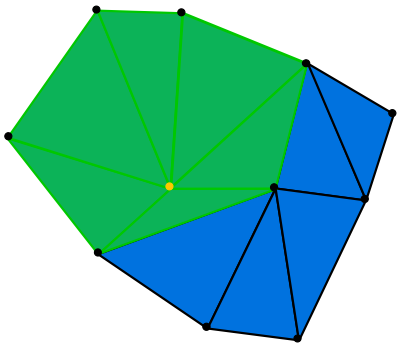
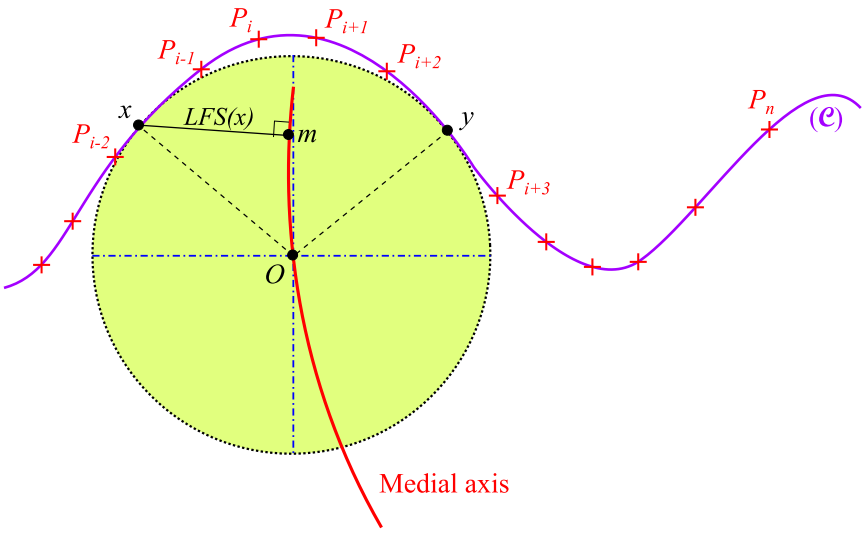
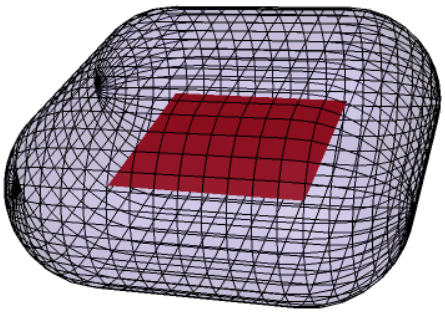


Fig. 104 Star (green triangles) of a point (yellow dot) in a mesh (blue triangles).

A tool to assess topological correctness of a mesh has been introduced by Amenta *et al* [171]. This tool, called the ε-sampling criterion, relies on the knowledge of the theoretical surface that we are trying to reconstruct as it relates notions on its local curvature with notions on the sampling density in order to prove that the reconstructed mesh is a 2-manifold. To clarify this idea, let us define the medial axis of curves knowing that this definition holds also for surfaces (Fig. 105). The medial axis of a curve is the locus of all points, such as O, having more than one closest point (x and y) on the curve. Additionally, the orthogonal distance from a point x on the curve to the medial axis (point m) is denoted as the Local Feature Size of x and written as $LFS(x)$. Dey *et al* [178] prove the existence of a strong correlation between a parametric curve and its medial axis but state that only an approximation of the latter is possible. The exact computation of a medial axis is very difficult.



(a)



(b)

Fig. 105 (a) Medial axis of a curve (red). (b) Medial axis of a surface (red) [179].

Let P be a set of points sampled on the underlying curve denoted by $\mathcal{C}$. The ε-sampling condition is based on the relationship between the sampling density of the underlying curve and its Local Feature Size (LFS). In general, it states that for any point $x \in \mathcal{C}$, there should be a point $p_i \in P$ in the neighborhood of x (35):

$$\forall x \in \mathcal{C}, \exists p_i \in P \text{ such that } \frac{\|x - p_i\|}{LFS(x)} \leq \varepsilon. \tag{35}$$

We can deduce from this inequality that more points are needed where the curvature of the underlying curve is high and less points are needed elsewhere. ε is a real number which characterizes the ε-sampling criterion, thus a small value of ε means that the sampling should be dense. Fig. 106 illustrates this principle on a curve: point x is a point that verifies the ε-sampling condition because x has at least one point in P (in this case two points p_5 and p_6) closer than $r(x) = 0.5 \times LFS(x)$ when $\varepsilon = 0.5$. However, point y does not verify the ε-sampling condition because no point in P is close enough. The sampling in this specific region on $\mathcal{C}$ is not an ε-sampling.

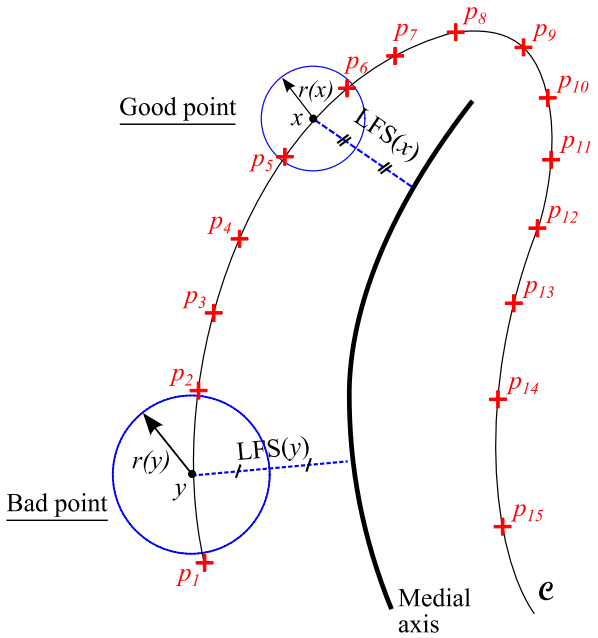


Fig. 106 ε-sampling condition with $\varepsilon = 0.5$. x is a point that respects the ε-sampling condition while y does not.

It is important to note that the maximum distance separating two sample points around any given point $x \in \mathcal{C}$ will not exceed $2 \times \varepsilon \times LFS(x)$. Dey *et al* [180] prove that, generally, a

value of $\varepsilon \leq 0.1$ guarantees that the mesh reconstruction algorithms return a mesh which is homeomorphic to the underlying curve topology. They also demonstrate that all of the above holds for surfaces.

The ε-sampling condition is a necessary but not sufficient condition as we will show in some further examples (c.f. section 4) the ability to perform correct mesh reconstruction without necessarily having this condition fulfilled. Moreover, based on the study that will follow in section 2, we suggest that there should be some lower bound ε_0 to the ε-sampling condition. The value of ε_0 is very difficult to determine but this bound is going to be a major limitation to the mesh reconstruction algorithms due to degenerate Delaunay configurations.

b) Provided that the topological equivalence is verified, some geometrical requirements must be met. Firstly, geometrical equivalence is about proximity and is quantified by the Hausdorff distance [181]. The Hausdorff distance describes the distance that separates two topological spaces, in this case, these spaces are the mesh and the underlying surface which is generally unknown:

$$\delta_H(X,Y) \coloneqq \max\left\{ \max_{x\in X} \min_{y\in Y} d(x,y),\ \max_{y\in Y} \min_{x\in X} d(x,y) \right\}. \tag{36}$$

Let $x \in X$ such that $\operatorname{argmin}_{y'\in Y} d(x,y') = y$, then $\operatorname{argmin}_{x\in X} d(x,y) = x^*$. As we can see in Fig. 107, in general, $x \neq x^*$.

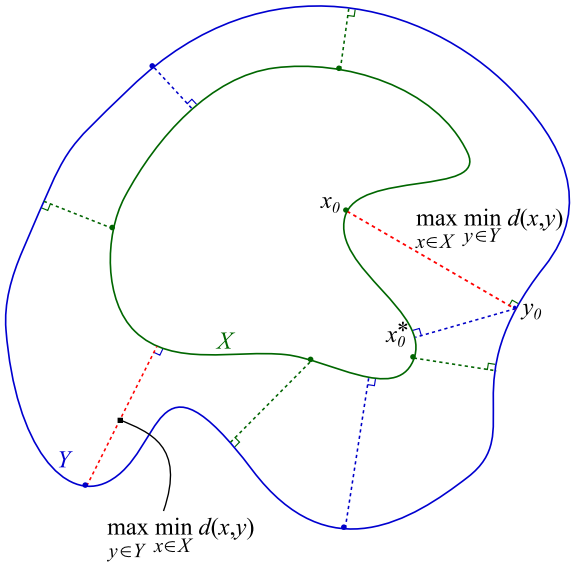


Fig. 107 Hausdorff distance between two spaces X and Y. The distance is the longest among both dashed red lines.

Geometrical equivalence is also about mesh orientation. Each facet of the mesh should necessarily have a normal orientation that is very close to the orientation of the normal of the underlying curve at the same location (Fig. 108). Each facet corresponds to a portion of the curve so the average normal orientation n_s of that portion and the normal orientation of the corresponding facet n_m should be within an angle difference proportional to ε. Dey *et al* [180], demonstrate that this limit angle must satisfy the following inequality:

$$\angle(n_s, n_m) \leq 2.\sin^{-1}\left(\frac{\varepsilon}{1-\varepsilon}\right). \tag{37}$$

This inequality can differ from one algorithm to another, but the important thing to keep in mind is that the normal orientation condition is a function of ε.

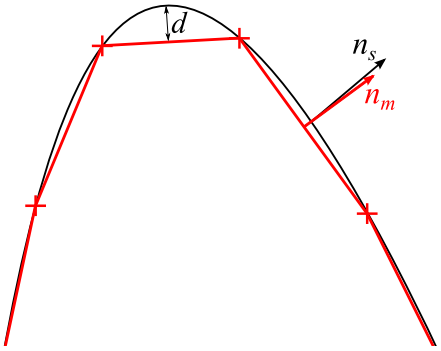


Fig. 108 Geometrical equivalence between a mesh (red segments) and the underlying curve (black). d is the chord error between the curve and the mesh. Vectors n_s and n_m are the normal directions to the curve and the mesh respectively.

2- Robustness is the capability of a result to remain unchanged even in the presence of added disturbances or noise in the input. This criterion guarantees that the algorithm can reconstruct a surface from noisy point clouds that might also contain outliers and be randomly scattered in 3D space. We also refer to robustness as in regard to sampling, meaning that the algorithm should also be robust to various sampling conditions.

3- Complexity of an algorithm is expressed in function of the size of the input data, the algorithm's instructions and the size of the output. Complexity is measured by two quantities, computational time and memory storage.

1. Combinatorial structures

The first family of surface reconstruction techniques is based on the Voronoi and Delaunay structures. These two dual representations are suitable for data structuring. The fundamentals of these combinatorial structures have been initiated at the beginning of the 20^{th} century through the work of Gregory Voronoi in 1908 followed by the work of Boris Delaunay in 1934 [182]. The Voronoi diagram delimits space with respect to a neighborhood criterion; whereas the Delaunay triangulation is a particular triangulation of the point set which connects points that are exclusively neighbors among themselves.

1.a Voronoi Diagram

Constructing the Voronoi diagram of a point set P is the process of dividing space into subspaces according to the concept of neighborhood as described in (38). Each point is allocated its exclusive subspace called the Voronoi cell. The Voronoi cell of a point $x \in P$ is the subspace in E^k that covers all points p that are closer to x than any other point q is.

$$Vor(p) = \{x \in E^k | \, d(x,p) \leq d(x,q), \forall q \in P\} \tag{38}$$

The graphical representation of the Voronoi diagram is illustrated in the two dimensional Euclidean space (Fig. 109). The diagram is generated using the Matlab function *voronoi*. The blue graph is the Voronoi diagram of the point set P. The properties of this diagram are such that a Voronoi edge is the bisector of the segment formed by the couple of points in P which are delimited by this edge according to a notion of neighborhood. For instance, the Voronoi edge that passes through A in Fig. 110a delimits space between points P_1 and P_2, which is in fact the bisector of segment $[P_1P_2]$. Furthermore, a Voronoi vertex, i.e. the intersection of three Voronoi edges, is equidistant from three points in P and is therefore the center of the circumscribed circle to these three points. In Fig. 110b, Voronoi vertex B is equidistant from P_1, P_2 and P_3.

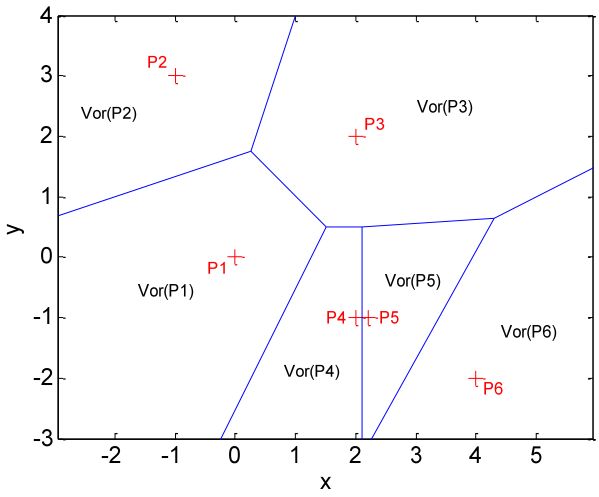
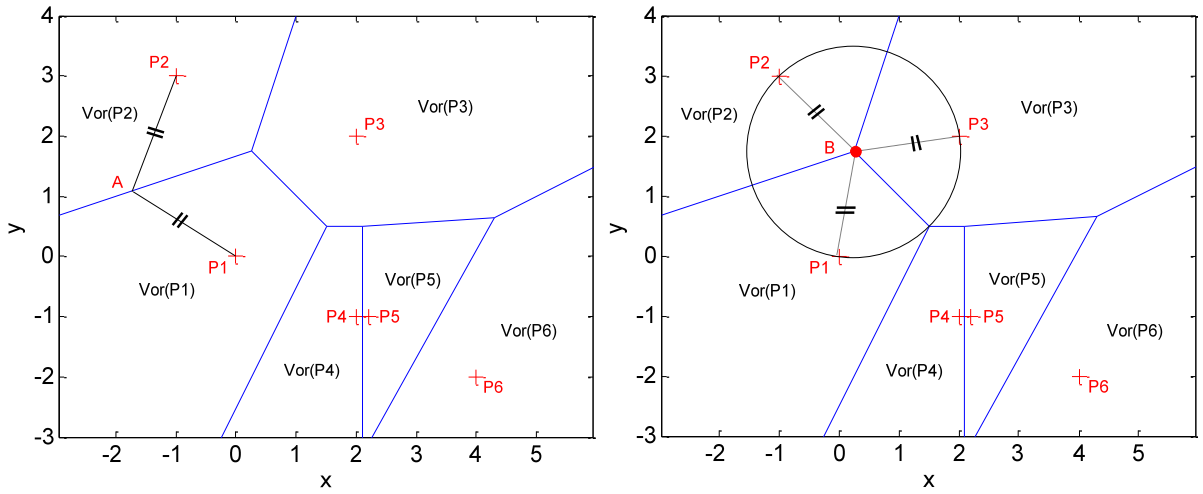


Fig. 109 The Voronoi diagram of a set of points in the plane ($Vor(P)$: Voronoi cell of P).



(a) (b)

Fig. 110 Voronoi properties: (a) a point on a Voronoi edge A is equidistant from the adjacent data points P1 and P2; (b) a Voronoi vertex B, center of a circumscribing circle passes through 3 data points P1, P2 and P3.

The restricted Voronoi diagram of a set of points is a subset of the Voronoi diagram. It is obtained by adding a restriction to the initial Voronoi diagram. Considering the subspaces $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$ in Fig. 111, a Voronoi diagram is said to be restricted to these subspaces if what belongs to the Voronoi representation is intersected by $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$. Here, all cells $Vor(p_1)$, $Vor(p_2)$, $Vor(p_3)$ and $Vor(p_4)$ intersect with the subspace $\mathbf{X_2}$. This implies that these four Voronoi cells belong to the restricted Voronoi diagram of the subspace $\mathbf{X_2}$.

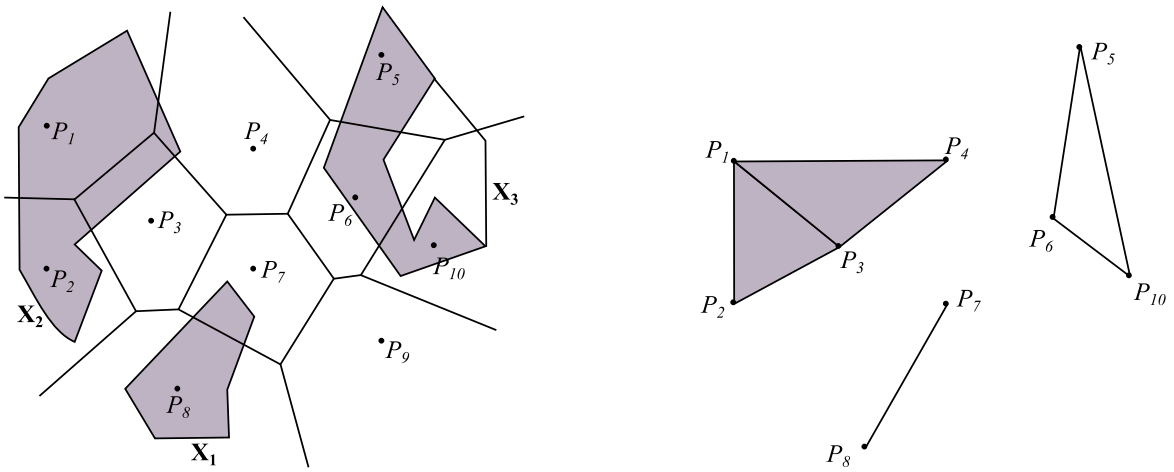


Fig. 111 Restricted Voronoi diagram (left) and corresponding Delaunay graph (right) (adapted from [180].

As for the weighted Voronoi diagram, also called the power diagram, is a Voronoi delimitation of the space of a point-set, with the additional restriction of assigning weights to points. In fact, weighted points attract more or less large neighborhood area depending on their value and the Voronoi diagram is a combinatorial structure of these points. The complexity to construct such a structure for N points is of the order of $O(N^2)$ in the worst case.

$$Vor(p) = \{x \in E^k | \, \pi_p(x) \leq \pi_q(x), \forall q \in P\} \tag{39}$$

1.b Delaunay triangulation

The Delaunay triangulation is a dual representation of the Voronoi diagram (Fig. 112). In 2D Euclidean space, the Delaunay triangulation is a 2-simplicial complex formed of simplices of lower or equal dimension. For instance, a 2D Delaunay triangulation is composed of vertices (0-simplex), edges (1-simplex) and triangles (2-simplex). The process of going from the Voronoi diagram to the dual Delaunay representation is straight forward and inversely. For the general case of space dimension n, duality suggests that a 0-simplex in Voronoi corresponds to a n-simplex in Delaunay, and a 1-simplex in Voronoi corresponds to a $(n-1)$-simplex in Delaunay, etc…

A Voronoi vertex is equidistant to exactly three points in a 2D point-set and to exactly four points in 3D. Among themselves, those points form a Delaunay triangle in 2D and a

Delaunay tetrahedron in 3D, respectively. In other words, each Voronoi vertex is the center of the circumscribing circle to a Delaunay triangle, and inversely, each Delaunay triangle connects three points which are exclusively neighbors among themselves and belong to the circumscribed circle of that triangle (Fig. 112). It is key to mention that the circumscribed circle of a Delaunay triangle is empty of other points in the dataset. The Delaunay triangulation can be directly constructed given a certain point-set as it can be seen as the triangulation of its convex hull (Fig. 112).

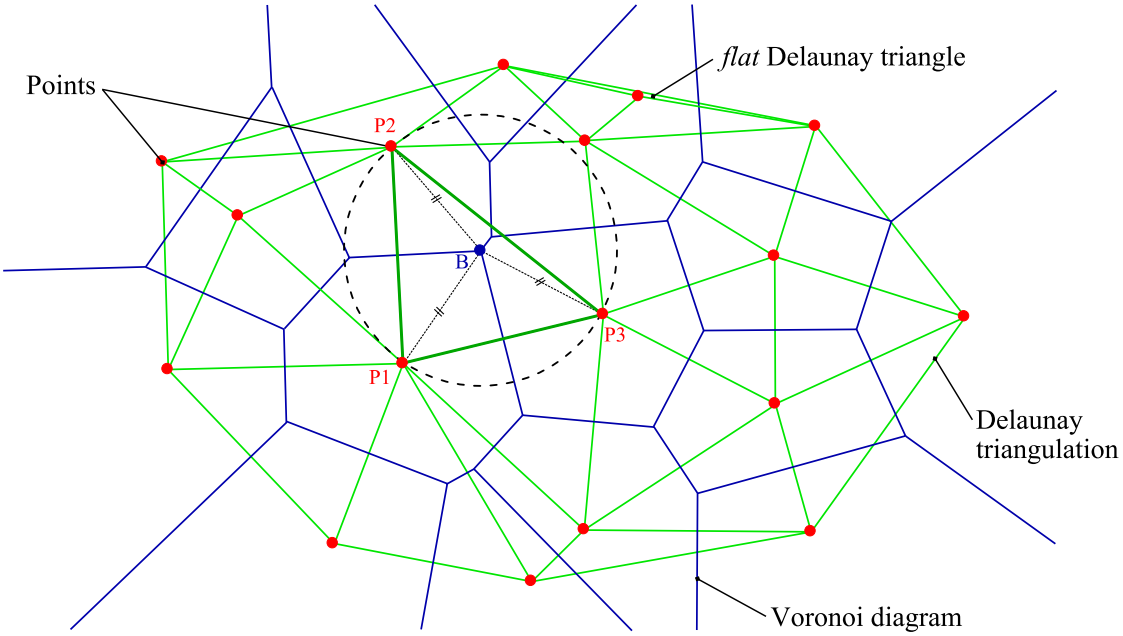


Fig. 112 2D Delaunay triangulation (green) and Voronoi diagram (blue) of points in the plane (red) [183]. The highlighted triangle is inscribed in the circle (dashed) centered at a Voronoi vertex B

In some cases, the Delaunay triangulation algorithm can be degenerate (Fig. 113) or manifest non-uniqueness of a solution. Degeneracies occur when more than three points in 2D are aligned or more than four points in 3D are co-planar [184]. We shall extend this notion of degenerate Delaunay simplices to what are called *flat* Delaunay triangles in 2D and *flat* Delaunay tetrahedra in 3D [185].

Non unique solutions occur when more than four points in 2D are co-circular or more than five points in 3D are co-spherical. Non-uniqueness is taken care of the recent Delaunay algorithm by optimizing for the shape regularity of the triangles/tetrahedra. The solution across all combinations is chosen so that the largest angle in each triangle/tetrahedron is minimal. The algorithm used to allow for such flexibility in the Delaunay triangulation is the

flip algorithm where edges can be flipped over and over to reach a state which satisfies the regularity condition [186]. For the case shown in Fig. 114, solutions (c) and (d) have more regular triangles with more homogeneous angles.

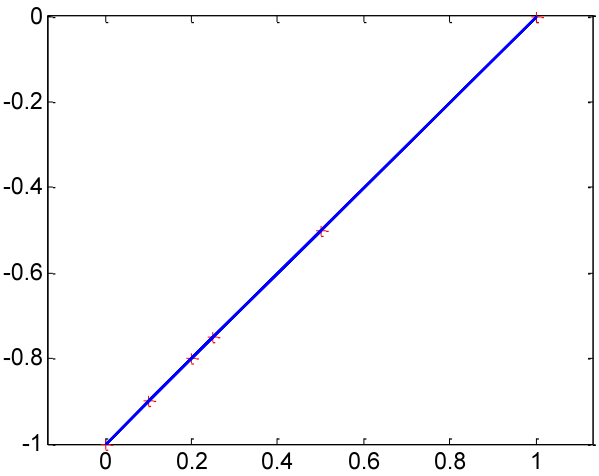
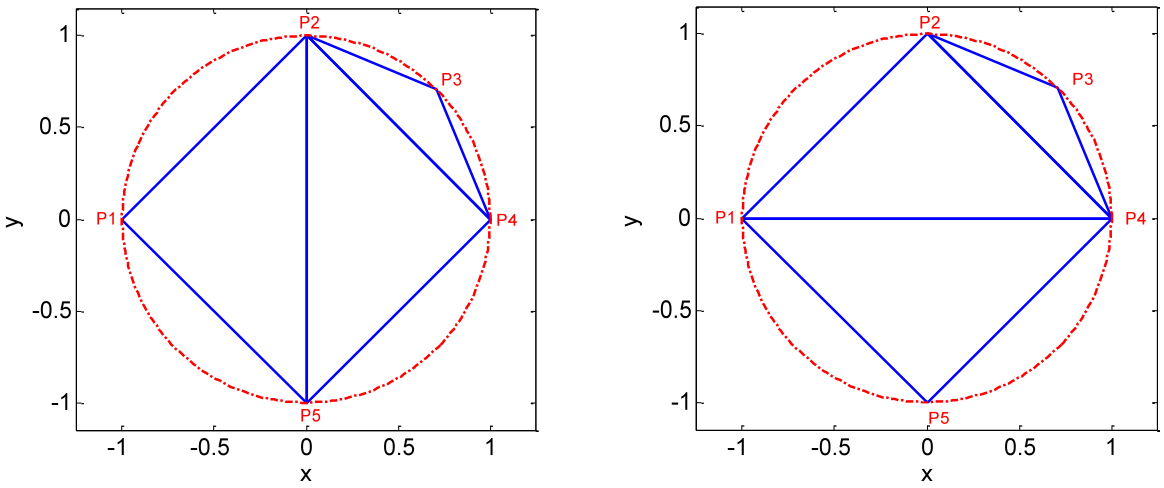


Fig. 113 Degenerate points (red) for a Delaunay triangulation (blue).



(a) (b)

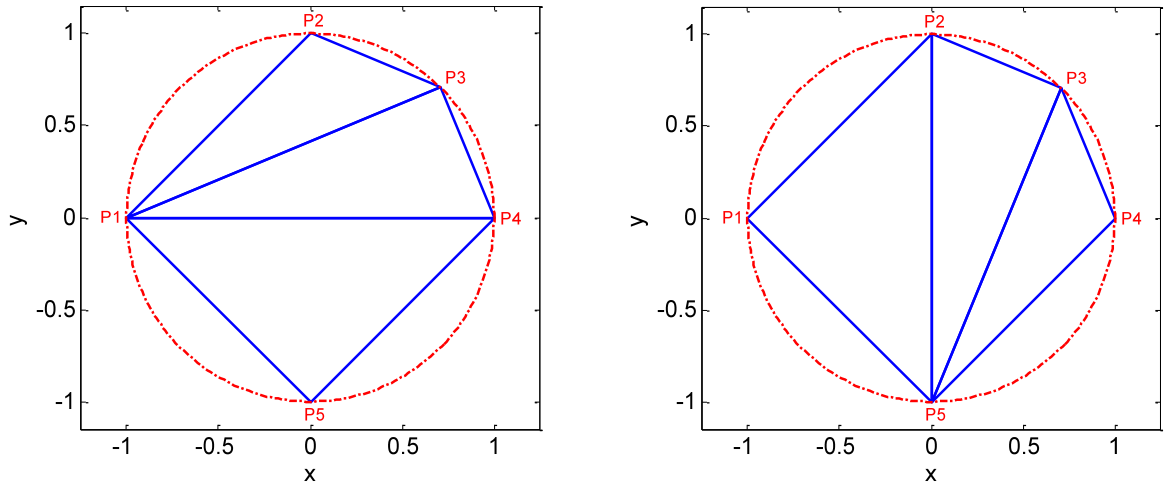


Fig. 114 Non unique solutions for 2D Delaunay (blue triangulations) when more than 4 points (P1, P2, P3, P4 and P5) are co-circular.

The restricted Delaunay triangulation (Fig. 111) is the dual representation of the restricted Voronoi diagram. From a surface reconstruction point of view, the restricted Delaunay triangulation represents the triangles which are restricted to the surface sought. However, in surface reconstruction applications, the surface is not known beforehand, and therefore the restricted Delaunay triangulation cannot exactly be built. Indeed, all mesh reconstruction algorithms seek to approximate the restricted Delaunay triangulation of the underlying surface by means of some assumptions and heuristics which will be presented in section 2.

The complexity of an algorithm reflects the time, in seconds, it would need to run and the space, in units of memory, it would need to use for storage. The complexity depends on all, the size of the input, the size of the output and the number of operations to perform during runtime [187]. The optimal algorithm for the 3D Delaunay triangulation is $O(N.logN)$, the worst case being $O(N^2)$ where N is the number of input data. In a more general case, and according to [188], Delaunay complexity in d-dimensions is as given in (40).

$$O(N.logN + N^{\frac{d}{2}}) \tag{40}$$

Optimality can be reached when either the *incremental* or the *divide-and-conquer* algorithms is used, but only for specific sampling conditions as described in [189], [190] and [191].

An incremental Delaunay triangulation algorithm is proposed in [192] for the reconstruction of surfaces embedded in spaces of arbitrary dimensions. CGAL libraries also offer an incremental Delaunay triangulation algorithm in two and three dimensional spaces (Fig. 112 and Fig. 115).

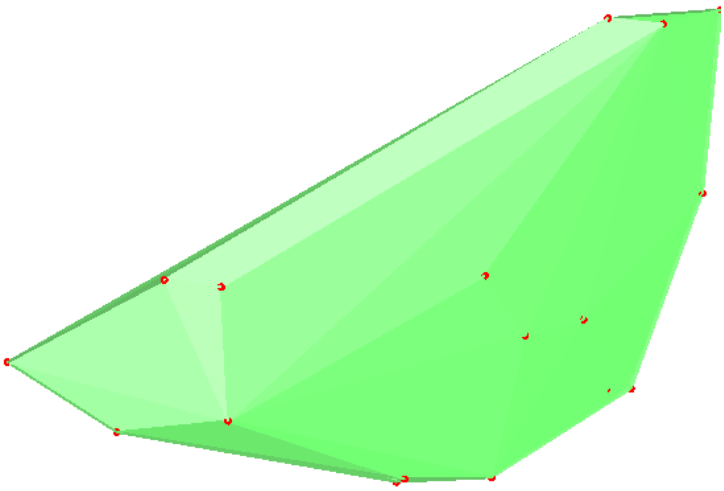


Fig. 115 3D Delaunay triangulation (green) of points in the space [183].

1.c Gabriel graph

The Gabriel graph is another graph for data points. It is not a basis for surface reconstruction algorithms but a tool serving as a criterion or inspiration to some of the surface reconstruction algorithms such as the geometric convection algorithm developed by Chaîne [173]. The Gabriel graph is also a graph linking points based on neighborhood. However, unlike the notion of neighborhood of the Voronoi diagram, the points that are linked together are the points that form a segment defined as the diameter of an empty circle in 2D (Fig. 116b); or the points that form a triangle defined as the inscribed triangle of the great circle of an empty sphere in 3D. For the sake of simplicity, consider points in Euclidean space E^2 and then refer to the intuitive algorithm of Gabriel graph as the process of passing a circle having the segment, linking each couple of points, as its diameter. The boundary δC of C passes by 2 points, say p_i and q_i. If the circle is empty of other points in the dataset, then the segment $p_i q_i$ belongs to the Gabriel graph (Fig. 116b). The Gabriel graph is a subset of the Delaunay triangulation of a given dataset. It is therefore sufficient to pass a circle through Delaunay segments only.

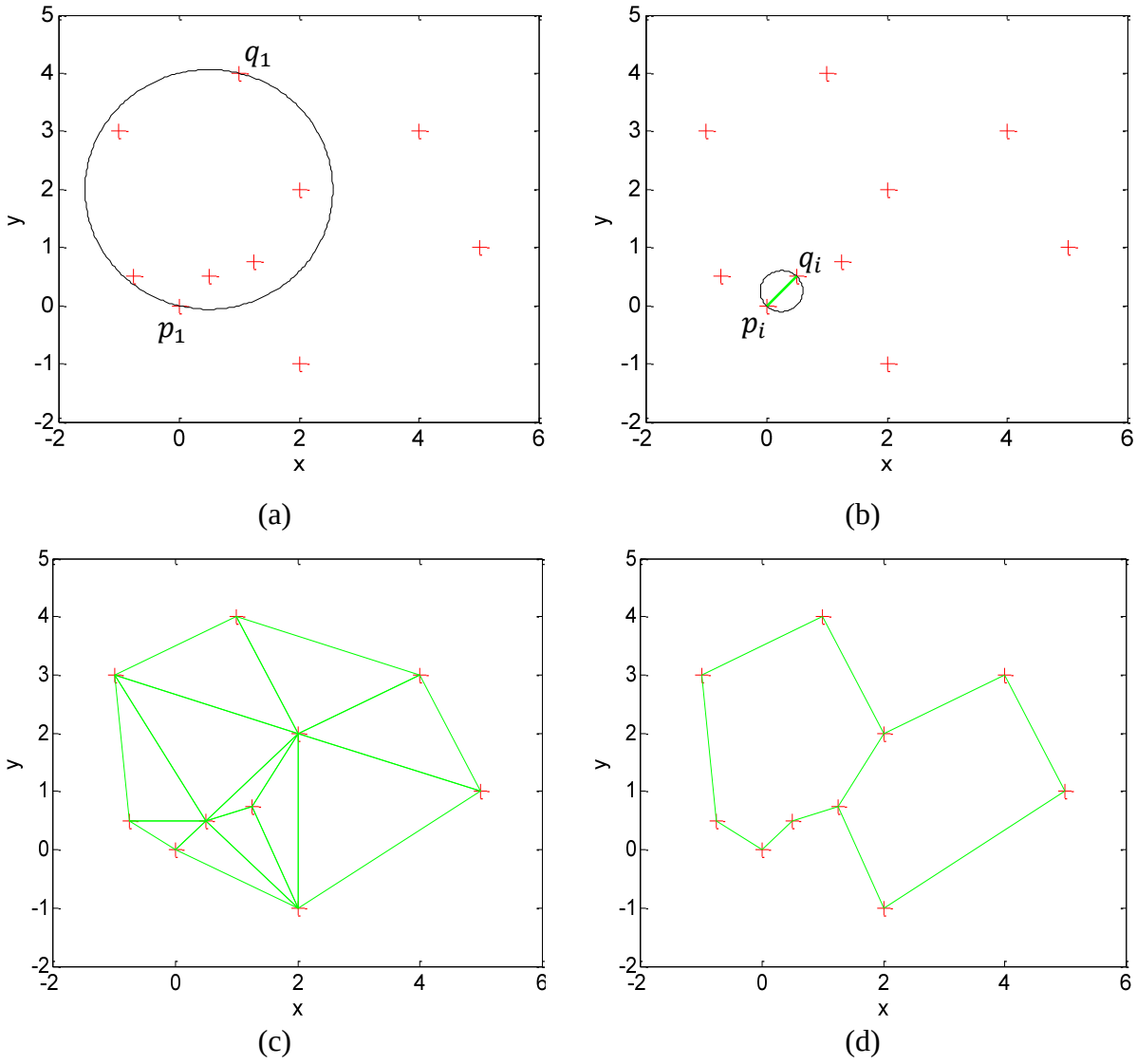


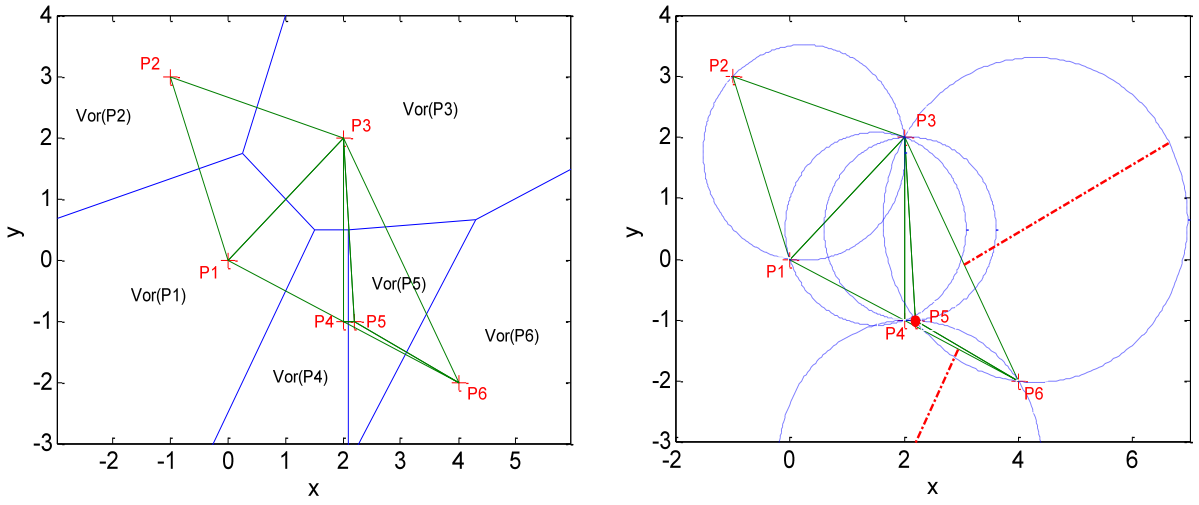
Fig. 116 (a) A non empty circle of diameter p_1q_1; (b) an empty circle of diameter p_iq_i; (c) Delaunay triangulation; (d) The Gabriel graph, a subset of the Delaunay triangulation: only Delaunay edges are considered as diameters of Gabriel circles.

2. Mesh reconstruction by combinatorial structures

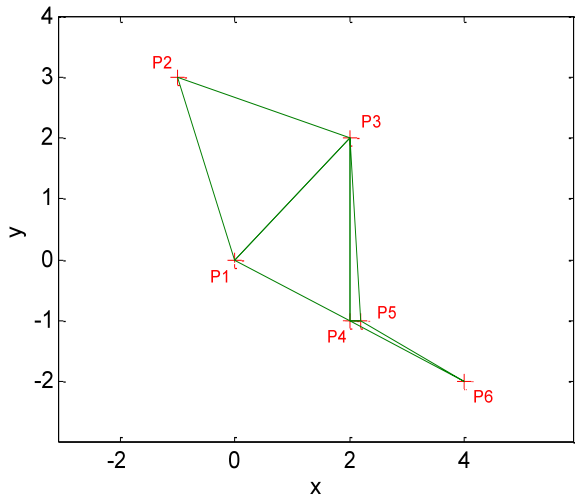
The mesh reconstruction algorithms that will be presented here rely on the construction of the Voronoi diagram and the Delaunay triangulation. These algorithms are studied and compared in the aim of assessing their capacities and eventually applying them to measured data of aspherical and freeform surfaces.

2.a Sculpture

The sculpture algorithm is among the first algorithms in the history of surface reconstruction and computer graphics. Boissonnat in [193] uses a volumetric approach to the problem by starting with the 3D Delaunay triangulation of the points. The basic idea behind this algorithm is that points must all belong to the boundary of the end-result convex polytope (CP). A polytope is a geometric object with flat sides and is a term that refers to a generalized form of polygons to any dimension. In 3D, a polytope is a polyhedron. In order to maintain the shape as a polytope, tetrahedra having at least one vertex inside the convex hull of the points and the greatest circumscribing sphere at the same time must be sculpted out. The condition to be satisfied is that CP must remain a polytope all along the iterations. It is to note that the interior of the convex hull is also the interior of the Delaunay triangulation. The remaining tetrahedra approximate the shape of the measured object. Boissonnat proved that performing Sculpture comes back to the following rule [193]: "The only tetrahedra that can be eliminated are those with exactly one face, three edges and three points on CP, or those with exactly two faces, five edges and four points on CP". In 2D, the idea behind the algorithm goes back to searching for interior points to the Delaunay complex and testing incident facets (Fig. 117a). When an interior point is detected, all incident facets/triangles which have at least one edge belonging to CP are examined and the triangle with largest circumscribing circle is eliminated (Fig. 117b). The resulting shape is a volumetric approximation of the measured object (Fig. 117c).



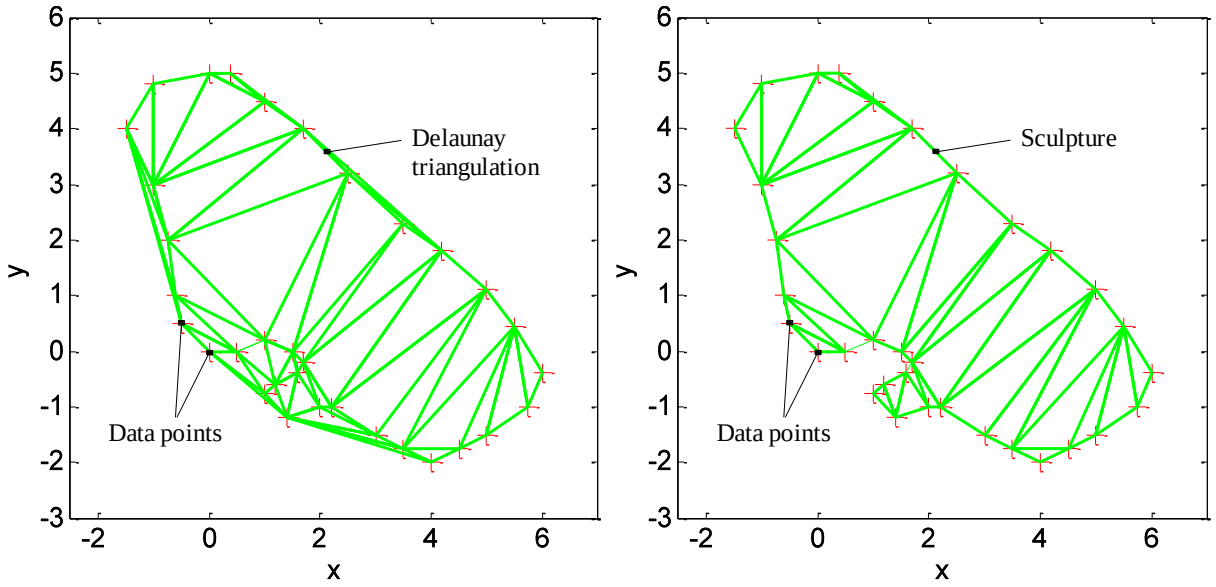
(a) (b)



(c)

Fig. 117 Sculpture algorithm. (a) Delaunay triangulation; (b) Detection of an interior point in the Delaunay triangulation; (c) Elimination of the incident triangle with largest circumscribing circle.

In order to compare the algorithms we are presenting in this section, we consider a simple closed curve represented by points in the plane. The Sculpture algorithm is firstly applied and the result is shown in Fig. 118. As we can see in this example, the interior point criterion is not sufficient for having a proper mesh reconstruction because it returned a rather volumetric reconstruction.



(a) (b)

Fig. 118 Sculpture (green) applied on a closed curve represented by the red points. (a) the Delaunay triangulation of the points; (b) the mesh reconstruction containing a lot of interior segments.

2.b Alpha Shapes

The method of α-shapes introduced by H. Edelsbrunner and E. Mücke [185] is inspired by the work of Boissonnat discussed in the previous paragraph. α-shapes, however, do not seek to build a volumetric mesh on the points like the sculpture algorithm but the boundary of the volume that would be the boundary of the shape sought. α-shapes are a generalization of the convex hull of the points and the shape retrieved is denoted by S_α. S_α is not necessarily manifold nor convex, and can contain simplices of lower dimensions than the ambient dimension of the points' space.

The idea behind the α-shapes is intuitive. Not only can it be seen as a surface reconstruction algorithm, but also as a morphological filtering technique. In other words, a parameter α represents the diameter of a sphere that must be passed across triangles and be maximal, meaning that it must be empty of other points in the set when passing through 3 points. We refer to the same example mentioned by Edelsbrunner and Mücke in [185]. Imagine that the space of the points is a mass of ice-cream and that the points are chocolate pieces. Say the chocolate grains are fixed and cannot move or be removed and that ice-cream can be carved away using an unchanging spoon diameter. The algorithm of α-shapes is pretty much like carving away as much ice-cream as possible, emptying the space around the chocolate grains. By analogy, α-shapes seek to keep triangles and segments that have a circumscribing sphere empty of other points in the dataset. For a k-simplicial complex, k-simplices are examined. Fig. 119 clarifies how simplices are kept or rejected. In 2D, a circle C is passed through edges. The boundary of C contains points which form the k-simplex σ_T studied. If C is empty of other points, σ_T is said to be α-exposed. The graph obtained after passing through all simplices contains all and only α-exposed simplices.

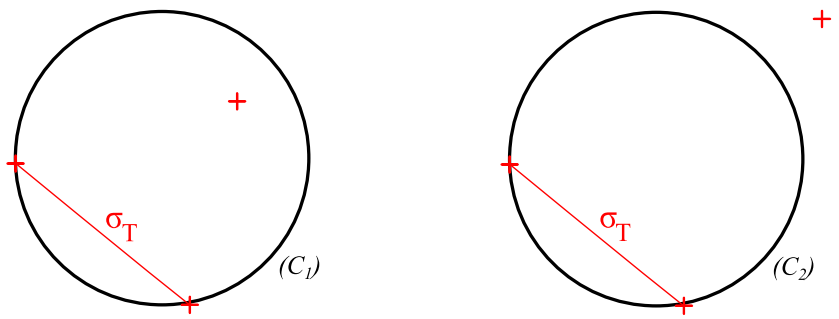


Fig. 119 α-shape condition featuring a 1-simplex σ_T (red segment). (a) non α-exposed simplex; (b) α-exposed simplex. (red points belong to the dataset)

The α-shape of a point-set can be seen as a sub-complex of the Delaunay triangulation. Therefore, a more efficient and simpler approach is to solve for the α-shape by extracting α-exposed simplices from the Delaunay triangulation of the dataset. The simplices that are tested are $(k-1)$-simplices of the Delaunay triangulation. A sphere of pre-defined diameter α is passed through Delaunay triangles (not tetrahedra) and removes non α-exposed simplices. The value of α is chosen to be the optimal value calculated based on as many intervals as there are $(k-1)$-simplices in the Delaunay complex. Each Delaunay triangle defines an interval for which, a sphere, of a certain diameter value within the interval, is empty of points in the dataset. The combination of all intervals gives the best estimate for α. Remark that the computed α might not satisfy all intervals. In fact, this explains why, most of the time, an α-shape gives a non-manifold surface. Once α is chosen the intervals determined for each σ_T make the algorithm directly infer the class of a simplex. First, let's define the α-complex, a simplicial complex deduced from the Delaunay triangulation.

Let Γ_T be a k-simplex ($k = 1, 2, 3$) of the Delaunay triangulation, and r_T the radius of the sphere b_T containing Γ_T. Let all the spheres be maximal and define α as the smallest value corresponding to the smallest radius of all the Delaunay spheres. The α-complex C_α is the collection of simplices which belong to $G_{k,\alpha}$ and the ones which are incident to $(k+1)$-simplices of C_α.

k	3	2	1
b_T	Sphere containing Γ_T	Great circle of b_T containing Γ_T	Circle with diameter Γ_T
$G_{k,\alpha}$	All tetrahedra with $r_T < \alpha$	All triangles with $r_T < \alpha$	All edges with $r_T < \alpha$

According to the classification proposed by Edelsbrunner and Mücke [185], the simplices are classified such that, a k-simplex σ_T that belongs to the α-complex C_α is said to be:

1. Interior if $\sigma_T \notin \delta S_\alpha$
2. Regular if $\sigma_T \in \delta S_\alpha$ and is incident to 2 higher dimension simplices of C_α
3. Singular if $\sigma_T \in \delta S_\alpha$ but is not incident to higher dimension simplices of C_α

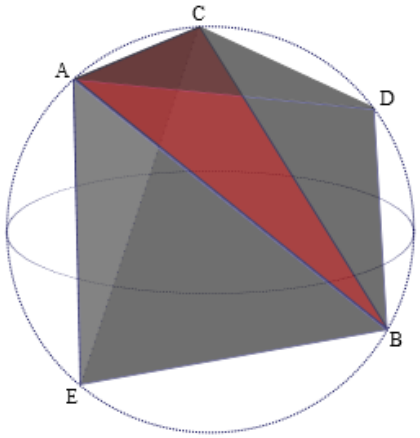
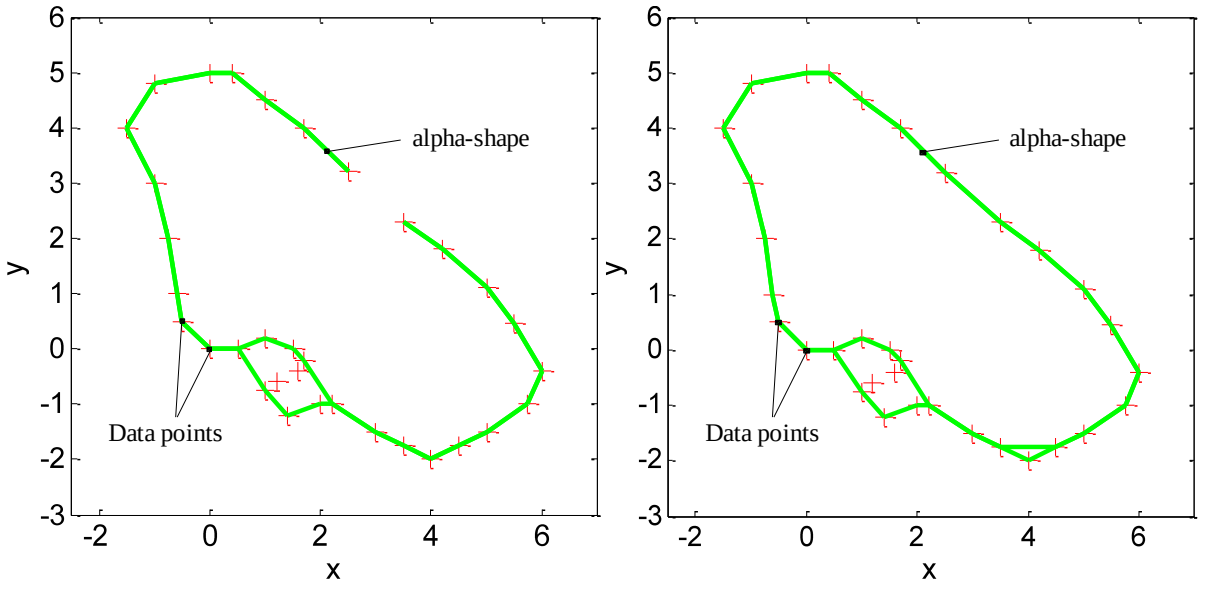


Fig. 120 $(k-1)$-simplex as a common face between 2 k-simplices

As shown in Fig. 120, for a $(k-1)$-simplex such as $\sigma_T \notin Del(P)$ which is nothing but triangle ABC, the algorithm seeks to determine whether σ_T is interior, regular or singular. σ_T is adjacent to two k-simplices, tetrahedron ABCE and tetrahedron ABCD. By computing the circumscribing spheres to each of the tetrahedra, the respective radii of those spheres can be deduced. Refer to ABCD as σ_{T_1} and to ABCE as σ_{T_2} and their respective radii as r_{T_1} and r_{T_2}. Suppose that $r_{T_1} < r_{T_2}$.

- Case 1: $\alpha > r_{T_2} \rightarrow \sigma_{T_1} \in C_\alpha$ and $\sigma_{T_2} \in C_\alpha$, and σ_T is not α-exposed so it cannot belong to the α-shape being an interior simplex.
- Case 2: $r_{T_1} < \alpha < r_{T_2} \rightarrow \sigma_{T_1} \in C_\alpha$ but $\sigma_{T_2} \notin C_\alpha$, and σ_T is α-exposed so it belongs to the α-shape being a regular simplex (σ_T delimits $\sigma_{T_1} \in C_\alpha$).
- Case 3: $\alpha < r_{T_1} < r_{T_2} \rightarrow \sigma_{T_1} \notin C_\alpha$ and $\sigma_{T_2} \notin C_\alpha$. What determines the class of σ_T is the radius r_T of its circumscribing circle. Here two cases are possible: if $r_T < \alpha$, then σ_T is α-exposed and belongs to the α-shape being a singular simplex ($\sigma_{T_1} \notin C_\alpha$ and $\sigma_{T_2} \notin C_\alpha$). σ_T is not α-exposed otherwise.

The simplices constituting the α-shape are the regular and singular simplices which are α-exposed. The application of the algorithm on the same previous curve example is plotted in Fig. 121 and starts with the Delaunay triangulation of the points (Fig. 118a). The best value for α in this example is equal to 0.7 and is selected after trying out multiple values. The reconstructed mesh is in all cases non-manifold and for low values of α (Fig. 121a), not continuous.



(a) (b)

Fig. 121 α-shape (green) applied on a closed curve represented by the red points. (a) for $\alpha = 0.6$; (b) for $\alpha = 0.7$.

Provided that an optimal value for α is hardly found, and that finding it requires user intervention, the α-shape's algorithm cannot be made automatic. Finding a global α-interval satisfying all individual α-intervals of every simplex is practically unachievable. The output mesh has locally-manifold zones where α is within the α-intervals of the simplices belonging to this zone.

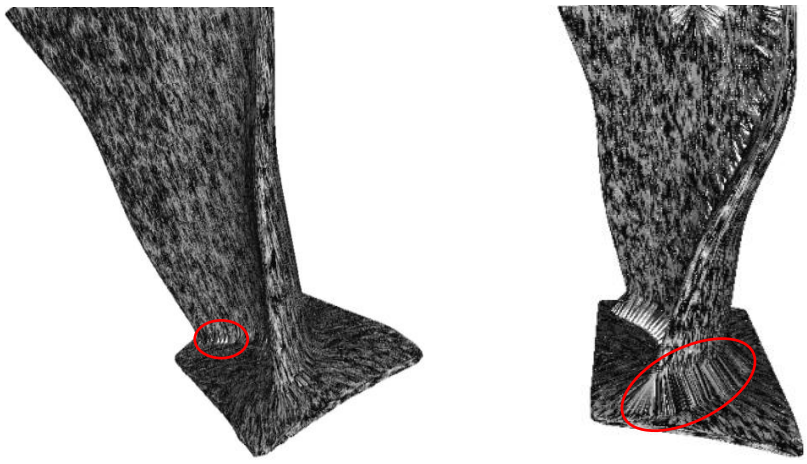
The complexity of the α-shape's algorithm is also considerable as it depends on three algorithm key steps: the Delaunay triangulation of the points, the search for α-intervals of all Delaunay k-simplices and the sorting of these intervals and the extraction of α-exposed simplices. All three steps have complexity $O(N^2)$ in the worst case where N is the number of points in the dataset. Therefore, for a large dataset of some hundred thousands of points, the α-shape algorithm is demanding in terms of units of computational time and memory storage. Finally, this algorithm presents no guarantees, meaning that no theory proves any correct reconstruction (no relation to the ε-sampling condition).

α-shapes have seen improvements, yet, guarantees are still lacking. Some works have been done on elliptical α-shapes and conformal α-shapes [194]. Non-spherical α-shapes provide a higher flexibility on individual α-intervals, and by that, make them span larger ranges. Elliptical α-shapes increase the likelihood of finding a value for α which covers more

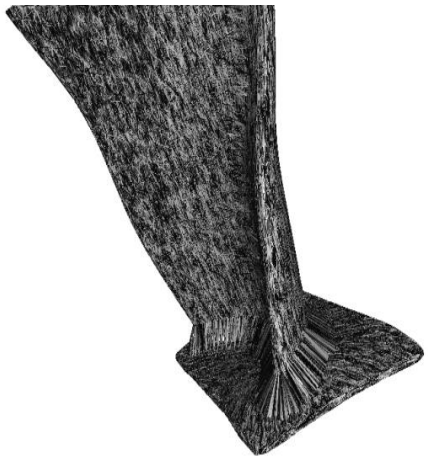
α-intervals than regular α-shapes do. Another step up on α-shapes is the Ball Pivoting Algorithm developed by Bernardini *et al* [195].

α-shapes are sometimes used as a filtration technique such as stated by Zhang *et al* in [25]. α-shapes can indeed serve as a filter for point-sets according to points spacing. For a given value of α, a sphere filters out points that come out to be inside the sphere when it is passed through three points.

The α-shapes algorithm is implemented in MeshLab 1.3 as well as in CGAL libraries [183,196]. Some applications of the α-shapes in MeshLab are shown in Fig. 122. The machine used has a 2.1 GHz 8-cores processor and 2 Gb of RAM. The α-shape reconstruction of the turbine blade dataset of some $127{,}000$ points is done in about 25 s. The figure shows three different reconstructions for three different values of α all chosen by the user. Fig. 122b illustrates the α-shape with $\alpha = 0.5$. The value is very close to the samples density in the z-direction and that is why it outputs a relatively fine mesh with some missing triangles. Fig. 122c takes a large value of $\alpha = 1.5$. The output mesh exhibits non-fine regions encircled in red. The reason is that the value of α is very large and that the sphere of diameter α cannot fit to the samples density in those regions, i.e., the sphere touches distant points and cannot go any further where points are closer because the sphere would contain other points and the simplices under test would be non α-exposed.



(a) (b)



(c)

Fig. 122 α-shape reconstruction of the blade using MeshLab. (a) for $\alpha = 0.5$ (missing triangles); (b) for $\alpha = 1.5$ (large sized triangles, i.e. bad connectivity); (c) for $\alpha = 1$.

2.c Crust

The Crust and all of the improvements of this technique have been developed by Amenta *et al* [170,171,179,197–199]. The Crust is another mesh reconstruction algorithm based on the Voronoi diagram and the Delaunay triangulation. This algorithm is the first to give geometrical and topological guarantees derived from mathematical proof based on the medial axis and the ε-sampling condition [171]. Recalling the definition of the medial axis of a curve/surface, it is the locus of the centers of circles/spheres that are tangent to the curve/surface in two or more points (Fig. 105). The particularity of those centers is that they belong to the Voronoi diagram of the dataset (Fig. 123). They are in fact some of its Voronoi vertices and the centers of circles/spheres circumscribing Delaunay k-simplices. This intrinsically means that some Delaunay $(k-1)$-simplices which are facets of Delaunay k-simplices constitute the restricted Delaunay triangulation of the underlying curve/surface. Consequently, by using an approximation of the medial axis given by a subset V_P of Voronoi vertices, the Crust algorithm computes an approximation of the object being discretized.

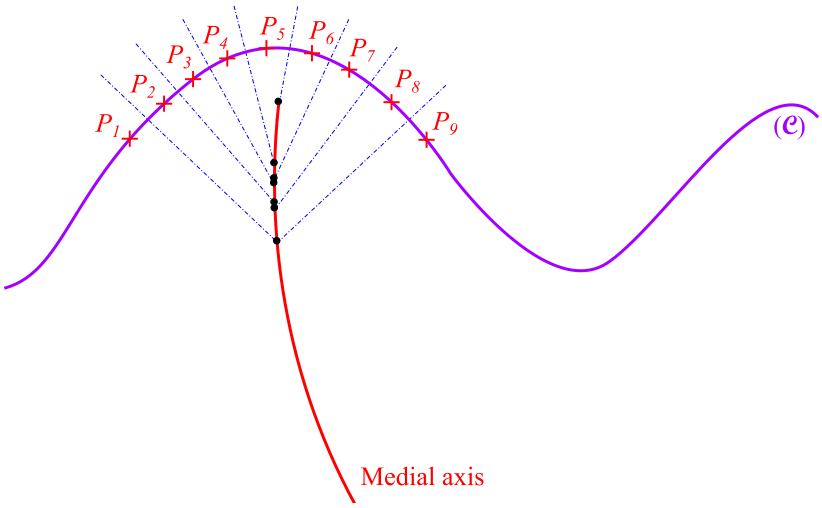


Fig. 123 Medial axis of a curve approximated by Voronoi vertices.

In 2D, all the Voronoi vertices approximate the medial axis therefore they all belong to V_P, but in 3D, only part of them does. In this case, Amenta *et al* [179] introduced a method to identify those vertices, called the poles of P, and state that they are the set of couples of farthest Voronoi vertices from $p \in P$, located on opposite sides in the Voronoi cell of p (Fig. 124). The poles are denoted by p^- and p^+.

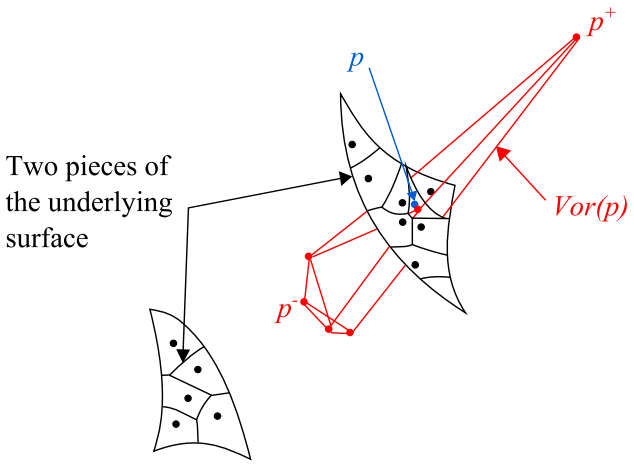


Fig. 124 A 3D Voronoi cell $Vor(p)$ of a point p with the poles p^- and p^+, one on each side.

The Crust is tested on the same set of points representing a closed curve to assess its outcome. The result is plotted in Fig. 125d after showing the major steps of the algorithm (Fig. 125a, b and c). The algorithm starts with the Voronoi diagram of the cloud of points P which is already implemented in CGAL and Qhull libraries [183]. Then the union of P and the Voronoi vertices V_P make up an augmented point-set $P_{aug} = \{P \cup V_P\}$ and the Delaunay

triangulation of P_{aug} is constructed. The Crust proceeds by keeping the Delaunay $(k-1)$-simplices whose vertices are points of P (Fig. 125c).

This procedure does not necessarily lead to a proper reconstruction, therefore an additional step cleans the mesh from eventually remaining non-manifold elements by checking out triangles which are badly oriented according to the geometrical equivalence condition on normal directions. The normal to the surface is approximated by the poles (Fig. 124) and the vector $\overrightarrow{pp^+}$ is a good indication of the normal orientation at point p on the surface. So if the angle between the normal to a given triangle and the normal to the corresponding p is greater than the limit angle, the triangle is suppressed.

The Crust algorithm requires a Voronoi diagram construction and another Delaunay triangulation. Both are equivalent in complexity since they are dual representations. So the overall complexity of Crust is $O(N^2)$.

1. Voronoi diagram	$O(N^2), N$ points
2. Identification of poles and normal approx.	$O(\log m), m = 2N$ poles
3. Delaunay Triangulation	$O(N^2), 3N$ points
4. Normal filter	$O(T), T$ triangles

Table 19 Complexity of the Crust algorithm's steps.

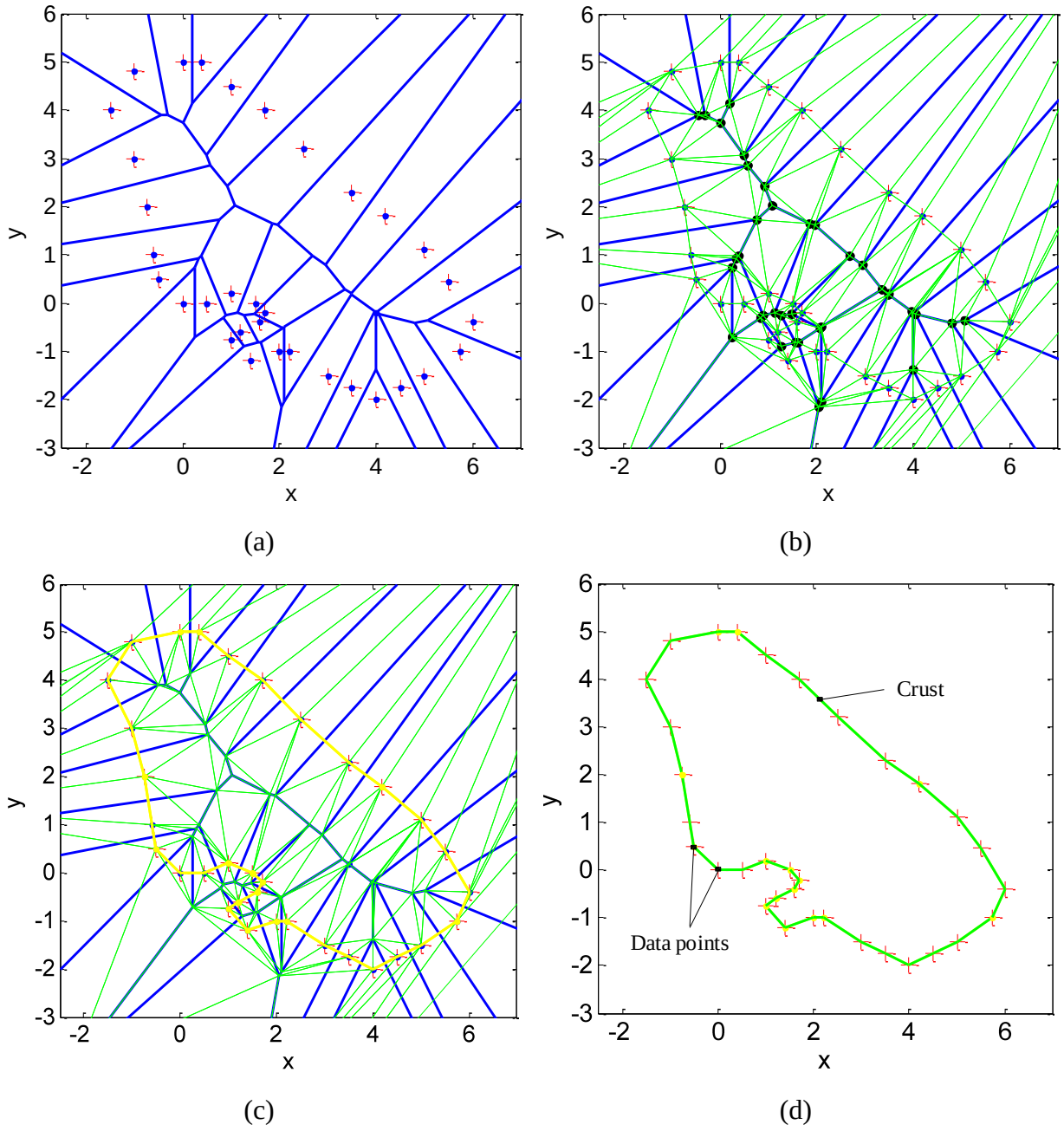


Fig. 125 (a) Points (red) and their Voronoi diagram (blue). (b) Delaunay triangulation (green) of the union of the points (red) and the Voronoi vertices (black). (c) Crust, a subset of Delaunay (highlighted in yellow). (d) Crust of a closed curve in 2D (green) without any interior segments.

The outcome of the Crust mesh reconstruction on the example of Fig. 125 is correct as the output approximation curve is 1-manifold when the ε-sampling condition is satisfied as well as its lower bound ε_0. Fig. 126 illustrates a counter example where ε_0 is not satisfied leading to the formation of *flat* Delaunay tetrahedra. The problem with *flat* tetrahedra is that more than one facet of a tetrahedron are almost coplanar so they all approximate the restricted Delaunay of the curve and are all output in the mesh.

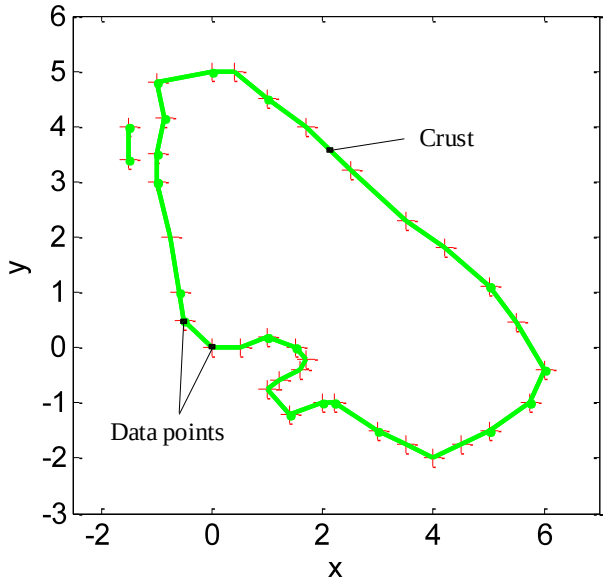


Fig. 126 Example of Crust non-manifold mesh reconstruction (double layer) where the lower bound on the ε-sampling, ε_0, is not satisfied.

The Crust algorithm is implemented in MeshLab, thus, a test on the turbine blade and another on the aspherical lens are run and shown in Fig. 127 and Fig. 128. The blade reconstruction is obviously non-manifold containing large triangles and an improper connectivity among points. The lens reconstruction is also non-manifold, however, with much less non-manifold elements than in the example of the blade (Fig. 128a). Again, this non-manifoldness is due to oversampling which creates irregularities in the Delaunay triangulation. We will elaborate more on this matter in the next paragraph because it is easier to understand this issue with the Cocone algorithm. A cleaning operation is available in MeshLab to remove triangles in the vicinity of non-manifold edges and the result of this operation is plotted in Fig. 128b. The latter is shown to contain missing triangles in the output mesh. The overall process of reconstruction takes about 25 seconds for 90,000 points. With the complete original cloud of points which contains 1,500,000 points, the process does not reach completion, knowing that the threshold that we found on the number of points for performing a Crust reconstruction in MeshLab is around 300,000 points.

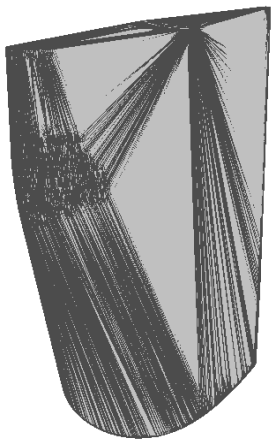
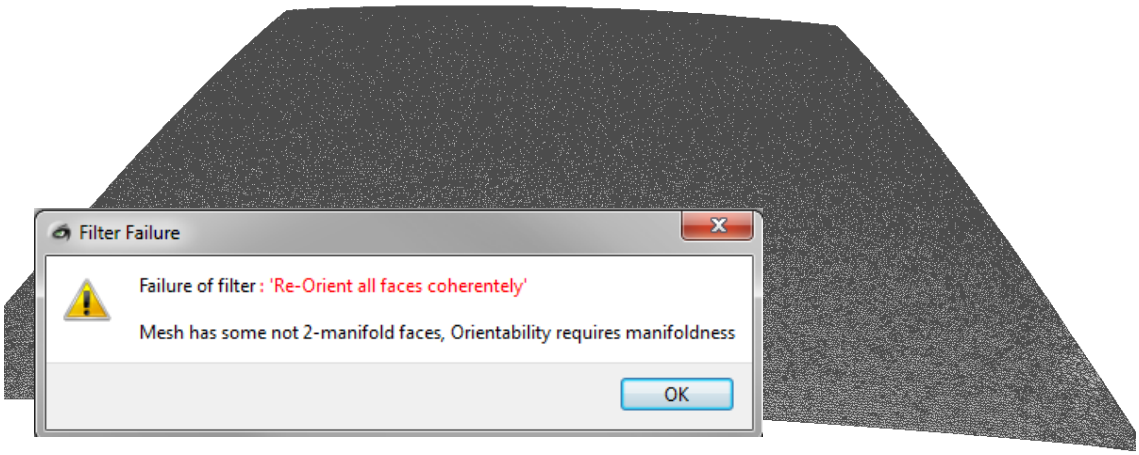
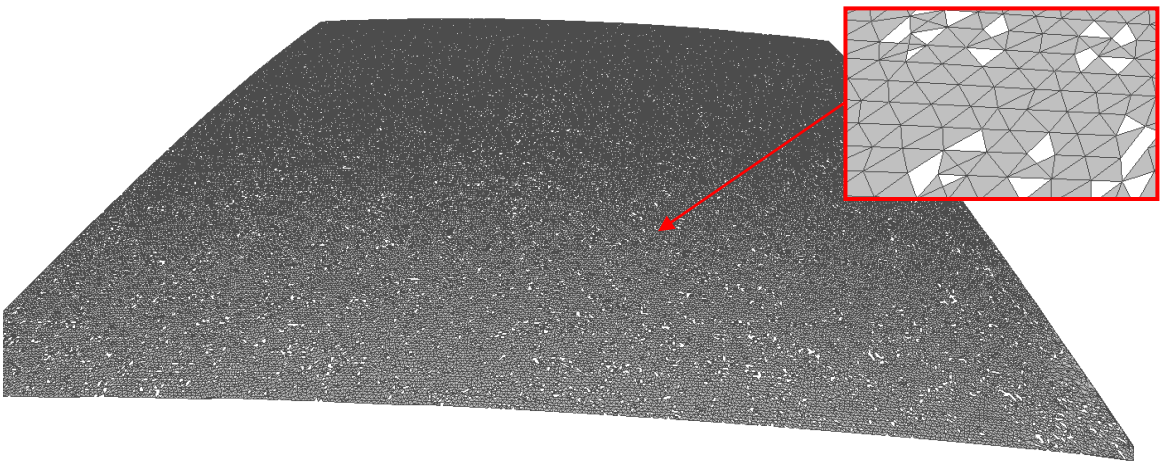


Fig. 127 Blade mesh reconstruction using Crust in MeshLab.



(a)



(b)

Fig. 128 The aspherical lens with $N = 90{,}000$ points and snapshots of its Crust mesh reconstruction in MeshLab: (a) original mesh containing non-manifold edges; (b) cleaned mesh after removing triangles near non-manifold edges (output mesh contains some missing triangles).

2.d Cocone and its variants

On the basis of the Crust algorithm, Amenta *et al* introduce the Cocone algorithm [170] which will be in turn followed by some heuristics and improved variants [174,200–202]. Cocone is presented as a simplified version of the Crust with reduced algorithmic complexity. Up to step 2 where the identification of poles occurs, both algorithms are identical. However, instead of building the Delaunay triangulation of the augmented point-set such as in step 3 of Crust, the Cocone directly performs a test called the Cocone test. This test consists of verifying whether a Delaunay triangle having a vertex $p_i \in P$ is a good approximation of the tangent plane at p_i. A good approximation means that the angle between the normal vector at p_i and the triangle lies within a pre-defined tolerance angle of $\frac{\pi}{2} \pm \theta$, where θ is set to a certain value and is referred to as the Cocone angle. Cocone is also based on the ε-sampling criterion and by that states that for an ε-dense sampling P, a point $p_i \in P$ has an elongated Voronoi cell in the direction of the normal $\vec{n}_i$ at p_i. The normal at p_i is the vector $\overrightarrow{p_i {p_i}^+}$ from p_i to its corresponding outside pole ${p_i}^+$. The tangent plane is nothing but the plane having the normal vector $\overrightarrow{p_i {p_i}^+}$. Since this vector is an approximation of the normal, the tangent plane is also an approximation. By allowing a tolerance angle θ derived from the ε-sampling condition, Amenta *et al* demonstrate that any triangle having p_i as a vertex and found within this angle tolerance, is a valid triangle that passes the Cocone test [170]. The Cocone can be thought of a double cone representing this angle tolerance within which the restricted Delaunay triangles must theoretically be found. The double cone has apex p_i and axis $\overrightarrow{p_i {p_i}^+}$ (Fig. 129). Not every triangle that passes the Cocone test belongs to the restricted Delaunay triangulation of the points. Only triangles having passed the test with respect to their three vertices are part of the output surface mesh.

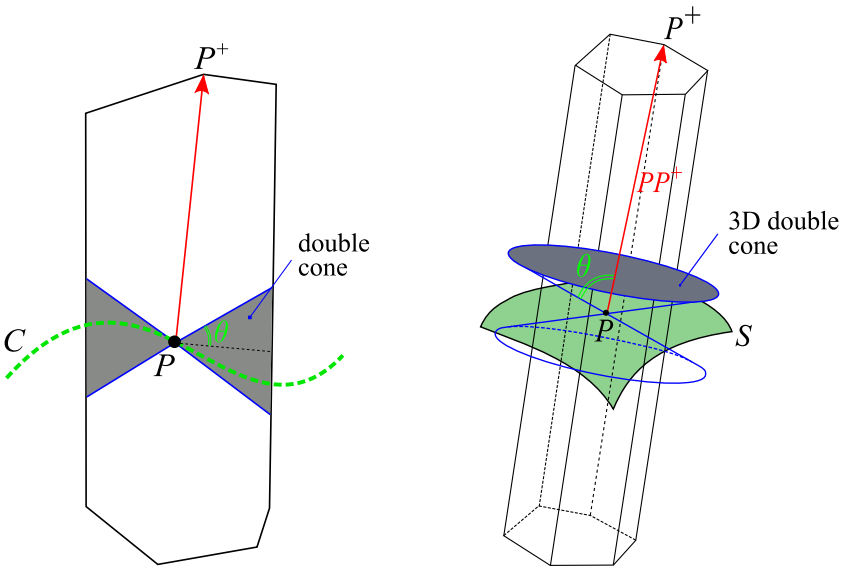


Fig. 129 The Cocone test at a given point in its Voronoi cell (adapted from [180]).

The Cocone algorithm starts with a Delaunay triangulation of the point-set. This convex representation of the set of points, like any convex representation, allows for the identification of each of the interior and the exterior subspaces. When inside and outside poles are identified, the normal orientation of the point-set is done and only positive normals oriented towards $p_i{}^+$ are considered (Fig. 129). The Cocone test then traverses all Delaunay triangles with a predetermined angle θ. The test on the closed curve is shown in Fig. 130.

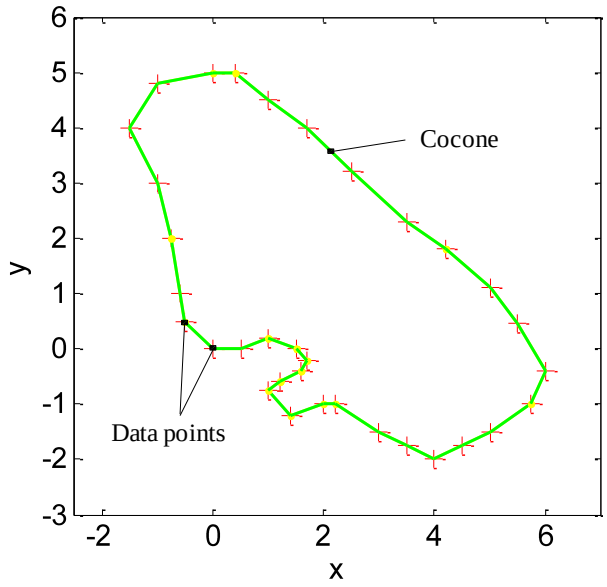


Fig. 130 Cocone applied on a closed curve represented by the red points with $\boldsymbol{\theta} = 67.5\,°$. The reconstruction is $\mathbf{1}$-manifold and an angle range for θ varying from $61\,°$ to $84\,°$ works just fine.

Among the facets which pass the Cocone test, only a subset belongs to the restricted Delaunay representation of the underlying surface. The remaining ones are either found in the vicinity of sharp edges or *flat* Delaunay tetrahedra. The reason why these pass the Cocone test is explained based on the following example which we restrict to *flat* Delaunay tetrahedra only (Fig. 131).

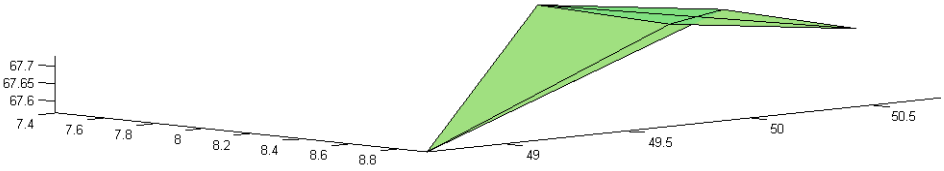


Fig. 131 2 Flat Delaunay tetrahedra in 3D explaining the issue of having missing triangles in the output mesh.

In fact, the facets of a flat Delaunay tetrahedron are almost coplanar, meaning that when they are approximated by the Cocone test, they have a very high probability of falling within the bounds of the double cone. Consequently, the choice of θ is critical in this configuration as an extreme value for the angle becomes required ($\theta \sim 0°$ in 2D and $\theta \sim 90°$ in 3D). Choosing an extreme value evidently deteriorates the reconstruction in the other regions. Keeping a moderate value results in superimposed layers of triangles like it is the case for both the Crust and the Cocone algorithms.

Nonetheless, the Cocone algorithm applies a heuristic in order to remove triangles that are adjacent to sharp edges (Fig. 132) and to extract the outer layer of triangles in regions where triangles are superimposed as a result of the existence of *flat* tetrahedra (Fig. 133a).

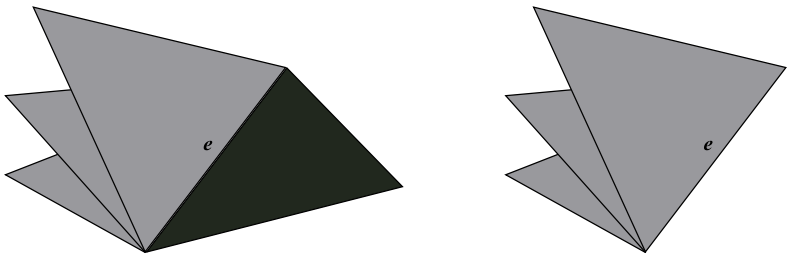
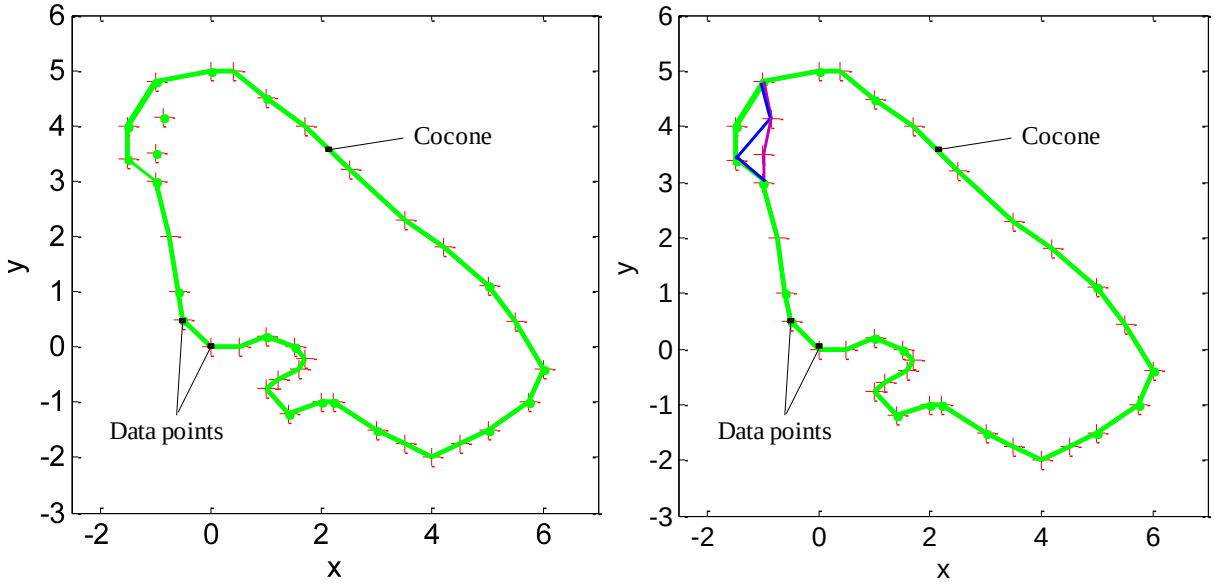


Fig. 132 left: *e* is a normal edge. Right: *e* is a sharp edge [180].

In particular cases, this heuristic solves the problem encountered by the Crust algorithm shown in Fig. 126. However, this heuristic opts for the outer layer (by choice) and does not

rely on a founded methodology for selecting the facets that are part of the output mesh. If we confront this issue to real measured data, the superposition of layers could be either the result of the presence of noise or outliers or the result of scan merging (case of the turbine blade measurement). Therefore, how to really reconstruct such regions, even on a curve, is not clear nor robust (Fig. 133b).



(a) (b)

Fig. 133 Mesh reconstruction is a region of superimposed layers of points: (a) A Cocone heuristic that opts for the outer layer. (b) Several ways are possible and none can be judged to be a better solution.

The algorithm is then tested on both the aspherical lens and the turbine blade but gives incomplete non-manifold reconstructions. For the aspherical lens, some triangles have been eliminated by the Cocone reconstruction process (heuristic) in regions where sharp edges and *flat* Delaunay tetrahedra are present (Fig. 134). For the turbine blade mesh reconstruction, not only there are missing triangles, but also non restricted Delaunay triangles and therefore the mesh is also non-manifold (Fig. 135).

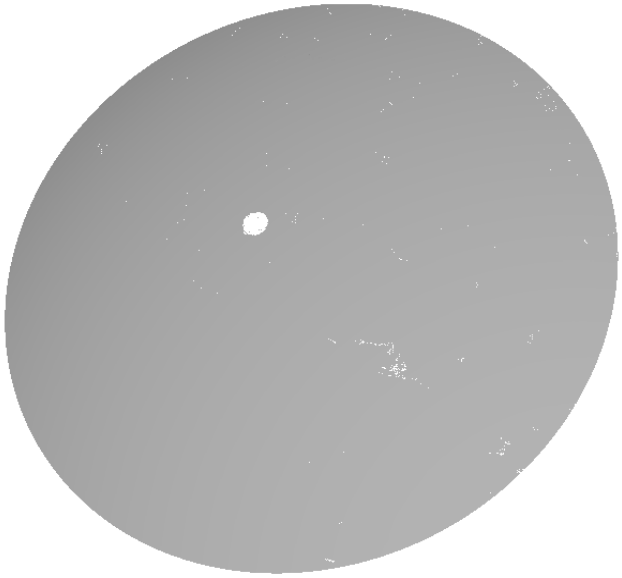


Fig. 134 Aspherical lens reconstruction using the Cocone approach for a dataset that does not satisfy the lower bound on the sampling density.

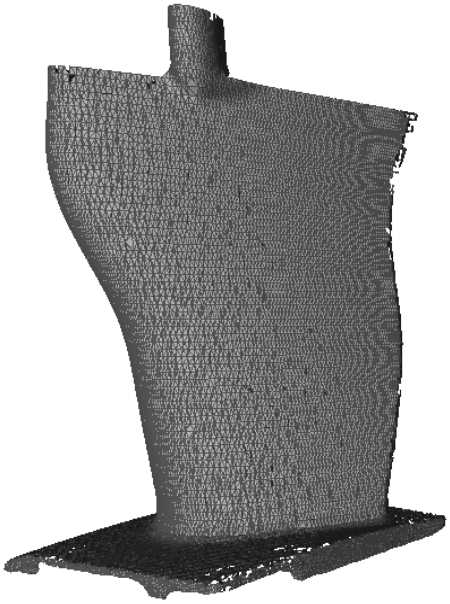


Fig. 135 Turbine blade mesh reconstruction using the Cocone algorithm.

When simulating an ε-sampling condition satisfying the lower bound (ε_0 found after many tests), the Cocone reconstruction of the aspherical lens becomes correct and manifold (Fig. 136).

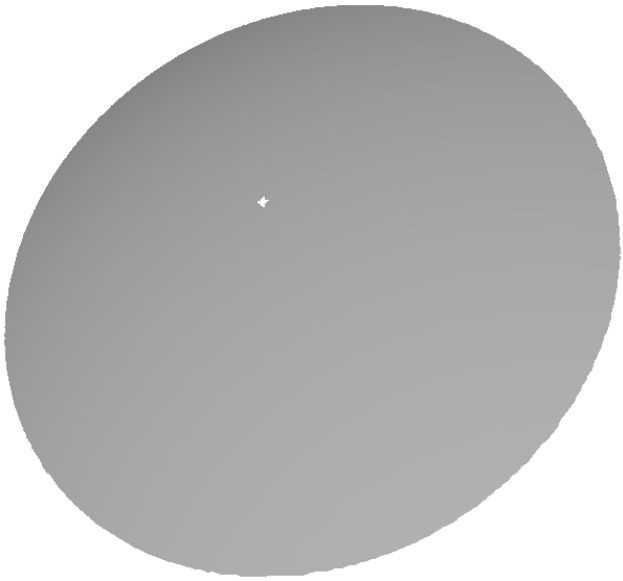


Fig. 136 Aspherical lens reconstruction using the Cocone approach for a dataset that satisfies the lower bound on the sampling density.

Variants of the Cocone algorithm are proposed with some additional heuristics to deal with under-sampling, large volumes of data and noise [180]. The SuperCocone [174], or more recently the LocCocone (Localized Cocone) [201] algorithms are an implementation of the Cocone algorithm with local resolution which allows processing large volumes of data. They are a kind of "*Divide and Conquer*" algorithms in which an Octree subdivision is applied [203]. It splits the space of the points into cells and then the reconstruction occurs in each individual cell B. In order to have a well-connected mesh from the subdivided partitions, the local reconstruction accounts for a small neighborhood in the adjacent cells to B. This results in an augmented cell B' such as shown in Fig. 137. The LocCocone algorithm presents guarantees based on the ε-sampling condition with an Octree subdivision that controls the number of points in B and therefore the size of B. According to the results published by Dey *et al*, the runtime for LocCocone as compared to SuperCocone is 10 times slower [201].

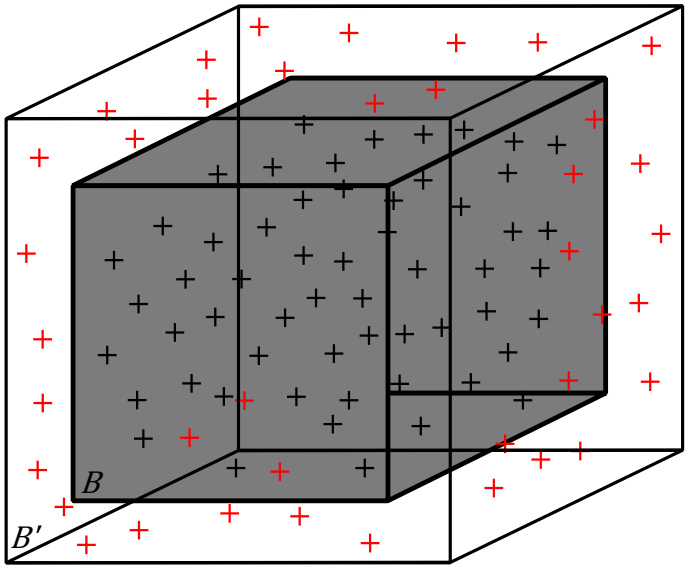


Fig. 137 A subdivided cell B and the augmented cell B'. The black points are points contained in B and red points are points added from the neighboring cells.

Other extensions to the Cocone algorithm exist and these are the TightCocone and the RobustCocone [200,202]. These algorithms were especially developed for datasets with undersampling and noise, respectively. The TightCocone adds a routine to detect undersampled regions and update them by identifying Voronoi cells which are not long and thin. The RobustCocone deals with noisy data up to a certain scale proportional to the sampling density and following a Gaussian distribution. Both algorithms have very specific sampling conditions to return reliable results, thus they are limited to particular cases which cannot be easily found in metrology applications.

2.e Natural Neighbors Interpolation (N.N.I)

This method is developed by Boissonnat and Cazals [204] at the French Institute for Research in Computer Science and Control (INRIA). It has been implemented in Catia V5 but was rapidly improved and modified leaving no trace of what is actually running in this software in matter of mesh reconstruction. The particularity of the Natural Neighbors Interpolation (N.N.I) is that it uses a blend of implicit techniques and combinatorial structures. Like the previously discussed algorithms (Crust and Cocone), N.N.I mesh reconstruction technique assumes a C^2-smooth, closed 2-manifold with no boundary and no self intersections. It also assumes an ε-sampling, but, the value of the parameter ε is slightly

more flexible. It also assumes that the point-set orientation is either given or can be robustly computed so that interior and exterior subspaces are identified.

When the data points do not have a uniform distribution, it is not easy to establish the surface neighborhood of each point. The k-nearest neighbors technique and the Voronoi diagram determine the 3D neighborhood but do not provide notions about surface neighborhood. Nevertheless, natural neighbors, introduced by Sibson [205] and used by Boissonnat and Cazals [204] are a good tool to determine surface neighborhood. The natural neighbors are calculated from the Voronoi diagram of the point-set and are computed using (41). Fig. 138 illustrates the technique on a 2D Voronoi graph. A point x is arbitrarily inserted in the Voronoi diagram of the points and the intersection of its Voronoi cell, $Vor(x)$, with the original Voronoi diagram of P defines the natural coordinate of x. $Vor(x)$ cuts the Voronoi cells of points p_1 to p_7 and by that concludes that points p_1 to p_7 are natural neighbors among themselves.

$$\lambda_{p_i}(x) = \frac{w_{p_i}(x)}{\sum w_{p_i}(x)}, \tag{41}$$

where $w_{p_i}(x)$ is the area (respectively the volume in 3D) measure of the portion $Vor(p_i)$ intersected by $Vor(x)$. Therefore the natural coordinate is nothing but the ratio of this portion with respect to the total area (respectively the volume in 3D) of $Vor(x)$. The closer the data point p_i is to x, the larger is its area/volume measure. In fact, x can be seen as the center of gravity of its natural neighbors. An exception to the area/volume measure occurs when x intersects a Voronoi cell at infinity, in which case, the value of $w_{p_i}(x)$ is infinite. Boissonnat and Cazals suggest adding a bounding box around the cloud of points in order to limit the workspace and throwing out exceptions.

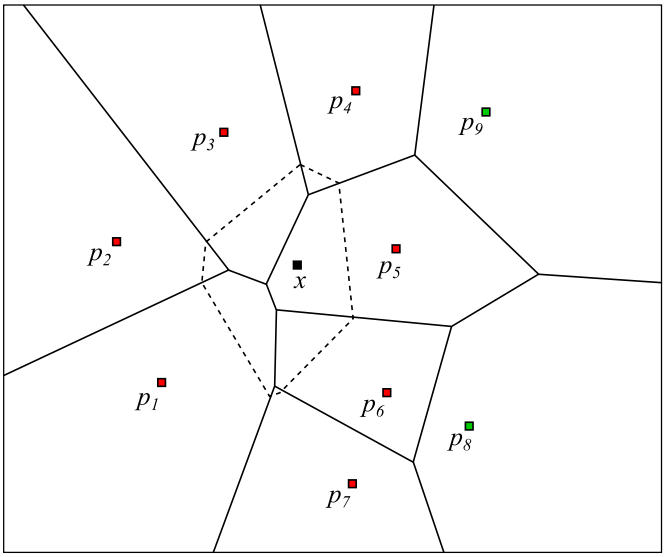


Fig. 138: Natural neighbors. Point insertion in a Voronoi diagram in 2D.

Once the natural coordinates have been assigned, they serve as weighting coefficients in the N.N.I which is in turn calculated on the basis of an implicit distance function $\mathbb{R}^3 \rightarrow \mathbb{R}$ weighted by $\lambda_{p_i}(x)$ (42). The interpolating function is a first approximation of the underlying surface of the point-set. The function naturally takes a value of zero at the data points since $\lambda_{p_i}(x) = 0$ at p_i.

$$h(x) = \sum \lambda_{p_i}^{1+w}(x)\, h_{p_i}(x), \tag{42}$$

Where $h_{p_i}(x)$ is the signed distance function calculated from the points [168]. The assumption about point-set orientation indicates the sign of $h(x)$ and states whether a point is on one side (interior) or the other side (exterior) of the convex object that is reconstructed. Hence, once h is calculated, an implicit surface is created and delimits interior and exterior sides of the point-set. Finally, the mesh is extracted by searching for Delaunay triangles which are close to the implicit function. The method to identify those triangles goes by searching for bipolar Voronoi edges having one vertex outside and another inside the confinement of the implicit function.

Recall that a Voronoi edge is orthogonal to its dual Delaunay triangle. So if a Voronoi edge is an approximation of the normal direction at some location, it follows that its dual representation approximates the tangent plane. The Voronoi edges that have each vertex on one side of the implicit surface would not only infer that their dual Delaunay triangles approximate the tangent plane, but also that these triangles are close to the surface and are a

good approximation of it. Detecting bipolar Voronoi edges reduces to performing a simple product of the h values of their vertices. Let a and b be the vertices of a Voronoi edge. A Voronoi edge is said to be bipolar if the product $h(a) \times h(b) < 0$ meaning that a and b are on opposite sides of the surface and their h values take opposite signs (Fig. 139).

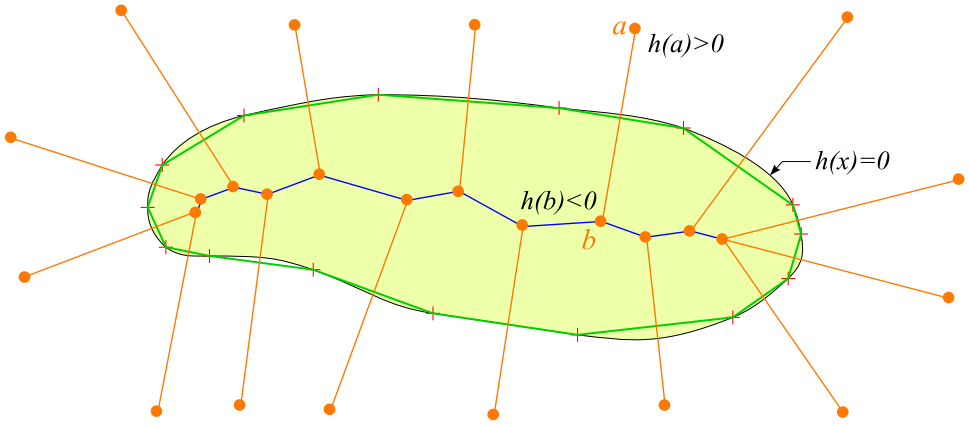


Fig. 139: Natural Neighbors Interpolation: bipolar Voronoi edges (orange highlighted segments) and mesh extraction (green). (Legend: Data points (red); implicit function $h(x) = 0$ (black curve); Voronoi diagram (blue)).

The same example of the closed curve is tested with N.N.I and the result is shown in Fig. 140. We show the Voronoi diagram (blue) because it is the basis for N.N.I. mesh reconstruction. The reconstruction is proper for this example since the ε-sampling is satisfied.

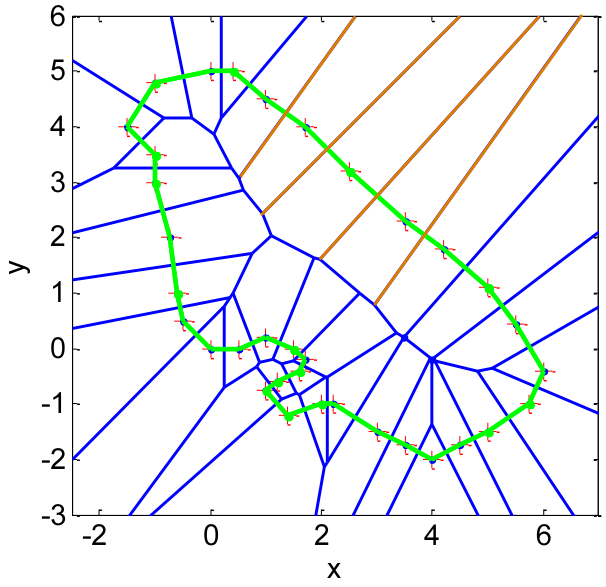


Fig. 140 N.N.I (green) applied on a closed curve represented by the red points. Some bipolar Voronoi edges are highlighted. The reconstruction contains no interior segments.

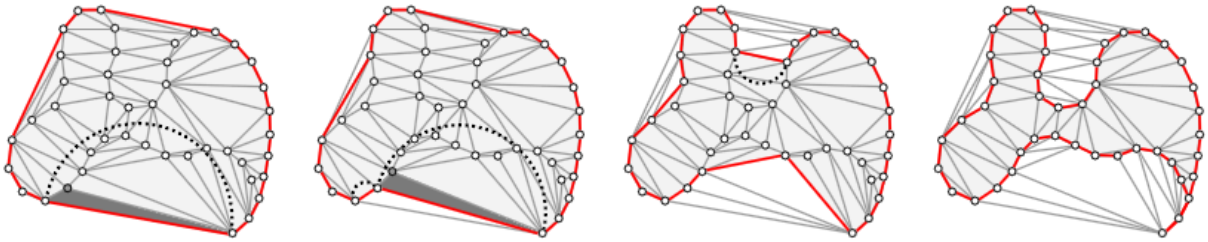
The complexity of the N.N.I algorithm is covered by the complexity of the Voronoi/Delaunay structures, therefore, it is quadratic (Table 20). Since the algorithm calculates areas/volumes and then computes for the natural coordinates this adds to the complexity of the point insertion operation done over the Delaunay triangulation. As compared to the Cocone algorithm, N.N.I has a larger complexity. Point insertions and bipolar Voronoi edges search are more complex operations than the filtration step of Cocone. The algorithm assumes an ε-sampling, consequently, it offers the same surface reconstruction guarantees as the Cocone or any other algorithm based on this assumption. The natural neighbors might come up being not exclusively close points on the real surface as this problem occurs for under-sampled data. The main limitation of this algorithm is that it assumes that the normal directions at each data point are well approximated or known in advance.

1. Voronoi diagram/Delaunay triangulation	$O(N^2), N$ points
2. Normal approximation (if any)	$O(\log m), m = 2N$ poles
3. Natural coordinates/implicit function	$O(v.\log v), v$ point insertions
4. Mesh extraction	$O(e), e$ Voronoi edges

Table 20: Complexity of the N.N.I algorithm steps.

2.f Geometric Convection

The geometric convection algorithm developed by Chaîne is a geometric tool for surface reconstruction based on the Delaunay triangulation of the point-set [173,206]. Given a closed surface $\hat{S}$ enclosing a point-set P, the convection of this surface along the gradient field of the distance function $-\nabla g(x)$ computed on P converges to a piecewise linear "pseudo-surface" that approximates P. The term "pseudo-surface" is picked from Chaîne in [173] and denotes a piecewise linear surface made of oriented Delaunay triangles which can join or disjoin after each iteration. The initialization of $\hat{S}$ are the Delaunay facets constituting the convex hull of P for which the normal directions can be computed easily. Consequently, the facets are oriented inwards and undergo the Gabriel test. Facets that fail the Gabriel test are the facets that are removed from $\hat{S}$ and replaced by their complementary facets in the corresponding Delaunay tetrahedron (Fig. 141a and Fig. 141b).



(a) (b) (c) (d)

Fig. 141 The geometric convection algorithm and the Gabriel test for a 2D curve.

The Gabriel test consists of checking the enclosing of the inner half-ball of each of the Delaunay facets. If the half-ball contains points from P, it means that the current Delaunay facet should be removed and replaced by new facets which should be re-oriented inwards if they happen to be exteriorly oriented. The process is repeated for all facets of the evolving $\hat{S}$ until all facets pass the test. The normal direction to each facet belonging to $\hat{S}$ is updated at each iteration.

The complexity of the algorithm is equivalent to that of the Delaunay triangulation as it is its most complex routine (Table 21).

1. Delaunay triangulation	$O(N^2), N$ points
2. Convex Hull of P	$O(N), N$ points
3. Oriented Gabriel test	$O(t), t$ tetrahedra

Table 21: Complexity of the geometric convection algorithm steps.

The conditions to use the geometric convection algorithm are that the underlying surface to the points must be closed and embedded in $\mathbb{R}^3$. The ε-sampling criterion is required in order to guarantee geometrical closeness and topological equivalence. The major limitation of the geometric convection algorithm are highly concave regions which are also referred to as pockets and shown in Fig. 142. The actual Delaunay facet under test (segment in 2D) passes the Gabriel test although it must not belong to the output mesh.

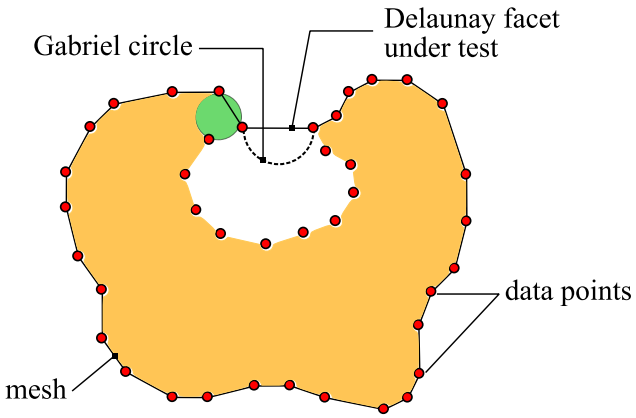


Fig. 142 Pockets: high concavity regions not solved by geometric convection.

2.g Other algorithms

The *Wrap* and the *Flow complex* are other developed algorithms in the subject matter of mesh reconstruction [169,207], [208]. The algorithms differ in the mesh reconstruction technique from the previous ones although the Delaunay triangulation is still the basis for this class too. For both techniques, functions describe a flow relation among Delaunay simplices, a subset of which constitute the mesh [135]. Comparison of those algorithms with all the above is summarized in Table 22.

There has also been the *Tangential Delaunay Complexes* algorithm developed by Boissonnat and Ghosh which relies on the notions of tangential neighborhood [42], [209]. Instead of solving the problem according to natural neighbors interpolation (N.N.I) and having to reason based on a global Voronoi diagram, the method of tangential complexes uses a technique to create a local system of coordinates.

2.h Comparative study

All of the above algorithms belong to the family of algorithms which require the computation of a Voronoi diagram and a Delaunay triangulation. Although they seem different, they all aim at approximating the restricted Delaunay triangulation to the underlying surface. In theory, if the surface is known, the restricted Delaunay set is unique. But since the surface is unknown and only unorganized points are given, mesh reconstruction falls within the family of approximation methods. The above presented algorithms are all based on the 3D global Delaunay triangulation of the point-set, except for the Localized Cocone method which computes local Delaunay sets after hashing the points space. The algorithms are all equivalent

with respect to complexity as they depend on the Delaunay triangulation which is the most computationally expensive task ($O(N^2)$). The sculpture algorithm and the α-shapes are the earliest algorithms in this field and are excluded from the previous statement. Table 22 summarizes the performances of each algorithm with respect to the criteria defined at the beginning of this chapter, the most important being the guarantees that each algorithm provides, its complexity and the initial conditions it requires. Note that we have chosen the most relevant algorithms without redundancies, i.e. we put Cocone to represent the basic Cocone algorithm and its variants, TightCocone and RobustCocone. Localized Cocone is the most recent Cocone variant and an enhanced SuperCocone algorithm and performs surface reconstruction locally.

	Complexity	**Guarantees**	**Large datasets**	**Input requirement**
Sculpture	$N^2.logN$	×	×	×
α-shapes	N^2	×	×	Uniform sampling
Cocone	N^2	$\varepsilon \leq 0.05$	×	Θ
N.N.I	N^2	$\varepsilon \leq 0.25$	✓	Points normal orientation
Wrap	N^2	$\varepsilon \leq 0.01$	×	×
Convection	N^2 (local)	×	✓	Points normal orientation
LocCocone	N^2 (local)	$\varepsilon \leq 0.05$	✓	Θ

Table 22 Comparison of mesh reconstruction techniques based on combinatorial structures.

The algorithms without guarantees are the algorithms that are not based on any assumption with respect to sampling density. The Sculpture algorithm and the α-shapes are as such. The other algorithms in the list present guarantees based on the ε-sampling density criterion. However, as it has been seen in this chapter, the guarantees remain theoretical because they are based on the knowledge of the theoretical surface and its medial axis.

For all the algorithms based on combinatorial structures, an additional key condition being a lower bound on sampling density must also be satisfied. This lower bound ε_0 is not easily achievable along with the ε-sampling condition. In practice, for the measurement of an

unknown freeform surface, the sampling density cannot be controlled because the surface curvature and normal orientation are usually unknown [210][211][212][10][11].

Having a quadratic complexity in the number of points, large datasets can be processed as mentioned in Table 22. Nevertheless, computational time will not, in general, be acceptable for metrology applications requiring fast processing, especially that further operations are yet to be performed in terms of the classical freeform surface reconstruction (Fig. 102).

Some of the algorithms source codes are found online or implemented in software and are tested for comparison on a cylindrical artifact (Table 23) [215][196][183]. For this example, we look for geometrical and topological correctness through statistical information on the generated mesh. The number of triangles in the mesh should be minimal ($< N^2$) without having undesired holes in the mesh. The size of the triangles should be proportional to the sampling density.

	α-shape MeshLab	**α-shape CGAL**	**SupCoc**	**TigCoc**	**N.N.I CATIA v5**	**Wrap Geomagic**
Number of points	54,790					
Number of mesh triangles	137,915	109,576	110,709	109,576	109,576	109,576
Mean triangle area	1.048	0.715	0.715	0.715	0.715	0.715
Standard deviation	0.762	0.337	0.320	0.320	0.321	0.320
Total mesh area (mm²)	72,318	78,397	79,180	78,320	78,323	78,320
Computational time (s)	10	62	21	320	4	4

Table 23 Comparison of selected algorithms software for the mesh reconstruction of a cylindrical artifact with respect to computational time and some elements that infer about geometry and topology correctness (SupCoc: SuperCocone; TigCoc: TightCocone; N.N.I: Natural Neighbors Interpolation).

For the example in Table 23, the sampling is taken as uniform, points are sampled on a theoretical CAD cylinder and the distance separating any two consecutive sampled points is about $\sim$1.2 mm. So, based on the mesh triangles area it can be observed that the algorithms return a reasonable value (almost equal to $(1.2)^2 / 2$. Inconsistencies and undesired holes can however be present in the output mesh for all the listed algorithms.

For the case of the aspherical lens, the algorithms have been tested and the results are reported in Table 24. The mesh triangles average area is slightly larger in order of magnitude

than the order of magnitude of the sampling density (point spacing is about ~4×10^{-4} mm) for the case of the α-shape ($\frac{0.0004^2}{2} \ll 4.5 \times 10^{-7}$).

	α-shape MeshLab	**α-shape CGAL**	**SupCoc**	**TigCoc**	**N.N.I CATIA v5**	**Wrap Geomagic**
Number of points	1,500,000					
Number of mesh triangles	5,055,453	Cannot handle	2,085,084	2,088,537	2,084,684	2,084,681
Mean triangle area	4.5×10^{-7}	Cannot handle	5.4×10^{-8}	1.1×10^{-7}	5.4×10^{-8}	5.4×10^{-8}
Standard deviation	1.5×10^{-6}	Cannot handle	1.8×10^{-8}	2.5×10^{-5}	1.8×10^{-8}	1.8×10^{-8}
Total mesh area (mm²)	2.253	Cannot handle	0.113	0.225	0.113	0.113
Computational time (s)	129	Cannot handle	155	5040	120	18

Table 24 Comparison of selected algorithms software for the mesh reconstruction of the asphere with respect to computational time and elements that infer about geometry and topology correctness (SupCoc: SuperCocone; TigCoc: TightCocone and N.N.I: Natural Neighbors Interpolation).

For more complex surfaces such as the blade, the mesh reconstruction algorithms return bad results due to the complexity of the shape of the surface. The tip of the blade as well as its leading and trailing edges are all regions of high curvature containing overlapping points. Moreover, the nature of the scanning is done in lines of scan along which the longitudinal density is higher than the lateral density (Fig. 143).

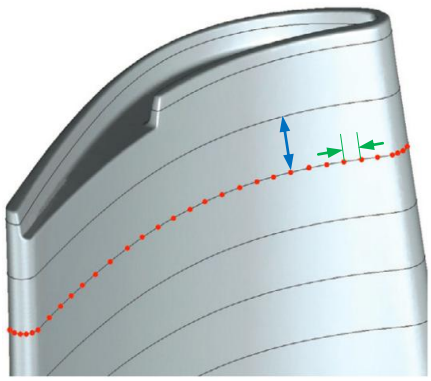
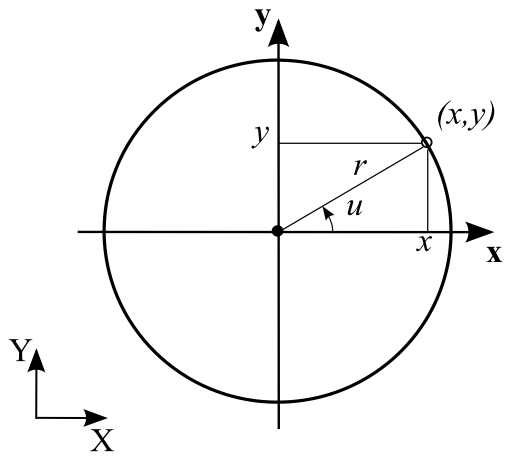


Fig. 143 Scanning of the turbine blade in lines of scan. Lateral point spacing (blue) is much larger than longitudinal spacing (green).

3. Mesh reconstruction by implicit techniques

As opposed to the techniques detailed previously, implicit mesh reconstruction techniques use implicit functions to approximate the restricted Delaunay triangulation of the underlying surface which is generally unknown [168,175,216–219]. These techniques have the advantage to adapt to the modeling of complex topologies [220] and maintain a simple data structure [175]. While explicit surfaces can localize and describe a point on the surface in an exact manner, implicit techniques approximate the surface by calculating an iso-contour associated to a scalar c of an implicit distance function. Contours with a scalar smaller than c define the interior space of a surface and contours having scalars greater than c define the exterior space of the surface. The computation of a point on the surface is very difficult with implicit representations, but the computation of the normal vector and tangent plane at a given location is easy [168].

Fig. 144 illustrates the difference between an explicit representation of a circle and the implicit one. Also, the parametric representation which is explicit is also shown for this example. It can be observed that the implicit representation of the circle in Fig. 144 has a scalar equal to 0. This means that the values of x and y which match the value of r^2 are points on the circle of radius r. Points that satisfy $x^2 + y^2 - r^2 = 1$ are points on another circle of radius $r^2 + 1$.



Explicit: $y = \pm\sqrt{r^2 - x^2}$

Parametric: $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos(u) \\ r\sin(u) \end{pmatrix}, 0 \leq u \leq 2\pi$

Implicit: $x^2 + y^2 - r^2 = 0$

Fig. 144 Explicit, implicit and parametric definitions of a circle.

In a more general way, an implicit function associates a scalar to a set of points in space $f: \mathbb{R}^3 \rightarrow \mathbb{R} \mid f(x_i) = c$. For mesh reconstruction problems, f is usually a distance function that must be zero at the data points. Other implicit functions exist and are the radial basis

functions [216], [221], the polynomial bi-variate functions [217] and indicator functions such as Poisson functions [218].

Normal orientation of the point-set is key for the implicit function methods except for the Level-set methods. Normal directions are the first order approximation of the surface and are used to identify interior and exterior regions of the underlying surface. When the implicit function is determined and the surface is implicitly approximated, efficient methods can generate the mesh [222,223].

3.a Tangent planes method

Among the first published works in implicit mesh reconstruction, the tangent plane approximation method has been introduced by Hoppe *et al* and is the reference for the implicit mesh reconstruction methods [168]. It uses an implicit distance function $d(x)$ to approximate the tangent plane at each point. For all $x \in F$ where F is the neighborhood of a data point $p_i \in P$, the method computes the distance separating x from p_i. Naturally, this function is zero when $x = p_i$, i.e., $d(x = p_i) = 0$, and we refer to the set of solutions to this equation as the zero set of $d(x)$. With this procedure, interior and exterior regions can be locally identified (Fig. 145), and, a contour tracing method for the zero set, such as the marching cubes, allows to obtain the mesh [224].

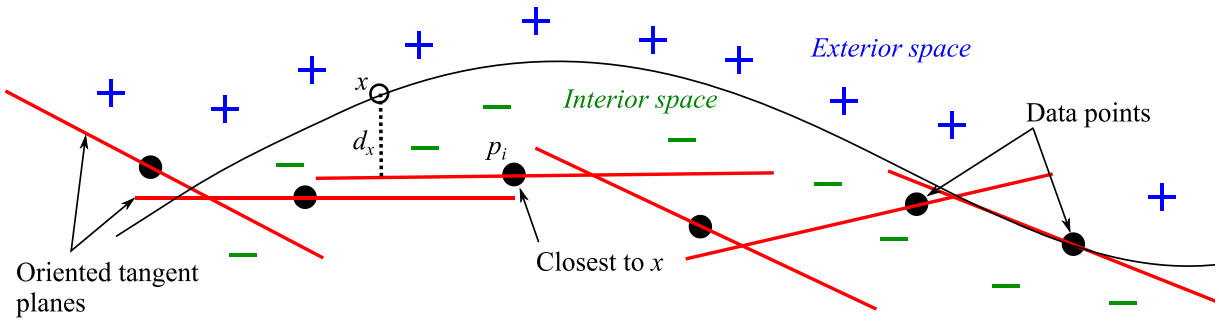


Fig. 145 The tangent planes method on a curve example. d_x is the distance separating a point x from its closest data point $p_i \in P$.

The algorithm starts with the assumption that the data may contain noise and thus are expressed as $p_i = x_i + e_i$ where, e_i represents the noise vector and x_i is the projection of a data point p_i on the theoretical surface. Therefore e_i is the error vector oriented along the

normal direction to the surface. According to Hoppe *et al*, an acceptable sampling is given by the following inequality:

$$\mathrm{Card}(\|x - x_i\| \leq \rho) \geq 1 \tag{43}$$

where, ρ is the radius of the sphere centered at a given point x on the theoretical surface. The condition is that the sphere contains at least one projection point x_i of a data point p_i. This is illustrated on a curve in Fig. 146.

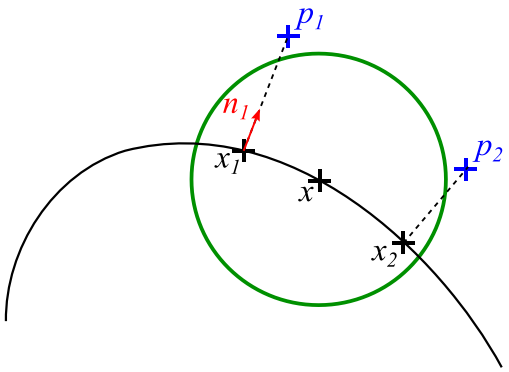


Fig. 146 The noise condition for the tangent planes method: a curve example.

The signed distance given by $d(x)$ is calculated in the normal direction to each data point $p_i \in P$. Normal estimation is deduced from the tangent planes estimation based on a discrete Principal Component Analysis (PCA) of P. For each point $p_i \in P$, the k-neighborhood of each point is considered and a Least-Squares plane is fitted to the cluster of points and denotes the tangent plane T_{p_i} at p_i. The normal direction n_i and the center of gravity of the cluster of points o_i can be deduced. Among all n_i with point of application o_i, the algorithm initializes the normal orientation with the orientation of n_i of the point having the largest z-component of o_i. All the other normal vectors are re-oriented according to the orientation chosen for this first vector according to the normal propagation method proposed by Hoppe *et al* [168].

With the normal orientations of the points being approximated, each data point p_i is now projected onto T_{p_i} along n_i. The distance separating p_i from its projection is the distance d_x (Fig. 145). If $d_x < \rho + \delta$, where δ is a small predetermined tolerance, then $d(x) = d_x$, otherwise, $d(x)$ is undefined. All the undefined values of $d(x)$ define the other contours having specific scalar values different from 0 as the contour with scalar value 0 determines

the curve/surface sought. Since the normal orientation is computed, the distance is signed and interior and exterior points are distinguished.

1. Tangent planes approximation	$O(N), N$ points
2. Tangent planes re-orientation	$O(N), N$ points
3. Orientation propagation	$O(N^2), N$ points
4. Signed distance function calculation	$O(N), N$ points

Table 25 Complexity of the Tangent Planes method.

The complexity of this algorithm depends on the complexity of the steps listed in Table 25. The most complex of all being the normal orientation propagation routine, the algorithm's complexity follows. Knowing that this step requires an Euclidean Minimum Spanning Tree routine (EMST) the complexity amounts to $O(N^2)$ for N points in the dataset. Nevertheless, the algorithm can be simplified by applying an Octree or a k-D tree subdivision to the set of points and reduce the EMST complexity to the order of $O(Nk)$.

The tangent planes algorithm requires the estimation of normal directions of the point-set and this issue is not so simple. Although the tangent planes approximation is relatively simple, defining a globally consistent orientation for the surface is the major difficulty of the algorithm as stated by Hoppe *et al* [168].

3.b Level-set

The Level-set method developed by Zhao *et al* is an implicit mesh reconstruction method that relies on the calculation of a distance function and the solution of a variational partial differential equation [175]. The formulation of the surface shape includes the minimization of oscillations between data points allowing for an implicitly smooth reconstruction. This method has inspired the work of Chaîne [173] presented in section Chapter 3 - II. 2.f and other works that are based on making an initial surface evolve to the underlying shape of the points.

The main idea is to start from an initial enclosing surface that roughly approximates the underlying surface to the points and transform it to a minimal representation of the points. The process involved in this surface representation evolution is based on minimizing the gradient flow of an energy functional and is called the Level-set method. It can be perfectly assimilated to an optimization problem based on the gradient descent criterion, only that here,

the functional implicitly implies notions about the distance between the surface at a given state and the data points. Zhao *et al* propose a surface initialization procedure that starts with the computation of a signed distance function $d(x)$ calculated by resolving the Eikonal equation given in (44):

$$|\nabla d(x)| = 1. \tag{44}$$

Then the initialization of the implicit surface can be done through a fast tagging algorithm that identifies interior from exterior subspaces. A more efficient way of solving the Eikonal equation (44) and finding numerical approximations to it is the Fast marching Method (FMM) introduced by Sethian [225], [226].

It is important that the initial surface be close to the datapoints so that the solution to the partial differential equation (PDE) (46) is efficiently computed. This PDE comes from the expression of the problem as the minimization of an energy functional that includes notions about both distance and mean curvature (45). It is independent of parameterization and invariant under rigid transformation.

$$E(s) = \left(\int d^e(x)\, ds \right)^{1/e}. \tag{45}$$

Minimizing $E(s)$ reduces to solving the PDE of $E(s)$ either following the gradient flow of the surface or the time dependent convection model given in [175].

$$d^{e-1}(x).\left(\nabla d(x).\vec{n} + \frac{1}{e} d(x).\kappa \right) = 0 \tag{46}$$

where, s is the surface at a given state (or iteration), $d(x)$ is the distance separating a point x on the surface from a data point $p_i \in P$, e is an exponent to determine the extent of the distance, i.e., $e = 1$ means that the distance is calculated between x and its closest data point p_i. $\vec{n}$ is the unit outwards normal vector and κ is the mean curvature of s at x.

The PDE can be seen as an equilibrium of forces equation in which the terms are forces to be balanced out. The first term represents an attraction that attracts the surface towards the points and the second term represents a surface tension that minimizes oscillations and smoothens the surface (Fig. 147). Any point on s at a given iteration undergoes a force in the direction of the gradient towards its closest point $p_i \in P$ and a lateral force in the tangent

direction to the surface. The algorithm stops at equilibrium state when the forces cancel each other meaning that (46) is solved. PDE allows to naturally build the connectivity among the points without having resort to any combinatorial structures.

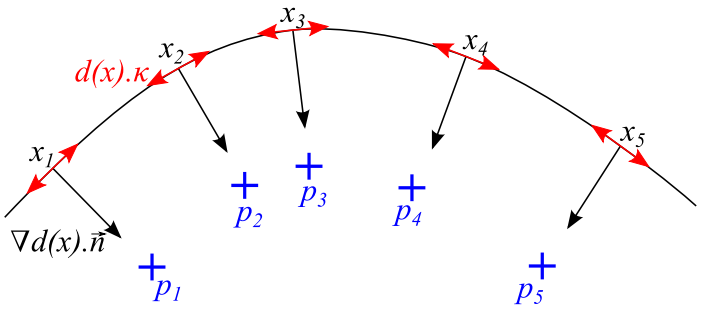


Fig. 147 The Level-set forces, attraction (black vectors) and curve tension (red vectors).

The steps involving the calculation of the distance function and initializing the implicit surface for the gradient flow require that the data points P are within a rectangular grid consisting of cells having a dimension h. This dimension transcribes the resolution of the approximation to the surface as it evolves across the grid cells. An illustration showing this evolution for a 2D curve is shown in Fig. 148.

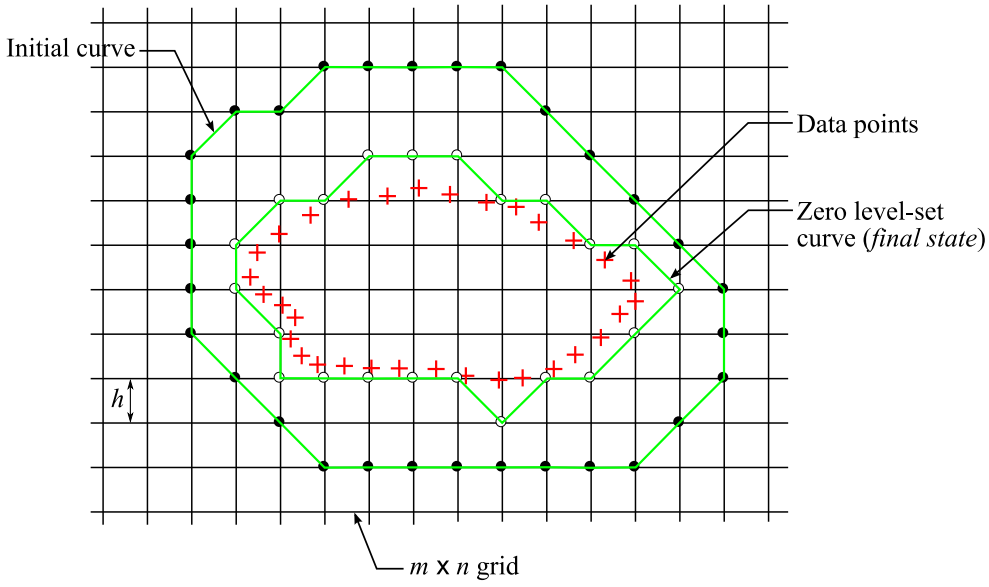


Fig. 148 The Level-set principle and evolution of the curve based on a rectangular grid.

The complexity of the Level-set algorithm depends on the initial curve/surface. The steps of the algorithm as well as their respective complexities are shown in Table 26.

1. Distance function calculation	$O(M+N), N$ points, M cells
2. Initial implicit surface	$O(M+N), N$ points, M cells
3. PDE solution	$O(M+N), N$ points, M cells

Table 26 Complexity of the Level-Set method.

If the curve/surface is too far from the optimal shape, complexity might happen to be expensive in time. So in order to have a good initial guess, the condition is to have all the $d(x)$ terms close to zero because theoretically, the zero Level-set represents the true surface. Following this logic, the terms $d(x).\vec{n}$ should be minimal. Nevertheless, they should not be too small, i.e., smaller than the point-set sampling density (c.f. ε-sampling [171]), else, the contour would be a set of small spheres around the data points. Zhao *et al* claim that a good initial curve/surface (initial contour) intersects the Voronoi diagram of the point-set based on [204]. To guarantee a correct value for $d(x)$, the condition in relation to the sampling density states that: $\max(l) < d(x) < \frac{1}{\min LFS(x)}$, where l is the spacing between each couple of theoretically connected data points and $\min LFS(x)$ is the minimum local feature size of the theoretical curve/surface.

When the curve/surface is initialized, the points $x \in s$ are also initialized and distributed in the form of a grid. The distance calculation from x to its corresponding data point $p_i \in P$ requires $O(M+N)$ operations for M grid points and N data points. With a grid resolution comparable to the sampling density, $N \approx M^{2/3}$, the total complexity of the Level-set routine is $O\left(N^{3/2}\right)$. Moreover, if the PDE is solved following the gradient flow, the result is a smooth implicit minimal curve/surface. However, this is computationally even more expensive. On the contrary, if the PDE is solved following the convection model, the output is a mesh that does not exactly pass by the data points but is within a pre-defined tolerance from the dataset. Our applications involve very high density samplings, so they require grids of high resolution, leading to the multiplication of the complexity by an order of 10 or more.

The Level-set method guarantees a good convergence provided that the surface is well initialized and this is based on the ε-sampling density criterion. Since that in inverse problems such as mesh reconstruction, the theoretical curve/surface is unknown, it is complicated to guarantee an acceptable initial distance function and an initial curve/surface because the local feature size is also unknown. The most important issue to point out here is that the smooth result of the Level-set is an implicit curve/surface approximated by discrete points on a grid

but which helps building a mesh on the dataset. It is only in the case where the mesh is manifold that a parametric smooth representation of the surface can be performed according to the framework illustrated in Fig. 102.

3.c Multi-level partition of unity

The Multi-level Partition of Unity (MPU) algorithm is another type of shape representation that is based on implicit quadratic functions [217]. This algorithm, proposed by Ohtake *et al*, is designed for problems with large data and for shapes that present regions of high curvature or even sharp edges. MPU is among the few algorithms ([174,201]) that were designed to subdivide the space of the points. These are segregated into clusters and each cluster is processed separately, making the algorithm runtime considerably shorter than the case where points are processed all at once. Nonetheless, the major difficulty regarding subdivision is the step in which the partial results are merged in order to generate the result. Subdivision can be a uniform hashing of the space, an Octree subdivision or a k-D tree subdivision. Each cluster of points contains a finite number of data points to which a quadratic function is associated (fitted). Here, Ohtake *et al* [217] use an iterative Octree subdivision that is subject to a condition on the residual errors of the fitting and this is what is called a multi-level partition of unity. If the local mean residual error is smaller than a pre-defined tolerance, the Octree subdivision stops.

The algorithm starts by assuming that the points are oriented, i.e., the normal directions at each point are known. Since the space of the points needs to be subdivided, it is limited to the bounding box of the points. These can therefore be scaled down until the bounding box diagonal has a unit length. Then the Octree subdivision is applied and subdivides the bounding box into smaller boxes indexed by j, having a diagonal length d_j, a center c_j and containing a finite number of points N_j. The sphere s_j centered at c_j and having a radius $r_j = \alpha d_j$ should contain a minimum number of data points N_{min} (a typical value for $\alpha = 0.75$). Hence, if s_j contains less than N_{min} points, r_j is increased. Based on this setting, implicit quadratic functions are fit to the points in each cell and the local mean residual error is compared to the pre-defined tolerance. Cells where the error is still large are further subdivided and the process is the same.

Since the bounding box diagonal has been set to unit length, the subdivided cells represent a partition of unity. Hence, the fitted implicit functions to each cell are also a

partition of unity and their fusion creates a global implicit function f_g describing the shape of the underlying surface to the points (47). The local implicit function that is used is either a quadric, a set of quadratic polynomials or a piecewise quadric (48) and is weighted according to a smooth B-Spline weighting model (49).

$$f_g(x) = \frac{1}{\sum_{i=1}^{n} w_i(x)} \sum_j w_j(x)\, F_j\,(x) \tag{47}$$

where,

$$F_j(x) = \begin{cases} \text{quadric} & N_j > 2N_{min}\ \&\ \theta \geq {}^{\pi}/_{2} \\ \text{bivariate quadratic} & N_j > 2N_{min}\ \&\ \theta < {}^{\pi}/_{2} \\ \text{piecewise quadric} & N_j \leq 2N_{min} \end{cases} \tag{48}$$

and

$$w_j(x) = b(u)\left(\frac{3|x - c_j|}{2r_j}\right). \tag{49}$$

The variability of the choice for the local functions $F_j(x)$ allows to fit different surface features such as, locally smooth surface parts (bi-variate quadratic polynomials) or sharp features (piecewise quadrics).

The complexity of this algorithm depends on the size of the output (number of functions and fittings to perform) and is of the order of $O(N\alpha^2)$ (Table 27). Since no assumptions are made on the sampling density and that the choice of the local implicit function is approximate, the MPU algorithm does not provide any reconstruction guarantees.

1. Point-set scaling	$O(1)$
2. Octree subdivision	$O(\alpha^2), \alpha =$ constant
3. Weighting	$O(k^2), k$ neighbors
4. Local implicit functions fit	$O(N\alpha^2), N$ points

Table 27 Complexity of the Multi-Level Partition of Unity algorithm.

The major issue about the MPU algorithm is that it reconstructs an approximation to the underlying curve/surface with piecewise models which might be a mixture of different implicit forms. This does not answer the requirements for a metrology application where form and dimensions need to be evaluated.

3.d Poisson

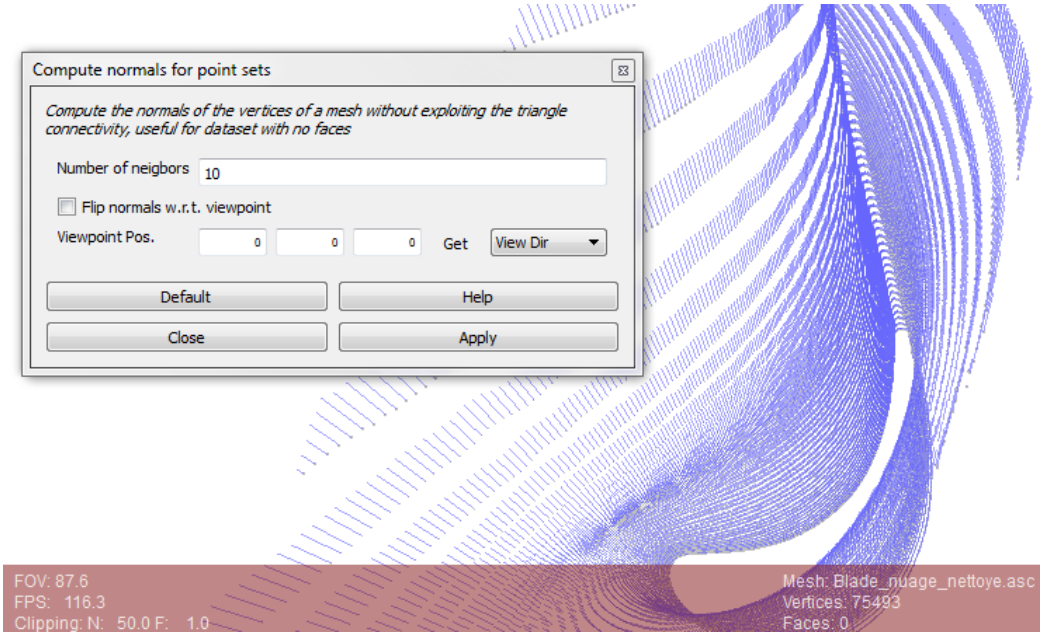
The Poisson surface reconstruction method is the mostly known implicit technique [218,227] and is implemented in both MeshLab and CGAL [183,196]. This method, introduced by Kazhdan *et al* [181], is remarkably robust to noise, however, at the expense of solving the problem in a global approach and processing data all at once. The output surface is expressed as the solution to a Poisson equation and is obtained by extracting the iso-surface from an implicit function f_i. More specifically, f_i is an indicator function that distinguishes outside ($f_i = 0$) from inside spaces of the surface ($f_i = 1$). While the value of f_i is constant on the outside and inside of the surface, its gradient is null in these regions. The gradient of f_i is thus only different from zero at the surface level and is nothing but the normal vector at that location on the surface. This reduces to solving for f_i by minimizing the following functional:

$$\min_{f_i} \left\| \Delta f_i - \nabla . \vec{N} \right\| \tag{50}$$

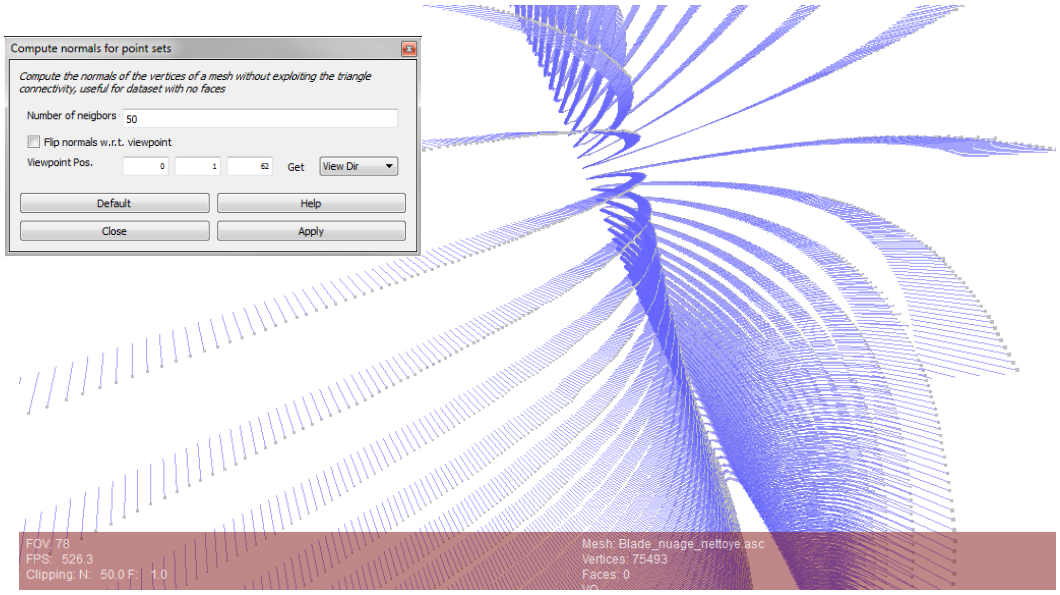
where $\vec{N}$ is the normal vector field.

The solution to the Poisson problem in (50) is obtained when the divergence of f_i and $\nabla . \vec{N}$ tend to equality, i.e., $\Delta f_i = \nabla . \vec{N}$. The algorithm works well in practice, however, with no geometrical or topological guarantees due to the fact that no assumptions on sampling density are made. The surface must necessarily be closed so that the notions of interior and exterior are meaningful. There exists an enhanced version of the Poisson mesh reconstruction algorithm for which the space of points is subdivided and all clusters are solved simultaneously. It is the parallel Poisson mesh reconstruction using an Octree subdivision [227] and is the version currently implemented in MeshLab. We tried out this algorithm for both the aspherical lens and the turbine blade datasets. Since this method needs to identify interior from exterior regions to the surface, the reconstruction occurs in two steps. In the first step, we use a MeshLab routine to approximate the normal orientation of the points. This routine cannot be successful without user input on the number of neighboring points to

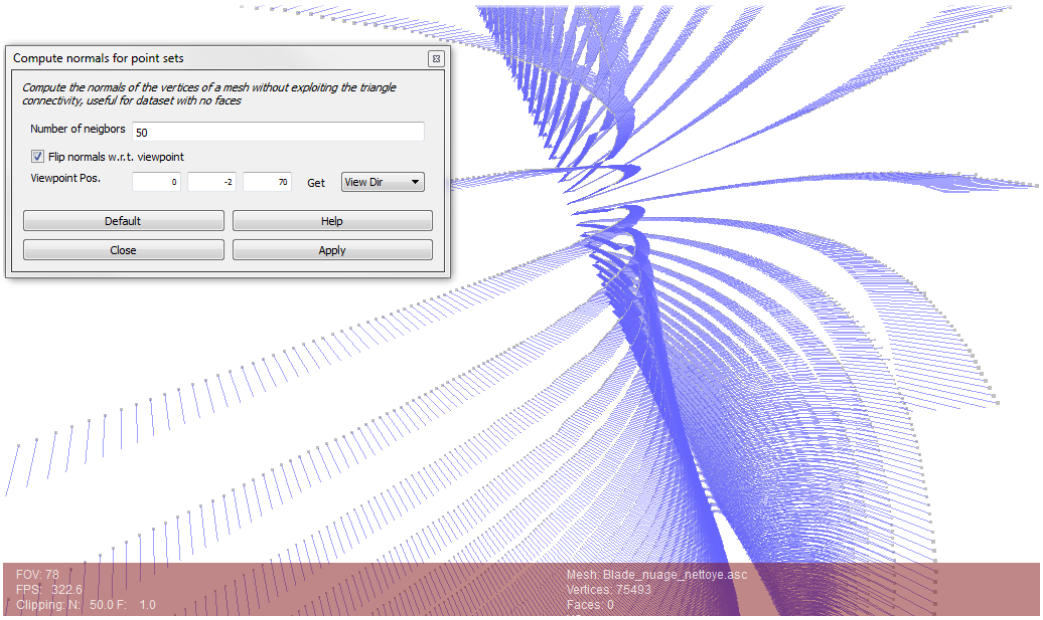
consider in the normal estimation and the possibility to re-orient normal directions according to a universal interior point selected by the user (Fig. 149).



(a)



(b)



(c)

Fig. 149 Normal estimation of the point-set of the turbine blade: (a) based on 10 neighbors without any re-orientation; (b) based on 50 neighbors and a re-orientation routine about an exterior point $(0, 1, 62)$; (c) based on 50 neighbors and a re-orientation routine about an interior point $(0, -2, 70)$.

In the second step, we proceed to the mesh reconstruction of the blade with the default parameters suggested by MeshLab (Fig. 150). The reconstruction of the blade is obviously geometrically not close to the underlying surface of the blade although being manifold.

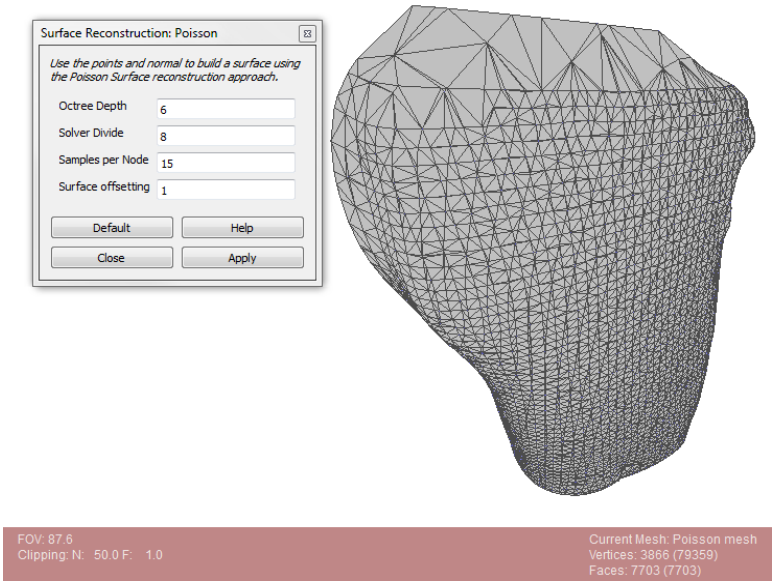


Fig. 150 Poisson mesh reconstruction using default settings on MeshLab.

We also try out this algorithm on a subset of the aspherical lens and find similar results (Fig. 151). We further conclude that datasets having more than about 300,000 points cannot be handled by the Poisson reconstruction algorithm in MeshLab as the number of points considered for the following example is around 90,000.

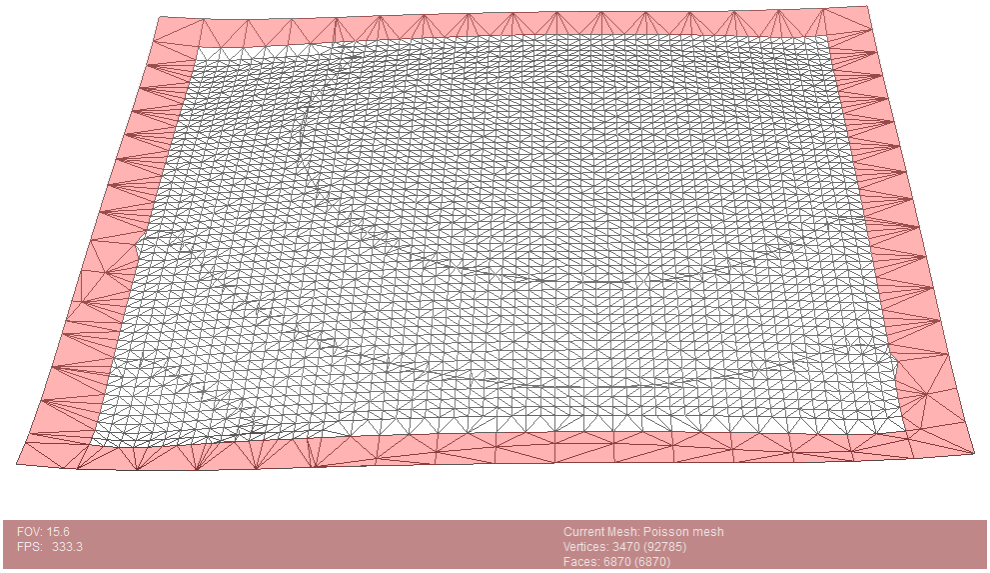


Fig. 151 Aspherical lens Poisson reconstruction which works fine at the points but adds an additional layer of triangles at the boundaries (highlighted large triangles).

4. Discussions

In this section, a thorough review of the mesh reconstruction algorithms that build a first order approximation of the surface has been achieved. The methods that are based on the Voronoi and Delaunay graphs are subject to an adaptive sampling condition which cannot be guaranteed in practice. Therefore, the reconstruction guarantees that these methods provide become insignificant.

Although the test on the aspherical lens shows that reconstruction using the Cocone approach with an ε-dense sampling of the data works (Fig. 136), it does not make the algorithm robust. For instance, Fig. 134 shows a case where the ε-sampling density criterion is satisfied whereas the ε_0 lower bound is not. For the special case of aspherical lenses and axis-symmetric surfaces, a 3D mesh reconstruction approach (Cocone, N.N.I, etc…) can be replaced by a 2D approach relying only on the Delaunay triangulation only. These surfaces have the particularity of being open surfaces that can be mapped, using a bijection, onto the plane that is normal to the axis of symmetry without loss of information or topological

superposition (Fig. 152). Otherwise, such as in the case of Fig. 152d, the right hand side portion of the curve will generate a surjection on the projection line in Fig. 152e and will create a superposition of some curve parts. The process consists of mapping the data points on a plane and performing the 2D Delaunay triangulation on the mapped points. This will surely generate a manifold mesh.

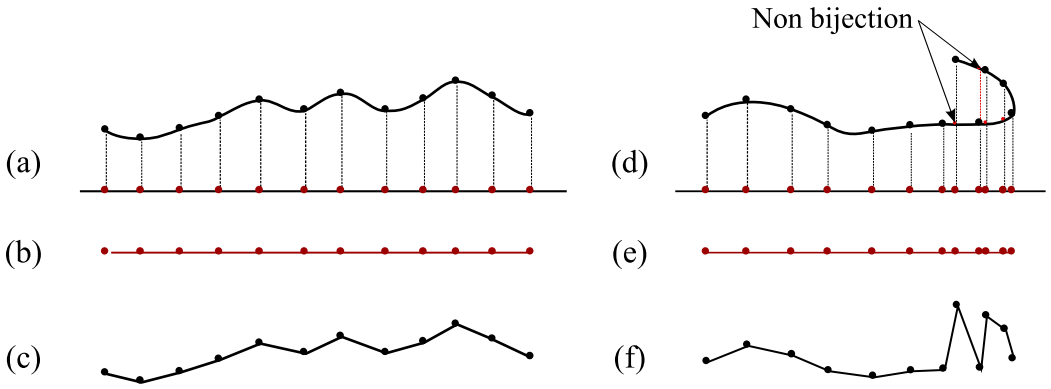


Fig. 152 Curve mapping on a straight line. (a),(d) Selected points mapping; (b),(e) Meshing of the mapped points; (c),(f) Inverse mapping function.

Fig. 153 shows the mesh reconstruction of points that do not satisfy the ε-sampling condition using this procedure calling a 2D Delaunay triangulation of the mapped points. The mesh is manifold and complete and this proves the claim that the ε-sampling condition is a necessary but not sufficient condition.

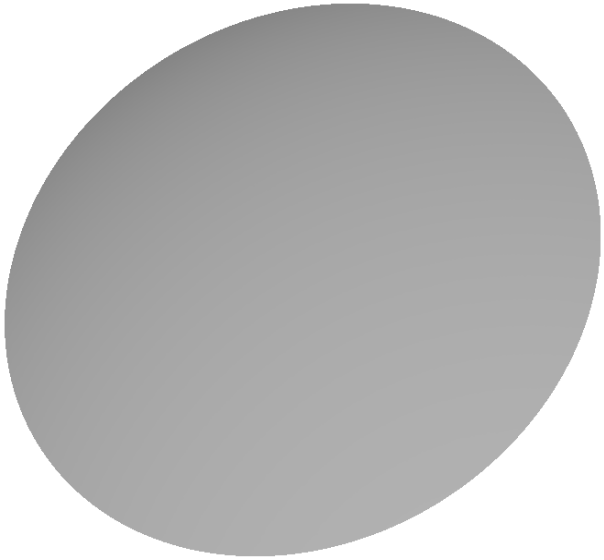


Fig. 153 Aspherical lens reconstruction using a 2D Delaunay approach with data that do not respect the ε-sampling condition.

Due to its complicated shape, the turbine blade cannot be mapped onto a 2D plane as described in Fig. 152, thus the only possible mesh reconstruction is a 3D one. We can recall from Fig. 135 that a reconstruction using the Cocone algorithm fails at sharp edges and where the sampling is inappropriate. For our application, if the mesh procedure is not guaranteed and does not solve the problem, especially that the complexity is of the order of $O(N^2)$, the above mesh reconstruction techniques are not adequate and must be avoided. Despite the fact that it creates a structure on the points when those are initially unorganized, a mesh is not a perfect descriptor of form and the classical framework for freeform surface reconstruction we evoked at the beginning of this chapter must be followed truthfully (Fig. 102). In this scheme, meshes are transformed to smooth representations by associating parametric surface patches among which the continuity is an additional difficulty.

While meshing techniques relying on combinatorial structures need additional steps for smoothing the data, implicit methods allow for that but the surface is always approximated by discrete data. Although they seem to work well in practice, especially in the presence of noise, outliers and even sharp features, implicit mesh reconstruction techniques mainly lack of reconstruction guarantees except for the Level-set method. Plus, they rely on the knowledge of point-set orientation and that is not achievable in an automatic way when data are unorganized. Unfortunately, for most implicit techniques studied in this thesis except for the Poisson's method, no implementation was available in commercial software. The Poisson reconstruction does not output a mesh with geometrical equivalence to the underlying surface. The requirement of most methods about normal orientation of the dataset has also been proven not to be straightforward in MeshLab.

Provided that the freeform surface reconstruction process consists of more operations when passing through the meshing phase, and that mesh reconstruction is complex, not robust, and lacks of guarantees in practice, we find it more legitimate, for our applications, to put the effort into an approach that discards the meshing phase and seeks the direct fitting of a single parametric surface (not in the form of patches) to the points. We therefore bring up the state-of-the-art techniques that exist in this field and then propose a Discrete B-Spline Active Contour Deformation (DBACD) algorithm. We start by a demonstration on curves in the plane and then pass on to an application on surfaces.

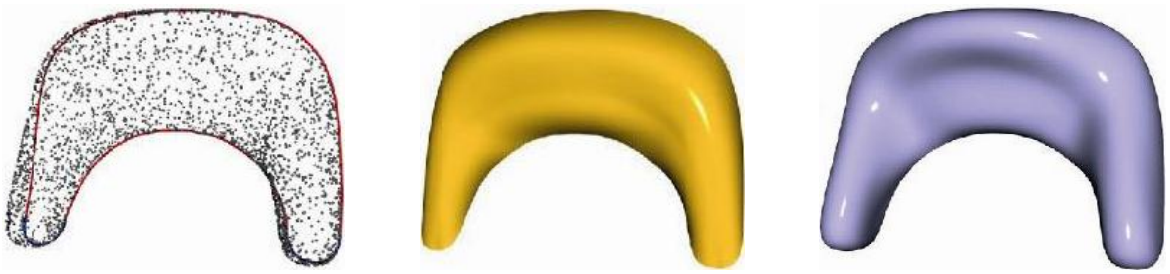
III. Discrete B-Spline Active Contour Deformation (DBACD)

1. Active contour deformation

The principle of active contour deformation originates from the works of Kass *et al* [228] on the detection of image contours. The concept was adapted later to many other applications such as implicit surface reconstruction, algorithmic geometry, computer aided geometry design algorithms, etc… [229]. We cite the Level-set method [175] described earlier in section II. 3.b , the geometric convection algorithm [173] described in section II. 2.f , and the active B-Spline curves and surfaces approximation of Pottmann *et al* [230]. Active contour deformation is the process of evolving a base surface around a given dataset so that it approximates the data points without necessarily having any knowledge about the underlying shape of the data points. The surface continuously deforms and progresses along a given direction (not necessarily the gradient direction) and gets closer to the desired shape along the iterations.

The algorithm that we propose here is a discrete active contour deformation algorithm applied to closed B-Spline curves and surfaces. A discrete approach makes the distance calculations (point-to-segment in the case of curves and point-to-triangle in the case of surfaces) much faster than distance calculations using a continuous model. Indeed, for continuous and smooth models, location parameters must be initialized and then optimized before computing the data points projections. One can obtain different approximation results by modifying the initial parameterization. The initialization of both the location parameters and the surface is usually cumbersome as it is not easy to determine a relatively *good* initial surface especially when the shape of the surface is unknown. Initial parameterization techniques can be found in [148,159,230–235]. According to Kineri *et al* [232], initial parameterization can be different based on whether the problem is an interpolation or an approximation. In interpolation problems, the centripetal method or the chord length method can be used [167]. When it comes to approximation problems in which data are randomly distributed, the above initialization methods do not apply because no order on the data is assumed. Here instead, a base surface (BS) is generated according to the method of Ma *et al* [233] by approximating four boundary curves with the smallest number of control points possible (Fig. 154a). The boundary curves are fitted from points digitized for this purpose and form a tensor product surface which is nothing but BS (Fig. 154b). For complex shapes, Kineri *et al* use the Dynamic Base Surface method introduced by Azariadis to construct the base surface [236].

However, this method assumes the existence of four boundary curves and this is not straightforward when shapes get more complex. Provided that BS is constructed, the parameterization is done by projecting the data points onto BS along the normal to BS or some optimized direction with respect to BS. Then the error vector between a data point and its projection on the base surface is calculated and the sum of squared errors is minimized and the solution at each iteration gives the amount by which each control point of the B-Spline surface must translate in order to approach the final shape (Fig. 154c).



(a) (b) (c)

Fig. 154 B-Spline surface fitting methodology of Kineri *et al* [232]. (a) input points with the four boundary curves approximation; (b) base surface; (c) resulting B-Spline surface.

In a different work, Minh and Forbes describe a geometrical approach to parameters initialization for CAD models fitting but then they assume that a reference surface (CAD) exists [237]. They propose that the initialization be done based on combinatorial structures information given by the Delaunay triangulation of sampled points on the reference model. The reference surface is sampled and a Delaunay triangulation is built. Points are organized thanks to the Delaunay triangulation, and the matching of a data point to a sampled point gives a footpoint (parameter) initialization. Yet, the necessity to have a reference model is a limitation because in Reverse Engineering applications the model is not always known. In the works of Zheng *et al* [159], where a L-BFGS algorithm for B-Spline curve fitting is proposed, the automatic generation of the initial curve is out of the scope of the paper. Authors rather start with a hand-drawn curve which somehow follows the shape of the underlying curve of the dataset. Hereafter, a sufficiently dense sampling of the curve is performed so that location parameters are found fast and as close to optimum as possible. Zheng *et al* follow the scheme proposed by Wang *et al* [235] which relies on a Gauss-Newton method for iteratively computing footpoints. Since the initial curve is very close to the dataset the authors assume that footpoints will not change across the iterations. Unless the residual errors after a L-BFGS

fitting is larger than a threshold value, the footpoints computation is only performed once. This suggests that the location parameters update and the distances minimization occur simultaneously.

As parameterization seems cumbersome, different approaches such as the works of Pottmann *et al* [230] and Yang *et al* [238] deal with the direct approximation of point-sets, curves and surfaces by active B-Spline models using the Squared Distance Minimization method (SDM). The advantage of SDM is that it does not require computing a parameterization of the data points in case of point-set approximation. As an alternative, a technique using local quadratic approximants of the squared distance function is used to help moving the active B-Spline to lower levels of squared distance without having to specify which point of the active B-Spline should move to which point in the dataset. Nevertheless, the latter technique requires the calculation of discrete curvatures and consequently assumes that the points are dense and organized. Because it is based on local optimization, the SDM method's sensitivity with respect to the initial B-Spline curve/surface is not negligible. To address this issue, Yang et al [238] propose a procedure to insert and remove knots and adjust the locations of control points when necessary. The SDM method with those added improvements appears to work well in practice but still assumes that the data are organized.

From what has just been said, and the observation that curve/surface initialization as well as parameters initialization are very critical, we come to propose a new approach that consists of disregarding the issues of a necessary good initialization as well as parameterization. For that we test one initialization that starts with a curve for which the control points are located on the circumference of a circle around the data and another initialization that starts from an offset of the bounding box of the dataset. The originality of our algorithm is that it fits a B-Spline curve to a cloud of points which can be unorganized and noisy, using a topological association and a subsequent distance calculation which does not need any parameterization methodology. Additionally, working with a discrete form of the B-Spline makes the processing very fast as differential calculations and footpoints projections, which are normally time consuming on smooth parametric models, are not required anymore. Our method does not entail any organization of the data points.

2. Planar active contour deformation with a B-Spline curve

2.a Initialization

A B-Spline is defined by three elements. The degree of the curve, a knot vector and a control polygon. Cubic B-Spline curves are usually enough to represent complex shapes, therefore we choose a degree $d = 3$. The control polygon consists of points that locally control the shape of the curve. By moving a control point, the curve is locally tailored within the knot intervals that are under the influence of that control point. Therefore, the idea here is to test both initialization configurations of a B-Spline control polygon around the given data points in the plane. Since we deal with closed profiles, periodic control points are needed. So if we geometrically want n_C control points to start with, $n_C + d$ control points are implicitly needed for periodicity issues.

The initial *base* knot vector is deduced from the initial non-periodic control polygon with the degree specified. In the *base* knot vector, there are $n_C + 1$ knots and n_C knot intervals. Therefore there are as many knot intervals as non-periodic control points. Taking into account periodicity, $2 \times d$ additional knots are needed in such a way that d knot intervals are appended at each end of the *base* knot vector and that they are equal in pairs as illustrated in Fig. 155. The initial periodic knot vector consists of an ordered uniform sequence of parameters which satisfy (51):

$$u_i{}^{(0)} = \frac{i - d}{n_C}\ ,\ \ 0 \le i \le n_C + 2d \qquad (51)$$

Fig. 155 shows a periodic knot vector for $d = 3$ and $n_C = 10$. In fact, what actually determine periodicity are the knot intervals between periodic knots. The key is to always make sure that the intervals, from inner-most to outer-most, are pairwise equal, i.e., the respective green, blue and yellow pairs of knot interval are equal.

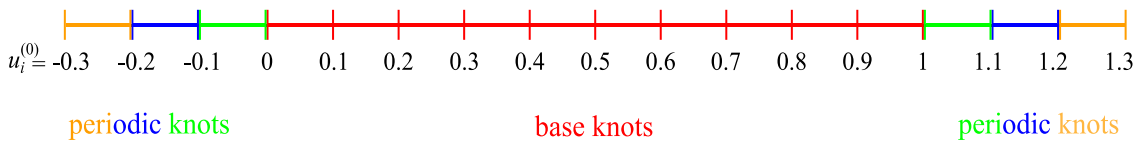
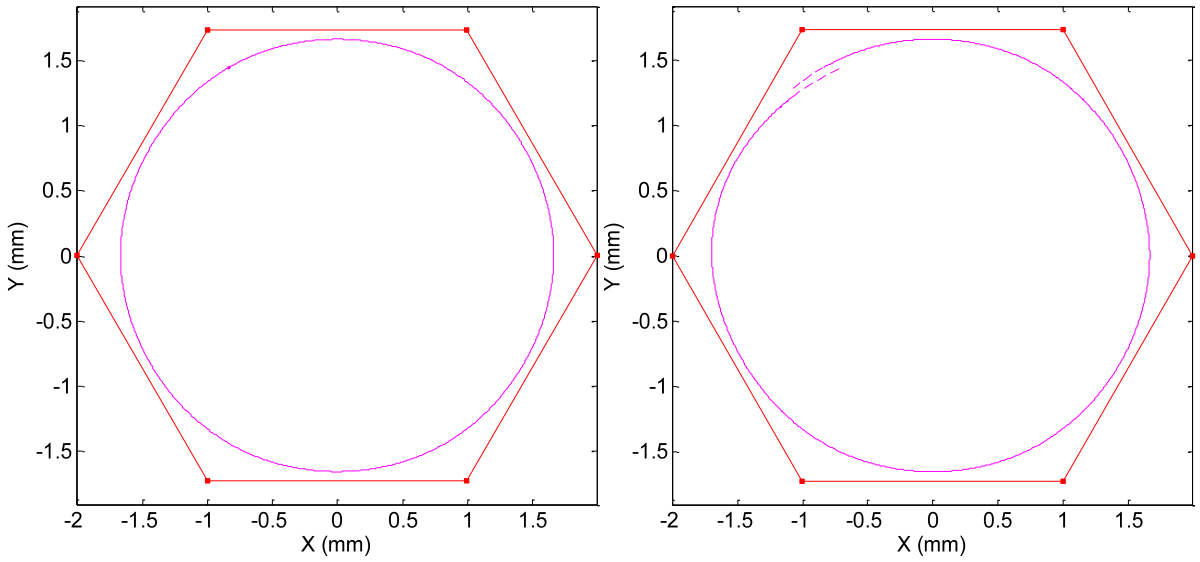


Fig. 155 The initial periodic knot vector denoted by $\tau^{(0)} = \{u_0^{(0)}, \cdots, u_{n_{c+2d}}^{(0)}\}$.

If the previous rule of conserving the pairwise equality between periodic knot intervals is not satisfied, the closure of the B-Spline curve is not properly generated (Fig. 156b).



(a) (b)

Fig. 156 Initial periodic B-Spline curve (magenta): (a) knot vector respecting the rule on the periodic knot intervals; (b) knot vector failing the rule on the periodic knot intervals.

2.b Subdivision points

The piecewise linear model that we use results from a subdivision of the control polygon of the B-Spline obtained using the Oslo algorithm which proceeds on the basis of global knot insertion [239]. We denote by L the number of global insertions. Then, the number of subdivided points depends on L and the number of control points ($n_S = 2^L \times n_C$). So if we choose one level of insertion, i.e. $L = 1$, it means that we will only insert knots at the middle of all the *base* knots intervals once and that the number of subdivided points will be double. The relationship between control points and subdivided points, in matrix form, is governed by equation (52) and shown in Fig. 157. In this plot, the blue polygon is the subdivided version of the initial red control polygon.

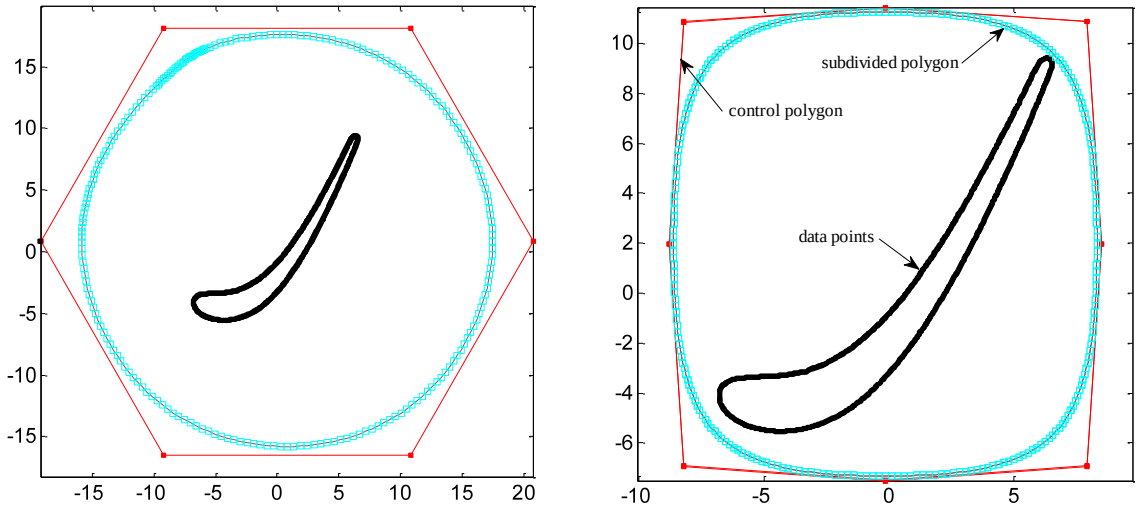
$$\mathbf{Q}^{(k)} = \mathbf{M}^{(k)}\mathbf{C}^{(k)} \tag{52}$$

where,

$$\mathbf{C}^{(k)} = \left(c_1^{(k)}, c_2^{(k)}, \ldots, c_{n_C}^{(k)}\right)^{\mathrm{T}} = \begin{pmatrix} c_{x_1}^{(k)} & c_{y_1}^{(k)} \\ c_{x_2}^{(k)} & c_{y_2}^{(k)} \\ & \vdots \\ c_{x_{n_C}}^{(k)} & c_{y_{n_C}}^{(k)} \end{pmatrix},$$

$$\mathbf{Q}^{(k)} = \left(q_1^{(k)}, q_2^{(k)}, \ldots, q_{n_S}^{(k)}\right)^{\mathrm{T}} = \begin{pmatrix} q_{x_1}^{(k)} & q_{y_1}^{(k)} \\ q_{x_2}^{(k)} & q_{y_2}^{(k)} \\ & \vdots \\ q_{x_{n_S}}^{(k)} & q_{y_{n_S}}^{(k)} \end{pmatrix},$$

and $\mathbf{M}^{(k)}$ is the $n_S \times n_C$ subdivision matrix that computes the subdivision of the control points at iteration k. As long as there are no local knot insertions and the addition of control points in $\mathbf{C}^{(k)}$ (c.f. section 2.e), $\mathbf{M}^{(k)}$ maintains a constant size. Since the control points will move at each iteration, the subdivided points will also follow according to $\mathbf{M}^{(k)}$ (52).



(a) (b)

Fig. 157 Initialization of the B-Spline piecewise-linear curve. (a) Initialization based on a generic circular form around the data; (b) Initialization based on an offset of the bounding box of the dataset.

2.c Point-segment association for distance calculations

The subdivided polygon $\mathbf{Q}^{(k)}$ is piecewise linear. The distances $\boldsymbol{\delta}^{(k)}$ are point-to-segment distances from the data points $\mathbf{P}$ to the segments of $\mathbf{Q}^{(k)}$, denoted by $\mathrm{Seg}\left(\mathbf{Q}^{(k)}\right)$. Nonetheless, these distances are not just geometric Euclidean distances. Since the initialization of the B-

Spline curve is generic, it might highly occur that geometrically nearest segments of $\mathbf{Q}^{(k)}$ to $\mathbf{P}$ are topologically incorrect (Fig. 158). Point-to-segment correspondences might be erroneous and be established on wrong sides. Therefore, a heuristic is applied and builds a correspondence between data points and B-Spline segments in $\mathbf{Q}^{(k)}$ with a preservation of topology. The order of the data points is initially unknown, but their topology can be coarsely set by constructing a polygonal mesh based on the topology of $\mathrm{Seg}(\mathbf{Q}^{(k)})$. A procedure is therefore introduced and consists of an *inverse* association phase (Fig. 159) followed by a *direct* association phase (Fig. 160).

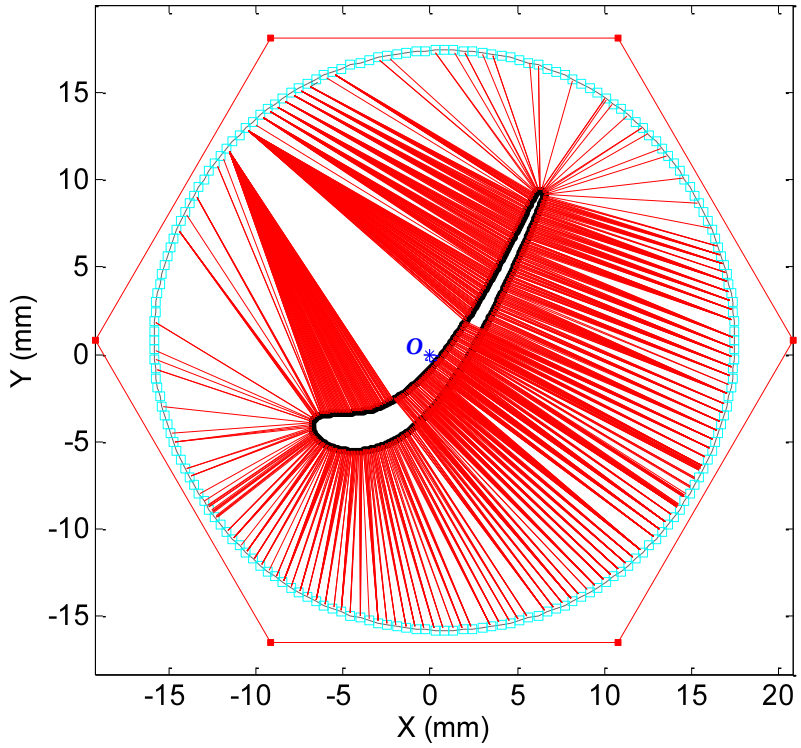


Fig. 158 Some topologically incorrect point-to-segment associations (O is the center of the circle circumscribing the control points).

The *inverse* association routine is the part of the algorithm that guarantees the preservation of the topology of the points. A surjective mapping of the discrete points $q \in \mathbf{Q}^{(k)}$ (blue squares in Fig. 159) to points in $\mathbf{P}$ (black dots) is created based on nearest neighbor search. The naïve algorithm of nearest neighbor search where the distances are computed from each point $q \in \mathbf{Q}^{(k)}$ to all points $p \in \mathbf{P}$ is computationally expensive ($O(n^2)$). To avoid this we apply a uniform point space partitioning strategy which reduces computational time by at least a factor of 10. Other optimized point space partitioning

methods exist such as Octree [203] or kD-tree [240] but are not yet integrated in our algorithm. Since the points of $\mathbf{Q}^{(k)}$ are ordered, their respective nearest points in $\mathbf{P}$ (magenta dots in Fig. 159) are also ordered. This results in a surjection function between $\mathrm{Seg}\left(\mathbf{Q}^{(k)}\right)$ and $\mathrm{Seg}(\mathbf{P})^{(k)}$ (magenta segmentation in Fig. 159).

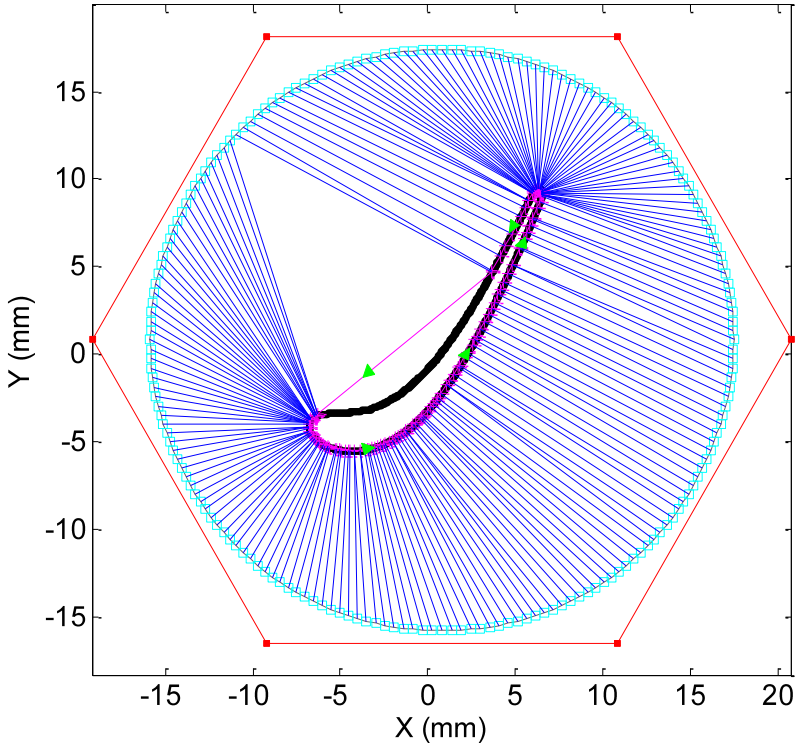


Fig. 159 Inverse association phase (blue) building a certain structure and orientation on the point-set (magenta segments): calculating the distance from each point in $\mathbf{Q}^{(k)}$ to the nearest data point in $\mathbf{P}$.

The *direct* association routine is the part of the algorithm that matches data points to subdivided polygon segments in $\mathrm{Seg}\left(\mathbf{Q}^{(k)}\right)$ (Fig. 160). As a consequence of the segmentation built on the points, each data point $p \in \mathbf{P}$ is firstly associated to its nearest segment in $\mathrm{Seg}(\mathbf{P})^{(k)}$. The order on $\mathbf{Q}^{(k)}$ which is consequent on $\mathrm{Seg}(\mathbf{P})^{(k)}$ insures that $\mathrm{Seg}(\mathbf{P})^{(k)}$ has a well-known orientation (green arrows in Fig. 159). This orientation helps increasing the probability of having perfect point-to-segment associations. It consists of making the association of a data point to a segment in $\mathrm{Seg}(\mathbf{P})^{(k)}$ robust, i.e. the data points in concave regions, for example, will be associated to the nearest segment that is consistently oriented.
Then, because of the surjection between $\mathrm{Seg}\left(\mathbf{Q}^{(k)}\right)$ and $\mathrm{Seg}(\mathbf{P})^{(k)}$, each point $p \in \mathbf{P}$ can be associated to a segment in $\mathrm{Seg}\left(\mathbf{Q}^{(k)}\right)$. It might occur that the actual correspondence is not

optimal, so we tolerate the search for the minimum distance also over the previous and the next segments of the associated segment in $\mathrm{Seg}(\mathbf{Q}^{(k)})$.

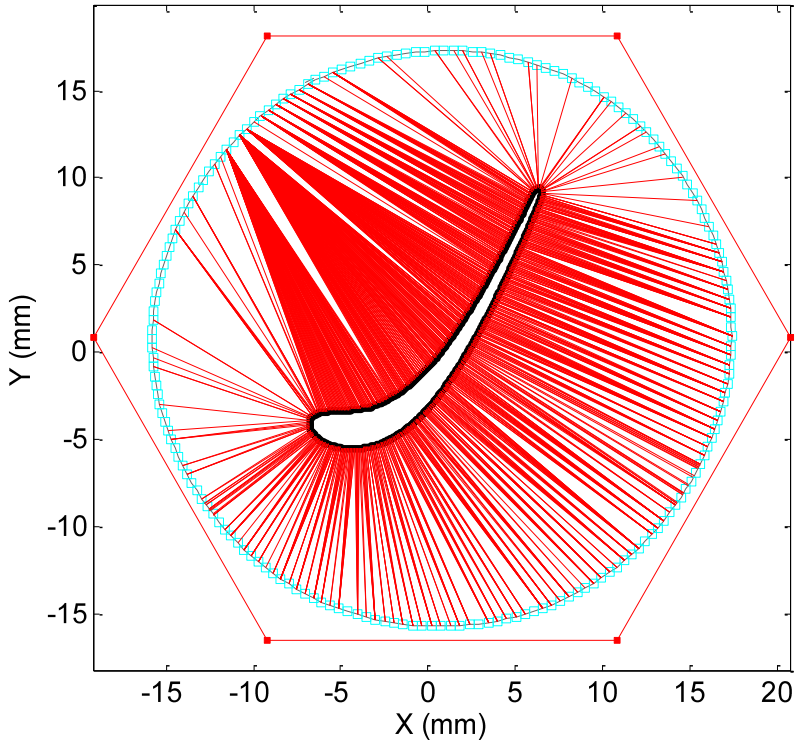


Fig. 160 Direct association phase (red) deduced from the inverse association phase.

2.d Active contour deformation scheme

The proposed association process between datapoints and subdivided control polygon segments establishes a good correspondence between both sets. We carried out many tests that revealed that this association method infers geometrically and topologically correct distances to be passed on to optimization problem schematized in Fig. 161.

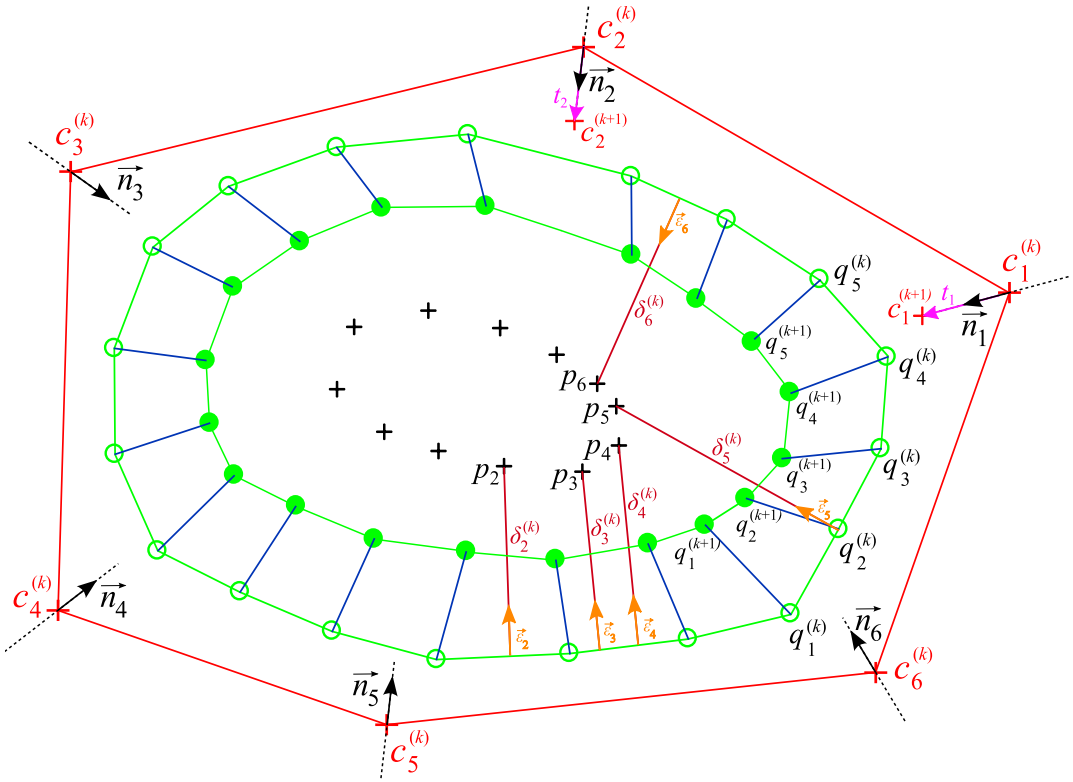


Fig. 161 Methodology of the DBACD algorithm. The control polygon is represented in red, normal directions at all control points c_j are the vectors denoted by $\vec{n}_j$, $\delta_i^{(k)}$ are the computed point-to-segment distances at iteration k, $\vec{\varepsilon}_i$ are the normalized distance vectors and t_j are the magnitudes by which the control points must be translated.

Optimization in the normal direction of the control points

The objective function to minimize (53) is the sum of squared differences between the n_d distances separating the data points in $\mathbf{P}$ from their respective segments in $\text{Seg}\left(\mathbf{Q}^{(k)}\right)$ and the B-Spline curve evolution between iteration k and $k+1$. A control point $c_j^{(k)} \in \mathbf{C}^{(k)}$ is allowed to move in the direction of its normal $\vec{n}_j^{(k)}$ updated at each iteration. It follows that the new control points are obtained such that $c_j^{(k+1)} = c_j^{(k)} + t_j^{(k)}\vec{n}_j^{(k)}$, where $t_j^{(k)}$ is a scalar. $\vec{n}_j^{(k)}$ for $j = 1, 2, \ldots, n_C$ is taken to be the bisector of the adjacent segments to $c_j^{(k)}$ at iteration k.

$$\min_{t_1^{(k)},\dots,t_{n_c}^{(k)}} \sum_{i=1}^{n_d}\sum_{j=1}^{n_c} \left(\langle\left(q_i^{(k+1)} - q_i^{(k)}\right),\ \vec{\varepsilon}_i^{(k)}\rangle - \delta_i^{(k)}\right)^2 \tag{53}$$

where $\vec{\varepsilon}_i^{(k)}$ is the unit vector of the i^{th} distance separating a data point from its associated segment in Seg $\left(\mathbf{Q}^{(k)}\right)$. Equation (53) written in matrix form gives:

$$\min_{\mathbf{t}^{(k)}} \left\| \langle\left(\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\right), \vec{\boldsymbol{\varepsilon}}^{(k)}\rangle - \boldsymbol{\delta}^{(k)} \right\|_2^2, \tag{54}$$

with $\mathbf{t}^{(k)} = \begin{bmatrix} t_1^{(k)} \\ t_2^{(k)} \\ \vdots \\ t_{n_c}^{(k)} \end{bmatrix}$, $\vec{\boldsymbol{\varepsilon}}^{(k)} = \begin{bmatrix} \vec{\varepsilon}_1^{(k)} \\ \vec{\varepsilon}_2^{(k)} \\ \vdots \\ \vec{\varepsilon}_{n_d}^{(k)} \end{bmatrix}$ and $\boldsymbol{\delta}^{(k)} = \begin{bmatrix} \delta_1^{(k)} \\ \delta_2^{(k)} \\ \vdots \\ \delta_{n_d}^{(k)} \end{bmatrix}$.

Recall the relationship between the control points and their subdivision points given in (52), we get:

$$\min_{\mathbf{t}^{(k)}} \left\| \langle\left(\mathbf{M}^{(k)}\mathbf{C}^{(k+1)} - \mathbf{M}^{(k)}\mathbf{C}^{(k)}\right), \vec{\boldsymbol{\varepsilon}}^{(k)}\rangle - \boldsymbol{\delta}^{(k)} \right\|_2^2 \tag{55}$$

What we need to find is the required translations that would move the control points $\mathbf{C}^{(k)}$, along their normal directions, from a current position to a new position while insuring that $\mathbf{C}^{(k+1)}$ brings the curve closer to the data points after each iteration (56):

$$\mathbf{C}^{(k+1)} = \mathbf{C}^{(k)} + \mathbf{t}^{(k)}\mathbf{I}_{n_c}\mathbf{N}^{(k)}, \tag{56}$$

where $\mathbf{N}^{(k)} = \left(\vec{n}_1^{(k)}, \cdots, \vec{n}_{n_C}^{(k)}\right)^{\mathrm{T}}$ and $\mathbf{I}_{n_c} = \mathrm{diag}\left(\underbrace{1, \cdots, 1}_{n_c}\right)$

Consequently, by blending (55) and (56) gives an expression of the objective function in terms of the solution vector $\mathbf{t}^{(k)}$ as follows (57):

$$\min_{\mathbf{t}^{(k)}} \left\| \langle \mathbf{M}^{(k)}\mathbf{t}^{(k)}\mathbf{I}_{n_c}\mathbf{N}^{(k)}, \vec{\boldsymbol{\varepsilon}}^{(k)} \rangle - \boldsymbol{\delta}^{(k)} \right\|_2^2$$
$$\min_{\mathbf{t}^{(k)}} \left\| \mathbf{A}^{(k)}\mathbf{t}^{(k)} - \boldsymbol{\delta}^{(k)} \right\|_2^2 \tag{57}$$

where $\mathbf{A}^{(k)} =$

$$\begin{bmatrix} m_{1,1}\left(\vec{\varepsilon}_1^{(k)}.\vec{n}_1^{(k)}\right) + m_{1,2}\left(\vec{\varepsilon}_1^{(k)}.\vec{n}_2^{(k)}\right) + \cdots + m_{1,n_C}\left(\vec{\varepsilon}_1^{(k)}.\vec{n}_{n_C}^{(k)}\right) \\ m_{2,1}\left(\vec{\varepsilon}_1^{(k)}.\vec{n}_1^{(k)}\right) + m_{2,2}\left(\vec{\varepsilon}_1^{(k)}.\vec{n}_2^{(k)}\right) + \cdots + m_{2,n_C}\left(\vec{\varepsilon}_1^{(k)}.\vec{n}_{n_C}^{(k)}\right) \\ \vdots \\ m_{i,1}\left(\vec{\varepsilon}_i^{(k)}.\vec{n}_1^{(k)}\right) + m_{i,2}\left(\vec{\varepsilon}_i^{(k)}.\vec{n}_2^{(k)}\right) + \cdots + m_{i,n_C}\left(\vec{\varepsilon}_i^{(k)}.\vec{n}_{n_C}^{(k)}\right) \\ m_{i+1,1}\left(\vec{\varepsilon}_i^{(k)}.\vec{n}_1^{(k)}\right) + m_{i+1,2}\left(\vec{\varepsilon}_i^{(k)}.\vec{n}_2^{(k)}\right) + \cdots + m_{i+1,n_C}\left(\vec{\varepsilon}_i^{(k)}.\vec{n}_{n_C}^{(k)}\right) \\ \vdots \\ m_{n_S,1}\left(\vec{\varepsilon}_{n_S}^{(k)}.\vec{n}_1^{(k)}\right) + m_{n_S,2}\left(\vec{\varepsilon}_{n_S}^{(k)}.\vec{n}_2^{(k)}\right) + \cdots + m_{n_S,n_C}\left(\vec{\varepsilon}_{n_S}^{(k)}.\vec{n}_{n_C}^{(k)}\right) \\ m_{1,1}\left(\vec{\varepsilon}_{n_S}^{(k)}.\vec{n}_1^{(k)}\right) + m_{1,2}\left(\vec{\varepsilon}_{n_S}^{(k)}.\vec{n}_2^{(k)}\right) + \cdots + m_{1,n_C}\left(\vec{\varepsilon}_{n_S}^{(k)}.\vec{n}_{n_C}^{(k)}\right) \end{bmatrix} \tag{58}$$

and $m_{i,j}$ are the elements of the subdivision matrix $\mathbf{M}^{(k)}$.

Here, the solution $\mathbf{t}^{(k)}$ is computed by solving the linear system of equations written in normal form in function of the optimization matrix $\mathbf{A}^{(k)}$ at iteration k. Note that $\mathbf{A}^{(k)}$ is an augmented matrix that includes for each point-to-segment distance δ_i a pair of entries corresponding to each of the vertices of the current segment $q_i^{(k)}$ and $q_{i+1}^{(k)}$ (58).

$$\left(\mathbf{A}^{(k)}\right)^{\mathrm{T}}\mathbf{A}^{(k)}\mathbf{t}^{(k)} = \left(\mathbf{A}^{(k)}\right)^{\mathrm{T}}\boldsymbol{\delta}^{(k)}. \tag{59}$$

Solving the system in a Least-Squares sense produces a vector of scalars by which the control points $\mathbf{C}^{(k)}$ of the curve must move in order to approach the dataset. Each control point is translated by an amount proportional to $t_j^{(k)}$, $\alpha_j^{(k)}t_j^{(k)}$, where $\alpha_j^{(k)} = t_j^{(k)}/\max\left(\mathbf{t}^{(k)}\right)$ is a vector of scalars between 0 and 1 proportional to $t_j^{(k)}$ and which can be assimilated to the step of gradient descent in usual optimization algorithms (Gauss-Newton, gradient descent, etc…). This choice of $\alpha_j^{(k)}$ is motivated by the fact that it slows down the deformation of the B-Spline curve, and by that allows to avoid overshoots and curve self-intersections.

Optimization in the separate x and y directions

Another interesting solution to the problem of DBACD is to add some more degrees of freedom to the problem by letting the control points move independently in x and y directions. In this case, the vector $\mathbf{t}^{(k)}$ is decomposed into two separate components, $\mathbf{t}_{\mathbf{x}}^{(k)} = \left(t_{1_x}^{(k)}, \dots, t_{n_{c_x}}^{(k)}\right)^{\mathrm{T}}$ and $\mathbf{t}_{\mathbf{y}}^{(k)} = \left(t_{1_y}^{(k)}, \dots, t_{n_{c_y}}^{(k)}\right)^{\mathrm{T}}$ and two systems are solved separately, one along the x-direction and another along the y-direction (60):

$$\min_{\mathbf{t}_{\mathbf{x}}^{(k)}} \sum_{i=1}^{n_d} \sum_{j=1}^{n_C} \left(\langle \left(q_{x_i}^{(k+1)} - q_{x_i}^{(k)} \right), \vec{\varepsilon}_{x_i}^{(k)} \rangle - \delta_i^{(k)} \vec{\varepsilon}_{x_i}^{(k)} \right)^2$$

and (60)

$$\min_{\mathbf{t}_{\mathbf{y}}^{(k)}} \sum_{i=1}^{n_d} \sum_{j=1}^{n_C} \left(\langle \left(q_{y_i}^{(k+1)} - q_{y_i}^{(k)} \right), \vec{\varepsilon}_{y_i}^{(k)} \rangle - \delta_i^{(k)} \vec{\varepsilon}_{y_i}^{(k)} \right)^2.$$

The suitable scheme for this alternate approach is illustrated in Fig. 162.

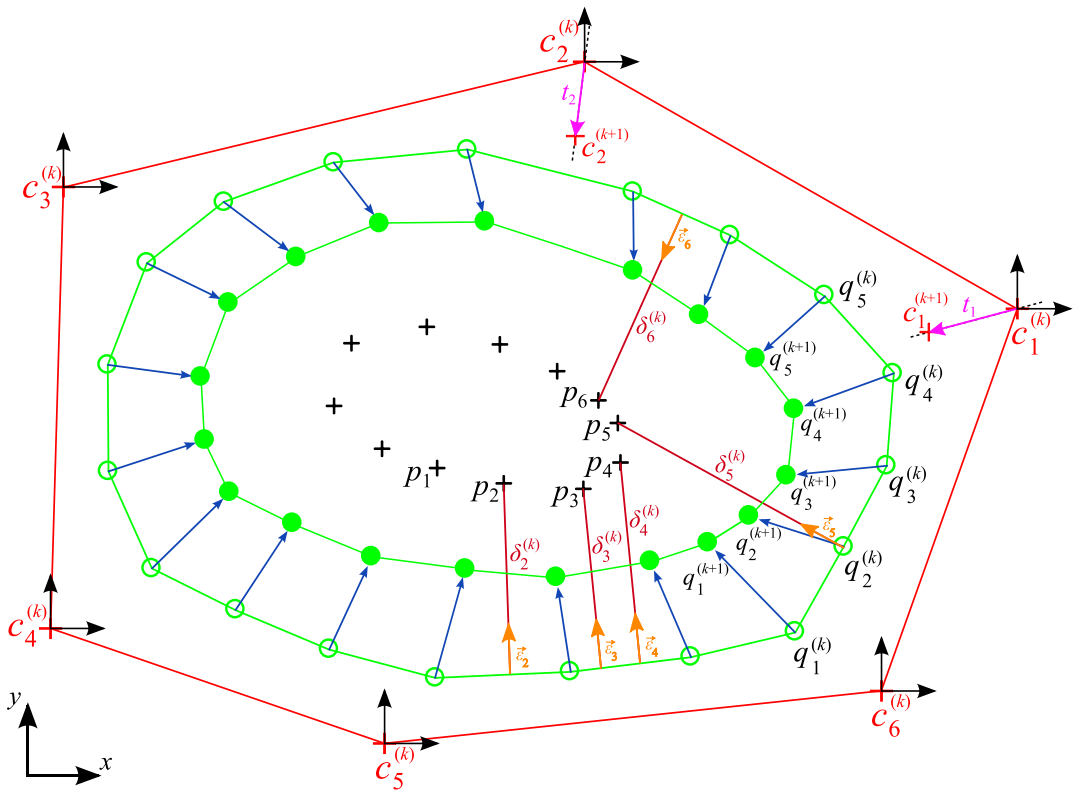


Fig. 162 Methodology of the DBACD algorithm by optimizing separately along x and y directions ($t_1 = (t_{1x}; t_{1y})$).

Written in matrix form the two systems to solve are as such:

$$\min_{\mathbf{t}_\mathbf{x}^{(k)}} \left\| \mathbf{A}_\mathbf{x}^{(k)} \mathbf{t}_\mathbf{x}^{(k)} - \boldsymbol{\delta}_\mathbf{x}^{(k)} \right\|_2^2,$$

and (61)

$$\min_{\mathbf{t}_\mathbf{y}^{(k)}} \left\| \mathbf{A}_\mathbf{y}^{(k)} \mathbf{t}_\mathbf{y}^{(k)} - \boldsymbol{\delta}_\mathbf{y}^{(k)} \right\|_2^2.$$

where $\mathbf{A}_\omega^{(k)} = \begin{bmatrix} m_{1,1}\left(\vec{\varepsilon}_{1_\omega}^{(k)}\right) + \cdots + m_{1,n_C}\left(\vec{\varepsilon}_{1_\omega}^{(k)}\right) \\ m_{2,1}\left(\vec{\varepsilon}_{1_\omega}^{(k)}\right) + \cdots + m_{2,n_C}\left(\vec{\varepsilon}_{1_\omega}^{(k)}\right) \\ \vdots \\ m_{i,1}\left(\vec{\varepsilon}_{i_\omega}^{(k)}\right) + \cdots + m_{i,n_C}\left(\vec{\varepsilon}_{i_\omega}^{(k)}\right) \\ m_{i+1,1}\left(\vec{\varepsilon}_{i_\omega}^{(k)}\right) + \cdots + m_{i+1,n_C}\left(\vec{\varepsilon}_{i_\omega}^{(k)}\right) \\ \vdots \\ m_{n_S,1}\left(\vec{\varepsilon}_{n_{S_\omega}}^{(k)}\right) + \cdots + m_{n_S,n_C}\left(\vec{\varepsilon}_{n_{S_\omega}}^{(k)}\right) \\ m_{1,1}\left(\vec{\varepsilon}_{n_{S_\omega}}^{(k)}\right) + \cdots + m_{1,n_C}\left(\vec{\varepsilon}_{n_{S_\omega}}^{(k)}\right) \end{bmatrix}$, with $\omega \in \{\mathrm{x}, \mathrm{y}\}$

2.e Knot insertion

Both systems presented above can be solved and each one of them iteratively gives out a solution for $\mathbf{t}^{(k)} = \left\{t_1^{(k)}, t_2^{(k)}, \ldots, t_{n_C}^{(k)}\right\}$ with a fixed number of control points n_C. Now it may occur that n_C is not always sufficient to reach a mean of errors below the specified tolerance ϵ. Therefore, the algorithm is designed to admit local knot insertion for adding control points. Knot insertion is applied locally where more flexibility is needed for the B-Spline curve to match high curvature regions. A control point is added at the locations where the distance between the B-Spline curve and the points is still larger than the specified threshold ϵ. We compute the value of the knot to be added in the knot vector by identifying the B-Spline segment with largest error and by calculating its position on the piece-wise linear curve. The ratio of the distance from the curve starting point to the identified segment over the length of the entire curve gives the knot value to be inserted in the knot vector. The position of the knot value within the knot vector translates into an insertion of an additional control point in the control polygon [167].

Knot insertion is a very well known process for B-Spline curves and surfaces and has been clearly detailed by Piegl and Tiller and Farin *et al* [167], [241]. The scheme is described as follows assuming that the knot value u to be added $\in [u_l,\ u_{l+1}[$ and that for a cubic B-Spline, the addition of a control point replaces 2 current control points by 3 new control points whilst the remaining ones are kept unchanged (62). Denoting by $\mathbf{C}^{new}$ the set of new control points after insertion we get that each one of them is a barycentric combination of 2 successive current control points (Fig. 163 and Fig. 164).

$$\begin{aligned} c_{l-d}^{new} &= c_{l-d} \\ c_j^{new} &= \rho_j c_j + (1-\rho_j) c_{j-1} \qquad l-d+1 \le j \le l \\ c_{l+1}^{new} &= c_l \end{aligned} \tag{62}$$

where $\rho_j = \frac{u-u_j}{u_{j+d}-u_j}$.

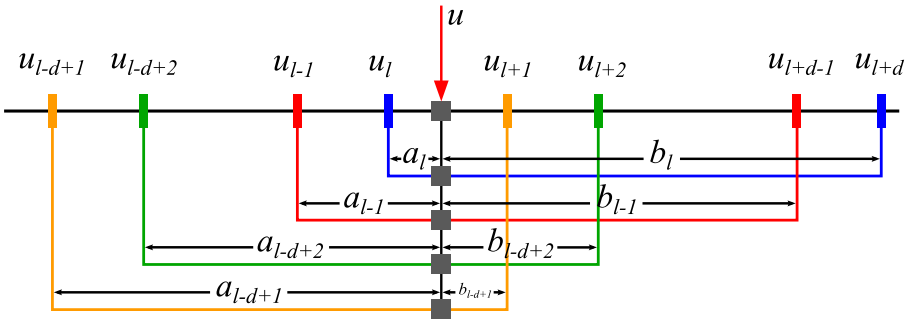


Fig. 163 General knot insertion scheme with the ρ_j computed as the ratio of a_j/b_j.

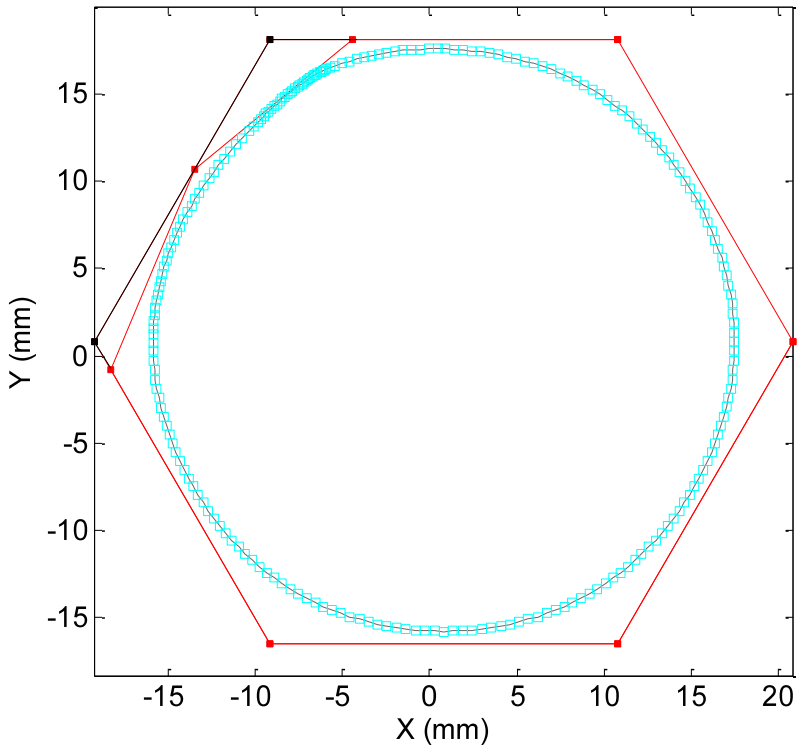


Fig. 164 Knot insertion in the case of a curve in 2D. Two current control points (black) are replaced by three control points (red).

Applications to simple curves

Our algorithm, shown in Fig. 165, is implemented in C++ and takes as input the degree of the curve d, the number of control points n_C, the level L of subdivision desired in order to approximate the curve by control points and the stop criterion ϵ. A cubic degree is chosen as a compromise between degrees of freedom and curve smoothness. A higher degree would make the curve too flexible resulting in undesired undulations of the curve and a lower degree would be too rigid to follow up with freeform shapes. The initial number of control points must be small so that the least flexibility is allocated. Too much flexibility at the start of the active contour deformation algorithm can degenerate into undulations and curve loops and self intersections. Finally the level of subdivision is a parameter that must be relatively high for better resolution, but it does not influence the system solution considerably. Only the residual errors remain large for small levels of subdivision below 4. This parameter has an incident on the complexity of the algorithm since it directly affects the size of matrix $\mathbf{M}^{(k)}$. Because there are repeatable matrix multiplications and matrix inversions at each iteration of the algorithm, the computational time and memory allocation are directly impacted by the

value of L. The compromise between complexity and accuracy is established and the numerous tests confirm that the values 4 and 5 are satisfactory.

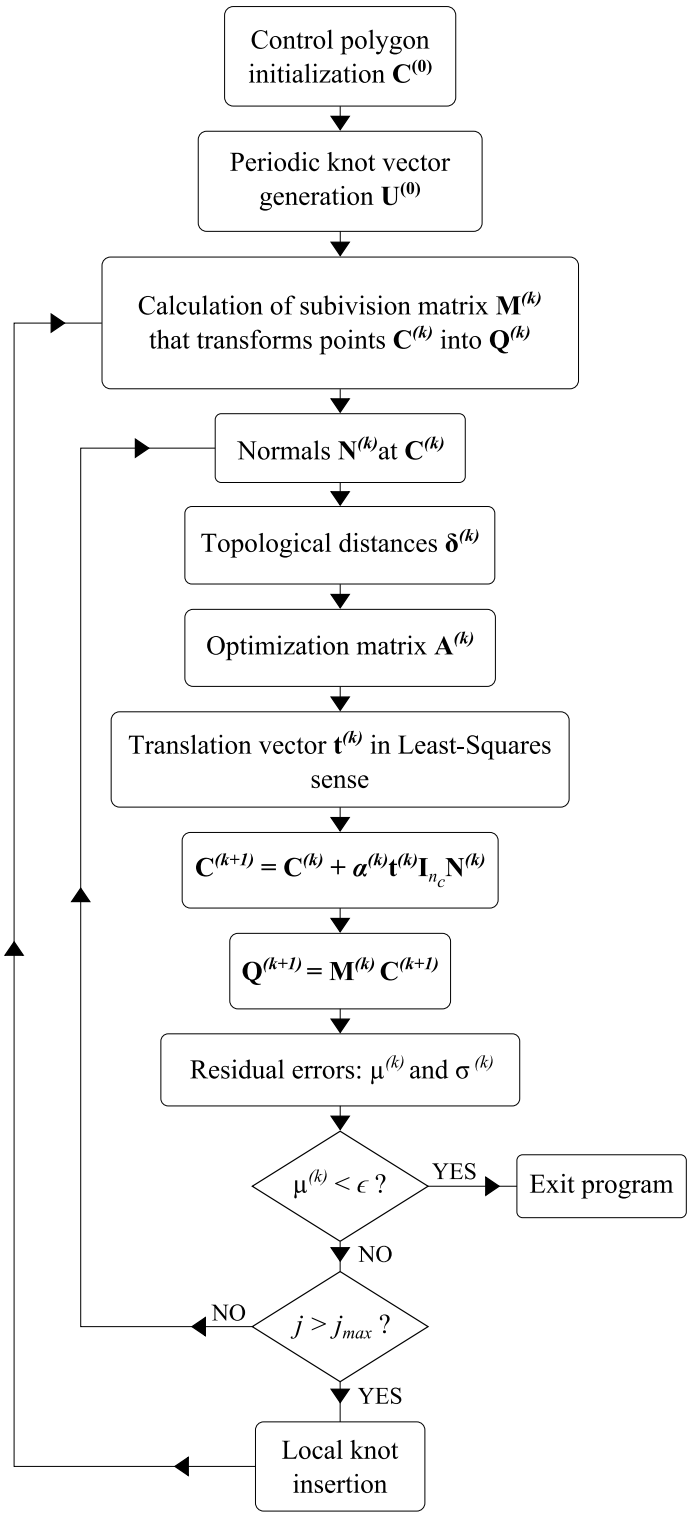
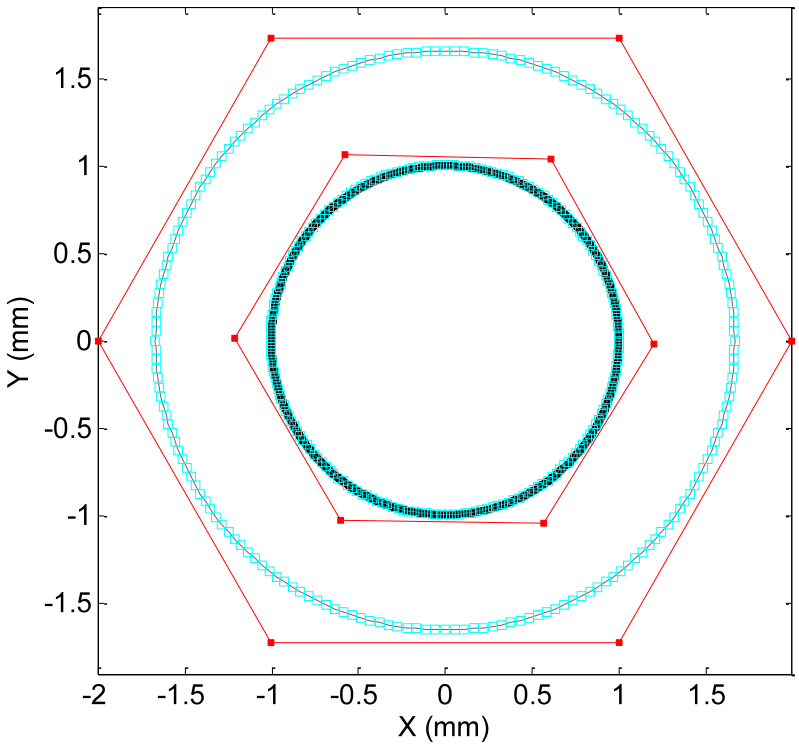
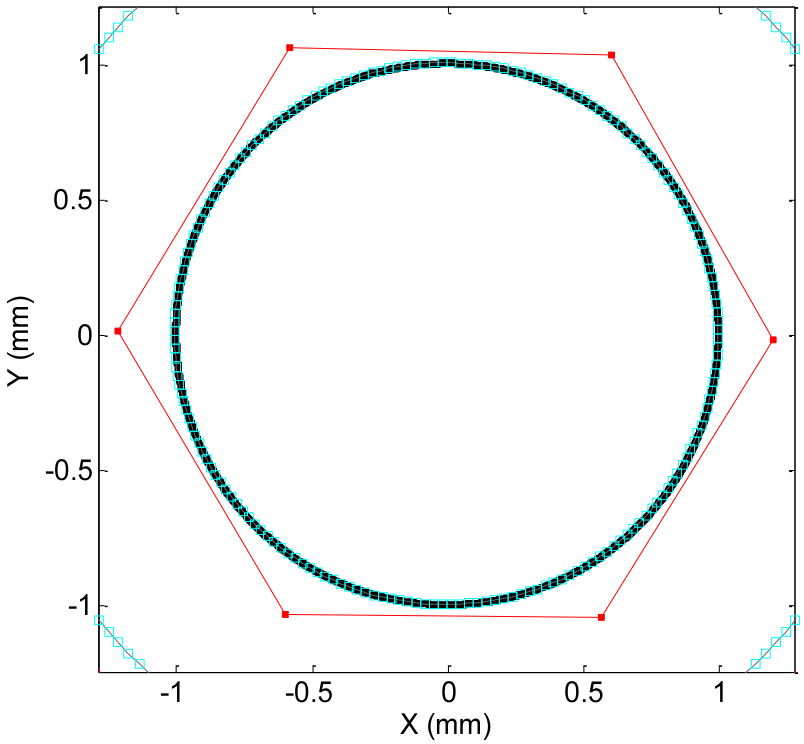


Fig. 165 DBACD algorithm ($\mu^{(k)}$: mean of residual distances and $\sigma^{(k)}$: standard deviation of residual distances).

In order to assess the effectiveness and correctness of our algorithm we firstly try it out on simple shapes with two different generic initializations that do not require knowledge about the underlying shape. These would be a generic initialization with control points on a circle around the data and an offset polygon of the bounding box of the data. For the following dataset, the points are measured on a cylinder at a given cross section with a tactile probe generating around 3600 points. These points are noisy and the measurement uncertainty is of some tens of nanometers. We therefore set the tolerance criterion of mean residual errors ϵ to be 10 nm and the results are illustrated in Fig. 166 and Fig. 168. The algorithm runs in 1.5 seconds and performs 10 iterations in the case where 6 control points are initialized to be on the circumference of a generic circle around the data (Fig. 166). In the other case, where the initialization is an offset of the bounding box and where 8 control points are considered, the algorithm runs in 2 seconds and performs 23 iterations (Fig. 168). The respective residual errors of each of the configurations are plotted in Fig. 167 and Fig. 169.



(a)



(b)

Fig. 166 DBACD on data representing a circular profile with generic initialization of 6 control points around the data. No knot insertions required. (a) Initial state and final state; (b) Zoom on the final state.

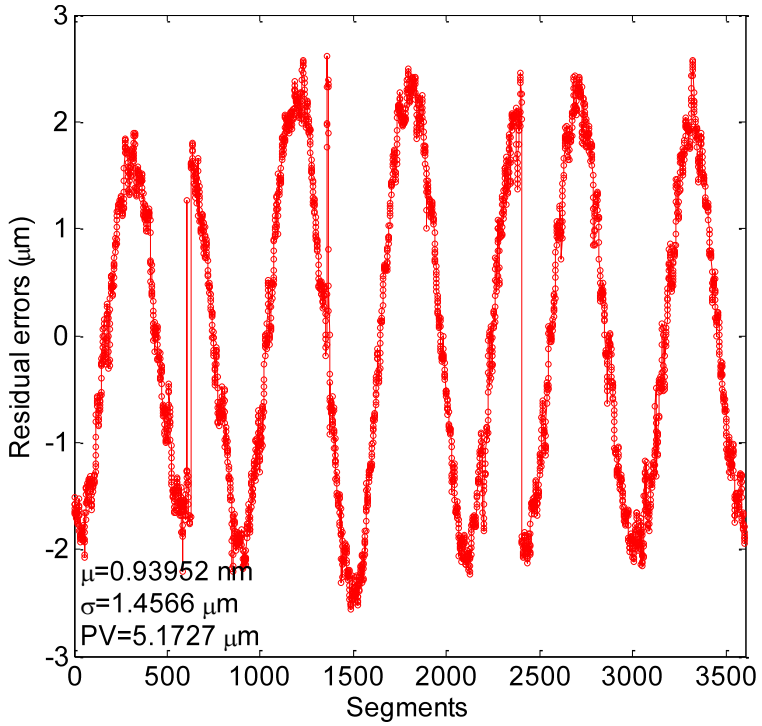
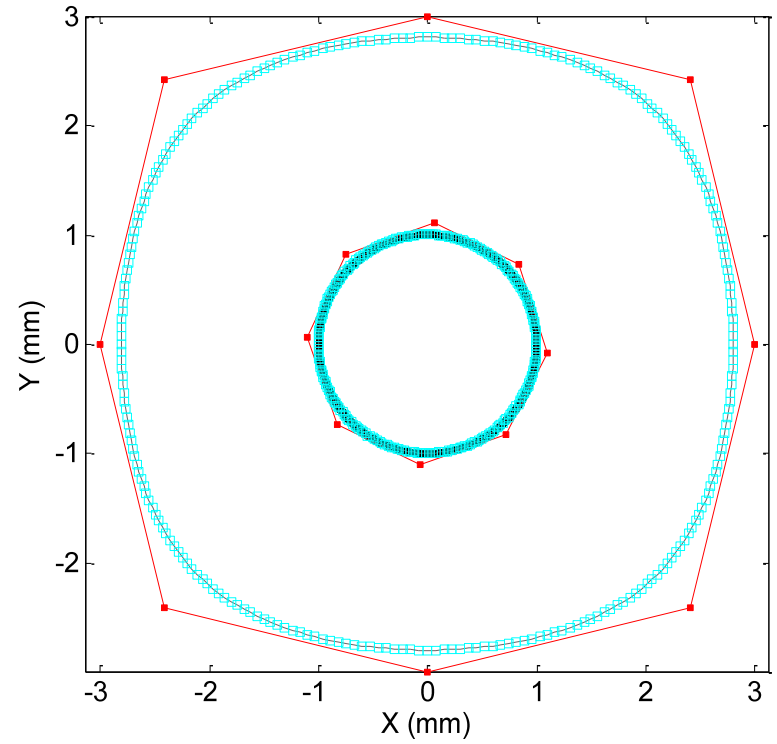
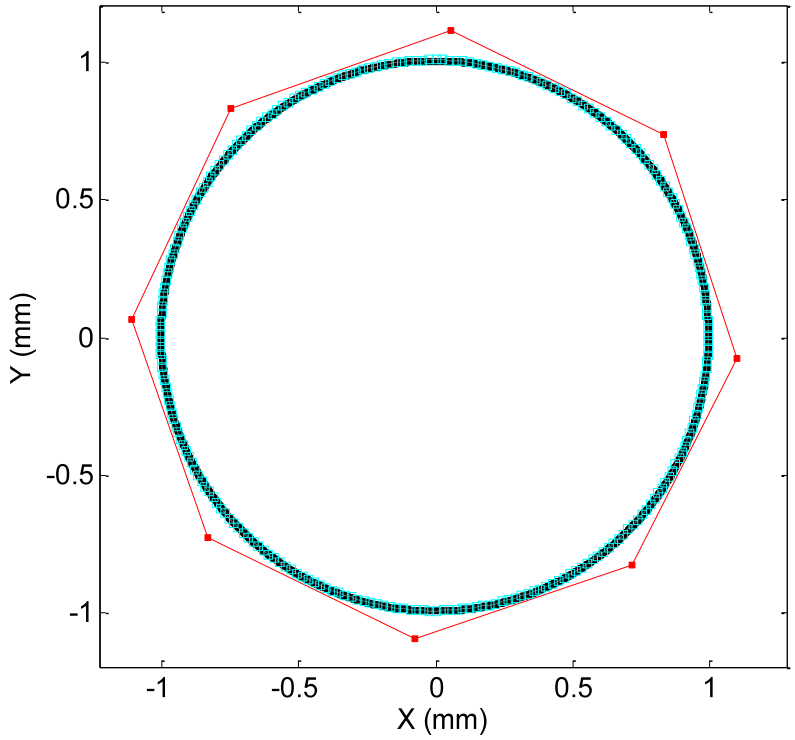


Fig. 167 Residual errors of the fitting based on a generic initialization for the circular profile dataset.



(a)



(b)

Fig. 168 DBACD on data representing a circular profile with an offset bounding box initialization of 8 control points around the data. No knot insertions required. (a) Initial state and final state; (b) Zoom on the final state.

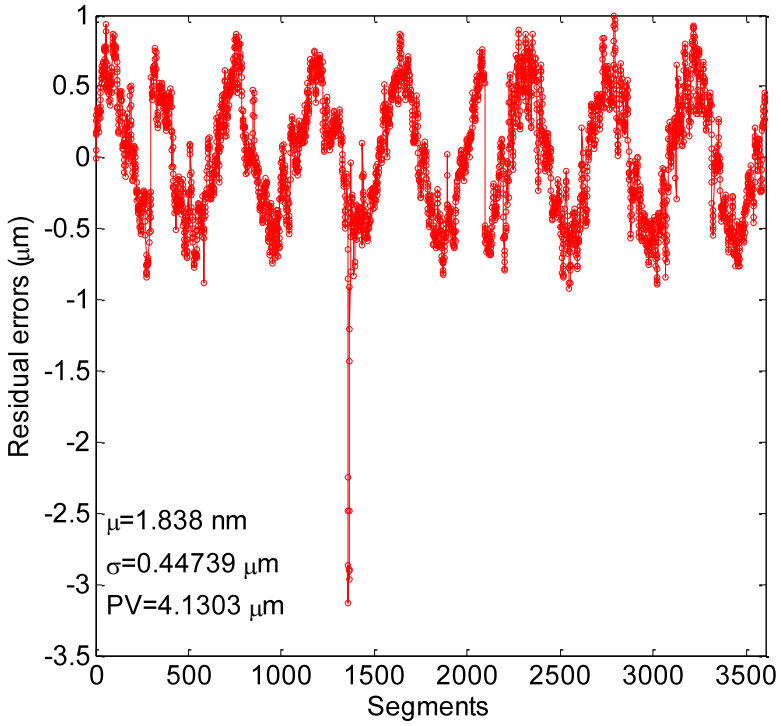


Fig. 169 Residual errors of the fitting based on an offset bounding box initialization for the circular profile dataset.

The graphs showing the residual errors with respect to the B-Spline subdivided segments reveal that the algorithm converges in the case of the circle dataset for both initializations (generic and offset of bounding box) and the mean of the residual errors is smaller than ϵ. It is to note, however, that the resulting B-Spline curve is not exactly the same for each of the cases of initialization.

We consider another typical dataset with some 400 points which is slightly more complex in shape because of concave features. Applying the DBACD algorithm on this dataset replicated from a dataset found in [159], we get to observe that the algorithm operates properly (Fig. 170). Nonetheless, it does not converge if we inhibit knot insertion, residual errors remain large as the mean of residual errors is about -33 μm and the PV is about 5 mm (Fig. 171).

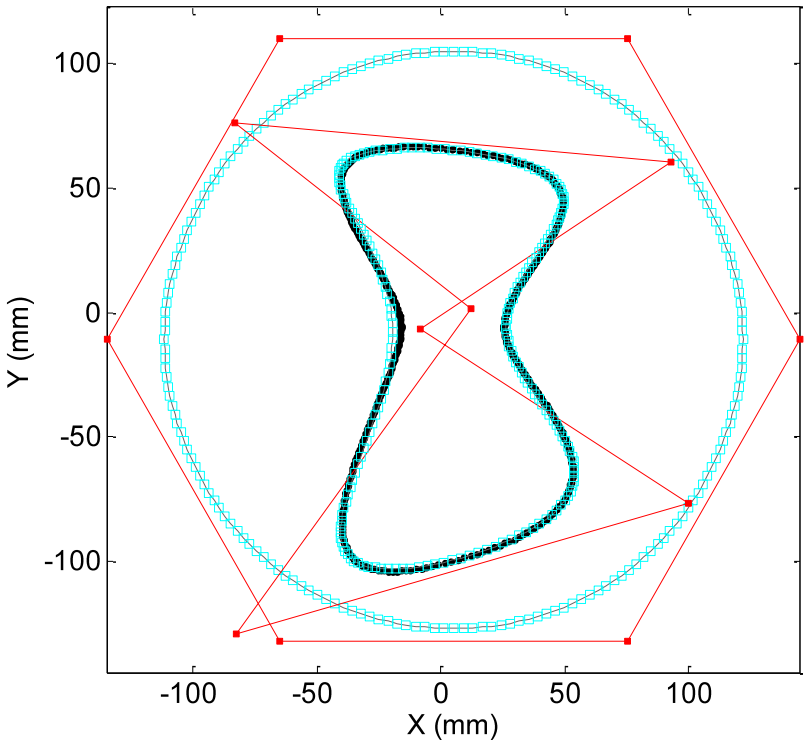


Fig. 170 DBACD on data representing a profile with concavities starting from generic initialization of 6 control points around the data. Knot insertions are inhibited.

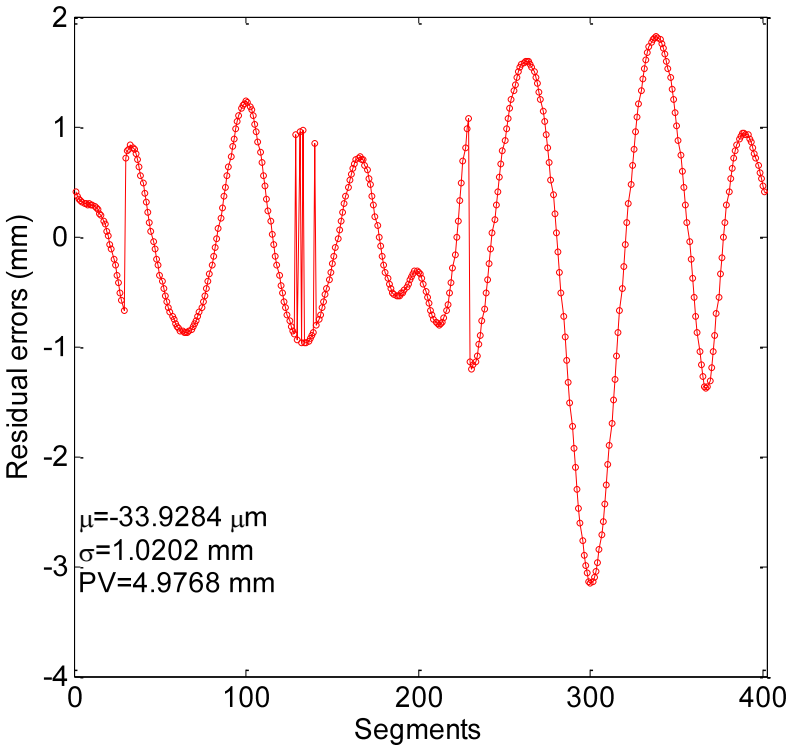


Fig. 171 Residual errors of the fitting based on a generic initialization for the concave profile dataset.

Now we let local knot insertions where the local error is still larger than the threshold $\epsilon = 10$ nm. Runtime is about 7 seconds for 40 inner iterations and 3 outer iterations (Fig. 172), giving a total of 10 final control points.

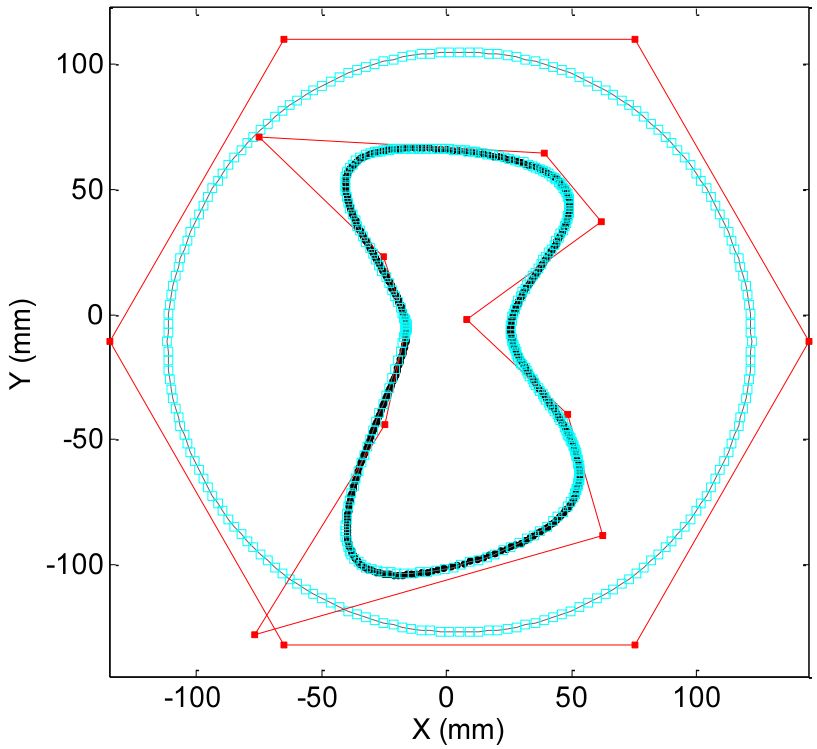


Fig. 172 DBACD on data representing a profile with concavities starting from generic initialization of 6 control points around the data with local knot insertion.

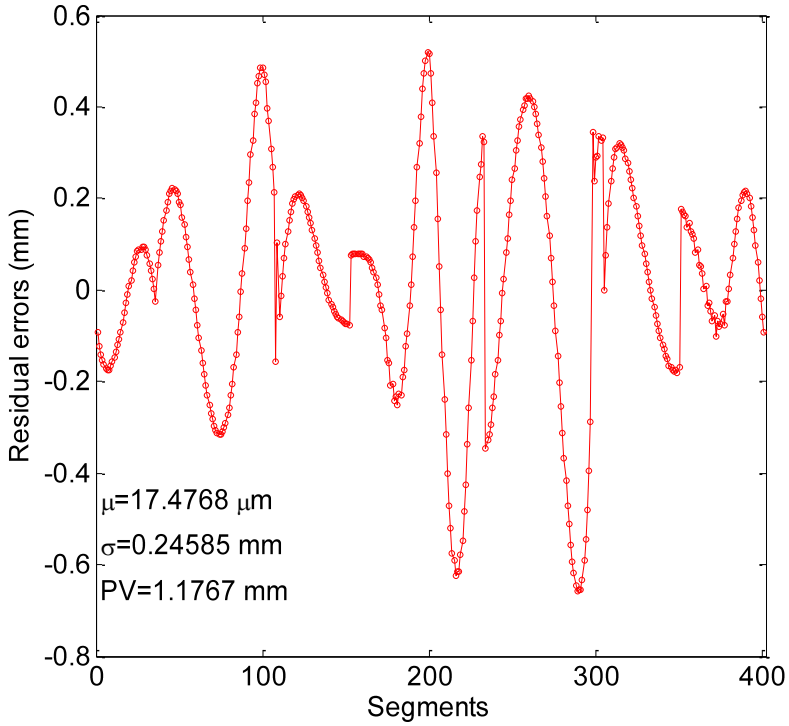


Fig. 173 Residual errors of the fitting based on a generic initialization for the concave profile dataset.

The graph of Fig. 173 shows that the residual errors improve with local knot insertion. Despite the fact that the mean of errors has reduced to 17 μm and the PV to 1 mm, the mean of residual errors is still far from the specified threshold ϵ. For this specific shape, the algorithm does not converge. If we allow more knot insertions, the active B-Spline starts having self-intersections, a problem that will be addressed later (Fig. 174).

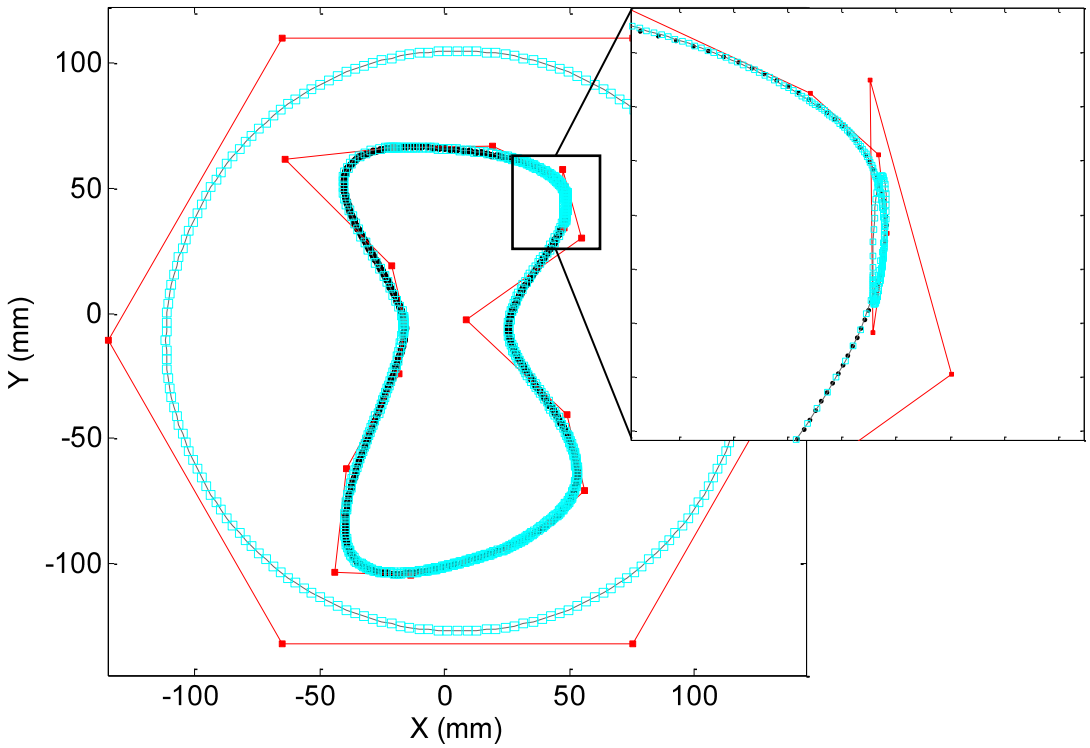


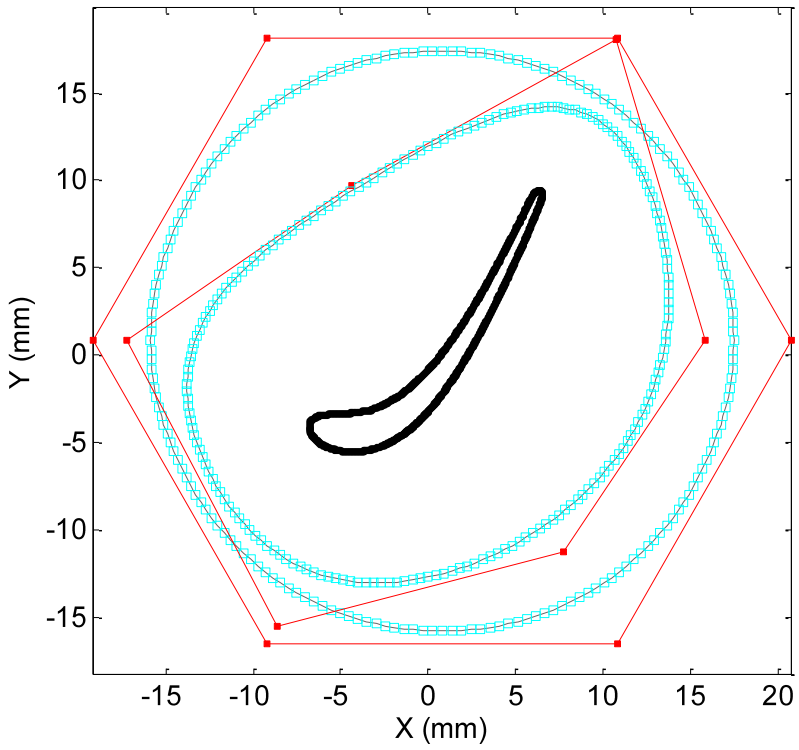
Fig. 174 DBACD on data representing a profile with concavities starting from a generic initialization of 6 control points around the data with local knot insertion.

Local knot insertion improves the outcome, as seen for the example of the shape representing some concavities, but does not necessarily make the algorithm converge. A lot of knot insertions might cause self-intersections in the B-Spline curve.

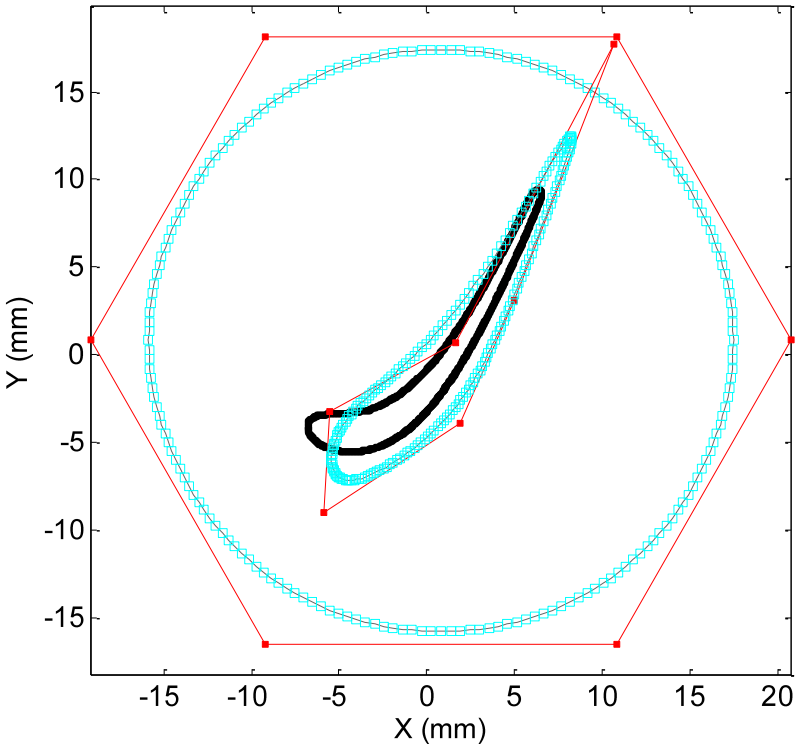
Application to the turbine blade profile

For the next example, we take a more complex shape which consists of a region of high curvature making a suitable fitting near that region difficult. The dataset contains about 1,200 points and we start with 6 control points uniformly distributed on a circle around the dataset (generic initialization) with a randomly chosen circle radius and center. The radius must however be such that all the control points are on the outside of the dataset without crossings. The objective tolerance ϵ is set to 0.5 µm, a value that corresponds to turbine blades profile form tolerance (Chapter 1 - II.).

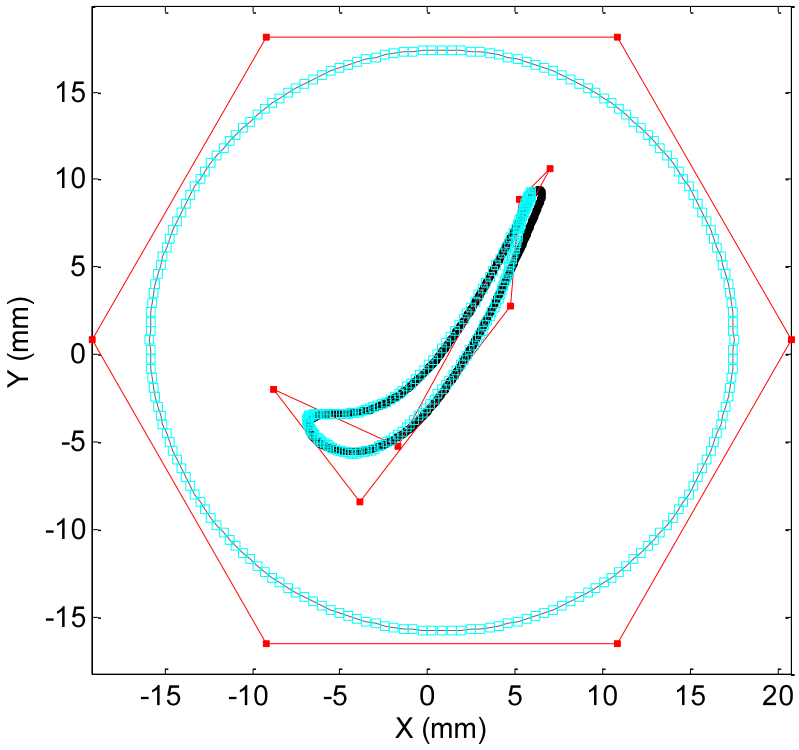
After fitting, the residual errors have a mean of -29.3 µm before local knot insertions (Fig. 175d), and with 2 additional knots, the mean of errors drops down to 0.79 µm (Fig. 175e). The process requires a total of 100 inner iterations and runs in 7 s.


15
10
5
0
-5
-10
-15
Y (mm)
-15
-10
-5
0
5
10
15
20
X (mm)

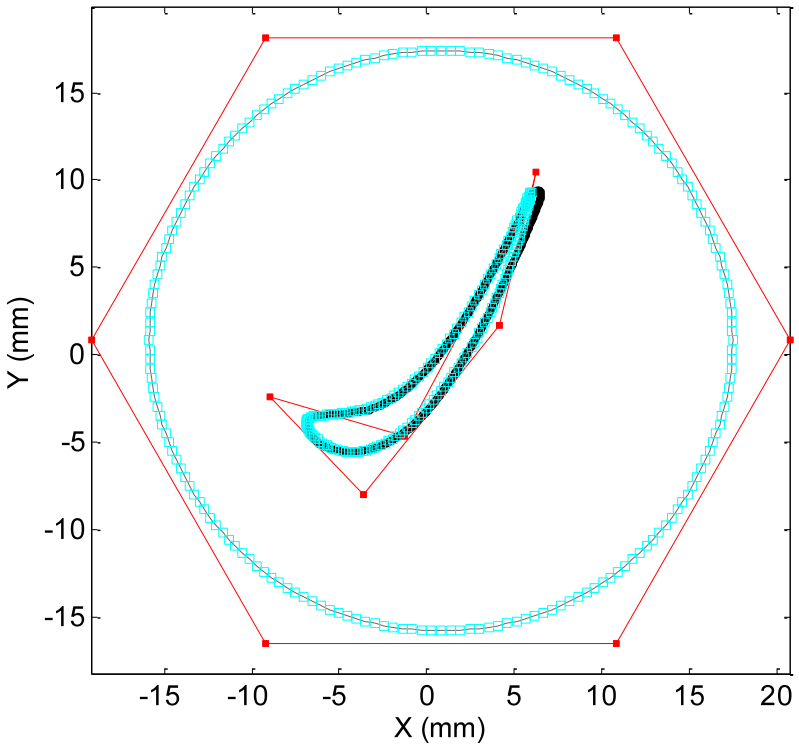
(a)


15
10
5
0
-5
-10
-15
Y (mm)
-15
-10
-5
0
5
10
15
20
X (mm)

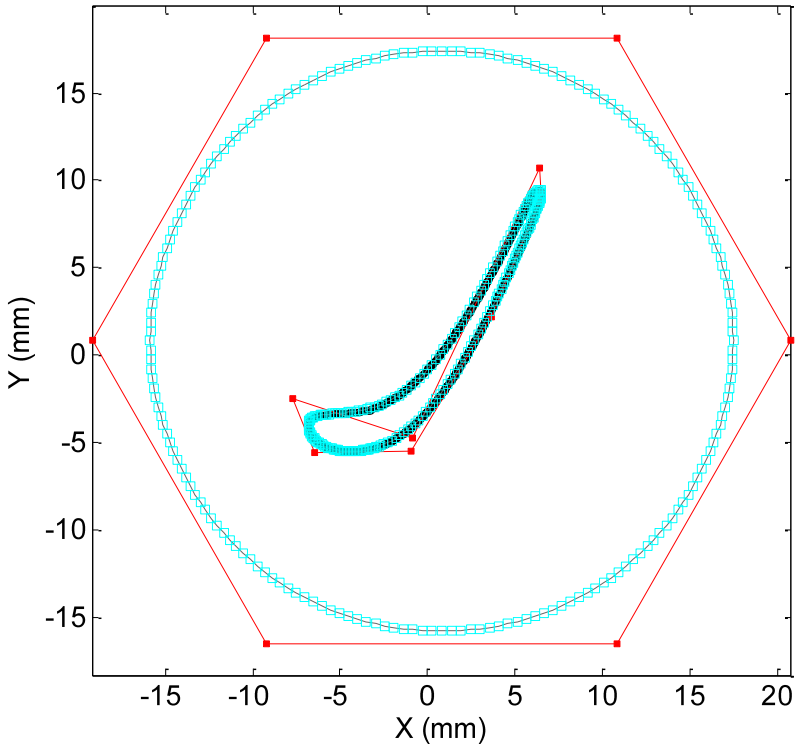
(b)


15
10
5
0
-5
-10
-15
Y (mm)
-15
-10
-5
0
5
10
15
20
X (mm)

(c)


15
10
5
0
-5
-10
-15
Y (mm)
-15
-10
-5
0
5
10
15
20
X (mm)

(d)



(e)

Fig. 175 DBACD algorithm for the turbine blade profile containing 1200 points. (a) step 1; (b) step 2; (c) step 20; (d) step 50 all without knot insertions. (e) one additional outer iteration of knot insertions giving 8 final control points.

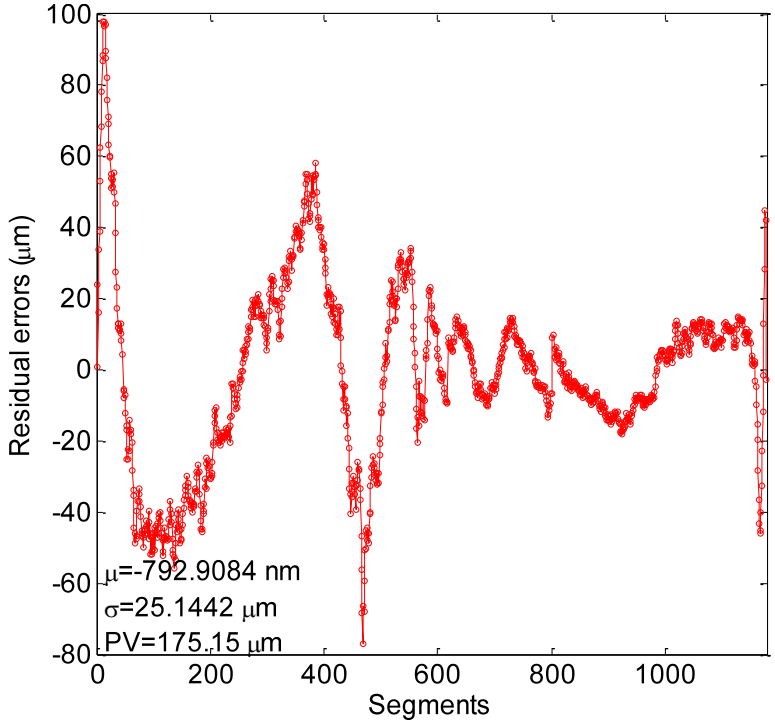
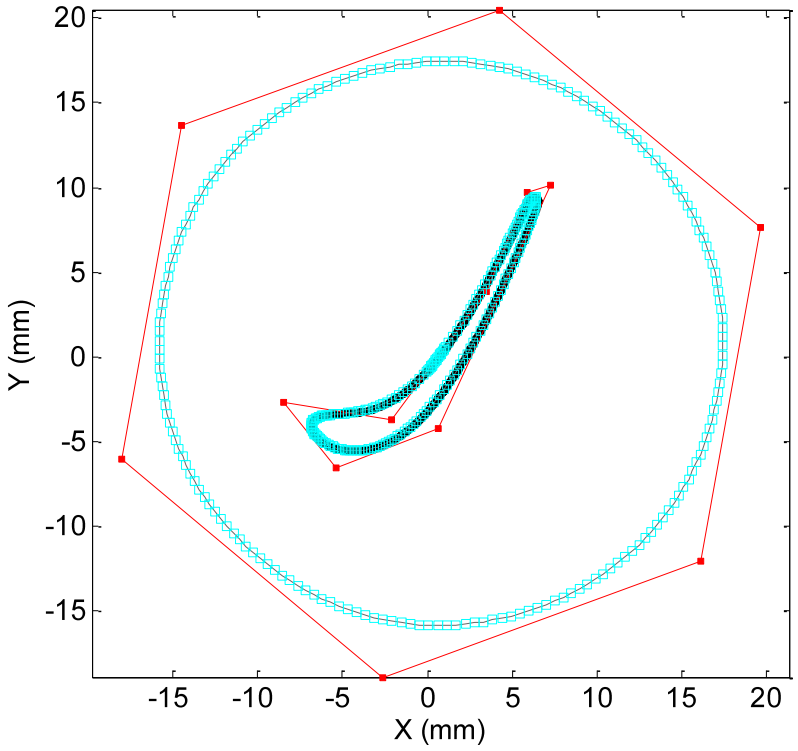


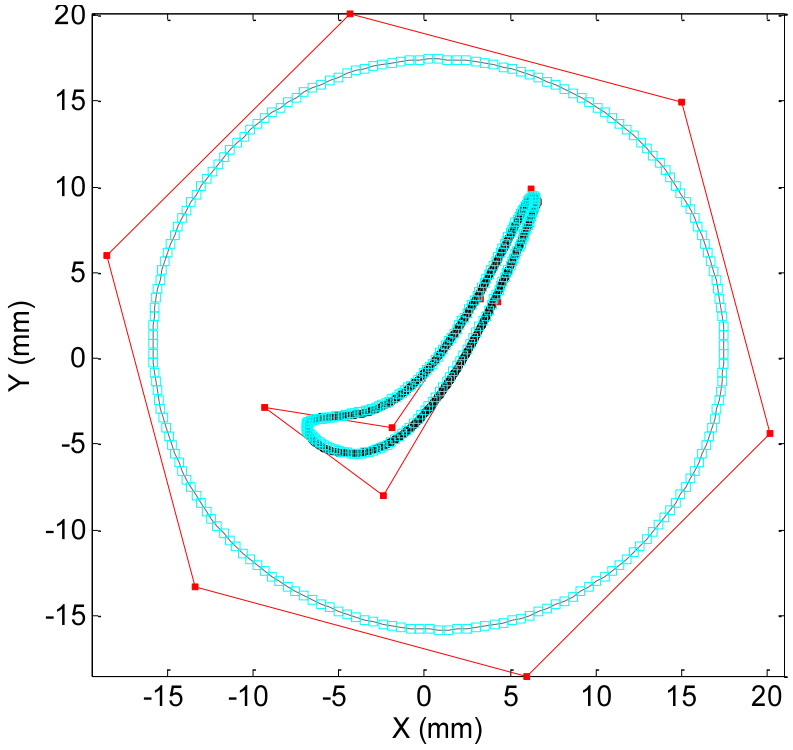
Fig. 176 Residual errors of the fitting based on a generic initialization for the blade profile dataset.

The end-result approaches the underlying shape of the data points but does not converge as the mean of residual errors is larger than the threshold $\epsilon = 0.5$ µm (Fig. 176).

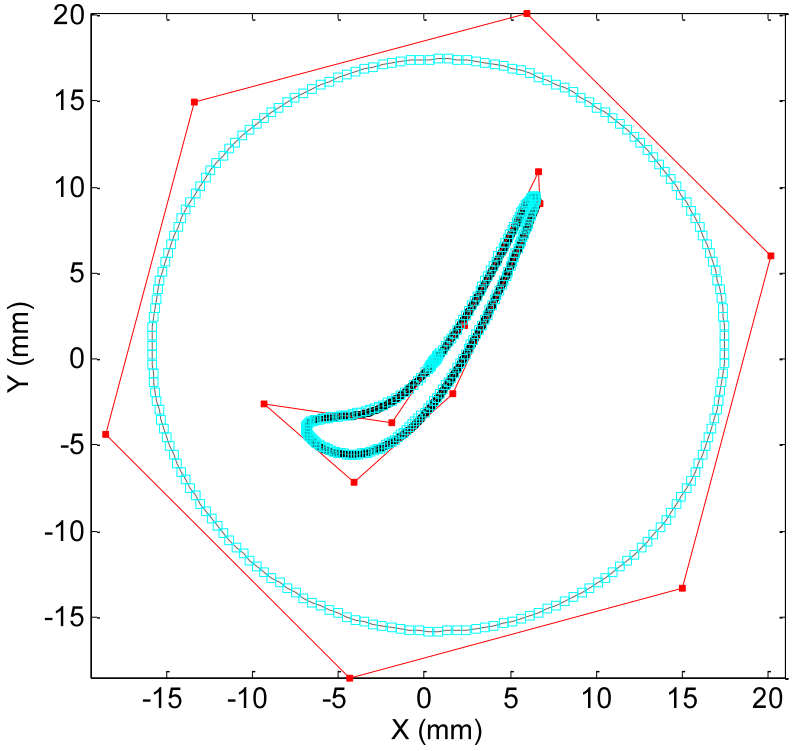
As a matter of robustness to initialization, we test several initial orientations of the initial control polygon. We take the same polygon rotated by 20 °, 45 ° and 70 ° (Fig. 177). The residual errors are reported in .



(a)



(b)



(c)

Fig. 177 DBACD algorithm for the blade profile applied for several initial control polygons rotations with respect to the dataset: (a) 20 °; (b) 45 °; (c) 70 °.

Angle	0 °	20 °	45 °	70 °
μ (μm)	-0.793	0.121	-0.739	0.164
σ (μm)	25.144	12.655	19.416	11.424
PV (μm)	175.15	78.01	118.54	72.07

Table 28 Residual errors of the different relative initial rotations of the control polygon for the blade's profile.

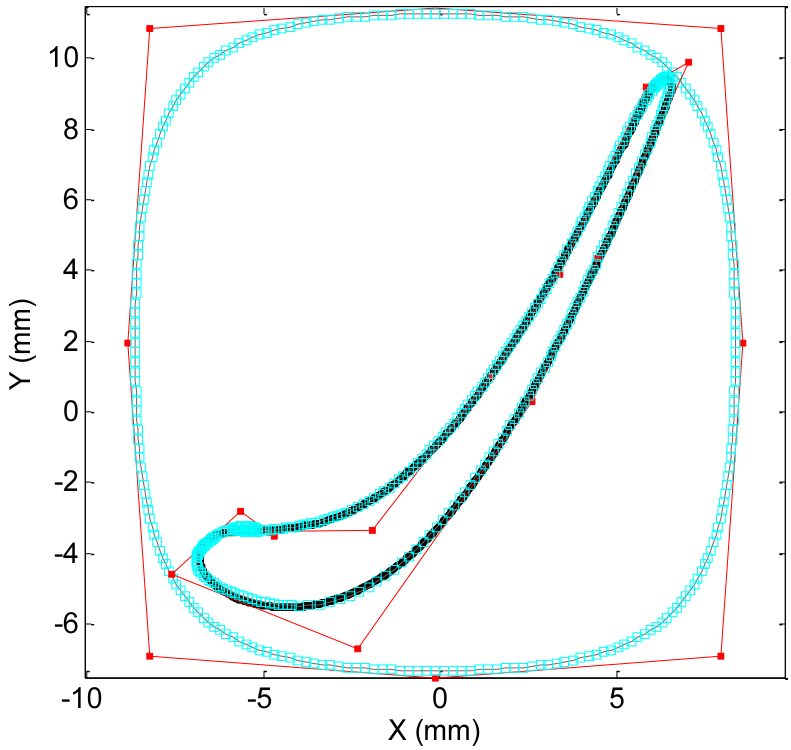


Fig. 178 DBACD algorithm for the blade profile applied for the bounding box offset initialization.

The results of Fig. 178 show once again that our algorithm is robust to control polygon initialization. For all the above tests, DBACD was applied by alternating between optimization directions. Optimization was once in both x and y directions and once in the normal direction to each control point. The independent x and y optimization directions, instead of the single normal direction, offer an additional degree of freedom for control points to move in the plane. Optimizing along the normal direction restrains the control points to move along one axis and by that require the need to insert more knots. The drawback of the algorithm up to this point is that having too many control points (excessive knot insertions) leads to self-intersections within the B-Spline curve and/or the control polygon (Fig. 179).

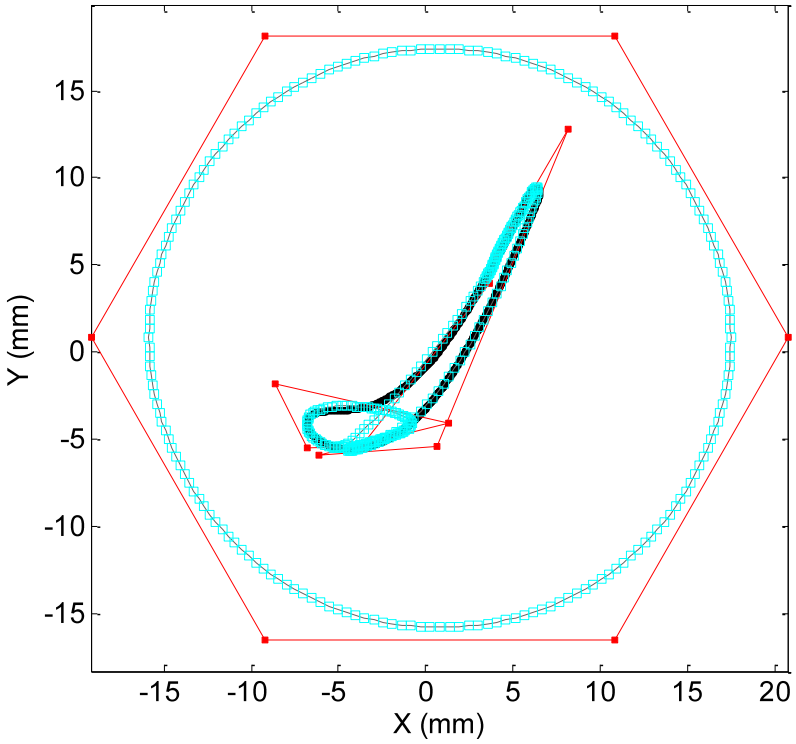


Fig. 179 Excessive knot insertions create buckles in the resulting B-Spline curve.

Although our algorithm is robust to initialization, robustness is still questionable in regard to convergence as the DBACD algorithm does not deterministically achieve a mean of residual errors that is below the pre-requisite tolerance ϵ, even with the knot insertion pattern. In the aim of solving this problem, we now propose to investigate about adding a fairing term to the objective function which would allow to simulate surface smoothness and minimize curve self-intersections.

2.f Fairing

Principle and implementation

Fairing has been used in many parametric fitting approaches [159] and even implicit methods [175]. We will introduce this principle here and describe how we append it to our minimization functional and implement it. Normally, the fairing term transcribing curve smoothness is expressed as the integral of the L_2 norm of curvature of a curve $\mathcal{C}$ (63). For parametric B-Spline curves, fairing can be applied onto the control polygon and the expression can be written in terms of the control points. In order that the newly computed

polygon takes into account notions of smoothness, fairing must be applied directly to the points $\mathbf{C}^{(k+1)}$ at iteration $k+1$:

$$\int_0^1 \|\mathcal{C}''(u)\|^2 du = \int_0^1 \left\| \sum_{j=1}^{n_C} N_{j,\tau^{(k)}}^{d\,''}(u)\ c_j^{(k+1)} \right\|^2 du\,, \tag{63}$$

where $\tau^{(k)}$ is the knot sequence at iteration k.

Since we work with a discretization of the B-Spline control polygon, we write the above expression in discrete form:

$$\sum_{s=0}^{\mathrm{S}-1} h \times \left\| \sum_{j=1}^{n_C} N_{j,\tau^{(k)}}^{d\,''}(sh)\ c_j^{(k+1)} \right\|^2 \tag{64}$$

where S is the number of sampled points, $h = 1/\mathrm{S}$ is the sampling step and $c_j^{(k+1)}$ are the control points at iteration $k+1$.

The second derivative B-Spline basis functions $N_{j,\tau^{(k)}}^{d\,''}(u)$ are computed using a simple algorithm found in [167]. Adding the fairing term in (64) to the original objective function we get that the new functional to minimize in the normal direction of control points is the following:

$$\min_{\mathbf{t}^{(k)}} \left(1-\lambda^{(k)}\right)\left\|\mathbf{A}^{(k)}\mathbf{t}^{(k)} - \boldsymbol{\delta}^{(k)}\right\|_2^2 + \lambda^{(k)} h \sum_{s=0}^{\mathrm{S}-1} \left\|\boldsymbol{\mathcal{N}}_{\tau^{(k)}}(sh)\left(\mathbf{C}^{(k)} + \mathbf{t}^{(k)}\mathbf{I}_{n_c}\mathbf{N}^{(k)}\right)\right\|_2^2 \tag{65}$$

where $\boldsymbol{\mathcal{N}}_{\tau^{(k)}}(sh)$ is the diagonal matrix consisting of $N_{j,\tau^{(k)}}^{d\,''}(sh)$ terms such that:

$$\boldsymbol{\mathcal{N}}_{\tau^{(k)}}(sh) = \mathrm{diag}\left(N_{1,\tau^{(k)}}^{d\,''}(sh), \cdots, N_{n_c,\tau^{(k)}}^{d\,''}(sh)\right).$$

The parameter $\lambda \in [0\ 1]$ is called the fairing parameter and gives a certain weight to each of the terms of the objective function. The larger λ is chosen to be, the more the influence of curve smoothness is and the less the influence of distances becomes. In our implementation, λ is set to vary according to the average distance that separates the B-Spline model from the

data points at each iteration. If the average distance is small, λ takes a small value so that the B-Spline curve which is now close to the data progresses mainly under the influence of distances.

The expression in (65) can be reduced to:

$$\min_{\mathbf{t}^{(k)}} \left\| \mathbf{B}^{(k)}\mathbf{t}^{(k)} - \boldsymbol{\zeta}^{(k)} \right\|_2^2 \tag{66}$$

where, $\mathbf{B}^{(k)} = \begin{bmatrix} \left(1-\lambda^{(k)}\right)\mathbf{A}^{(k)} \\ \lambda^{(k)} h\, \mathfrak{N}_{\tau^{(k)}}(0) \\ \lambda^{(k)} h\, \mathfrak{N}_{\tau^{(k)}}(h) \\ \vdots \\ \lambda^{(k)} h\, \mathfrak{N}_{\tau^{(k)}}\left((\mathrm{S}-1)h\right) \end{bmatrix}$, $\boldsymbol{\zeta}^{(k)} = \begin{bmatrix} \left(1-\lambda^{(k)}\right)\boldsymbol{\delta}^{(k)} \\ -\lambda^{(k)} h\, \mathcal{N}_{\tau^{(k)}}(0)\mathbf{C}^{(k)} \\ -\lambda^{(k)} h\, \mathcal{N}_{\tau^{(k)}}(h)\mathbf{C}^{(k)} \\ \vdots \\ -\lambda^{(k)} h\, \mathcal{N}_{\tau^{(k)}}\left((\mathrm{S}-1)h\right)\mathbf{C}^{(k)} \end{bmatrix}$

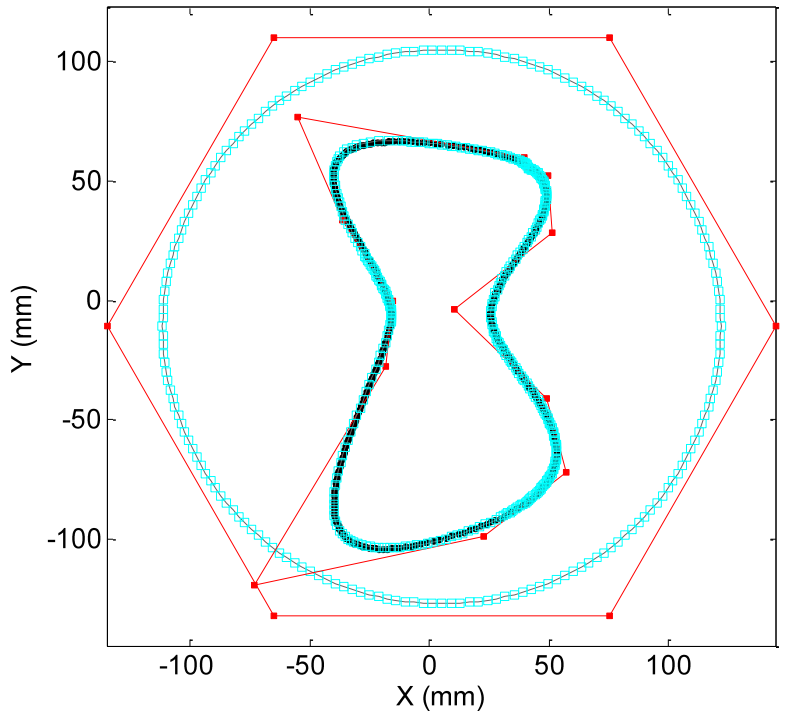
and $\mathfrak{N}_{\tau^{(k)}}(sh)$ is the matrix such that: $\mathcal{N}_{\tau^{(k)}}(sh)\mathbf{t}^{(k)}\mathbf{I}_{n_c}\mathbf{N}^{(k)} = \mathfrak{N}_{\tau^{(k)}}(sh)\mathbf{t}^{(k)}$.

The linear system to solve at iteration k, written in normal form, is then:

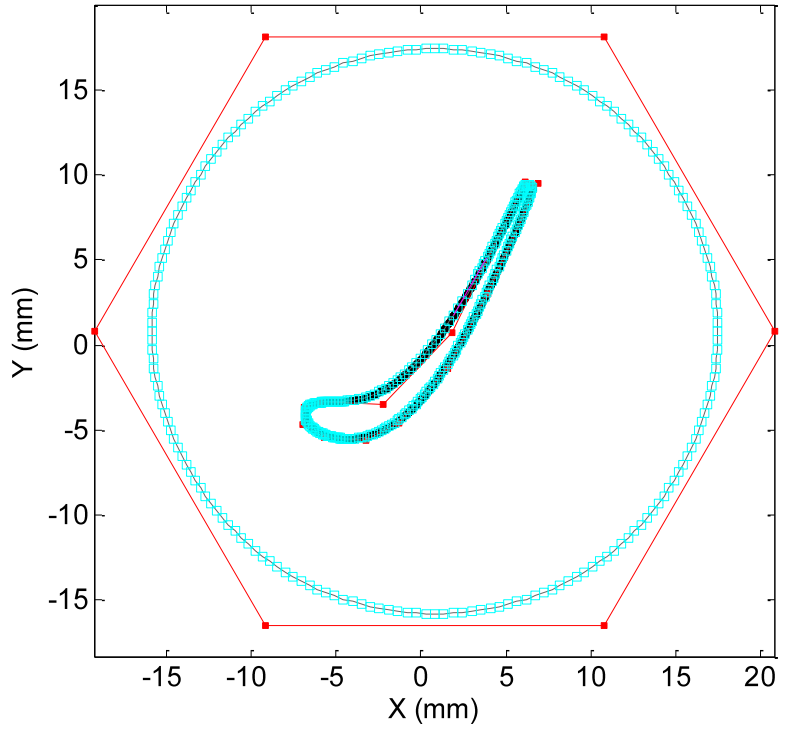
$$\left(\mathbf{B}^{(k)}\right)^{\mathrm{T}}\mathbf{B}^{(k)}\mathbf{t}^{(k)} = \left(\mathbf{B}^{(k)}\right)^{\mathrm{T}}\boldsymbol{\zeta}^{(k)}. \tag{67}$$

Application

We now present a result on the examples of the concave shape and the turbine blade profile with the addition of the fairing term. We show that with fairing, curve self-intersections tend to disappear and convergence is improved. Fig. 180 shows the B-Spline curve fit to the data with a fairing parameter starting value of 0.8. The value starts to decrease when the average distance is below 10 µm until it reaches, in this example, a value of 0.2.

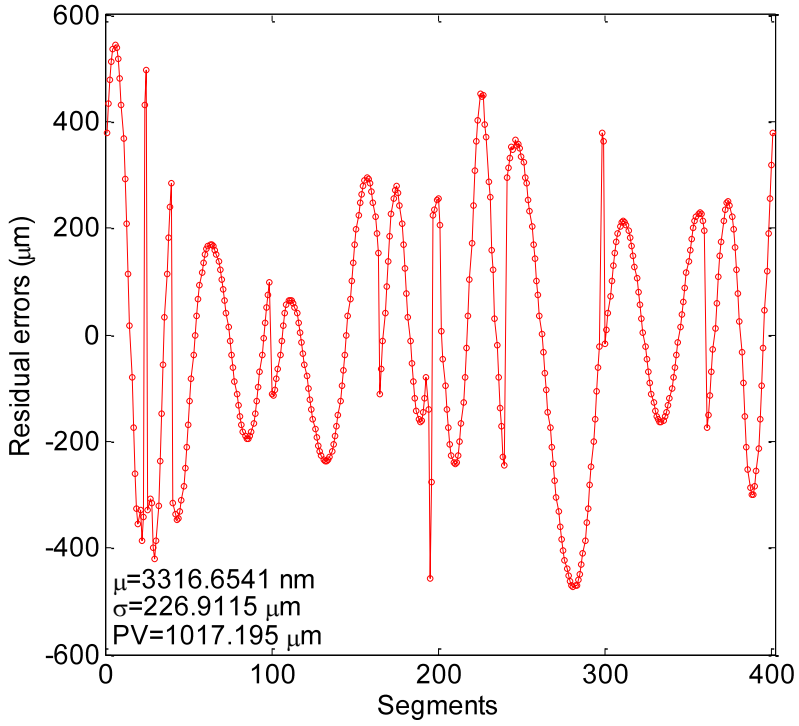


(a)

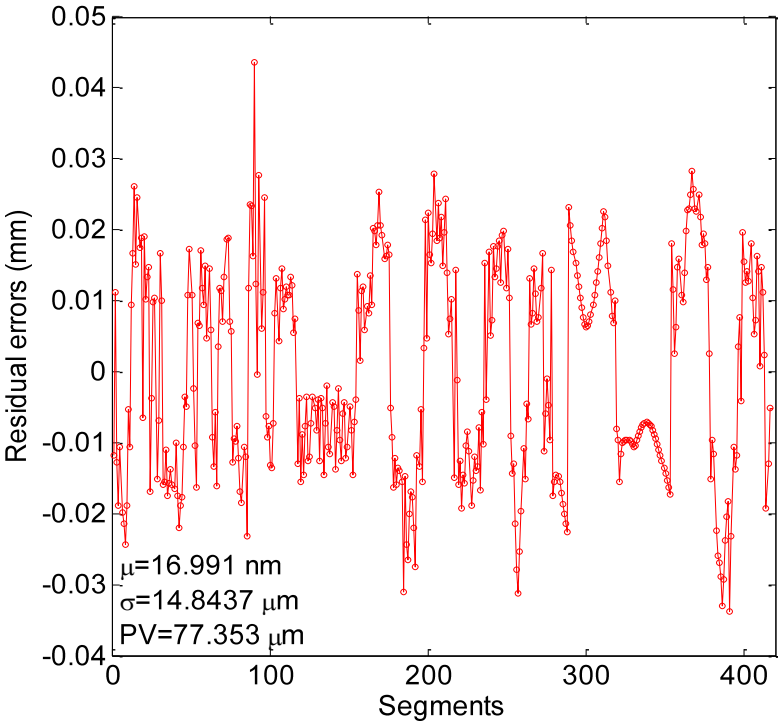


(b)

Fig. 180 DBACD algorithm with fairing: $\lambda = 0.8$ and varies automatically according to the average distance: (a) 13 final control points for the concave shape; (b) 17 final control points for the blade profile.



(a)



(b)

Fig. 181 Residual errors of the fitting with fairing: (a) of the shape with concavities; (b) of the turbine blade profile.

We can see from Fig. 181 that the algorithm has an improved convergence with fairing and gives far better residual errors than without it with a mean $\mu = 3.3$ μm for the shape with

concavities and $\mu = 0.017$ µm without having any self-intersections in the resulting B-Spline subdivision curve. For the turbine blade profile, the mean of residual errors is below the threshold. Fig. 182 shows that the subdivided curve is actually very close to the smooth B-Spline curve. Adding the fairing term to the minimization increases computational time by around 10 to 15 %.

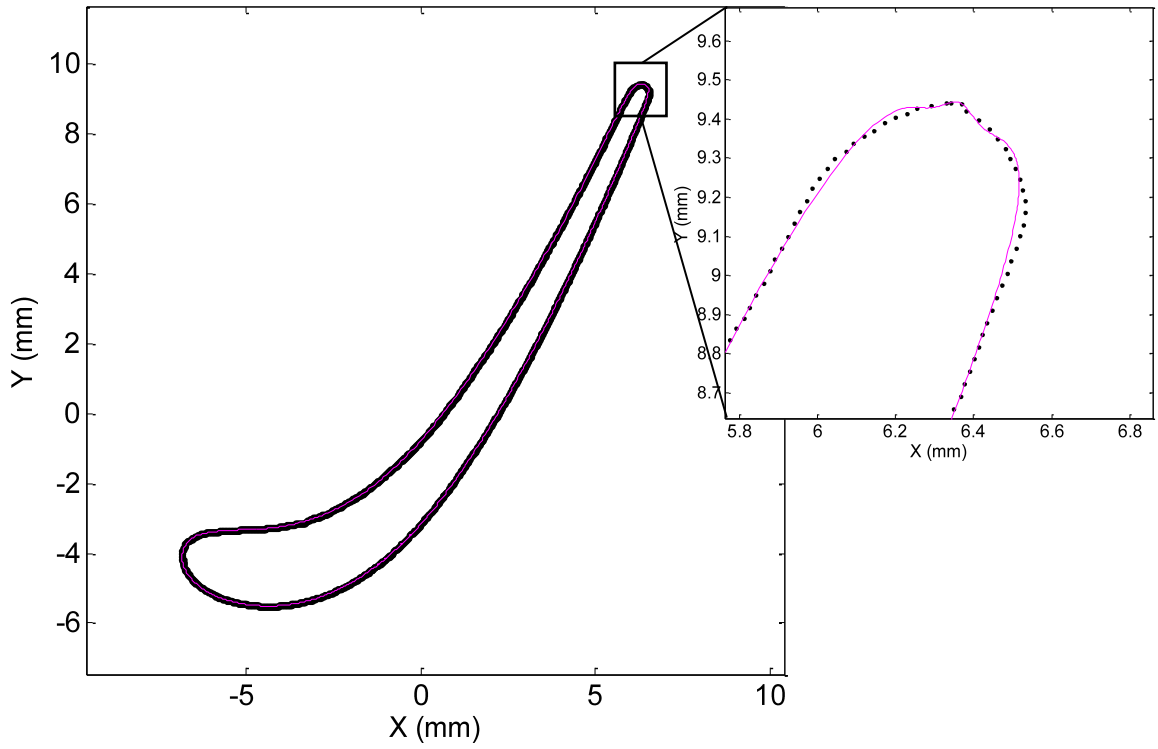


Fig. 182 The smooth B-Spline curve resulting from the fitting with our algorithm for the turbine blade profile.

3. Perspectives for the extension to surfaces

The DBACD algorithm that we developed for curves can be extended to surfaces but the work is still underway. The approach that we particularly proposed to address reconstruction problems related to turbine blades is somehow a first attempt to turbine blade surface reconstruction. The surface problem can be seen in two perspectives. Either a surface reconstruction that is based on lofting methods, i.e. the junction of the section curves in order to form a smooth lofted surface [242], or a surface reconstruction approach that resembles the proposed curve approach but in three dimensions.

The extension of the DBACD algorithm to surfaces following the approach proposed for curves is clear since the passage from B-Spline curves to surfaces is well-defined and

straightforward such as given in (6). For parametric surfaces, bidirectional parameters are needed and we will refer to these directions by u and v (5). The steps of the DBACD algorithm for surfaces are similar to the procedure of the algorithm for curves so it will be presented following the same sequence for the example of the turbine blade. The dataset contains about $90{,}000$ points.

Initialization

The three elements that defined a B-Spline curve also define a B-Spline surface. The degree of the unidirectional curves, the knot vector and the control polygon must be specified for each direction u and v. Cubic B-Spline surfaces are enough to represent complex shapes, therefore we choose a degree $d = 3$ for both u and v directions.

Since we are dealing with a special example of surfaces, the turbine blade, we assume that the unidirectional curves are closed in the u- direction and that the unidirectional curves in the v-direction are open. The set of control polygons in both directions will be referred to as the control net. The control net is such that the control points are periodic in u-direction and non-periodic in v-direction. The initial knot vectors are deduced from the degree and the control points in each direction in the same way as in (51). For the periodic knots, $2 \times d$ additional knots are appended to the *base* knot vector in such a way to reproduce periodicity (Fig. 155). For the non-periodic knots, the end-knots are repeated with a multiplicity equal to d so that the v-direction curves coincide with the u-direction curves at their end points.

Subdivision points

By applying a global knot insertion in u-direction first, exactly as indicated in the previous section (Oslo algorithm), we can deduce the subdivision in this direction from the relation given by equation (52). Then, taking the subdivided points in the v-direction and preserving the order in u, we apply a subdivision matrix $\mathbf{M_v}$ in the v-direction across all u corresponding values. The relationship between control points and subdivided points sums up to (68):

$$\mathbf{Q}^{(k)} = \mathbf{M_v}^{(k)}\mathbf{M_u}^{(k)}\mathbf{C}^{(k)} = \mathbf{M}^{(k)}\mathbf{C}^{(k)} \tag{68}$$

where, $\mathbf{C}^{(k)} = \begin{pmatrix} c_{11} \\ c_{21} \\ c_{31} \\ \vdots \\ c_{n_v 1} \\ c_{12} \\ c_{22} \\ c_{32} \\ \vdots \\ c_{n_v 2} \\ \vdots \\ c_{n_v n_u} \end{pmatrix}$, $\mathbf{Q}^{(k)} = \begin{pmatrix} q_{11} \\ q_{21} \\ q_{31} \\ \vdots \\ q_{n_{s\_v} 1} \\ q_{12} \\ q_{22} \\ q_{32} \\ \vdots \\ q_{n_{s\_v} 2} \\ \vdots \\ q_{n_{s\_v} n_{s\_u}} \end{pmatrix}$ (69)

and $\mathbf{M_u}$ and $\mathbf{M_v}$ are block matrices.

Having this in hand, we can proceed just like in the curves problem, except that the matrices are much larger. The result of subdivision is a coherent subdivided control net as shown in Fig. 183. For curves in the plane, it was easy to compute the distance from a data point to subdivided segments formed by each subdivided point and the next one. In 3D, the segments are no longer valid geometrical elements and must be replaced with triangular facets.

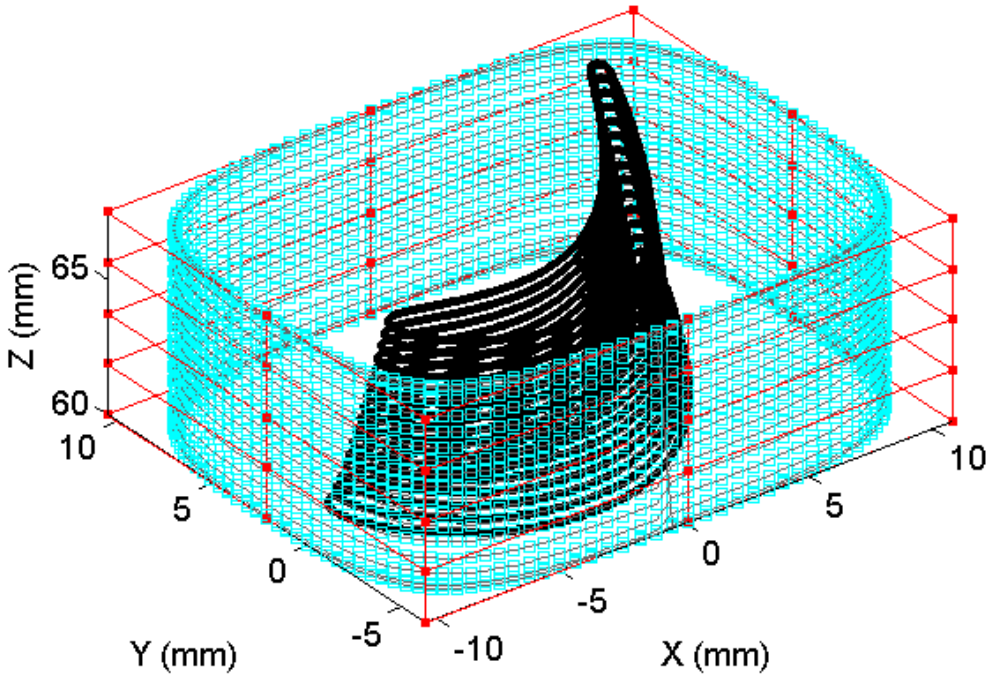


Fig. 183 Initial control polygon (red) and initial subdivision (blue squares) of the turbine blade dataset containing about 90,000 points.

Point-triangle association for distance calculations

The problem of distance calculation falls back to a point-to-triangle problem just like the ICP method discussed in Chapter 2. After the subdivided points $\mathbf{Q}^{(k)}$ are calculated, a triangular mesh is built on the ordered points and this triangulation is referred to as $\mathrm{Tri}(\mathbf{Q}^{(k)})$. $\mathrm{Tri}(\mathbf{Q}^{(k)})$ is represented in magenta color in Fig. 184.

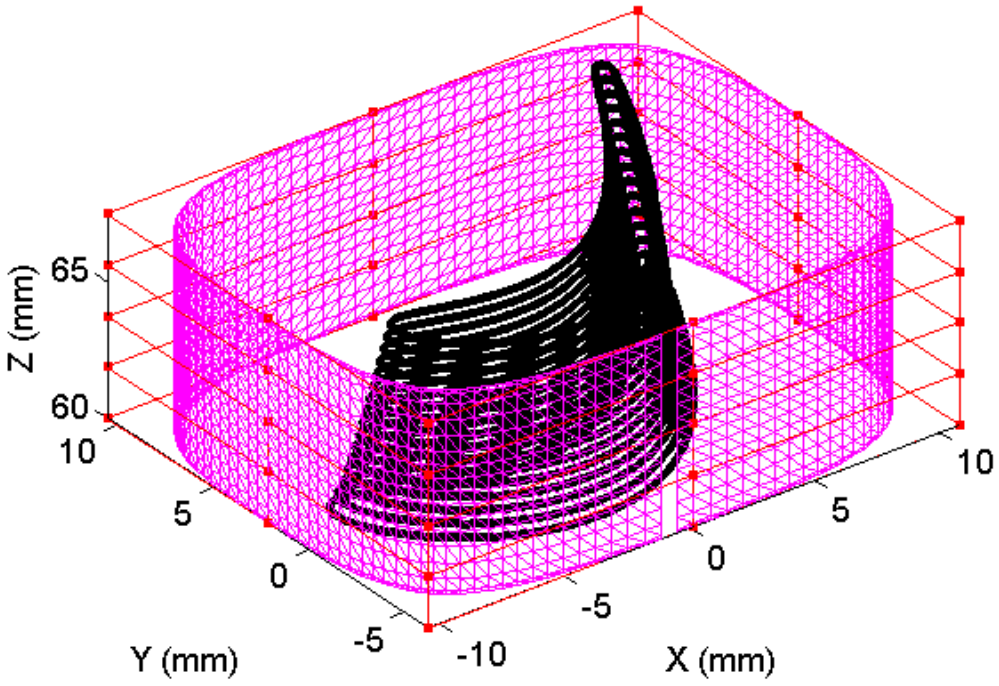


Fig. 184 Triangular mesh of the initial subdivided net.

The subdivided net $\mathbf{Q}^{(k)}$ is piecewise linear. The distances $\boldsymbol{\delta}$ are point-to-triangle distances from the data points $\mathbf{P}$ to the triangular facets $\mathrm{Tri}(\mathbf{Q}^{(k)})$. Just like for curves, these distances are not just geometric distances but they also account for topology. The *inverse* and *direct* associations seen previously are sequentially applied to give a good correspondence between data points and B-Spline facets with a preservation of topology. The *inverse* association routine associates to each triangle's barycenter, the nearest data point in the dataset. The *direct* association matches, in a backward correspondence, the data points to the subdivided net's triangular facets $\mathrm{Tri}(\mathbf{Q}^{(k)})$. In the intermediate step in which a data point is associated to its closest facet in $\mathrm{Tri}(\mathbf{P})^{(k)}$, which is the segmentation of the data points, the nearest facet in $\mathrm{Tri}(\mathbf{P})^{(k)}$ is selected carefully by taking into account the orientation of the facets and their order. With a topological consideration, the association process gives the shortest distance

$\delta_i{}^{(k)}$ separating a data point p_i from its closest triangular facet in the subdivision net. Knowing that each facet in Tri($\mathbf{Q}^{(k)}$) consists of 3 vertices that belong to the subdivided net of the B-Spline surface, the correspondence between a data point and 3 subdivided points is considered in the construction of an augmented optimization matrix $\mathbf{A}^{(k)}$ in which each vertex is taken into account with the same distance magnitude.

Active contour deformation scheme

The distance $\delta_i{}^{(k)}$ is signed and its value is integrated in the objective function to be minimized in a Least-Squares sense. Here again, the optimization direction can be chosen by the user and it can either be an optimization in the normal directions of the control points or an optimization in the three independent directions, x, y and z. In the first case, normal vectors are calculated as the resultants of the normal vectors to the incident facets at the control points (Fig. 185). In the second case, the vector $\mathbf{t}^{(k)}$ is decomposed into three separate components, $\mathbf{t}_\mathbf{x}^{(k)}$, $\mathbf{t}_\mathbf{y}^{(k)}$ and $\mathbf{t}_\mathbf{z}^{(k)}$ and three systems are solved separately, along each of the concerned directions.

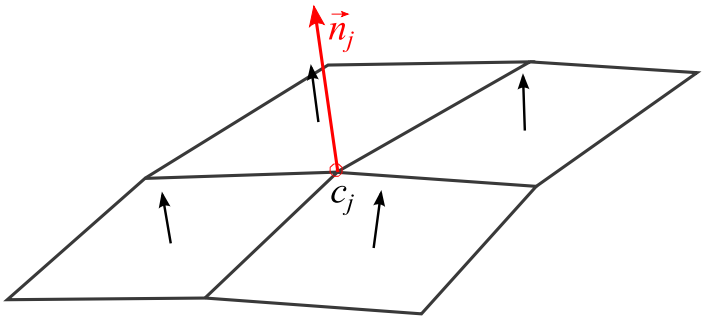


Fig. 185 Approximating the normal direction at a control point.

The system to solve here is written in a similar way as for curves but the matrices are much larger in size. Each distance δ_i is considered thrice, each time for one vertex of the corresponding triangular facet of the subdivided net. This is primordial because the transformation matrix $\mathbf{M}^{(k)}$ correlates the subdivided points $\mathbf{Q}^{(k)}$ to the control net and not the subdivided facets Tri$\left(\mathbf{Q}^{(k)}\right)$ to the control net (68). Instead of taking the index of a single line in matrix $\mathbf{M}^{(k)}$ to construct $\mathbf{A}^{(k)}$ for each data point p_i, the indices of three lines corresponding to the vertices $\{q_{i,1}, q_{i,2}, q_{i,3}\}$ of a triangular facet in Tri$\left(\mathbf{Q}^{(k)}\right)$ are selected for

the same distance δ_i (Fig. 186). The elements of $\mathbf{A}^{(k)}$ for each data point consist of a block of 3 entries at a time.

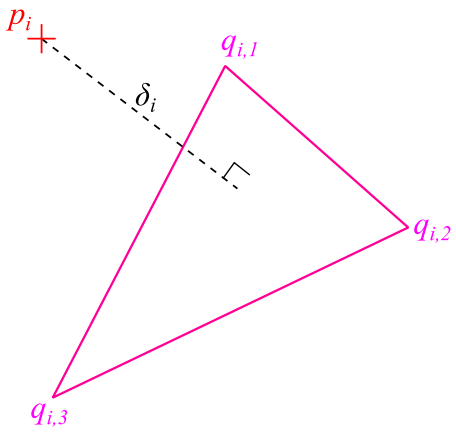


Fig. 186 Correspondence between a data point and a triangular facet of the subdivided net. The distance δ_i is orthogonal and the 3 vertices of the facet are considered in the construction of the optimization matrix $\mathbf{A}^{(k)}$.

Knot insertion

The DBACD algorithm iterates until either convergence or a maximum number of inner iterations has been reached. In the latter case, where the current number of control points is not sufficient to fit the B-Spline surface to the data points accurately (average of residual errors must be below the tolerance ϵ), local knot insertions are applied. For B-Spline surfaces, an inserted knot at a given u-value should be inserted and applied on all v-direction polygons. Similarly, an inserted knot at a given v-value should be applied on all u-direction polygons.

IV. Conclusion

Freeform surface reconstruction techniques are well developed especially in the fields of computer graphics and reverse engineering. In computer graphics, meshing techniques are employed in order to approximate the surface with a piecewise linear model that is the surface mesh of the data points. Usually in these applications, the data points are simulated on a CAD model and therefore the sampling strategy and density can be controlled in order to fulfill the requirements related to the key condition of the reconstruction method, the ε-sampling condition. All the reconstruction methods that are based on the combinatorial Voronoi and

Delaunay structures need this condition to guarantee a geometrical and topological equivalent surface that is also manifold. However, this condition is not sufficient because a lower bound on it is also shown to be necessary. This lower bound is not easily computable and can be variable from one example to another.

The reconstruction methods that are based on implicit techniques and the calculation of implicit functions mainly require that the normal direction at each data point is known. Some methods assume the dataset orientation to be known while some other methods compute it either by principal components analysis (PCA) or by some techniques relying on the Voronoi diagram.

Whilst the purpose of the computer graphics community is mainly visual, the dedicated methods are well adapted and their outcomes can be useful in other applications. For instance, in reverse engineering, some methods that attempt to reconstruct surfaces are based on meshes. A mesh is constructed on the dataset and then it is transformed into a quadrilateral mesh before being approximated by B-Spline patches or NURBS. These methods are generally not robust.

Other techniques in reverse engineering fit B-Spline or NURBS curves and surfaces directly on the dataset. These techniques are more and more efficient but not completely robust. A major limitation to these techniques is that they require some knowledge about the underlying curve or surface to the data points. It is assumed that an initial curve/surface can be constructed in such a way that it is close to the data points and close to the shape of the target curve/surface. By being close, the location parameters which are the parameters of the projections of data points onto the model are computed correctly. Nevertheless, this assumption cannot be guaranteed, especially when the underlying curve or surface is unknown.

Consequently, we have come to propose active contour deformation algorithm for curves that does not require any knowledge of the original curve. Curve initialization is not inevitable anymore. Additionally, the DBACD algorithm is a discrete approach to fit B-Spline curves to points in the plane and does not require any computation of location parameters. Point projections are calculated based on an association procedure that takes into account geometrical distances with a preservation of topology: an inverse association matches B-Spline subdivided segments to data points and then a direct association matches all remaining data points to subdivided B-Spline segments.

The different tests we have carried out show that the algorithm is robust to relative positions of the dataset and the initial B-Spline control polygon. Knot insertion allows

improving the accuracy of the fit but is not sufficient because convergence tolerance is not necessarily met. Moreover, excessive knot insertions (addition of control points) might end up creating self-intersections within the active B-Spline curve. The remedy to these issues is the addition of a fairing term that would take into account the smoothness of the active B-Spline curve at each iteration and allow attaining the required tolerance in general. In the case of the turbine blade profile, the computational time is considerably low (few seconds) for a dataset containing 1,200 points and can be made even faster with parallelization. In the case of the turbine blade surface, where about 100,000 points are involved, computational time for the major steps of the algorithm (initialization and distances computation) remains very low of the order of a few tens of seconds.

The DBACD algorithm can perform the approximation of unorganized and noisy datapoints in the plane measured on a freeform closed curve. The application can be easily extended to open curves in the plane. However, the algorithm requires further adjustments, namely in regards to robustness of convergence and computational time. When those improvements are met, we believe that the DBACD algorithm can be extended to curves and surfaces in 3D quite simply.

General conclusion

Complex surfaces can be classified according to their geometrical shape complexity and their tolerance specification. Aspheres belong to the class of surfaces that are not very complex in shape but which require a nanometric precision in manufacturing and measurement. Turbine blades belong to the class of surfaces that are very complex in shape but necessitate less precision than optical surfaces. The tolerance sought is sub-micrometric.

Although high-precision manufacturing technologies exist and are applied today, an equal-level-of-precision measurement remains inevitable in order to inspect parts that come out of production lines. Measurement is what gives an insight of the dimensional characteristics of a shape as well as of its form. The UHPMM for aspherical and freeform optics, as well as the CMM for turbine blades, are widely used in metrology of freeform parts. These machines can handle different measuring instruments from a variety of scanning speeds and precisions. The mostly used probing system is the stylus which can achieve measurements with nanometric level of uncertainty. Despite the advances of stylus profilometry and the fact that it measures points in an ordered sequence/strategy, this technique's inherent limitation is the contact with the target. In contrast, some new optical measuring instruments and probing systems can generate very large volumes of data exceeding a million points within a much shorter amount of time. However, the data can be highly unorganized, making it impossible to infer notions about the geometry or the topology of the underlying surface to the points.

Aspherical surfaces specification is clearly defined in ISO standards and states that an asphere shall be specified with respect to form tolerance according to a known mathematical model of the asphere. Turbine blades do not have known mathematical models and their specifications in the standards are rather according to cross-sectional profiles or at best according to partial surface features, such as pressure or suction surface and leading or trailing edge surface. In all cases, the specification of turbine blades without a known model makes the approach of surface reconstruction very different than the approach to aspherical surface reconstruction. Therefore, we have divided surface reconstruction into two separate aspects: an aspect involving fitting techniques when an analytical model of the surface is known and another aspect consisting of active contour deformation when the surface model is either a CAD model or unknown. In both cases, we did not assume any organization of the data points to remain in the general case that disregards measurement strategies.

The present thesis report has addressed the issues related to the EMRP project IND10: FORM entitled "*Optical and tactile metrology for absolute form characterization*". We have

proposed a fitting algorithm for aspherical surfaces based on the L-BFGS method. The processing scheme can handle very large volumes of data and can fit an aspherical shape to the raw data within a few seconds in an automated manner. This conclusion is founded on a thorough comparative study based on simulating data containing errors and validating the use of some algorithms for aspherical surface fitting. Precisely, we have compared our newly used L-BFGS algorithm to classical algorithms used in metrology such as the Levenberg-Marquardt algorithm (LM) and the Iterative Closest Point algorithm (ICP). The comparison showed the superior performances of L-BFGS as compared to LM and ICP while preserving accuracy. We additionally studied the effect of having a variable sampling density in the dataset as well as having variable regions of the surface represented in the dataset. This was basically done in the aim of proposing an approach to compare different measurements of the same optical part performed by the different project partners of IND10: FORM. The simulation results revealed that the sampling density does not affect the residual errors when points are taken all over the surface, except when the number of points is considerably reduced ($N < 0.01$ % of the original points). However, the residual errors are slightly altered with the variation of the sampled region and the variation of the density in the respective sampled regions. Consequently, the comparison of the actual measurements done by the project partners was based on taking data points in a common region (aperture) of the asphere. The results revealed that the residual errors were not exactly similar although the same aspherical lens model was measured.

Furthermore, we have experimented on the type of aspherical model used to describe the measured lens. A comparison between using the classical model defined in ISO standards and the newly proposed Forbes conic model led to the following conclusions: the Forbes conic representation of aspherical surfaces consists of independent model parameters and by that was better suited for a fitting involving model parameters estimation. Conversely, the fitting runtime was almost twice slower. If the fitting does not need to be done for model parameters estimation but only for motion parameters, we recommend using the classical model definition.

In future works, the MinMax fitting of aspherical surfaces is going to be addressed in order to better evaluate form errors. The development of a robust and deterministic algorithm for this purpose is going to be the subject of the next European project: EMPIR.

The present thesis report has also addressed the issues related to the general problem of freeform shape reconstruction and more particularly, the reconstruction of turbine blades. For

that we have built a benchmark and thorough study of mesh reconstruction techniques: the techniques based on combinatorial structures and the techniques based on implicit functions. Although these techniques have been mainly developed for computer graphics applications, they drag interest to several aspects. The topological and geometrical guarantees that the techniques based on combinatorial structures can offer, raise the first aspect about sampling density, expressed in terms of the ε-sampling condition. We have shown that this condition is necessary to guarantee geometrical closeness and topological correctness but is not sufficient. In fact, a lower bound on the sampling density ε_0, was demonstrated to be also necessary in our applications.

Knowing that turbine blades are still specified today according to cross-sectional profiles, we have come to propose a new active contour deformation algorithm that consists of a planar B-Spline curve model that will iteratively deform until it matches the shape of the underlying curve to the points. Active contour deformation techniques exist in literature but are all based on at least a good initial parameterization or a good B-Spline curve initialization. Our proposed algorithm uses a discrete representation of the B-Spline and by that disregards the issues of a necessarily good initial parameterization or a good initial curve to the datapoints. Instead, a robust point-to-segment association procedure was introduced to preserve both the geometry and the topology of the underlying curve to the points. Computational time is quite effective but can be enhanced by parallel programming. In actual time, the algorithm can handle very small datasets of some thousands of points within negligible time and some hundreds of thousands of points within a few tens of seconds ($\approx$ 100,000 points in 130 s).

The Discrete B-Spline Active Contour Deformation (DBACD) algorithm can converge with a sub-micrometric tolerance which matches the form tolerance of turbine blades, however, the robustness of the algorithm regarding convergence is not yet guaranteed at all times.

So, this issue of convergence is going to be addressed in future works in order to have a robust fitting that works for any freeform curve. We would also like to extend this algorithm to surfaces in 3D in such a way to solve the turbine blade problem when a non measured cross-section needs to be characterized or when the specification of the blade is based on surface features (pressure surface, suction surface, leading edge surface and trailing edge surface features).

Bibliography

[1] M. Rodgers, K. Thompson, Benefits of Freeform Mirror Surfaces in Optical Design, in: Proc. ASPE Winter Top. Meet. Free. Opt., North Carolina USA, 2004: pp. 73–78.

[2] R. Henselmans, *Non-contact Measurement Machine for Freeform Optics*, Technische Universiteit Eindhoven, Ph.D. thesis, 2009.

[3] Bureau International des Poids et Mesure (BIPM), available: http://www.bipm.org.

[4] Guide to the expression of Uncertainty in Measurement (GUM-Part 3), , 2008.

[5] *International vocabulary of terms in legal metrology (VIML)*, Technical Report, 2013.

[6] Y. Li, P. Gu, Free-form surface inspection techniques state of the art review, Comput. Des. 36 (13) pp. 1395–1417 , 2004.

[7] E. Savio, L. DeChiffre, An artefact for traceable freeform measurements on coordinate measuring machines, Precis. Eng. 26 (1) pp. 58–68 , 2002.

[8] E. Savio, L. De Chiffre, R. Schmitt, Metrology of freeform shaped parts, CIRP Ann. - Manuf. Technol. 56 (2) pp. 810–835 , 2007.

[9] M.L. Culpepper, M.V. Kartik, C. DiBiasio, Design of integrated eccentric mechanisms and exact constraint fixtures for micron-level repeatability and accuracy, Precis. Eng. 29 (1) pp. 65–80 , 2005.

[10] L. Zhu, J. Barhak, V. Srivatsan, R. Katz, Efficient registration for precision inspection of free-form surfaces, Int. J. Adv. Manuf. Technol. 32 (5-6) pp. 505–515 , 2006.

[11] E.M. Bispo, R.B. Fisher, Free-form Surface Matching for Surface Inspection, in: Proc. 6th IMA Conf. Math. Surfaces, Clarendon Press, 1994: pp. 119–136.

[12] G. Sharp, S.W. Lee, D.K. Wehe, ICP Registration using Invariant Features, IEEE Trans. Pattern Anal. Mach. Intell. 24 (1) pp. 90–102 , 2002.

[13] C.F. Cheung, L. Kong, M. Ren, Measurement and characterization of ultra-precision freeform surfaces using an intrinsic surface feature-based method, Meas. Sci. Technol. 21 (11) pp. 115109 , 2010.

[14] H. Nouira, J. Salgado, N. El-Hayek, S. Ducourtieux, A. Delvallée, N. Anwer, Setup of a high-precision profilometer and comparison of tactile and optical, Meas. Sci. Technol. 25 , 2014.

[15] ISO 10110-12, Optics and Photonics - Preparation of drawings for optical elements and systems, Part 12: Aspheric surfaces, 2007.

[16] Geomnia SAS, available: www.geomnia.eu.

[17] E. Catmull, J. Clark, Recursively generated B-Spline surfaces on arbitrary topological meshes, Comput. Aided Des. 10 (6) pp. 350–355 , 1978.

[18] M. Eck, H. Hoppe, Automatic reconstruction of B-spline surfaces of arbitrary topological type, in: Proc. 23rd Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '96, ACM Press, New York, New York, USA, 1996: pp. 325–334.

[19] Y. Lai, L. Kobbelt, S. Hu, An Incremental Approach to Feature Aligned Quad Dominant Remeshing, in: Proc. 2008 ACM Symp. Solid Phys. Model., 2008: pp. 137–145.

[20] C. Loop, S. Schaefer, *Approximating Catmull-Clark subdivision surfaces with bicubic patches*, Technical Report, 2008.

[21] J. Peters, Patching Catmull-Clark Meshes, in: Proc. 27th Annu. Conf. Comput. Graph. Interact. Tech., ACM, 2000: pp. 255–258.

[22] J. Stam, Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values, in: Proc. 25th Annu. Conf. Comput. Graph. Interact. Tech., ACM Press, New York, New York, USA, 1998: pp. 395–404.

[23] L.. Zhang, R.. Zhou, J.. Zhu, X. Wu, Piecewise B-spline surfaces fitting to arbitrary triangle meshes, CIRP Ann. Technol. 51 (1) pp. 131–134 , 2002.

[24] X. Zhang, H. Zhang, X. He, M. Xu, X. Jiang, Chebyshev fitting of complex surfaces for precision metrology, Measurement. 46 (9) pp. 3720–3724 , 2013.

[25] X. Zhang, X. Jiang, P. Scott, A Minimax Fitting Algorithm for Ultra-Precision Aspheric Surfaces, in: J. Phys. Conf. Ser., 2011: p. 012031.

[26] A. Küng, A. Nicolet, F. Meli, Virtual CMM method applied to aspherical lens parameters calibration, in: Proc. 13th Int. Conf. Eur. Soc. Precising Eng. Nanotechnol., 2013: pp. 83–86.

[27] P. Alliez, D. Cohen-steiner, Y. Tong, M. Desbrun, Voronoi-based Variational Reconstruction of Unoriented Point Sets, in: A. Belyaev, M. Garland (Eds.), Eurographics Symp. Geom. Process., 2007: pp. 39–48.

[28] Q. Merigot, M. Ovsjanikov, L.J. Guibas, Voronoi-based Curvature and Feature Estimation from Point Clouds, IEEE Trans. Vis. Comput. Graph. 17 (6) pp. 1–14 , 2010.

[29] N.J. Mitra, A. Nguyen, L. Guibas, Estimating Surface Normals in Noisy Point Cloud Data, Int. J. Comput. Geom. Appl. 14 (04n05) pp. 261–276 , 2004.

[30] D. Ouyang, H.-Y. Feng, On the normal vector estimation for point cloud data from smooth surfaces, Comput. Des. 37 (10) pp. 1071–1079 , 2005.

[31] S. Liu, C.C.L. Wang, Orienting unorganized points for surface reconstruction, Comput. Graph. 34 (3) pp. 209–218 , 2010.

[32] I. Arbizu, C. Perez, Surface roughness prediction by factorial design of experiments in turning processes, J. Mater. Process. Technol. 143-144 pp. 390–396 , 2003.

[33] H. Hansen, K. Carneiro, H. Haitjema, L. DeChiffre, Dimensional Micro and Nano Metrology, CIRP Ann. - Manuf. Technol. 55 (22) pp. 721–743 , 2006.

[34] X. Jiang, D. Whitehouse, Technological shifts in surface metrology, CIRP Ann. - Manuf. Technol. 61 (2) pp. 815–836 , 2012.

[35] A. Weckenmann, P. Kraemer, J. Hoffmann, Manufacturing metrology – State of the art and prospects, in: 9th Int. Symp. Meas. Qual. Control, 2007.

[36] T. Arnold, G. Böhm, R. Fechner, J. Meister, A. Nickel, F. Frost, T. Hänsel, A. Schindler, Ultra-precision surface finishing by ion beam and plasma jet techniques—status and outlook, Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip. 616 (2-3) pp. 147–156 , 2010.

[37] Y. Shi, D. Zheng, L. Hu, Y. Wang, L. Wang, NC polishing of aspheric surfaces under control of constant pressure using a magnetorheological torque servo, Int. J. Adv. Manuf. Technol. 58 pp. 1061–1073 , 2012.

[38] P.R. Dumas, J. Fleig, G.W. Forbes, D. Golini, W.I. Kordonski, P.E. Murphy, A.B. Shorey, M. Tricard, Flexible Polishing and metrology solutions for free-form optics, in: ASPE Proceedings, Free. Opt. Des. Fabr. Metrol. Assem., ASPE, 2004.

[39] I. Fortmeier, M. Stavridis, A. Wiegmann, M. Schulz, G.B. Baer, C. Pruss, W. Osten, C. Elster, Results of a Sensitivity Analysis for the Tilted-Wave Interferometer, in: Fringe 2013 - 7th Int. Work. Adv. Opt. Imaging Metrol., Springer Berlin Heidelberg, 2014: pp. 701–706.

[40] X. Jiang, P. Scott, D. Whitehouse, L. Blunt, Paradigm shifts in surface metrology. Part II. The current shift, Proc. R. Soc. A Math. Phys. Eng. Sci. 463 (2085) pp. 2071–2099 , 2007.

[41] A. Weckenmann, G. Peggs, J. Hoffmann, Probing systems for dimensional micro- and nano-metrology, Meas. Sci. Technol. 17 (3) pp. 504–509 , 2006.

[42] A. Küng, F. Meli, R. Thalmann, Ultraprecision micro-CMM using a low force 3D touch probe, Meas. Sci. Technol. 18 (2) pp. 319–327 , 2007.

[43] T. Ruijl, *Ultra Precision Coordinate Measuring Machine – Design, Calibration and Error Compensation*, Technische Universiteit Delft, Ph.D. thesis, 2001.

[44] A. Vissière, H. Nouira, M. Damak, O. Gibaru, J.-M. David, A newly conceived cylinder measuring machine and methods that eliminate the spindle errors, Meas. Sci. Technol. 23 (9) pp. 094015 , 2012.

[45] M. Vermeulen, *High-Precision 3D-Coordinate Measuring Machine: Design and Prototype-Development*, Technische Universiteit Eindhoven, Ph.D. thesis, 1999.

[46] A. Weckenmann, T. Estler, G.N. Peggs, D. McMurtry, Probing Systems in Dimensional Metrology, CIRP Ann. - Manuf. Technol. 53 (2) pp. 657–684 , 2004.

[47] J. Chen, C. Ling, Improving the machine accuracy through machine tool metrology and error correction, Int. J. Adv. Manuf. Technol. 11 (3) pp. 198–205 , 1996.

[48] M. Conroy, J. Armstrong, A comparison of surface metrology techniques, J. Phys. Conf. Ser. 13 pp. 458–465 , 2005.

[49] G. Bönsch, E. Potulski, Measurement of the refractive index of air and comparison with modified Edlén's formulae, Metrologia. 35 (2) pp. 133 , 1998.

[50] I. Malinovsky, A. Titov, J.A. Dutra, H. Belaïdi, R. Dos Santos Franca, C.A. Massone, Toward Subnanometer Uncertainty in Interferometric Length Measurements of Short Gauge Blocks, Appl. Opt. 38 (1) pp. 101–112 , 1999.

[51] ISO 17450-1, Geometrical Product Specifications (GPS) - General Concepts, Part 1: Model for geometrical specification and verification, 2012.

[52] P.J. Scott, X. Jiang, Freeform surface characterisation: theory and practice, J. Phys. Conf. Ser. 483 pp. 012005 , 2014.

[53] M. McCarthy, S. Brown, A. Evenden, A. Robinson, NPL freeform artifact for verification of non-contact measuring systems, in: Proc. SPIE 7864, Three-Dimensional Imaging, Interact. Meas., 2011: p. 78640K–78640K–13.

[54] W. Khaled, S. Reichling, O. Bruhns, H. Boese, M. Baumann, G. Monkman, S. Egersdoerfer, H. Freimuth, H. Ermert, A New Haptic Sensor Actuator System for Virtual Reality Applications in Medicine, in: R. Ellis, T. Peters (Eds.), Med. Image Comput. Comput. Interv., 2003: pp. 132–140.

[55] P. Payeur, C. Pasca, A.M. Cretu, E.M. Petriu, Intelligent haptic sensor system for robotic manipulation, Instrum. Meas. 54 (4) pp. 1583–1592 , 2005.

[56] J. Chaves-Jacob, J.M. Linarès, J.M. Sprauel, Increasing of surface quality in friction free-form surfaces of knee prosthesis, CIRP Ann. - Manuf. Technol. 60 (1) pp. 531–534 , 2011.

[57] S. Marcos, S. Barbero, I. Jiménez-Alfaro, Optical quality and depth-of-field of eyes implanted with spherical and aspheric intraocular lenses, J. Refract. Surg. 21 (3) pp. 223–35 , 2005.

[58] N.E. Claytor, D.M. Combs, O.M. Lechuga, J.J. Mader, J. Udayasankaran, An overview of freeform optics production, in: Proc. ASPE Winter Top. Meet., North Carolina USA, 2004.

[59] F.Z. Fang, X.D. Zhang, A. Weckenmann, G.X. Zhang, C. Evans, Manufacturing and measurement of freeform optics, CIRP Ann. - Manuf. Technol. 62 (2) pp. 823–846 , 2013.

[60] E. Robert, What is the relevance of asphericity in today's ophthalmic practice ?, Philipp. J. Ophthalmol. 36 (1) pp. 5–6 , 2011.

[61] ISO 1101, Geometrical Product Specifications (GPS) - Geometrical Tolerancing - Tolerances of Form, Orientation, Location and Run-out, 2004.

[62] J.E. Makem, H. Ou, C.G. Armstrong, A virtual inspection framework for precision manufacturing of aerofoil components, Comput. Des. 44 (9) pp. 858–874 , 2012.

[63] ISO 10110-5, Optics and Photonics - Preparation od drawings for optical elements and systems, Part 5: Surface form tolerances, 2007.

[64] R. Schuhmann, Aspheres in Optical Design – Changes and Limits, , 2007.

[65] K.F. Beckstette, Trends in aspheres and freeform optics, , 2008.

[66] G.W. Forbes, Shape specification for axially symmetric optical surfaces., Opt. Express. 15 (8) pp. 5218–26 , 2007.

[67] G.W. Forbes, C.P. Brophy, Asphere, O Asphere, how shall we describe thee?, Opt. Syst. Des. 7100 (1) pp. 710002–710002–15 , 2008.

[68] G.W. Forbes, Effective Characterization of the Nominal Shape of Aspheric Optics, Opt. Des. Eng. , 2010.

[69] G.W. Forbes, Characterizing the shape of freeform optics., Opt. Express. 20 (3) pp. 2483–99 , 2012.

[70] QED Technologies Inc., available: http://www.qedmrf.com.

[71] R.J. Campbell, P.J. Flynn, A Survey Of Free-Form Object Representation and Recognition Techniques, Comput. Vis. Image Underst. 81 (2) pp. 166–210 , 2001.

[72] P. Besl, The free-form surface matching problem, in: Mach. Vis. Three-Dimensional Scenes, 1990: pp. 25–40.

[73] K. Garrard, T. Bruegge, J. Hoffman, T. Dow, A. Sohn, Design tools for freeform optics, in: Proc. SPIE 5874, Curr. Dev. Lens Des. Opt. Eng. VI, San Diego, 2005.

[74] A. Lasemi, D. Xue, P. Gu, Recent development in CNC machining of freeform surfaces: A state-of-the-art review, Comput. Des. 42 (7) pp. 641–654 , 2010.

[75] M. Petitcuenot, L. Pierre, B. Anselmetti, ISO specifications of complex surfaces: Application on aerodynamic profiles, in: 13th CIRP Conf. Comput. Aided Toler., 2014.

[76] H.-S. Kim, E.-J. Kim, B.-S. Song, Diamond turning of large off-axis aspheric mirrors using a fast tool servo with on-machine measurement, J. Mater. Process. Technol. 146 (3) pp. 349–355 , 2004.

[77] G. Wanders, Advanced techniques in freeform manufacturing, , 2006.

[78] E. Brinksmeier, D. Grimme, W. Preuss, Generation of Freeform Surfaces by Diamond Machining, in: Proc. 17th Annu. Meet. ASPE, 2002: pp. 542–545.

[79] E. Brinksmeier, L. Autschbach, Ball-end Milling of Free-form Surfaces for Optical Mold Inserts, in: Proc. ASPE Annu. Meet., Orlando, 2004: pp. 4–7.

[80] M. Weiser, Ion beam figuring for lithography optics, Nucl. Instruments Methods Phys. Res. Sect. B Beam Interact. with Mater. Atoms. 267 (8-9) pp. 1390–1393 , 2009.

[81] D. Walker, D. Brooks, R. Freeman, A. King, G. McCavana, R. Morton, D. Riley, J. Simms, The first aspheric form and texture results from a production machine embodying the precession process, in: Proc. SPIE 46th Annu. Meet., San Diego, 2001: pp. 267–276.

[82] C. Brecher, S. Lange, M. Merz, F. Niehaus, C. Wenzel, M. Winterschladen, NURBS Based Ultra-Precision Free-Form Machining, CIRP Ann. - Manuf. Technol. 55 (1) pp. 547–550 , 2006.

[83] Zeeko LTD, available: http://www.zeeko.co.uk.

[84] eo Edmund Optics, available: http://www.edmundoptics.com/technical-resources-center/optics.

[85] P.F. Bariani, S. Bruschi, T. Dal Negro, Integrating physical and numerical simulation techniques to design the hot forging process of stainless steel turbine blades, Int. J. Mach. Tools Manuf. 44 (9) pp. 945–951 , 2004.

[86] Sandvik Coromant's webiste, available: www.sandvik.coromant.com.

[87] N.S. Qu, Z.Y. Xu, Improving machining accuracy of electrochemical machining blade by optimization of cathode feeding directions, Int. J. Adv. Manuf. Technol. 68 (5-8) pp. 1565–1572 , 2013.

[88] P.S. Veers, T.D. Ashwill, H.J. Sutherland, D.L. Laird, D.W. Lobitz, D. a. Griffin, J.F. Mandell, W.D. Musial, K. Jackson, M. Zuteck, A. Miravete, S.W. Tsai, J.L. Richmond, Trends in the Design, Manufacture and Evaluation of Wind Turbine Blades, Wind Energy. 6 (3) pp. 245–259 , 2003.

[89] H. Nouira, R. Bergmans, A. Küng, H. Piree, R. Henselmans, H.A.. Spaan, Ultra-high precision CMMs as well as tactile and optical single scanning probes evaluation in dimensional metrology, Int. J. Metrol. Qual. Eng. (proof version) pp. 1–13 , 2014.

[90] E. Abbe, Messapparte für physiker, Zeits Fürinstrumenten-Kd. 10 pp. 446–448 , 1890.

[91] R. Leach, Book: *Optical measurement of surface topography*, Springer, 2011.

[92] D. Whitehouse, Book: *Surfaces and their measurement*, Elsevier, 2004.

[93] G.B. Baer, J. Schindler, C. Pruss, W. Osten, Comparison of alignment errors in asphere metrology between an interferometric null-test measurement and a non-null

measurement with the tilted-wave-interferometer, in: Proc. SPIE 8884, Optifab 2013, 2013: p. 88840T–88840T–7.

[94] T.H. Yoon, J. Ye, J.L. Hall, J.-M. Chartier, Absolute frequency measurement of the iodine-stabilized He-Ne laser at 633 nm, Appl. Phys. B. 72 (2) pp. 221–226 , 2001.

[95] I. Misumi, S. Gonda, T. Kurosawa, K. Takamasu, Uncertainty in pitch measurements of one-dimensional grating standards using a nanometrological atomic force microscope, Meas. Sci. Technol. 14 (4) pp. 463–471 , 2003.

[96] I. Widdershoven, R.L. Donker, H. a M. Spaan, Realization and calibration of the "Isara 400" ultra-precision CMM, J. Phys. Conf. Ser. 311 pp. 012002 , 2011.

[97] L. Cacace, *Development of a nanometer accuracy non-contact probe for NANOMEFOS*, Eindhoven University of Technology, Ph.D. thesis, 2006.

[98] A. Vissière, H. Nouira, M. Damak, O. Gibaru, J.-M. David, Concept and architecture of a new apparatus for cylindrical form measurement with a nanometric level of accuracy, Meas. Sci. Technol. 23 (9) , 2012.

[99] J.B. Bryan, D.L. Carter, Design of a new error-corrected co-ord,nate measur,ng mach,ne, Precis. Eng. 1 (3) pp. 125–128 , 1979.

[100] A. Vissière, *Mesure de cylindricité de très haute exactitude: Développement d'une nouvelle machine de référence*, Arts et Métiers ParisTech, Ph.D. thesis, 2013.

[101] G. Baer, C. Pruss, J. Schindler, W. Osten, *The Tilted Wave Interferometer (TWI): A quick and flexible approach to measure aspheric and freeform surfaces*, Technical Report, 2013.

[102] G. Baer, J. Schindler, C. Pruss, W. Osten, Measurement of aspheres and free-form surfaces with the Tilted-Wave-Interferometer, in: Fringe 2013, Springer Berlin Heidelberg, 2014: pp. 87–95.

[103] C. Pruss, G.B. Baer, J. Schindler, W. Osten, J. Siepmann, Flexible and Rapid Measurement, Opt. Photonik. 8 (4) pp. 61–64 , 2013.

[104] G.B. Baer, C. Pruss, W. Osten, Correction of misalignment introduced aberration in non-null test measurements of free-form surfaces, J. Eur. Opt. Soc. 8 (13074) pp. 1–5 , 2013.

[105] V. Zelený, V. Stejskal, The most recent ways of CMM calibration, in: XVI IMEKO World Congr., 2000: pp. 2–7.

[106] R. Thalmann, A New High Precision Length Measuring Machine, in: 8th Int. Congr. Metrol., Besancon, 1997: pp. 1–4.

[107] ISO 10360-6, Geometrical Product Specifications (GPS) - Acceptance and reverification tests for coordinate measuring machines (CMM), Part 6: Estimation of errors in computing Gaussian associated features, 2009.

[108] D. Flack, *Measurement Good Practice Guide No. 40-43*, Technical Report, 2001.

[109] D. Vaissiere, *Métrologie tridimensionnelle des états de surface par microscopie confocale à champ étendu*, Ph.D. thesis, 2003.

[110] Micro-epsilon, available: http://www.micro-epsilon.com.

[111] STIL S.A, available: http://www.stilsa.com/.

[112] J.-L. Charron, Book: *Mesures sans contact, Méthodes optiques (partie 1)*, CETIM, Paris, 1999.

[113] H. Nouira, N. El-Hayek, X. Yuan, N. Anwer, Characterization of the main error sources of chromatic confocal probes for dimensional measurement, Meas. Sci. Technol. 25 , 2014.

[114] ISO 25178-602, Geometrical product specifications (GPS) - Surface texture: Areal, Part 602: Nominal characteristics of non-contact (confocal chromatic probe) instruments, 2010.

[115] H. Nouira, N. El-Hayek, X. Yuan, N. Anwer, J. Salgado, Metrological characterization of optical confocal sensors measurements (20 and 350 travel ranges), J. Phys. Conf. Ser. 483 (1) pp. 012015 , 2014.

[116] W. Black, W. Carr, Application of a Differential Interferometer to the Measurement of Heat Transfer Coefficients, Rev. Sci. Instrum. 42 (3) pp. 337–340 , 1971.

[117] N. El-Hayek, H. Nouira, N. Anwer, O. Gibaru, M. Damak, Comparison of tactile and chromatic confocal measurements of aspherical lenses for form metrology, Int. J. Precis. Engineeing Manuf. 15 (5) pp. 821–829 , 2014.

[118] N. El-Hayek, N. Anwer, H. Nouira, O. Gibaru, M. Damak, P. Bourdet, 3D Measurement and Characterization of Ultra-precision Aspheric Surfaces, in: 13th CIRP Conf. Comput. Aided Toler., 2014.

[119] R. Hocken, P. Pereira, J. Bosch, D. Swyt, E. Morse, A. Wechenmann, J. Hoffmann, C.M. Shakarji, S.D. Phillips, Book: *Coordinate Measuring Machines and Systems, Second Edition*, Taylor and Francis Group, 2012.

[120] J.M. Sprauel, J.M. Linarès, J. Bachmann, P. Bourdet, Uncertainties in CMM Measurements, Control of ISO Specifications, CIRP Ann. - Manuf. Technol. 52 (1) pp. 423–426 , 2003.

[121] Renishaw, available: http://www.renishaw.com/.

[122] A. Tarantola, Book: *Inverse Problem Theory*, Society for Industrial and Applied Mathematics, Paris, 2005.

[123] C.M. Shakarji, V. Srinivasan, Theory and Algorithms for Weighted Total Least-Squares Fitting of Lines, Planes, and Parallel Planes to Support Tolerancing Standards, J. Comput. Inf. Sci. Eng. 13 (3) pp. 031008 , 2013.

[124] G. Lukacs, R.R. Martin, A.D. Marshall, Faithful Least-Squares Fitting of Spheres, Cylinders, Cones and Tori for Reliable Segmentation, in: Proc. 5th Eur. Conf. Comput. Vis., Springer, 1998: pp. 671–686.

[125] C.F. Cheung, H. Li, L. Kong, W.. Lee, S. To, Measuring ultra-precision freeform surfaces using a robust form characterization method, Meas. Sci. Technol. 17 pp. 488–494 , 2006.

[126] H. Aceves-Campos, Profile Identification of Aspheric Lenses, Appl. Opt. 37 (34) pp. 8149–8150 , 1998.

[127] Z. Chen, Z. Guo, Q. Mi, Z. Yang, R. Bai, Research of fitting algorithm for coefficients of rotational symmetry aspheric lens, in: 4th Int. Symp. Adv. Opt. Manuf. Test. Technol. Opt. Test Meas. Technol. Equip., The International Society for Optical Engineering, Chengdu, 2009.

[128] W. Sun, J.W. McBride, M. Hill, A new approach to characterising aspheric surfaces, Precis. Eng. 34 (1) pp. 171–179 , 2010.

[129] Z. Zhang, Parameter estimation techniques: A tutorial with application to conic fitting, Image Vis. Comput. 15 (1) pp. 59–76 , 1997.

[130] H. Park, A solution for NURBS modelling in aspheric lens manufacture, Int. J. Adv. Manuf. Technol. 23 (1-2) pp. 1–10 , 2004.

[131] The European Metrology Research Programme, available: http://www.emrponline.eu/2010.

[132] N. El-Hayek, H. Nouira, N. Anwer, M. Damak, O. Gibaru, A new method for aspherical surface fitting with large-volume datasets, Precis. Eng. 38 (4) pp. 935–947 , 2014.

[133] C.M. Shakarji, Least-squares fitting algorithms of the NIST algorithm testing system, J. Res. Natl. Inst. Stand. Technol. 103 (6) pp. 633 , 1998.

[134] H. Yu, M. Bogdan, Levenberg–Marquardt Training, in: Ind. Electron. Handb., 2010: pp. 1–16.

[135] K. Pulli, Multiview registration for large data sets, in: 3-D Digit. Imaging Model., 1999: pp. 160–168.

[136] S.-J. Ahn, Book: *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*, Springer-Verlag, Fraunhofer, 2004.

[137] V. Srinivasan, Computational Metrology for the Design and Manufacture of Product Geometry: A Classification and Synthesis, J. Comput. Inf. Sci. Eng. 7 (1) pp. 3–9 , 2007.

[138] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, Proc. Third Int. Conf. 3-D Digit. Imaging Model. pp. 145–152 , 2001.

[139] F. Pomerleau, F. Colas, R. Siegwart, S. Magnenat, Comparing ICP variants on real-world data sets, Auton. Robots. 34 (3) pp. 133–148 , 2013.

[140] P. Besl, N. McKay, A method for the registration of 3D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) pp. 239–256 , 1992.

[141] K. Borgwardt, Book: *The Simplex Method - Algorithms and Combinatorics*, Springer Berlin Heidelberg, Seattle, 1987.

[142] P. Lancaster, Error analysis for the Newton-Raphson method, Numer. Math. 9 pp. 55–68 , 1966.

[143] E. Süli, D. Mayers, Book: *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.

[144] M.K. Transtrum, J.P. Sethna, Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization, arXiv Prepr. arXiv1201.5885. , 2012.

[145] J. Moré, The Levenberg-Marquardt algorithm: implementation and theory, in: Numer. Anal. Lect. Notes Math., 1978: pp. 105–116.

[146] X. Jiang, X. Zhang, P. Scott, Template Matching of Freeform Surfaces Based on Orthogonal Distance Fitting for Precision Metrology, Meas. Sci. Technol. 21 (4) pp. 045101 , 2010.

[147] P. Halmos, What does the spectral theorem say?, Am. Math. Mon. 70 (3) pp. 241–247 , 1963.

[148] T. Speer, M. Kuppe, J. Hoschek, Global reparametrization for curve approximation, Comput. Aided Geom. Des. 15 (9) pp. 869–877 , 1998.

[149] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Book: *Introduction to Algorithms, 3rd Edition*, MIT Press, 2009.

[150] M. Greenspan, M. Yurick, Approximate k-d tree search for efficient ICP, in: Proc. IEEE 4th Int. Conf. 3-D Digit. Imaging Model., 2003: pp. 442 – 448.

[151] A. Nuchter, K. Lingemann, J. Hertzberg, Cached k-d tree search for ICP algorithm, in: Proc. 6th Int. Conf. 3-D Digit. Imaging Model., Montreal, 2007: pp. 419 – 426.

[152] M. Walker, L. Shao, R. Volz, Estimating 3-D Location Parameters Using Dual Number Quaternions, CVGIP Image Underst. 54 (3) pp. 358–367 , 1991.

[153] K. Arun, T. Huang, S. Blostein, Least-Squares Fitting of Two 3-D Point Sets, Pattern Anal. Mach. Intell. 9 (5) pp. 698–700 , 1987.

[154] A. Fitzgibborn, Robust Registration of 2D and 3D Point Sets, Image Vis. Comput. 21 (13) pp. 1145–1153 , 2001.

[155] H. Liu, J. Yan, D. Zhang, Three-dimensional surface registration: A neural network strategy, Neurocomputing. 70 (1-3) pp. 597–602 , 2006.

[156] S.-Y. Park, M. Subbarao, An accurate and fast point-to-plane registration technique, Pattern Recognit. Lett. 24 (16) pp. 2967 – 2976 , 2003.

[157] N. El-Hayek, H. Nouira, N. Anwer, M. Damak, O. Gibaru, Reconstruction of freeform surfaces for metrology, J. Phys. Conf. Ser. 483 (1) pp. 012003 , 2014.

[158] J. Nocedal, S. Wright, Book: *Numerical optimization*, 2nd Editio, Springer, 2006.

[159] W. Zheng, P. Bo, Y. Liu, W. Wang, Fast B-spline curve fitting by L-BFGS, Comput. Aided Geom. Des. 29 (7) pp. 448–462 , 2012.

[160] L. Dong, J. Nocedal, On the Limited Memory BFGS method for large scale Optimization, Math. Program. 45 pp. 503–528 , 1989.

[161] J. Nocedal, Updating Quasi-Newton Matrices with Limited Storage, Math. Comput. 35 (151) pp. 773 , 1980.

[162] Y. Lin, K. Damodharan, C.M. Shakarji, Standardised Reference Data Sets Generation for Coordinate Measuring Machine Software Assessment, Int. J. Adv. Manuf. Technol. pp. 819–830 , 2001.

[163] E. Marsh, Book: *Precision Spindle Metrology*, DEStech publishing, 2010.

[164] G. Davis, J. Roblee, A. Hedges, Comparison of freeform manufacturing techniques in the production of monolithic lens arrays, in: Proc. SPIE 7426, Opt. Manuf. Test. VIII, 2009: p. 742605.

[165] N. El-Hayek, N. Anwer, H. Nouira, M. Damak, O. Gibaru, Characterization of Ultra-precise aspherical surfaces using Forbes equation, in: Proc. Jt. Conf. Mech. Des. Eng. Adv. Manuf., Toulouse, France, 2014.

[166] W. Ma, J.-P. Kruth, NURBS Curve and Surface Fitting for Reverse Engineering, Int. J. Adv. Manuf. Technol. 14 pp. 918–927 , 1998.

[167] L. Piegl, W. Tiller, Book: *The NURBS Book*, 1997.

[168] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, ACM SIGGRAPH Comput. Graph. 26 (2) pp. 71–78 , 1992.

[169] H. Edelsbrunner, Surface Reconstruction by Wrapping Finite Sets in Space, Discrete Comput. Geom. 25 pp. 379–404 , 2003.

[170] N. Amenta, S. Choi, T.K. Dey, N. Leekha, A simple algorithm for homeomorphic surface reconstruction, in: Proc. Sixt. Annu. Symp. Comput. Geom. - SCG '00, ACM Press, New York, New York, USA, 2000: pp. 213–222.

[171] N. Amenta, M. Bern, The Crust and the β -Skeleton : Combinatorial Curve Reconstruction, Graph. Model Image Process. 60 (2) pp. 125–135 , 1998.

[172] F. Cazals, J. Giesen, *Delaunay Triangulation Based Surface Reconstruction : Ideas and Algorithms*, Technical Report, 2004.

[173] R. Chaine, A geometric convection approach of 3-D reconstruction, in: Eurographics Symp. Geom. Process., Eurographics Association, 2003: pp. 218–229.

[174] T.K. Dey, J. Giesen, J. Hudson, Delaunay based shape reconstruction from large data, in: Proc. IEEE 2001 Symp. Parallel Large-Data Vis. Graph., IEEE, 2001: pp. 19–146.

[175] H. Zhao, S. Osher, R. Fedkiw, Fast Surface Reconstruction Using the Level Set Method, in: Proc. IEEE Work. Var. Lev. Set Methods Comput. Vis., Vancouver, 2001: pp. 194–201.

[176] G. Moore, The evolution of the concept of homeomorphism, Hist. Math. 34 pp. 333–343 , 2007.

[177] R. Dyer, H. Zhang, M. Torsten, *A survey of Delaunay structures for surface representation*, Technical Report, 2009.

[178] T.K. Dey, W. Zhao, Approximate Medial Axis as a Voronoi Subcomplex, Comput. Aided Des. 36 (2) pp. 195–202 , 2004.

[179] N. Amenta, M. Bern, M. Kamvysselis, A New Voronoi-Based Surface Reconstruction Algorithm, in: Proc. 25th Annu. Conf. Comput. Graph. Interact. Tech., ACM, 1998: pp. 415–421.

[180] T.K. Dey, Book: *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*, Cambridge University Press, 2007.

[181] L. Scharf, *Computing the Hausdorff distance between sets of curves*, Freie Universität Berlin, Ph.D. thesis, 2003.

[182] F. Razafindrazaka, *Delaunay Triangulation Algorithm and Application to Terrain Generation*, United Nations University, Ph.D. thesis, 2009.

[183] Computational Geometry Algorithms Library (CGAL), available: www.cgal.org.

[184] H. Edelsbrunner, Triangulations and meshes in computational geometry, Acta Numer. pp. 1–81 , 2000.

[185] H. Edelsbrunner, E.P. Mücke, Three-dimensional alpha shapes, ACM Trans. Graph. 13 (1) pp. 43–72 , 1994.

[186] F. Hurtado, M. Noy, J. Urrutia, Flipping edges in triangulations, Discrete Comput. Geom. 22 pp. 333–346 , 1999.

[187] J. Boissonnat, M. Yvinec, Book: *Geometrie Algorithmique*, 1995.

[188] N. Salman, *From 3D point clouds to feature preserving meshes*, Nice-Sophia Antipolis, Ph.D. thesis, 2011.

[189] S. Funke, E.A. Ramos, Smooth-Surface Reconstruction in Near-Linear Time, in: SODA ’02 Proc. Thirteen. Annu. ACM-SIAM Symp. Discret. Algorithms, 2000: pp. 781–790.

[190] D. Attali, J.-D. Boissonnat, A. Lieutier, Complexity of the Delaunay Triangulation of Points on Surfaces : the Smooth Case, in: Proc. Ninet. Annu. Symp. Comput. Geom., 2003: pp. 201–210.

[191] T.K. Dey, S. Funke, E.A. Ramos, Surface Reconstruction in Almost Linear Time under Locally Uniform Sampling, in: EuroGraphics Work. Comput. Geom., 2001: pp. 129–132.

[192] D. Freedman, An incremental algorithm for reconstruction of surfaces of arbitrary codimension, Comput. Geom. 36 (2) pp. 106–116 , 2007.

[193] J.-D. Boissonnat, Geometric structures for three-dimensional shape representation, ACM Trans. Graph. 3 (4) pp. 266–286 , 1984.

[194] F. Cazals, J. Giesen, M. Pauly, A. Zomorodian, Conformal alpha shapes, in: Proc. Second Eurographics / IEEE VGTC Conf. Point-Based Graph., 2005: pp. 55–61.

[195] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, The Ball Pivoting Algorithm for Surface Reconstruction, Vis. Comput. Graph. IEEE Trans. 5 (4) pp. 349–359 , 1999.

[196] Meshlab, available: http://meshlab.sourceforge.net/2014.

[197] N. Amenta, M. Bern, Surface Reconstruction by Voronoi Filtering, Discrete Comput. Geom. 22 (4) pp. 481–504 , 1999.

[198] N. Amenta, S. Choi, R. Kolluri, The power crust, unions of balls and the medial axis transform, Comput. Geom. 19 (2-3) pp. 127–153 , 2001.

[199] B. Mederos, N. Amenta, L. Velho, L.H. De Figueiredo, Surface Reconstruction from Noisy Point Clouds, in: Eurographics Symp. Geom. Process., 2005: pp. 53–62.

[200] T.K. Dey, S. Goswami, Tight Cocone : A Water-tight Surface Reconstructor, in: Proc. Eighth ACM Symp. Solid Model. Appl., ACM, Seattle, 2003: pp. 127–134.

[201] T.K. Dey, R. Dyer, L. Wang, Localized Cocone surface reconstruction, Comput. Graph. 35 (3) pp. 483–491 , 2011.

[202] T.K. Dey, S. Goswami, Provable surface reconstruction from noisy samples, Comput. Geom. 35 (1-2) pp. 124–141 , 2006.

[203] D. Meagher, Book: *Octree Encoding: a New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*, 1980.

[204] J.-D. Boissonnat, F. Cazals, Smooth surface reconstruction via natural neighbour interpolation of distance functions, Comput. Geom. 22 (1-3) pp. 185–203 , 2002.

[205] R. Sibson, A vector identity for the Dirichlet tesselation, in: Math. Proc. Cambridge Philos. Soc., 1979: pp. 151–155.

[206] R. Allègre, R. Chaine, S. Akkouche, A flexible framework for surface reconstruction from large point sets, Comput. Graph. 31 (2) pp. 190–204 , 2007.

[207] E.A. Ramos, B. Sadri, Geometric and Topological Guarantees for the Wrap Reconstruction Algorithm, in: Proc. 18th Annu. ACM-SIAM Symp. Discret. Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, 2007: pp. 1086–1095.

[208] J. Giesen, M. John, The flow complex: A Data Structure for Geometric Modeling, Comput. Geom. 39 (3) pp. 178–190 , 2008.

[209] J. Boissonnat, A. Ghosh, *Manifold Reconstruction using Tangential Delaunay Complexes*, Technical Report, 2009.

[210] R. Edgeworth, R.G. Wilhelm, Adaptive sampling for coordinate metrology, Precis. Eng. 23 (3) pp. 144–154 , 1999.

[211] S.M. Obeidat, S. Raman, An intelligent sampling method for inspecting free-form surfaces, Int. J. Adv. Manuf. Technol. 40 (11-12) pp. 1125–1136 , 2008.

[212] J. Wang, X. Jiang, L. Blunt, Simulation of Adaptive Sampling in Profile Measurement for Structured Surfaces, in: Proc. Comput. Eng. Annu. Res., Huddersfield, 2010: pp. 48–53.

[213] A. Weckenmann, H. Eitzert, M. Garmer, H. Webert, Functionality-oriented evaluation and sampling strategy in coordinate metrology, Precis. Eng. 17 (4) pp. 244–252 , 1995.

[214] G. Lee, J. Mou, Y. Shen, Sampling strategy design for dimensional measurement of geometric features using Coordinate Measuring Machine, Int. J. Mach. Tools Manuf. 37 (7) pp. 917–934 , 1997.

[215] T.K. Dey, http://www.cse.ohio-state.edu/~tamaldey/.

[216] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, T.R. Evans, Reconstruction and Representation of 3D Objects with Radial Basis Functions, in: Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech., ACM, 2001: pp. 67–76.

[217] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H.-P. Seidel, Multi-level partition of unity implicits, ACM Trans. Graph. 22 (3) pp. 463–470 , 2003.

[218] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson Surface Reconstruction, in: K. Polthier, A. Sheffer (Eds.), Proc. Fourth Eurographics Symp. Geom. Process., 2006.

[219] P. Mullen, F. De Goes, M. Desbrun, D. Cohen-Steiner, P. Alliez, Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets, Comput. Graph. Forum. 29 (5) pp. 1733–1741 , 2010.

[220] J. Bloomenthal, B. Wyvill, Book: *Introduction to Implicit Surfaces*, Morgan Kauffman Publishers, San Fransisco, 1997.

[221] S. Liu, C. Wang, Quasi-interpolation for surface reconstruction from scattered data with radial basis function, Comput. Aided Geom. Des. 29 (7) pp. 435–447 , 2012.

[222] G. Wyvill, C. McPheeters, B. Wyvill, Data structure for soft objects, pp. 227–234 , 1986.

[223] P. Ning, J. Bloomenthal, An evaluation of implicit surface tilers, IEEE Comput. Graph. Appl. 13 (6) pp. 33–41 , 1993.

[224] D. Dobkin, S. Levy, W. Thurston, A. Wilks, Contour tracing by piecewise linear approximations, ACM Trans. Graph. 9 (4) pp. 389–423 , 1990.

[225] J. Sethian, A fast marching level set method for monotonically advancing fronts., Proc. Natl. Acad. Sci. 93 (4) pp. 1591–1595 , 1996.

[226] J. Sethian, Book: *Fast marching methods and level set methods for propagating interfaces*, 1999.

[227] M. Bolitho, M. Kazhdan, R. Burns, H. Hoppe, Parallel Poisson Surface Reconstruction, in: Adv. Vis. Comput., Springer, Las Vegas, 2009: pp. 678–689.

[228] M. Kass, A. Witkin, T. Demetri, Snakes : Active Contour Models, Int. J. Comput. Vis. 331 pp. 321–331 , 1988.

[229] A. Blake, M. Isard, Book: *Active Contours*, Springer, London, 2000.

[230] H. Pottmann, S. Leopoldseder, M. Hofer, Approximation with active B-spline curves and surfaces, 10th Pacific Conf. Comput. Graph. Appl. pp. 8–25 , 2002.

[231] J. Hoschek, F. Schneider, F. Wassum, Optimal approximate conversion of spline surfaces, Comput. Aided Geom. Des. 6 pp. 293–306 , 1989.

[232] Y. Kineri, M. Wang, H. Lin, T. Maekawa, B-Spline Surface Fitting By Iterative Geometric Interpolation/Approximation Algorithms, Comput. Des. 44 (7) pp. 697–708 , 2012.

[233] W. Ma, J.-P. Kruth, Parameterization of randomly measured points for leastsquares fitting of b-spline curves and surfaces, Comput. Aided Des. 27 (9) pp. 663–675 , 1995.

[234] H. Park, Choosing nodes and knots in closed B-spline curve interpolation to point data, Comput. Des. 33 (13) pp. 967–974 , 2001.

[235] W. Wang, H. Pottmann, Y. Liu, *Fitting B-Spline Curves to Point Clouds by Squared Distance Minimization*, Technical Report, 2004.

[236] P. Azariadis, Parameterization of clouds of unorganized points using dynamic base surfaces, Comput. Des. 36 (7) pp. 607–623 , 2004.

[237] H.D. Minh, A.B. Forbes, New Method for free-form surface fitting in precision metrology, in: XX IMEKO World Congr. Metrol. Green Growth, Busan, 2012.

[238] H. Yang, W. Wang, J. Sun, Control point adjustment for B-spline curve approximation, Comput. Des. 36 (7) pp. 639–652 , 2004.

[239] E. Cohen, T. Lyche, R. Riesenfeld, Discrete BSplines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics, Comput. Graph. Image Process. 14 pp. 87–111 , 1980.

[240] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM. 18 (9) pp. 509–517 , 1975.

[241] G. Farin, J. Hoschek, M.S. Kim, Book: *Handbook of Computer Aided Geometric Design*, 2002.

[242] K.-R. Koch, Uncertainty of NURBS Surface fit by Monte Carlo simulations, J. Appl. Geod. 3 pp. 239–247 , 2009.

List of figures

List of tables

Appendix

Résumé substantiel en français

Introduction générale

Les surfaces complexes sont des surfaces qui manifestent des variation de forme et de géométrie et qui posent des défis quant à leur fabrication et leur mesure. Les surfaces complexes sont normalement appelées surfaces gauches et celles-ci sont classifiées selon leur taille, la complexité de leur forme ainsi que la tolérance associée à leur spécification. Les surfaces optiques et aubes de turbines sont des cas de surfaces complexes et font l'objet de ce travail de thèse.

L'optique a connu un essor remarquable à travers les décennies induisant ainsi des performances inégalées dans innombrables domaines tels que la métrologie, la photonique, l'énergie, la médecine, l'ophtalmologie… [1]. Les progrès sont notamment dus à l'avancement des technologies de fabrication et de mesure. De nouvelles machines à mesurer de très haute précision sont désormais capables d'atteindre des niveaux d'incertitudes nanométriques pour la mesure de surfaces optiques asphériques. Les surfaces asphériques sont des surfaces complexes car elles sont composées de plusieurs formes, i.e., une forme conique et une forme polynomiale. Les surfaces gauches ou freeform ont elles aussi connu d'énormes progrès et se trouvent concernées par des applications diverses comme dans l'automobile, l'aéronautique, l'énergie (éoliennes), etc… L'utilisation de pièces freeform notamment en aéronautique et dans l'automobile a permis de réduire considérablement la consommation de carburants et les effets nuisibles à l'environnement. Les surfaces freeform sont fabriquées et mesurées avec une tolérance sub-micrométrique et donc à l'aide de machines à mesurer tridimensionnelles (MMT/CMM).

Dû à la différence concernant la tolérance recherchée nous allons traiter les surfaces asphériques et les aubes de turbines de deux manières distinctes. Le sujet de thèse est motivé par projet Européen portant sur les pièces asphériques et un projet industriel portant sur les aubes de turbines. Afin de promouvoir l'expertise Européenne en tout ce qui concerne les technologies optiques de pointe, l'European Metrology Research Programme (EMRP) a lancé le projet IND10: FORM intitulé "*Optical and tactile metrology for absolute form characterization*", et ce, dans le but de développer des méthodes pour mesurer et caractériser

des surfaces complexes à un niveau d'incertitude nanométrique. Le projet est coordonné par la laboratoire national de métrologie Allemand, le PTB, et regroupe différents laboratoires nationaux de métrologie Européens ainsi que des partenaires académiques et industriels. Aujourd'hui, il n'existe pas de méthodes universelle pouvant atteindre un niveau nanométrique lorsqu'il s'agit de la mesure et de la caractérisation de surfaces complexes telles que les surfaces asphériques et les surfaces gauches aussi appelées freeform. Le Laboratoire National de métrologie et d'Essais Français (LNE), en collaboration avec l'école d'Arts et Métiers ParisTech (ENSAM), l'école Normale Supérieure de Cachan (ENS) et l'entreprise Geomnia, est le leader d'un workpackage centré sur le traitement de données et le développement d'algorithmes pour la métrologie de forme de surfaces asphériques.

Geomnia, une PME avec des compétences en métrologie 3D, participe à ce projet pour son savoir-faire en ingénierie des systèmes métrologiques et son expertise en apport de solutions logicielles à des problèmes de métrologie et parce que l'une de ses problématiques actuelles concerne la qualification dimensionnelle d'aubes de turbines à gaz.

Dans les deux cas précédents, la caractérisation de surfaces complexes requiert du traitement de données et par conséquent la reconstruction de surfaces. La reconstruction de surfaces est une filière de recherche abondamment étudiée et inculquée dans différentes disciplines scientifiques telles que l'Informatique Graphique, l'Ingénierie Inverse (Reverse Engineering) et la Métrologie. Chacune de ces disciplines a ses propres méthodes et outils pour résoudre le problème inverse. En général, le processus de reconstruction part d'un ensemble de données cartésiennes, et vise à reconstituer la forme de la surface sous-jacente aux points. La communauté d'informatique graphique, par exemple, se suffit d'une reconstruction du premier ordre, c'est-à-dire, linéaire par morceaux, en construisant un maillage triangulaire. La communauté de l'ingénierie inverse effectue la régression de modèles Splines bidirectionnelles (B-Splines et/ou NURBS) comme une approximation de la surface sous-jacente aux points. En métrologie, une problématique d'association n'existe pas en l'absence d'une connaissance bien précise du modèle mathématique/analytique de la surface mesurée.

Avec l'avancée technologique des instruments et des systèmes de mesure, certains instruments optiques sont capables de générer de grands flots de données contenant plus d'un million de points, et ce, pour un temps de mesure très court. Le nuage de points qui en résulte est un ensemble de points non-organisés. La connectivité entre les points n'étant pas inhérente

au nuage de points, ce dernier ne communique pas d'informations suffisantes sur la géométrie et la topologie de la surface sous-jacente. D'autant plus, le nuage de points est bruité, peut contenir des points aberrants (outliers) et peut présenter des régions de recouvrement dans la cas où plusieurs scans séparés ont été réalisés et ensuite fusionnés pour constituer le nuage de points de la surface complète. Dans le but d'augmenter toujours en efficacité et en rendement, il est toujours inéluctable de développer des solutions automatisées et rapides d'où le l'objectif de ce travail qui est de développer un algorithme robuste et automatique pour la reconstruction de surfaces complexes avec un temps ne dépassant pas le temps de mesure.

Afin d'atteindre cet objectif, nous définissons brièvement le type de données que l'on va traiter dans ce travail et nous exposons les capacités liées à la métrologie actuelle et décrivons notre approche pour résoudre le problème auquel nous faisons affaire. L'entrée à notre problème est un nuage de points à très grand nombre de points (quelques millions). Un nuage de points est un ensemble de données non-organisées, bruitées, non-orientées et contenant des points aberrants. Idéalement, une mesure conduira à un ensemble de données qui se trouvent exactement sur la surface. Or en pratique, ceci est impossible et on observe toujours la présence de points parasites. Un bruit peut être interne, provenant de l'appareillage électronique, ou externe, lié à l'espace environnant. Tandis qu'un point aberrant (ou outlier) est un point qui est situé relativement loin du reste des points. Sa présence peut être due à une erreur de mesure aberrante (dépôt de poussières sur la surface) ou à une mesure erronée due à la présence d'un obstacle qui bloquerait la zone de mesure désignée.
De plus, la complexité des surfaces induit que le temps de mesure et de traitement est long. Cependant, nous cherchons un algorithme dont le temps de traitement serait au moins équivalent au temps de mesure. Par ailleurs, il faudra aussi pouvoir évaluer et valider l'algorithme dans le sens de ces fonctionnalités.

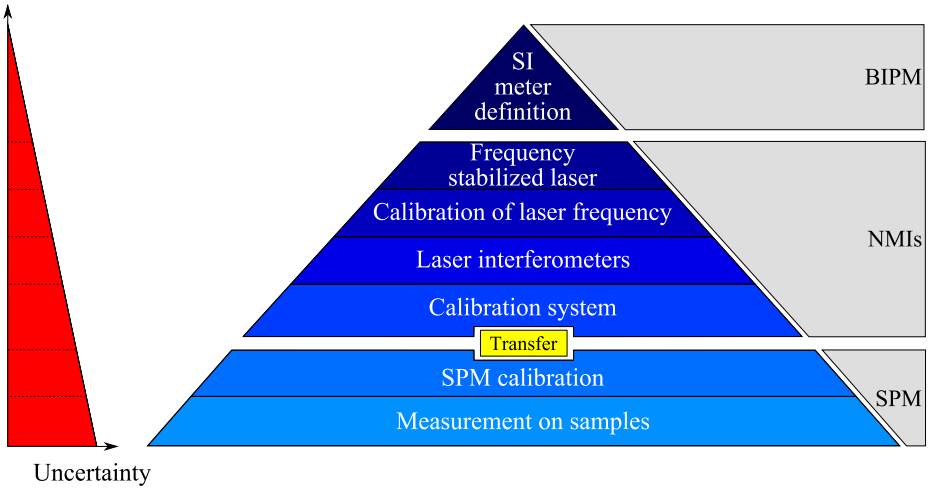
Comment obtenir un algorithme de reconstruction de surfaces à partir de grands flots de données non-organisées qui soit automatique, robuste aux données d'entrée et rapide par rapport au temps de mesure ?

Ce travail de thèse contribue à cette problématique et tente d'en proposer des solutions. Dans ce travail, nous traitons de mesures à contact, notamment les mesures avec stylet, et de mesures sans contact, notamment les mesures optiques confocales, interférométriques et à triangulation laser. Chaque système de mesure possède ses propres spécificités. Celles-ci

peuvent être, par exemple, la capacité de générer des mesures en lignes de scans structurées, en Z-buffer, en spirales, ou encore des mesures complètement aléatoires non-organisées. Ainsi, toute connaissance par rapport au système de mesure et le mesurande a une influence sur le processus de reconstruction et sur le choix de l'algorithme de reconstruction.

La métrologie révèle les aspects de forme et les grandeurs géométriques de l'objet mesuré et par conséquent permet de vérifier la conformité de ce dernier vis-à-vis des spécifications définies lors de sa conception. Les techniques de mesure traditionnelles se font sur MMT et sont encore très utilisées aujourd'hui puisqu'elles sont fidèles et traçables. Néanmoins, ceci n'est vrai que pour des formes simples et régulières et pour lesquelles il existe des étalons (sphères, plans, cylindres …). Ces étalons sont fabriqués avec une très bonne qualité de finition et étalonnés par le biais d'instruments primaires avant d'être utilisés comme étalons de référence pour l'étalonnage de MMT. Lorsqu'une machine est étalonnée à l'aide d'un étalon de référence, toute mesure sur le même type de forme que celle de l'étalon devient désormais traçable selon la définition du 'mètre' communiquée par le Bureau International des Poids et Mesures (BIPM). Le BIPM est un organisme qui a pour rôle de garantir l'uniformité universelle des mesures et leur traçabilité selon le Système International des unités (SI) [2]. En revanche, lorsque les surfaces complexes ou freeform sont traitées, le Guide pour l'expression de l'incertitude de mesure (GUM) n'est plus applicable dû à l'absence d'étalons de référence [3], [4]. Une sphère par exemple possède un paramètre de grandeur qui est son diamètre, ce qui peut être facilement mesurable par une machine dédiée à la mesure de diamètres. Etant donné qu'une surface freeform n'a pas de tels paramètres naturellement identifiables, il est très difficile d'en fabriquer des étalons de référence universels [5], malgré le fait que quelques uns commencent à être proposés [6].

Lorsque une très haute précision est requise, les MMT traditionnelles ne sont pas suffisantes et sont remplacées par des MMT d'extrême précision que l'on va abréger par UHPMM. L'atout majeur des UHPMM est qu'elles ne nécessitent plus d'étalonnage par des étalons car la mesure est directement contrôlée par des instruments primaires tels que des interféromètres laser.



Pyramide de traçabilité métrologique (SPM: Mesure par capteurs scan).

Les machines à mesurer (MMT et UHPMM) engendrent un processus de mesure en deux temps. La mesure consiste premièrement à recaler la pièce à mesurer dans le repère de la machine. Deuxièmement, la mesure consiste à scanner la pièce une fois recalée et là il existe plusieurs techniques telles que la mesure point-à-point ou en lignes de scans, pas toujours représentatives de la géométrie 3D de l'objet mesuré. Dans ces cas de mesure, la fréquence d'acquisition ne dépasse pas les 50 à 60 points par minute [5]. La métrologie d'aujourd'hui implique des instruments plus avancés qui sont indépendants de l'étape de recalage des repères permettant ainsi l'acquisition de données à des fréquences très élevées atteignant les 20000 points/s. Ceux-ci sont des instruments sans contact, en majorité optiques, générant plusieurs centaines de milliers de points en une durée de temps réduite. En particulier, les scanners 3D montrent une bonne efficacité quant à la mesure de surfaces complexes. Cependant, et surtout avec l'émergence continue de nouvelles technologies, les scanners 3D sont difficilement étalonnés et calibrés, leur mesure manquant de traçabilité. De plus, les nuages de points qui en résultent sont non-organisés et grands en nombre de points.

Deux types de surfaces complexes servent de support à ce travail de recherche:

I. Une surface asphérique AO775 fabriquée par Anteryon® :

- Une mesure réalisée au sein du LNE à l'aide d'un profilomètre de haute-précision et d'un capteur tactile. La mesure tactile contient 1500000 points.
- Une autre mesure sur le profilomètre du LNE à l'aide d'un capteur optique confocal [7]. La mesure confocale contient 1000000 de points. Le but de cette deuxième mesure est de comparer la mesure tactile à la mesure confocale.

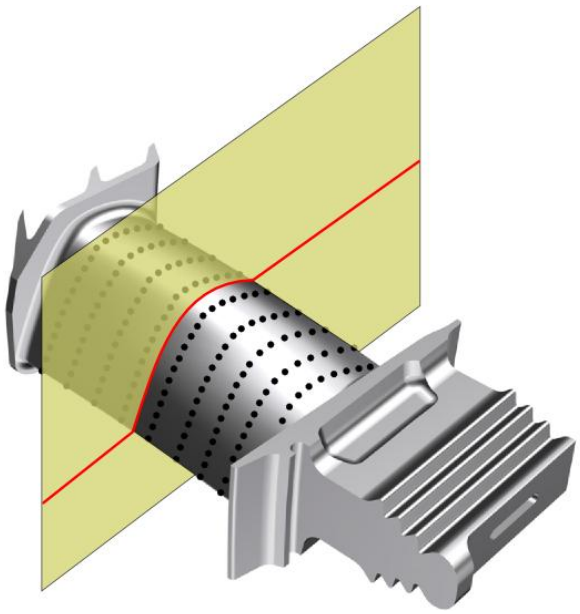
- Des mesures différentes par des partenaires du projet IND10: FORM pour l'évaluation des capacités de mesure de chacun des laboratoires concernés.

La difficulté concernant la reconstruction de surfaces asphériques découle du fait que leur modèle est constitué d'un bon nombre de paramètres. En rajoutant le fait qu'il y ait un très grand flot de données, rend les algorithmes d'optimisation longs.

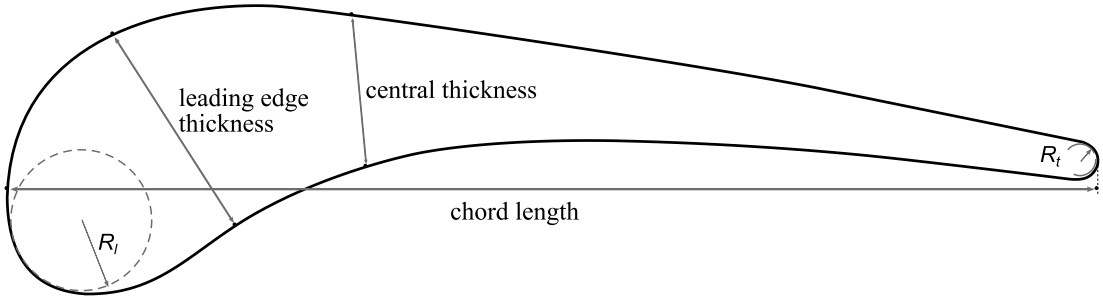
II. Une aube de turbine mesurée par triangulation laser. Pour cette mesure, quatre scans, résultant de quatre posages différents de la pièce, sont fusionnés et produisent un nuage de points à 127000 points distribués en lignes de scans.

Les difficultés liées à la reconstruction dans ce cas sont l'abondance du bruit de mesure, les régions de recouvrement dues à la fusion des données et la disparité de la fréquence spatiale des points dans les différentes directions de mesure.

Les surfaces freeform n'ont normalement pas de modèles mathématiques associés. En effet, elles sont souvent définies par des modèles CAO. Ce cas n'étant pas toujours vérifié, nous nous intéressons au cas où aucun modèle n'est fournis. C'est ainsi que, dans le cas des aubes de turbines où la caractérisation d'un profil à une hauteur donnée est requise, une représentation continue de la surface est primordiale afin de pouvoir extraire un profil à toute hauteur désirée (Fig. 5).



Caractérisation des aubes de turbine par section.



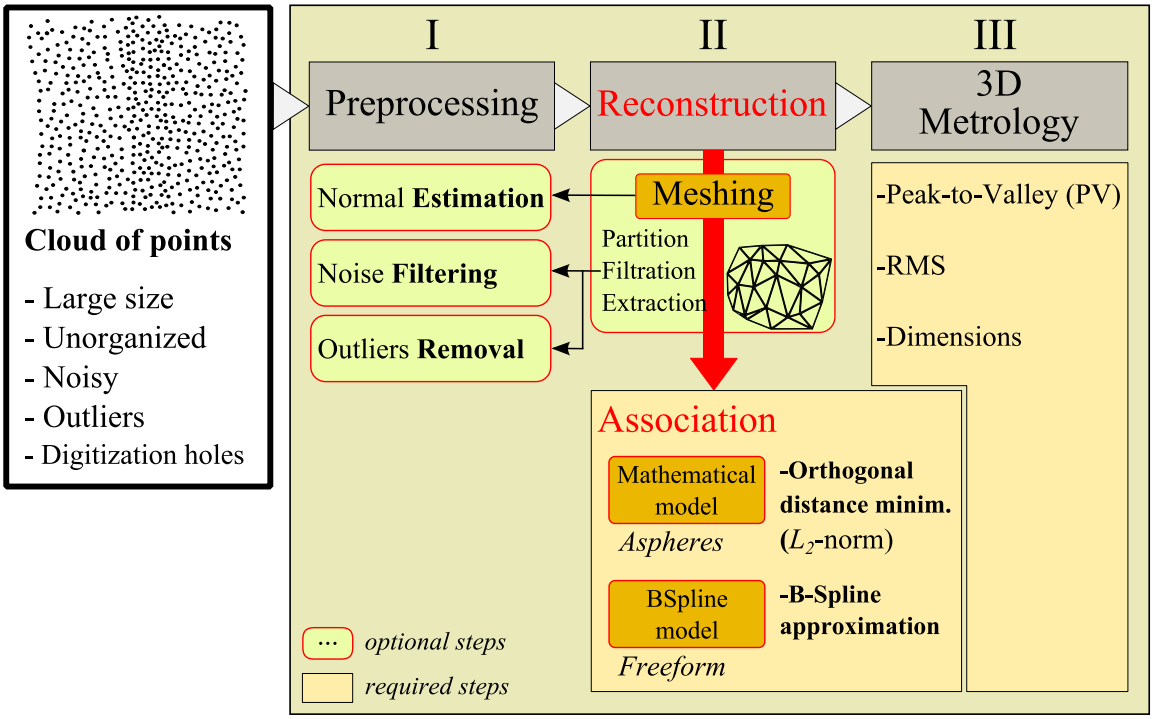
Des dimensions recherchées sur un profil d'aube de turbine.

Quelque soit la technique de mesure utilisée, le processus de caractérisation d'une surface représentée par des points induit la nécessité d'en reconstruire une approximation de celle-ci. Les raisons les plus simples sont que, par exemple, pour un nuage de points 3D non-organisés, la détermination du voisinage est compliquée. Les techniques proposées par la communauté de l'informatique graphique dénouent ce problème en créant un maillage des points. Le maillage est une structure de données à plusieurs utilités. D'un premier côté, le maillage permet de plus simplement partitionner l'espace des points permettant ainsi l'association de patchs paramétriques lisses de type Splines à chacune des partitions [9]–[15]. Ces algorithmes nécessitent de rendre les maillages triangulaires quadrangulaires et donc requièrent un temps de calcul considérable. Dans d'autres approches au problème, des travaux ont permis de montrer qu'il est aussi possible d'associer des surfaces paramétriques, directement sur des nuages de points sans passer par l'interpolation de premier ordre. Ces techniques font l'objet d'une bibliographie suivie d'une contribution par une méthode de déformation de contour actif de modèle B-Spline.

L'association, ou autrement dit, le fitting d'une surface à des points doit permettre de procéder à une analyse des déviations, d'estimer les défauts de forme et d'inspecter les surfaces complexes, qu'elles soient asphériques ou freeform et qu'elles aient un modèle mathématique connu à priori ou pas. La métrologie des surfaces asphériques se fait par le biais de l'association du modèle mathématique aux points en estimant les paramètres de transformations (rotation et translation) et éventuellement les paramètres du modèle. Suivant la norme d'association utilisée, la caractérisation de forme n'est pas la même. Selon les standards ISO, la métrologie de forme s'effectue en utilisant la norme L_∞ consistant à calculer une enveloppe minimale contenant les points. Par contre, cette méthode augmente en complexité plus le nombre de points en entrée est important surtout que les algorithmes qui existent sont instables et non-déterministes [16], [17]. Cela dit, le problème d'association peut

être résolu en utilisant la norme L_2. Malgré le fait que cette norme n'est pas appropriée pour faire de la métrologie de forme, les travaux actuels continuent à en faire service. Le défaut de forme est surestimé avec la norme L_2 mais l'erreur crête-à-crête ou Peak-to-Valley (PV) est un indicateur encore très utilisé et accepté [18].

L'approche globale à la caractérisation de surfaces complexes suit le schéma de la figure suivante.



Procédure de traitement des données mesurées d'une surface complexe.

La phase de prétraitement est importante dans le sens où elle permet d'éliminer les points aberrants et/ou les bruits de mesures. Normalement, cette étape est intégrée en tant qu'étape après-mesure dans le post-processeur de la machine à mesurer et tel est le cas de la machine du LNE. Eventuellement, cette étape de prétraitement peut être utile pour estimer l'orientation normale des points par une technique d'Analyse par Composantes Principales (PCA) ou par des approches utilisant le diagramme de Voronoï [19]–[21] mais cette phase du processus n'est pas notre point d'intérêt dans ce travail. L'étape suivante qui consiste à la reconstruction et l'association constitue notre centre d'intérêt. Nous étudions la robustesse et l'efficacité des techniques de reconstruction par maillage triangulaire avec une analyse et une comparaison de plusieurs méthodes existantes et nous définissons les éléments clés afin de valider l'une ou

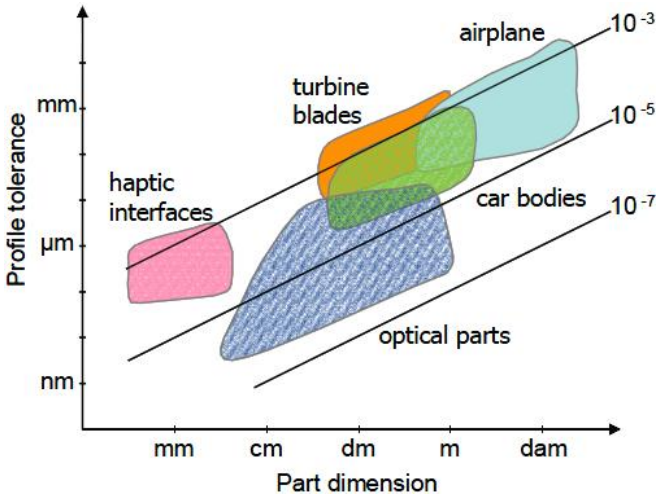
l'autre des méthodes en ce qui concerne la caractérisation des surfaces complexes. Le maillage est une étape que nous incluons dans l'étude mais il faut savoir qu'elle peut n'est pas indispensable pour la suite du traitement. Nous procédons ensuite à l'association de modèles analytiques lorsque ceux-ci existent ou de modèles B-Splines dans le cas contraire. Les données sont soit les points résultant du prétraitement soit les points ordonnés constituant les sommets du maillage si la structuration des données s'est avérée indispensable. La dernière étape consiste à analyser les écarts qui résultent de l'association et à qualifier ainsi la forme et/ou les grandeurs dimensionnelles requises de la surface mesurée.

Le manuscrit est divisé en trois chapitres. Le chapitre 1 présente une vue globale sur la conception, la fabrication et les techniques de mesures de surfaces complexes et fournit un état de l'art sur les machines à mesurer de très haute précision ainsi que sur les technologies de capteurs. Ce chapitre explique aussi comment les surfaces complexes citées ci-dessus sont mesurées. Dans le chapitre 2, nous mettons le point sur les techniques d'optimisation par le biais d'une étude bibliographie suivie d'une comparaison et nous proposons un algorithme de régression robuste pour l'association de modèles implicites à des points issus d'une mesure sur une pièce asphérique. L'algorithme proposé est validé sur des données simulées avant d'être appliqué à des données réelles. Dans le chapitre 3 nous traitons des surfaces gauches (freeform) qui ne sont définies ni par des modèles mathématiques ni par des modèles CAO. Nous commençons par une étude bibliographie sur les techniques d'association/fitting de modèles B-Splines tout en énumérant leurs limites et difficultés et proposons ensuite une méthode qui tient compte des difficultés et qui s'avère être une forte contribution.

Chapitre 1 - Conception, fabrication et mesure des surfaces asphériques et freeform

Conception et spécification des surfaces asphériques et freeform

Les surfaces complexes peuvent être classées tel que le montre la Fig. 18. Nous nous intéressons particulièrement aux classes de surfaces optiques et aux aubes de turbine. Ce graphe montre clairement qu'entre ces deux classes, la tolérance recherchée se trouve à différentes échelles. Les surfaces optiques requièrent une tolérance nanométrique alors que les aubes de turbines sont au niveau micrométrique (ou aujourd'hui sub-micrométrique).

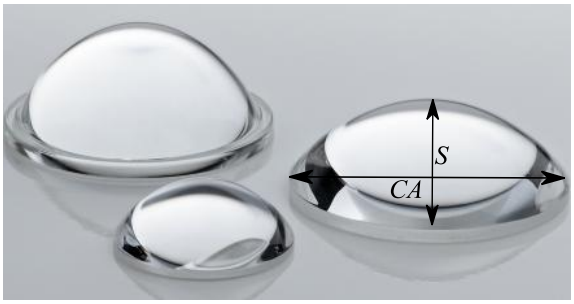


Tolérance en fonction de la dimension des surfaces complexes et freeform [22].

La fabrication et la mesure sont les deux étapes-clé qui suivent la conception sur la chaîne du cycle de vie d'un produit avant que celui-ci ne soit transmis au consommateur. A la toute première étape de conception, un concepteur peut aller jusqu'à définir des spécifications (tolérances) très fines, cependant, si le produit ne peut être fabriqué précisément ou si son inspection ne peut être réalisée avec les moyens de mesures existants, ce produit n'a aucune raison d'exister car ses fonctionnalités ne sont pas garanties [23]. L'avance technologique a permis de surmonter les difficultés liées à la mesure et à la fabrication de haute précision [24], [25], [26]. Non seulement des processus de fabrication avancés tels que la rectification et le polissage ont réduit les défauts de fabrication, mais aussi les technologies de mesures ont évolué vers une échelle micrométrique voire nanométrique que se soit en termes d'incertitude ou de répétabilité [27], [28], [29], [30], [31], [32]. Cela est vrai pour les formes simples avec des débuts de progrès pour les formes complexes. Ceci dit, il devient promettant d'en vouloir

demander davantage aux concepteurs et aller vers des produits aux formes plus complexes avec l'assurance d'évoluer en performance. Des surfaces asphériques aux aubes de turbines, nous décrivons comment ces surfaces complexes sont spécifiées, nous listons rapidement les procédés de fabrication et nous nous concentrons sur la métrologie dédiée à ces types de surfaces. Actuellement, il est devenu possible d'atteindre un niveau nanométrique dans la fabrication de surfaces asphériques et leur mesure [5], [33], [34].

Surfaces asphériques



Lentilles asphériques: illustrant les grandeurs physiques telles que l'ouverture ou le diamètre effectif (*CA*) et la hauteur (*S*).

Les surfaces asphériques (Fig. 20) sont formulées et définies dans l'ISO 10110-Part 12 [35]. En ce qui concerne les surfaces axisymétriques, elles sont de la forme suivante (1):

$$z = f(r) = -\left(\frac{cr^2}{1 + \sqrt{1 - (1+\kappa)c^2 r^2}} + \sum_{j=2}^{m/2} a_{2j} r^{2j} \right). \qquad (70)$$

où $r = \sqrt{x^2 + y^2}$ et z sont les coordonnées Cartésiennes; c est la courbure au sommet, κ est la conicité et $\boldsymbol{a} = (a_4, a_6, \dots, a_m)$ est le vecteur des paramètres de déviation asphérique d'ordre m. Pour tout couple $(r, z) \in \mathbb{R}^2$, $F(r, z; c, \kappa, \boldsymbol{a})$ est dérivable deux fois.

Dans notre cas nous traitons d'une surface asphérique (modèle AO775) d'ordre 10 qui n'est pas exactement définie comme dans l'ISO [35] en question. Les paramètres ont pour valeurs respectives: $c = 10^{-20}$ mm^{-1}, la conicité $\kappa = -1$ et les paramètres asphériques $a_2 = 0.0223, a_4 = 7.293 \times 10^{-6}, a_6 = 4.52 \times 10^{-9}, a_8 = -1.061 \times 10^{-11}, a_{10} = 9.887 \times 10^{-15}$. Cette lentille a les dimensions $S = 3.217$ mm et $CA = 11.74$ mm.

Il existe deux autres définitions mathématiques plus récentes pour les surfaces asphériques qui ont été proposées par Forbes [36], [37]. Ces nouvelles définitions sont intéressantes car elles sont fondées sur des polynômes à base orthogonale pour lesquels les coefficients, qui ne sont autres que les paramètres asphériques, sont indépendants [38]. Cela implique que chacun des coefficients joue sur une fréquence spatiale de la surface [39].

Aubes de turbines

Contrairement aux formes simples et/ou asphériques, les surfaces gauches n'ont pas de forme particulière et peuvent donc prendre toutes formes. Dans l'absence d'une définition formelle, Campbell et Flynn perçoivent les freeform comme étant des surfaces composées d'un ou plusieurs bouts de surfaces non planes et non quadriques [40]. Les surfaces gauches sont intéressantes dans le sens où elles remplissent un cahier de charge à la fois fonctionnel et esthétique [22], [41], [42]. Pour une turbomachine, une freeform permet d'optimiser les performances et de réduire les pertes d'énergie.

En optique et imagerie, les freeform sont capables d'atteindre des performances encore plus poussées que celles fournies pas des surfaces asphériques. En plus d'éliminer les aberrations optiques, les lentilles freeform augmentent le champs de profondeur et élargissent le champ de vue [43]. Les surfaces freeform sont rarement accompagnée de modèles mathématiques mais sont souvent décrites par des modèles paramétriques. Les modèles les plus utilisés en CAO sont les modèles B-Splines (6).

$$S(u,v) = \sum_i \sum_j N_i^d(u) N_j^l(v) C_{i,j} \,, \tag{71}$$

où $C_{i,j}$ les points de contrôle de la surface paramétrique, $N_i^d(u)$ et $N_j^l(v)$ les fonctions de mélanges d'ordre d et l dans les directions u et v, respectivement.

Fabrication des surfaces asphériques et freeform

Les techniques de fabrication de surface optiques asphériques et gauches sont listées d'une manière exhaustive dans les travaux de Fang *et al* in [43]. Pour les surfaces de révolution composées de matériaux non-ferrés ou céramiques, une méthode de fabrication très courante est le tournage à pointe de diamant unique (ou Single Point Diamond Turning (SPDT)) [44]. Le SPDT peut être utilisé pour des surfaces freeform à condition qu'une

adaptation soit possible telle que la Fast Tool Servo (FTS) [8], [45]. Beaucoup d'autres processus de fabrication existent et nous citons le fraisage rotatif diamanté (fly-cutting), le fraisage à bout sphérique, et le moulage de précision [46], [47]. Des technologies de polissage assistées par ordinateur font l'ample montée en précision et en répétabilité de la fabrication de surfaces complexes. Nous citons ici l'usinage par faisceaux ioniques et/ou plasma [27], [48], le polissage magnétorhéologique asservi [28] et le Precession Process [49].
D'autres techniques asservies permettent aussi d'atteindre des précisions importantes. Henselmans propose une technique qui consiste à réaliser une mesure après chaque étape de fabrication, en boucle fermée, afin d'obtenir la qualité de surface désirée [8].

Les surfaces gauches telles que les aubes de turbines sont généralement fabriquées sur des machines 5-axes à commande numérique [47], [50] ou par des techniques de forge [51]. Les matériaux utilisés pour les aubes de turbine, par exemple, sont le Prepreg ou des matériaux d'infusion ou encore l'acier inoxydable. Pour des performances plus poussées en terme de température, des techniques de fabrication par procédés électrochimiques sont utilisés et les matériaux concernés sont des alliages à base de Titane et de Nickel ou encore de l'aluminure de Titane.

Métrologie des surfaces asphériques et freeform

Dans le but d'améliorer leurs capacités de mesurer des pièces complexes, les laboratoires nationaux de métrologie, les laboratoires de recherche et quelques industries Européens ont développé des machines d'ultra-haute précision [52]. La clé de cette fidélité de mesure élevée et inégalée est dans le fait que ces machines respectent bien le principe d'Abbe, utilisent des matériaux à très faibles coefficients de dilatation thermique, des éléments de guidages robustes et précis et sont équipées d'interféromètres laser qui contrôlent la mesure. La traçabilité de ces machines est établie grâce à ces interféromètres laser eux-mêmes étalonnés en fréquence par comparaison avec une source laser He-Ne I2-stabilisée primaire [53], [54]. Finalement les instruments de mesure utilisés sont soit tactiles (stylet mécanique) soit optiques (confocal chromatique) [55][56][57]. Les machines évoquées sont la machine F25 de Zeiss utilisée par le VSL et le SMD, la µ-CMM du METAS, l'ISARA400 de chez IBSPE, NANOMEFOS de chez TNO, le TWI du PTB et finalement le profilomètre du LNE. Nous allons nous contenter de présenter le profilomètre de haute précision du LNE dans ce travail

résumé sachant que la description détaillée des autres machines se trouve dans l'article par Nouira *et al* [52].

Le profilomètre de haute précision du LNE pour la mesure de pièces asphériques

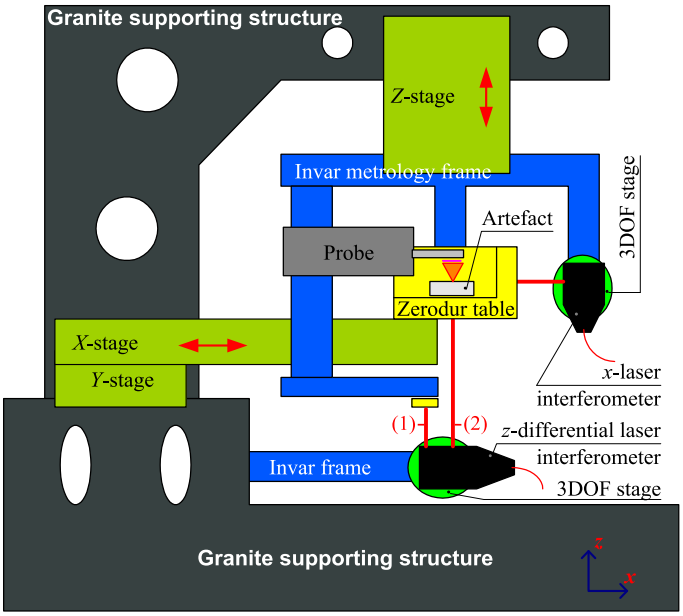
Le profilomètre du LNE a été conçu à la base pour la métrologie d'état de surfaces planes avec des incertitudes de l'ordre du nanomètre. Mais aujourd'hui cette machine est mise en œuvre pour la mesure de surfaces asphériques et ses capacités y sont évaluées. L'architecture de cette machine répond aux exigences du principe d'Abbe et comporte une chaîne métrologique courte et dissociée [58]. La machine admet deux types de capteurs, un capteur point tactile et un capteur point confocal chromatique. La machine est stable et précise grâce aux matériaux utilisés et aux éléments de guidage de très haute précision, complètement indépendants dans chacune des directions x, y et z. Pour 50 mm de course, la rectitude du mouvement des guidages est à moins de 1 nm d'erreur.

Une table en Zerodur qui se déplace dans le plan horizontal et sur laquelle la pièce à mesurer est posée, est contrôlée par deux interféromètres laser (en x et en y) dont l'incertitude de mesure ne dépasse pas le nanomètre. La mesure se fait dans la direction verticale de z et les déplacements dans cette direction sont aussi contrôlés mais par un interféromètre différentiel raccourcissant ainsi la chaîne métrologique et l'isolant des altérations thermiques de la structure porteuse de la machine. Les variations thermiques des éléments constituant la chaîne métrologique sont estimés à quelques centièmes de degrés conduisant ainsi à une dilatation thermique maximale faible de 2 nm pour l'aluminium et 0.5 nm pour le Zerodur.

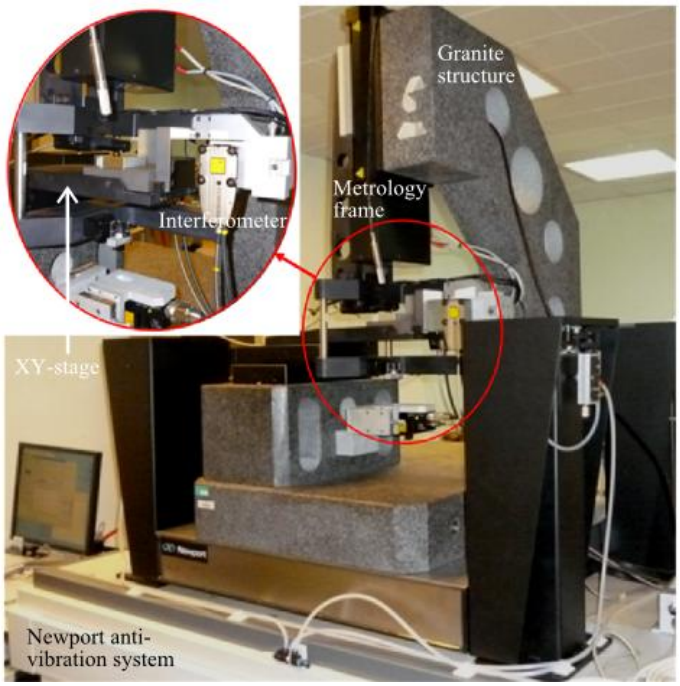
Le profilomètre adopte le principe de la structure métrologique dissociée qui consiste à désunir la structure métrologique de la structure porteuse de la machine [58], [59]. De ce fait, la structure métrologique est reliée à la structure porteuse par des liaisons isostatiques flexibles afin de limiter la transmission des déformations que peut subir cette dernière. Par conséquent, la structure métrologique n'est que sous l'effet de sa propre masse et n'a pour fonction que d'effectuer la mesure.

La machine est installée dans la salle blanche du LNE où les conditions environnementales sont optimales. La température ambiante est réglée à $20^{\pm 0.3}$ °C et l'humidité relative à $50^{\pm 5}$ %RH. Cette petite amplitude de variation de température est lente et conduit à de très faibles impacts sur les éléments de la machine. Les vibrations basses-fréquences venant du sol sont réduites grâce à un système antivibratoire. Le budget d'incertitude a été établit pour cette machine avec un facteur d'élargissement égal à 2 et en

conformité avec le GUM [3]. L'incertitude élargie est estimée à $15 + 10^{-6}\,L$ nm, où L est la distance mesurée avec un capteur tactile. Lorsqu'il s'agit d'un capteur confocal, cette incertitude est à ré-estimer.



(a)



(b)

Le profilomètre de très haute précision du LNE: (a) architecture de l'appareil. (b) photo de l'appareil.

La technique la plus couramment utilisée en métrologie dimensionnelle jusqu'à ce jour demeure la mesure à contact car elle est bien maîtrisée. Le stylet utilisé sur le profilomètre pour la mesure d'une lentille asphérique possède un angle au bout de 90 °, un rayon de la sphère de 2 µm et une force statique de mesure inférieure à 0.7 mN. Trois courses de mesures sont possibles avec ce capteur, le plus petite allant jusqu'à 100 µm, la moyenne allant jusqu'à 500 µm et la plus large allant jusqu'à 1 mm. La mesure à contact connaît une limitation majeure liée au contact entre le capteur et la pièce mesurée. Pour des objets non rigides, le contact peut engendrer des déformations et pour des objets rigides, des égratignures peuvent se manifester. C'est pourquoi les technologies évoluent davantage aujourd'hui dans le sens de la mesure sans contact avec des efforts multipliés pour rendre ces technologies très fiables.

La mesure confocale chromatique est l'une des plus répandues avec entre autres les techniques de triangulation laser et l'interférométrie. Le principe du confocal ne sera pas évoqué ici mais de bonnes références bibliographiques sont à la portée du lecteur [55], [60]. Le capteur confocal chromatique utilisé pour mesurer la pièce asphérique en jeu a été étalonné sur un banc d'étalonnage spécialement dédié à ce but [61].

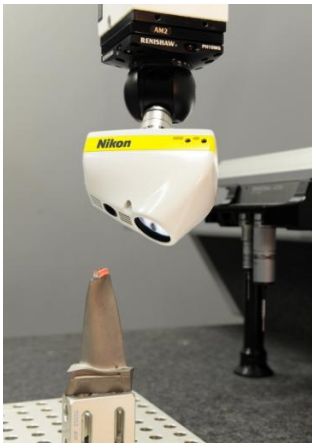
La pièce asphérique AO775 décrite précédemment est posée sur la table de la machine à mesurer du LNE et un recalage manuel de la pièce par rapport aux repères de la machine est effectué. Sachant que sur le profilomètre il n'est pas possible de faire un recalage parfait et d'aligner l'axe de la pièce avec l'axe z de la machine, le point de rebroussement de la surface peut être plus ou moins bien estimé. Ce point là représente le centre autour duquel une mesure symétrique en x et en y est effectuée. Une fois l'alignement accompli, la surface est scannée sous forme de grille XY et les coordonnées Cartésiennes des points sont enregistrées. Pour la mesure tactile, l'aire mesurée est de 6×6 mm² car nous avons restreint la course de mesure à l'intervalle $[0 - 100]$ µm pour une incertitude associée de ± 6.5 nm. Pour la mesure confocale, l'aire mesurée est de 5×5 mm², cette étendue étant limite par rapport à une incertitude associée qui soit faible, ou du moins du même ordre que celle de la mesure tactile. La comparaison et l'effet de mesure avec l'un ou l'autre des capteurs sont détaillés dans les travaux de El-Hayek *et al* [62], [63].

Mesure des aubes de turbine

Les aubes de turbines étant considérées comme fortement freeform, présentent un vrai défi quant à la mesure. Selon Savio *et al* [22], l'étape de recalage de la pièce sur le repère de la machine est compliqué et le choix de l'instrument de mesure est variable. Dans quelques cas,

l'inspection visuelle simple est encore employée alors que dans d'autres cas qui requièrent plus de fidélité (precision en anglais), des moyens de mesure à triangulation laser sont utilisés. Dans notre cas, l'aube de turbine est mesurée à l'aide d'un scanner à triangulation laser monté sur MMT. Le regroupement des points finaux est la fusion de scans issus de 4 posages différents de la pièce. Les scans sont des scans parallèles formant ainsi des lignes de scans tout au long de l'aube.

(a)



(b)

Mesure d'une aube: (a) point-à-point avec capteur à contact (machine METRIS) [64]; (b) triangulation laser sans contact (labo de métrologie de Nikon).

Conclusion

Les surfaces asphériques et freeform ont trouvé leur place sur la marché car elles offrent des performances inégalées. Ayant des géométries et des formes complexes, les surfaces asphériques et freeform doivent cependant être fabriquées mesurées avec très haute précision afin que leurs performances soient perçues. Pour les surfaces asphériques nous cherchons une précision à l'échelle du nanomètre tandis que pour l'aube de turbine nous sommes à l'échelle sub-micrométrique. Les moyens de mesure qui existent sur le marché sont multiples, le stylet étant le plus maîtrisé. Les technologies optiques voient leur essor aussi. Les machines à mesurer dédiées à la mesure de surfaces complexes sont très précises et peuvent atteindre des niveaux d'incertitude et de répétabilité à l'échelle du nanomètre. Concernant la mesure de la pièce asphérique, le LNE a effectué deux mesures, une mesure tactile et une mesure confocal sans contact et les partenaires Européens du projet ont eux aussi effectuer des mesures diverses. De nouvelles technologies de mesures certes très rapides mais qui manquent de l'étape de recalage de repères deviennent de plus en plus demandées. Les nuages de points y

résultants sont non organisés et ont des topologies non identifiables et c'est pour cela qu'il est impossible d'avoir un cadre de travail unique dans le cas des freeform. L'aube de turbine a été mesurée par triangulation laser sur MMT et le nuage de points résultant manque d'informations sur sa topologie et sa géométrie et ne peut être utilisé pour faire de la métrologie dimensionnelle. Les méthodes de caractérisation des surfaces asphériques ainsi que celles des surfaces gauches feront l'objet des chapitres 2 et 3, respectivement.

Chapitre 2 - Reconstruction de surfaces complexes à modèles connus: cas des surfaces asphériques

Les métrologues font référence au terme reconstruction par le terme association (qui veut dire fitting en anglais). Dans le cas de l'association, on associe un modèle mathématique (implicite ou explicite) connu à des données discrètes. C'est pourquoi que dans le cas des aubes, ce traitement n'est pas possible vu qu'aucun modèle mathématique n'existe. La caractérisation de surfaces asphériques via l'association de modèles asphériques n'est pas très abondante dans la littérature et seulement quelques travaux ont été recueillis [65], [66] et [67]–[71]. Ce chapitre traite d'un retour sur l'état de l'art des méthodes d'optimisation, les méthodes de caractérisation de surfaces asphériques mais aussi de la validation et de l'utilisation d'une nouvelle méthode que nous proposons à ce sujet, la méthode du L-BFGS. Cette méthode est validée sur des données simulées en s'appuyant sur la norme ISO 10360-6 [72] et puis appliquée à des données mesurées. Le L-BFGS montre des performances supérieures à celles des méthodes classiques [73].

Le processus d'association est décrit dans la norme ISO 17450-Part 1 [74]. L'association est l'opération utilisée pour ajuster un élément idéal (le modèle) à un élément non idéal (le nuage de points) selon un critère (tel que les moindres carrés). Le modèle peut être sous forme discrète telle qu'un nuage de points théorique ou un maillage, ou sous forme continue telle qu'un modèle CAO ou une formulation mathématique. Dans le cas des modèles discrets, l'Iterative Closest Point (ICP) est l'algorithme le plus utilisé et les distances sont calculées point-à-point et/ou point-à-triangle. Parmi les algorithmes d'optimisation qui sont les plus utilisés nous citons les algorithmes de minimisation de distances orthogonales, l'ICP [75]–[77], le simplex [78] et bien d'autres algorithmes tels que les algorithmes non déterministes comme l'évolution différentielle, l'algorithme génétique, etc… pour ce travail nous évoquons

l'ICP, la méthode Newton-Raphson, le Levenberg-Marquardt (LM). Nous mettons le point sur la méthode Limited memory- Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) qui sera la contribution de ce chapitre.

$$\Phi = \min_{\boldsymbol{x}}\big(\boldsymbol{\phi}(\boldsymbol{x})\big). \tag{72}$$

Le LM et le L-BFGS ainsi que l'ICP sont testés pour l'ajustement de surfaces asphériques en s'appuyant sur la norme ISO 10360-Part 6 [72] et les travaux de Lin *et al* [79]. Nous nous intéressons uniquement à l'optimisation des paramètres de transformations ($\boldsymbol{R}$ et $\boldsymbol{T}$) en comparant la minimisation des distances verticales et orthogonales au sens des moindres carrés. Nous étudions également l'effet du nombre de points considérés et l'effet de leur répartition sur le temps de calcul et la précision.

Validation des algorithmes L-BFGS, LM et ICP

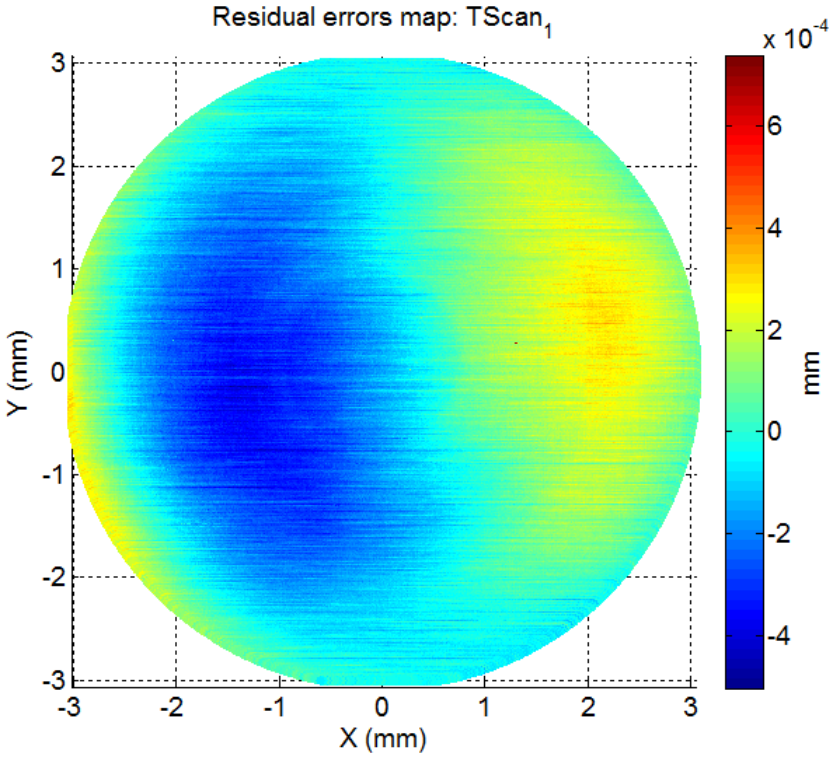
Afin de valider les algorithmes, de les comparer et de montrer l'efficacité de L-BFGS, le modèle asphérique est simulé en générant des points symétriquement distribués autour de l'axe de symétrie. Ensuite, à ces points simulés nous ajoutons du bruit Gaussien et superposons des erreurs systématiques pour modéliser le bruit, la rugosité et les défauts de forme. Le bruit aléatoire superposé aux données théoriques est un bruit Gaussien avec $\mu = 0$ et $\sigma = 8$ nm. La valeur d'écart-type est en cohérence avec l'état de surface qui peut se manifester sur des pièces réelles à cause de défauts de fabrications [80]. Les erreurs systématiques modélisant les défauts de forme sont représentées par des harmoniques de Fourrier et sont superposées aux données précédentes dans la direction orthogonale à la surface en chaque point. Ces erreurs peuvent survenir surtout des erreurs systématiques du mouvement de la machine de fabrication et/ou de mesure [92]. L'erreur crête-à-crête générée s'élève à environs 610 nm ce qui correspond à des erreurs de forme sur des surface asphériques.

Nous montrons à travers l'étude que les trois méthodes, ICP, LM et L-BFGS retournent des valeurs de RMS et PV très similaires, et ce, pour tous les cas de nombres de points tant que ce dernier ne descend pas sous le seuil des 1000 points. La complexité des algorithmes a été aussi étudiée et mesurée en termes de temps de calcul et de mémoire utilisée. En ce qui concerne l'allocation mémoire, tous les algorithmes ne dépassent pas les 1 Go de mémoire active. Par contre, concernant le temps de calcul, surtout si nous augmentons le nombre de

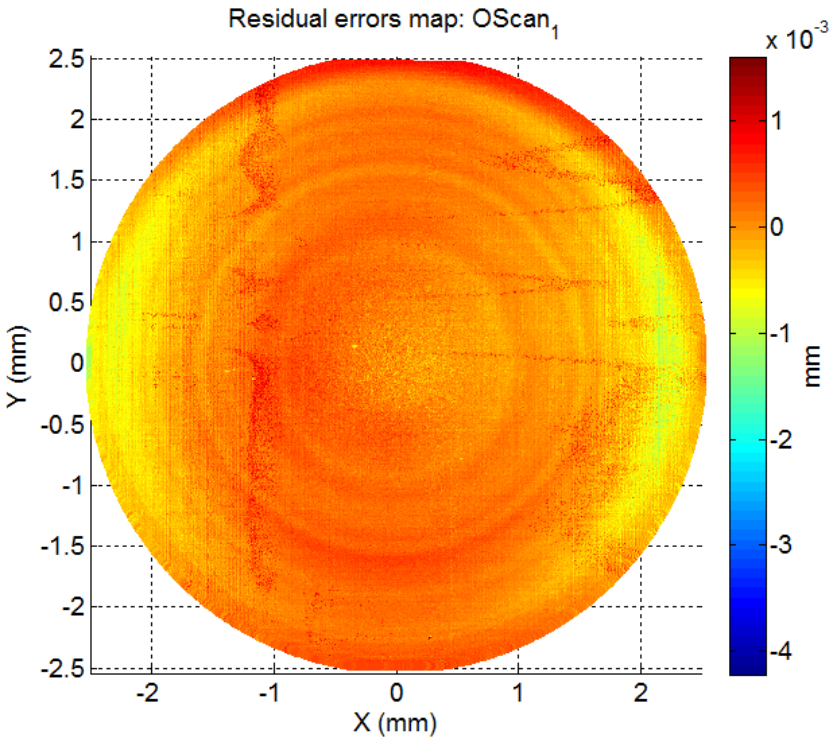
paramètres à calculer (on intègre l'estimation des paramètres du modèle dans le calcul), le L-BFGS montre clairement ses atouts. Finalement, des tests que nous avons effectués pour évaluer les effets des paramètres intrinsèques de L-BFGS montrent que ces paramètres n'influençaient pas la complexité ou la précision du résultat d'une manière significative.

Application à des données mesurées

Les algorithmes sont validés sur des données simulées pour l'ajustement de modèles asphériques, et sont alors appliqués à des données issues de mesure. Le L-BFGS montre encore une fois son efficacité et sa précision :



(a)



(b)

Cartographies des erreurs résiduelles après ajustement du modèle de la lentille AO775 avec le L-BFGS sur les données mesurées: (a) $TScan_1$, (b) $OScan_1$.

Chapitre 3 - Reconstruction de courbes et surfaces freeform

Les surfaces freeform ou gauches présentent un vrai défi car dans de nombreuses applications car elles peuvent ne pas être accompagnées d'un modèle CAO [81]. Au mieux, les surfaces freeform manquent d'une définition mathématique formelle mais sont représentées par des modèles B-Spline et/ou NURBS [82]. Nous nous appliquons aux modèles B-Splines afin d'effectuer une association sur des données initialement non-organisées et bruitées. Dans ce chapitre nous faisons un retour sur l'état de l'art concernant les techniques de maillage, les techniques de régression de modèles B-Splines existantes et enfin proposons une nouvelle approche par déformation de contour actifs de B-Spline. Nous exposons les avantages et les limites de cette approche afin de montrer son efficacité et les pistes de son amélioration.

Les techniques de maillage peuvent être classées sous deux grandes familles, une famille de techniques se basant sur des structures combinatoires et une autre famille de techniques se

basant sur le calcul de fonctions implicites. Les techniques appartenant à la première famille se basent sur les deux représentations duales, le diagramme de Voronoi et la triangulation de Delaunay. Ces graphes se sont avérés très appropriés quant à la structuration de données non-organisées. Ensuite, la construction du maillage surfacique se fait à l'aide de techniques qui cherchent à trouver une approximation des facettes triangulaires appartenant au Delaunay restreint à la surface. Nous citons, présentons et comparons quelques méthodes de maillage en se basant sur des critères de comparaison bien définis et qui sont de bons indicateurs de robustesse et de qualité de reconstruction [83]–[88].

Les techniques appartenant à la catégorie des surfaces implicites cherchent à extraire le même type de maillages non pas grâce à une approche géométrique ou topologique mais le calcul de fonctions implicites qui s'annulent au niveau de la surface sous-jacente [89]–[94]. Plus précisément, une fonction implicite f associe un scalaire, $c = 0$, à un ensemble de données dans l'espace. Généralement, les fonctions implicites sont des fonctions de distance mais peuvent prendre d'autres formes.

Etat de l'art

La raison pour laquelle nous étudions les techniques de reconstruction aboutissant à un maillage triangulaire se résume à la possibilité de convertir la surface reconstruite maillée en une surface paramétrique adaptée à la représentation d'objets et de topologies complexes. Classiquement, lorsque le modèle d'une surface n'est pas connu, et qu'il n'est surtout pas de forme canonique, la reconstruction de surfaces freeform se fait en passant par un maillage. Celui-ci permet la structuration des données et leur transformation en des patches paramétriques B-Spline ou NURBS [9]–[15]. Le but de ce chapitre est d'exposer les limites de ces techniques et de montrer qu'elles ne sont pas suffisantes pour accomplir notre objectif étant le développement d'un algorithme de reconstruction automatique, robuste et qui s'applique à des données bruitées de grandes tailles. Les critères utilisés pour évaluer les différentes méthodes de maillage sont les garanties, la robustesse et la complexité. En ce qui concerne les garanties, il est impératif de rappeler la condition nécessaire pour l'obtention d'un maillage géométriquement proche et topologiquement équivalent à la surface sous-jacente. Cette condition sur la densité d'échantillonnage est bornée par un seuil haut indiquant une densité minimale et un seuil bas indiquant une densité maximale. Cette condition appelée condition d'ε-échantillonnage est basé sur une théorie impliquant que les points échantillonnés appartiennent à une surface théorique connue. Comme dans notre cas les

données peuvent être bruitées et la surface peut être inconnue, la garantie d'un maillage manifold correct n'existe plus [83]–[88], [95]–[102].

Les techniques de maillage ne garantissent pas une reconstruction correcte et/ou manifold et la complexité dépend majoritairement de la triangulation de Delaunay qui est de l'ordre de $O(N^2)$. Pour nos applications de reconstruction de surfaces asphériques et freeform, nous montrons que ces techniques ne sont pas adaptées et nous optons alors à une méthode d'association de courbes et surfaces de type B-Spline aux points sans passer par un maillage. La littérature concernant ce sujet révèle des résultats promettant. L'une des méthodes les plus connues dans ce domaine est celle de la déformation de contours actifs en utilisant des modèles B-Spline. Cette technique introduite par Kass *et al* [103] et développée ultérieurement par Pottmann *et al* [104] consiste à faire évoluer une surface, d'une manière itérative, afin que celle-ci soit, à l'étape finale, une bonne approximation des points. La surface peut être initialisée de manières différentes mais les techniques d'initialisation ne sont pas assez robustes aujourd'hui et s'appuient sur le fait que la forme de la surface est connue au préalable. Il en découle aussi qu'une paramétrisation initiale ne peut être construite correctement que si la surface initiale est relativement proche des données. Selon Kineri *et al* [105] le problème d'initialisation de la surface se fait en construisant les courbes appartenant à la frontière de la surface B-Spline sur des points sélectionnés et supposés sur la frontière. Lorsque la forme de la surface n'est pas connue au préalable ou s'il est impossible de déterminer les points qui appartiennent à la frontière, la méthode de Kineri *et al* ne fonctionne plus. Dans les travaux de Minh et Forbes [106], la surface initiale est supposée connue grâce à la connaissance du modèle CAO de la surface sous-jacente et les auteurs proposent une technique géométrique (basée sur la triangulation de Delaunay) pour l'initialisation des paramètres. Dans les travaux de Zheng *et al* [107], la génération d'une surface B-Spline initiale n'est même pas traitée. Par conséquent, nous déduisons que l'initialisation est une étape très importante mais difficile.

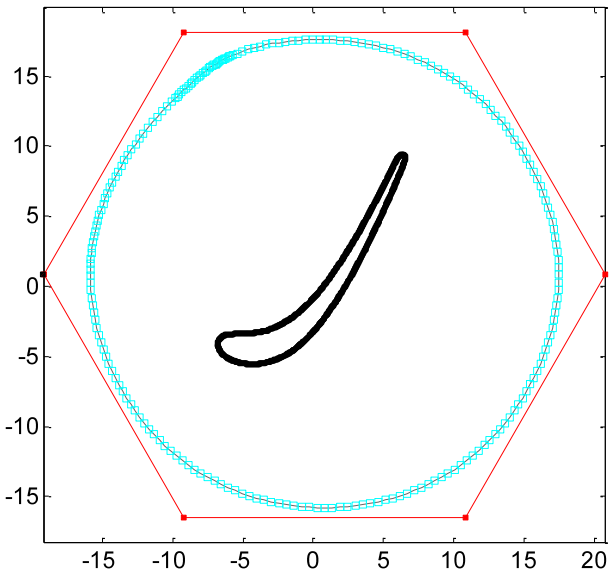
Déformation de contours actifs de courbes B-Spline dans le plan

Nous sommes donc amenés à proposer une méthode qui ne tient pas compte de l'initialisation et qui garantisse une bonne reconstruction quelque soit la forme de la surface et l'initialisation associée, évidemment sous réserve de certaines limites (fortes concavités, topologies de genre élevé, etc…). Notre approche ne tient donc pas compte d'une initialisation

qui soit nécessairement proche des points ni de la paramétrisation de ces derniers. L'originalité de l'algorithme que nous proposons est qu'il permet d'associer des courbes B-Spline à un nuage de points qui peut être non-organisé et bruité. De plus, étant discrète, notre approche offre un temps de calcul suffisamment rapide et permet d'éviter des calculs différentiels et des projections de points. Nous présentons pour le reste du travail, une application sur les courbes fermées dans le plan dans le but de caractériser les aubes de turbines.

Caractérisation des aubes de turbine

Les courbes fermées requièrent des modèles B-Splines dites périodiques dont le vecteur nœud a la particularité d'avoir des intervalles extrêmes égaux par paires. Une courbe B-Spline est définie par trois éléments, le degré de la courbe, le vecteur des nœuds et le polygone de contrôle. Comme nous ne nous intéressons pas à l'initialisation et que nous cherchons à résoudre le problème dans son cadre générique, l'initialisation de la B-Spline se fait en plaçant les points de contrôle sur un cercle autour des données. Cette initialisation s'affranchit de toute connaissance sur la forme de la surface sous-jacente et sur l'ordre des données. Lorsque le polygone de contrôle est construit, nous utilisons une forme très approchée de la courbe B-Spline en appliquant des subdivisions globales récursives du polygone de contrôle par l'algorithme d'Oslo [108]. Nous obtenons ainsi un polygone de contrôle subdivisé dont les sommets sont suffisamment et finement proches de la courbe réelle. En notant les points de contrôle par le vecteur de points 3D $\mathbf{C}$, et les points subdivisés par $\mathbf{Q}$, nous avons une relation très simple pour calculer les points subdivisés à partir des points de contrôle : $\mathbf{Q} = \mathbf{MC}$, où $\mathbf{M}$ est appelée la matrice de subdivision globale.



Initialisation générique d'une B-Spline périodique autour des données (noir) : le polygone de contrôle (rouge) et sa subdivision (bleu).

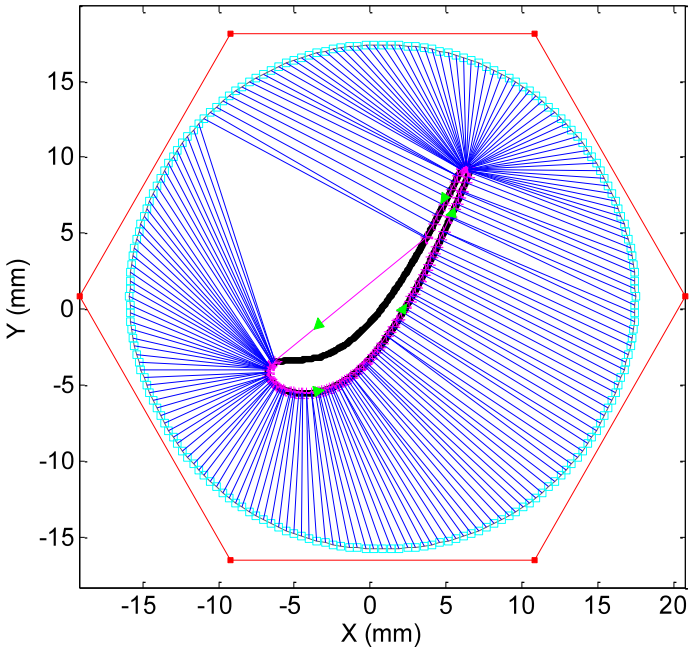
Le principe de notre algorithme se résume par une minimisation au sens des moindres carrés des distances qui tiennent à la fois compte de la géométrie et de la topologie. Plus explicitement, nous cherchons à minimiser la différence entre la variation de la position des points subdivisés d'une itération à l'autre et la distance séparant les données du polygone subdivisé :

$$\min_{t_1^{(k)},\dots,t_{n_c}^{(k)}} \sum_{i=1}^{n_d} \sum_{j=1}^{n_C} \left(\langle \left(q_i^{(k+1)} - q_i^{(k)} \right), \ \vec{\varepsilon}_i^{(k)} \rangle - \delta_i^{(k)} \right)^2, \tag{73}$$

où les points $q_i^{(k)}$ sont les points subdivisés, $\vec{\varepsilon}_i^{(k)}$ les vecteurs unitaires de distance et les $\delta_i^{(k)}$ les distances à l'itération k. Les scalaires $\left\{ t_1^{(k)}, \dots, t_{n_c}^{(k)} \right\}$ correspondent aux amplitudes par lesquelles les points de contrôle à l'itération k doivent se déplacer (dans la direction normale des points de contrôle) afin d'obtenir un polygone de contrôle $\mathbf{C}^{(k+1)}$ à l'itération $k+1$ qui garantira que la courbe B-Spline se rapproche des données.

Le calcul des distances $\delta_i^{(k)}$ se fait comme précisé précédemment en tenant compte de la proximité géométrique et de la topologie sous-jacente aux points. Ce processus commence par la recherche des points les plus proches aux points $q_i^{(k)}$ parmi les données. Cette première étape est l'étape de l'association *indirecte* dont nous montrons le principe dans la figure

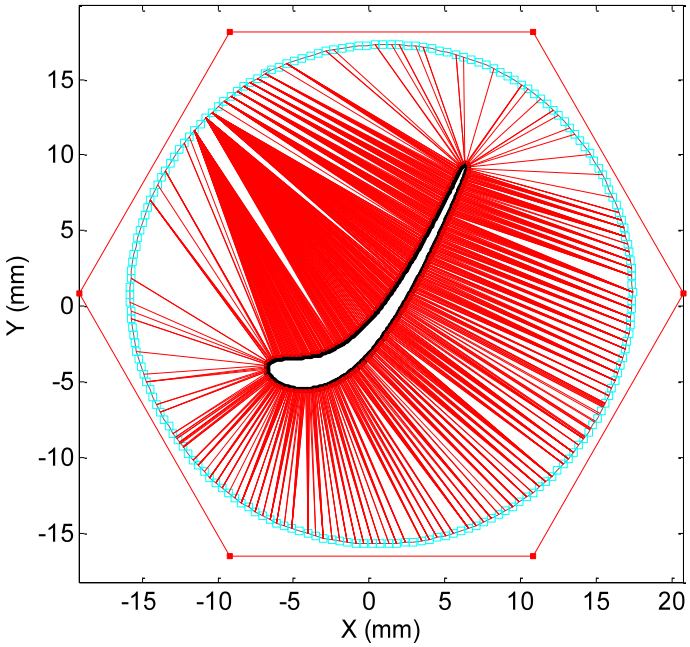
suivante. Les segments bleus relient les données (noir) à des points de la subdivision (bleu) par une simple recherche de proximité géométrique (distance Euclidienne).



Association inverse (segments bleus) : calcul des correspondances entre chaque point subdivisé et les données.

Cette première étape d'association permet d'avoir une première notion sur la topologie sous-jacente aux données et induit une certaine organisation sur ces dernières (segments magenta et flèches vertes).

Une étape d'association *directe* vient ensuite calculer les distances séparant les données des segments subdivisés de la B-Spline. A l'aide des informations recueillies à travers la correspondance des points de l'étape précédente, les distances séparant chaque point de son correspondant géométrique et topologique sont calculées. Ces distances rentrent dans le processus de minimisation donné par l'expression ci-dessus (4).

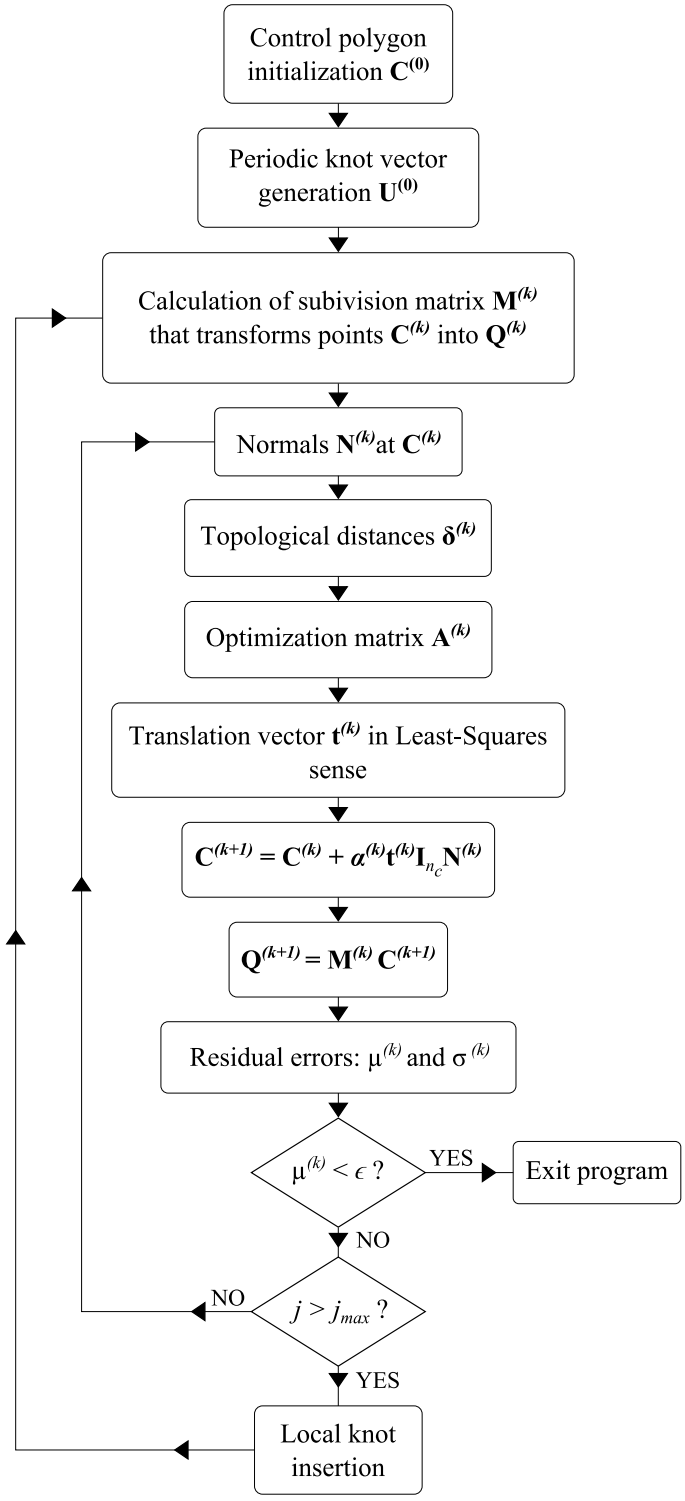


Association directe (segments rouges) déduite de l'association indirecte.

L'application de l'expression (73) ne mène généralement pas à une solution satisfaisante où la moyenne des écarts résiduels serait sous le seuil de tolérance prédéfini. C'est pour cela que notre algorithme admet deux améliorations :

- une amélioration par l'insertion de nœuds et traduite par l'ajout de points de contrôle dans les zones où l'erreur résiduelle est localement supérieure au seuil,
- une amélioration par le biais du "*fairing*" ou autrement dit de la tension de courbe.

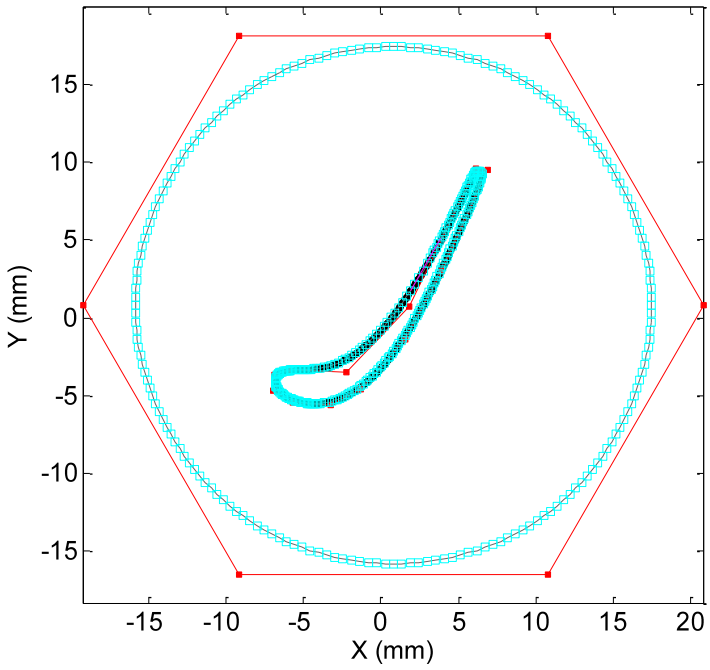
L'insertion locale de nœuds et par conséquent de points de contrôle se fait dans une boucle extérieure alors que la résolution itérative du problème de l'expression (73) se fait dans une boucle intérieure comme l'indique le pseudo algorithme de la figure ci-après. L'insertion locale de nœuds permet d'améliorer la précision de l'approximation mais présente des limites. L'insertion excessive peut créer des auto-intersections au niveau de la courbe B-Spline et ceci n'est pas souhaitable. Afin de remédier à ce problème, nous introduisons un terme de tension de courbe et le rajoutons à l'expression de minimisation.



Pseudo-algorithme de déformation de contours actifs de B-Spline discrètes.

Les tests montrent que les deux améliorations proposées sont indispensables à l'obtention de bons résultats. La caractérisation d'aubes de turbine et plus généralement de courbes fermées dans le plan est réussie dans la majorité des essais que nous avons réalisés. Ces essais incluent différentes positions et orientations initiales du polygone de contrôle par rapport aux données.

L'algorithme est rapide mais une importante optimisation de code peut nettement améliorer le temps de calcul. L'algorithme n'est pas totalement automatique car le choix des paramètres qui le contrôlent n'est pas encore au point d'être robuste. Par ailleurs, la tolérance recherchée est dans la majorité des cas assurée.



Algorithme DBACD avec terme de tension : $\lambda = 0.8$ qui varie automatiquement en accord avec la moyenne des écarts résiduels.

Conclusion générale

Les surfaces complexes posent des défis considérables quant à leur spécification, fabrication et mesurage ainsi que lors de l'évaluation de leur défaut de forme. Les processus de fabrication et de mesure de surfaces complexes sont fortement tributaires des dimensions, des tolérances et des formes spécifiées. Afin de maîtriser les données acquises par les systèmes de mesure, une étape importante de traitement s'impose. Il s'agit généralement de reconstruction de surface afin de reconstituer la géométrie et la topologie de la surface sous-jacente et d'en extraire les informations nécessaires pour des besoins de métrologie dimensionnelle (caractéristiques dimensionnelles et évaluation des défauts de forme). Pour la catégorie des surfaces asphériques, où un modèle mathématique y est associé, le processus de traitement de données géométriques, non nécessairement organisées, se fait par l'association du modèle sur les données. Les résidus d'association recherchés en optique sont typiquement

de l’ordre du nanomètre. Dans ce cadre, l'algorithme L-BFGS qui n'a jamais encore été utilisé en métrologie permet de résoudre des problèmes d'optimisation non-linéaires sans contraintes d'une manière fidèle, automatique et rapide. La méthode L-BFGS reste efficace pour des données constituées de plusieurs millions de points. La caractérisation des surfaces asphériques provient d'un besoin Européen qui s'est matérialisé par le projet Européen IND10: FORM. Par conséquent, ce travail de thèse comporte aussi une comparaison des moyens de mesures issues des différentes machines d'ultra-précision présentes dans chacun des instituts Européens partenaires du projet. Lors de cette comparaison menée, la méthode du L-BFGS s'est avérée être la plus performante.

Dans la catégorie des surfaces gauches et notamment les aubes de turbines, la fabrication, le mesurage et le traitement sont à toute une autre échelle, sub-micrométrique. Les surfaces gauches ne sont généralement pas définies par un modèle mathématique et sont donc représentées par des modèles paramétriques de type B-Spline et/ou NURBS. Dans ce cadre, nous exposons un état de l'art détaillé et proposons une nouvelle approche itérative de régression B-Spline. L'algorithme proposé est indépendant des problèmes liés à l'initialisation et au paramétrage initial et par conséquent, c'est une nouvelle contribution dans ce domaine. Nous établissons par la suite une étude approfondie en évoquant les avantages ainsi que les limites actuelles de cette approche sur des exemples de courbes fermées en 2D.

Ce travail de thèse a des perspectives variées. Sur la partie traitant des pièces asphériques, un prochain projet Européen EMPIR va se mettre en place afin de tenter de trouver une méthode d'association qui se baserait sur le critère du MinMax. En ce qui concerne le travail sur les aubes de turbine, les améliorations majeures se présentent dans l'optimisation de code, dans la maîtrise des paramètres de l'algorithme et dans l'extension de cette méthodes aux courbes et aux surfaces en 3D.

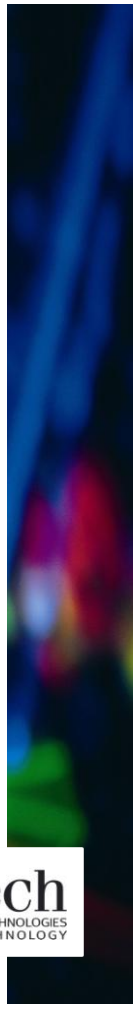
CONTRIBUTION A LA RECONSTRUCTION DE SURFACES COMPLEXES A PARTIR D'UN GRAND FLOT DE DONNEES NON ORGANISEES POUR LA METROLOGIE 3D

RESUME : Les surfaces complexes ont des applications dans divers domaines tels que ceux de la photonique, de l'énergie, du biomédical, du transport... Par contre, elles posent de véritables défis quant à leur spécification, fabrication et mesure ainsi que lors de l'évaluation de leur défaut de forme. Les processus de fabrication et de mesure de surfaces complexes sont fortement tributaires des dimensions, des tolérances et des formes spécifiées. Afin de rendre exploitable les informations données par le système de mesure, une étape importante de traitement s'impose. Il s'agit ici de la reconstruction de surfaces afin de reconstituer la géométrie et la topologie de la surface sous-jacente et d'en extraire les informations nécessaires pour des besoins de métrologie dimensionnelle. Dans la catégorie des surfaces asphériques pour lesquelles un modèle mathématique est associé, le processus de traitement de données géométriques, non nécessairement organisées, se fait par l'association du modèle aux données. Les résidus d'association recherchés en optique sont typiquement de l'ordre du nanomètre. Dans ce cadre, nous proposons l'utilisation de l'algorithme L-BFGS qui n'a encore jamais été utilisé en métrologie. Ce dernier permet de résoudre des problèmes d'optimisation non-linéaires, sans contraintes d'une manière robuste, automatique et rapide. La méthode L-BFGS reste efficace pour des données à plusieurs millions de points. Dans la catégorie des surfaces gauches et notamment des aubes de turbines, la fabrication, la mesure et le traitement sont à une toute autre échelle, sub-micrométrique. Les surfaces gauches ne sont généralement pas définies par un modèle mathématique mais sont représentées par des modèles paramétriques de type B-Spline et/ou NURBS. Dans ce cadre, nous exposons un état de l'art détaillé et proposons une nouvelle approche itérative d'association B-Spline. L'algorithme s'affranchit de tous les problèmes liés à l'initialisation et au paramétrage initial. Par conséquent, un tel algorithme constitue une nouveauté dans ce domaine. Nous établissons une étude approfondie en évoquant les avantages et les limites actuelles de cette approche sur des exemples de courbes fermées en 2D. Nous complétons ensuite cette étude par des perspectives d'amélioration et de généralisation aux surfaces en 3D.

Mots clés : métrologie dimensionnelle, caractérisation, surfaces asphériques, ailette, L-BFGS, déformation de contours actifs, B-Spline.

ABSTRACT : Complex surfaces exhibit real challenges in regard to their design specification, their manufacturing, their measurement and the evaluation of their manufacturing defects. They are classified according to their geometric/shape complexity as well as to their required tolerance. Thus, the manufacturing and measurement processes used are selected accordingly. In order to transcribe significant information from the measured data, a data processing scheme is essential. Here, processing involves surface reconstruction in the aim of reconstituting the underlying geometry and topology to the points and extracting the necessary metrological information (form and/or dimensional errors). For the category of aspherical surfaces, where a mathematical model is available, the processing of the data, which are not necessarily organized, is done by fitting/associating the aspherical model to the data. The sought precision in optics is typically nanometric. In this context, we propose the L-BFGS optimization algorithm, first time used in metrological applications and which allows solving unconstrained, non-linear optimization problems precisely, automatically and fast. The L-BFGS method remains efficient and performs well even in the presence of very large amounts of data. In the category of general freeform surfaces and particularly turbine blades, the manufacturing, measurement and data processing are all at a different scale and require sub-micrometric precision. Freeform surfaces are generally not defined by a mathematical formula but are rather represented using parametric models such as B-Splines and NURBS. We expose a detailed state-of-the-art review of existing reconstruction algorithms in this field and then propose a new active contour deformation of B-Splines approach. The algorithm is independent of problems related to initialization and initial parameterization. Consequently, it is a new algorithm with promising results. We then establish a thorough study and a series of tests to show the advantages and limitations of our approach on examples of closed curves in the plane. We conclude the study with perspectives regarding improvements of the method and its extension to surfaces in 3D.

Keywords : dimensional metrology, form characterization, aspherical surfaces, freeform, blade inspection, L-BFGS, Active contour Deformation, B-Spline.



ARTS ET MÉTIERS ParisTech

ParisTech INSTITUT DES SCIENCES ET TECHNOLOGIES PARIS INSTITUTE OF TECHNOLOGY