Journal of Mechanical Engineering and Automation 2015, 5(3): 94-111
DOI: 10.5923/j.jmea.20150503.02

Online Validation of Chebyshev Geometric Element Algorithms Using the TraCIM-System

Daniel Hutzschenreuter[1], Frank Härtig[1,*], Klaus Wendt[1], Ulrich Lunze[2], Harald Löwe[3]

[1]Physikalisch-Technischer Bundesanstalt, Braunschweig, Germany
[2]University of Applied Science Zwickau, Zwickau, Germany
[3]Technische Universität Braunschweig, Braunschweig, Germany

Abstract In various areas of industrial manufacturing the correct evaluation of Chebyshev geometrical elements has fundamental importance for inspection of products with Coordinate Measuring Machines (CMM). According to standard ISO 1101 these are the default algorithms for accessing form deviation and to calculate certain datums for the evaluation of position deviation. To validate the accuracy of these computations, a network of National Metrological Institutes (NMIs) and Designated Institutes (DIs) established an online service which, among other tests, allows manufacturers and users of CMM software to test their mathematical algorithms for Chebyshev. The test covers five important types of associated geometrical elements which are calculated by the so-called Minimum-Zone criterion (MZ). By registration at the TraCIM (traceability for computationally-intensive metrology) online service a user can order the test for validation of his algorithms. Each test contains fifty individual data sets. The users' evaluation results are compared against reference results provided by the institution offering the test. Procedures for the comparison cover numerical uncertainties of the reference results and maximum permissible errors (MPEs) specified by the user. For each test an online report including detailed test results is issued. Test data, test procedure and test evaluation are subjected to quality rules specified and controlled by TraCIM e.V. which is an international head organization for metrological algorithm validation.

Keywords Chebyshev, Minimum-Zone, TraCIM, Software validation, Geometrical elements, Coordinate Metrology

1. Introduction

3D Coordinate Metrology is an essential part of different industrial sectors such as automotive, mechanical and medical engineering in order to meet the steady growing demands on quality of products. A key requirement of 3D Coordinate Measuring Machine (CMM) software is therefore the correct fitting of ideal geometrical elements for point data that is extracted from the surface of features from work pieces. The underlying computational aims are mathematical approximation problems. In general solving these problems requires the application of different numerical algorithms.

Conventional calibration of CMM principally utilizes reference artefacts like block gauges, calibration cubes or spheres or complex artefacts like ball plates. Their associated measurands are connected with the meter definition by a chain of precedent calibration. Obviously calibration with artefacts uses only a small amount of algorithms that are available at CMM. Hence calibration results give limited conclusion on the overall accuracy and stability of the software used. Therefore, autonomous testing of software components with test data is a convenient and cost-efficient supplement for conventional calibration.

ISO 10360 series [1] and VDI 2617 [2] specify requirements for the calibration of CMM. A wide area of independent software testing remains unhandled by these standards. Only in ISO 10360 part 6 basic procedures for testing least squares (Gaussian) geometrical element algorithms are described.

Actual there is no test available according to that standard. A similar development is the least squares element test from [3]. The NMI of Germany – PTB (Physikalisch-Technische Bundesanstalt) has been offering this test to industry for more than 20 years. Other important fitting algorithms are Minimum-Zone best-fit (Chebyshev), Maximum-Inscribed and Minimum-Circumscribed geometrical elements. The calculation of these elements needs the application of considerably more complex numerical algorithms than those which are used for least squares evaluation. However, regulations on software tests for Chebyshev elements are least developed in standardization. Urgently needed tests have not been available for industrial applications until now.

To overcome this critical situation PTB in cooperation with the University of Applied Science in Zwickau has established a test for Chebyshev geometrical element algorithms according to the Minimum-Zone best-fit criterion.

* Corresponding author:
frank.haertig@ptb.de (Frank Härtig)

Published online at http://journal.sapub.org/jmea

The test is referred to as Chebyshev test and comprises the evaluation of five important geometrical elements. These are line and circle in 2D space as well as plane, sphere and cylinder in 3D space. In total one test comprises fifty individual data sets. The test is implemented within the scope of a software validation online service that was established in 2014 during an European research project named Traceability for computationally-intensive metrology (TraCIM) [4]. It offers a web interface including a web shop [5] and a client-server application for testing of metrological software. It is available from any location worldwide with an internet connection. This service is referred to as TraCIM service. Figure 1 depicts the general concept.

A NMI, which in this case is PTB, provides the test service. This is of paramount importance, since the tests are to be carried out – or at least monitored – by the supreme metrological authority of a country. PTB is organized with other metrology institutes under the umbrella of the TraCIM association (TraCIM e. V.). It's main task consists in describing unified quality rules and in defining the technical infrastructure under which the algorithm tests are to be run. Each provider is, however, sole responsible and also held liable for the correctness and reliability of the tests. The NMIs act autonomously in defining the test scope, the business workflow, in the maintenance of their datasets, the running of the server and in consultation and support for customers. For this reason, each metrology institute runs its own TraCIM web shop and server. Each server has to be addressed individually, which leads to a different extent of services offered depending on each metrology institute. The metrology institutes, however, have the possibility of providing algorithm tests mutually as subcontractors, which allows a test provider to enhance the extent of services offered.

The primary users of the TraCIM service are manufacturers of analysis and evaluation software for measuring instruments. The service allows them to have their analysis algorithms validated and officially approved by an national metrology institute. This mainly serves to increase confidence in the products they offer on the market. In principle, they can have this service unlocked for their customers in order to have, for example, updates validated directly on the user's computer. Software engineers can already test their algorithms during the development phase to be on the safe side and, thus, make development faster. New customers who want to access the service need to register once and can access individual tests via the internet afterwards. The service is available 24 hours a day at each day of the year at each location on the globe.

The following sections give details on the implementation of the Chebyshev test within the TraCIM system. After an introduction to the mathematical computation for the geometric elements of interest the TraCIM test design and requirements for an successful implementation of the test are presented. It comprises specific technical requirements and business aspects associated with testing. A marginal part provides details on the general TraCIM System that apply to all existing and future tests. These are the TraCIM quality rules, IT architecture and the basic test procedure which are provided for completeness of the explanations but are not the core issue of the paper. Further sections will give important details on the creation of the test data and the assessment of associated numerical uncertainties for Chebyshev geometric elements. Finally five public test data sets with reference results are presented.

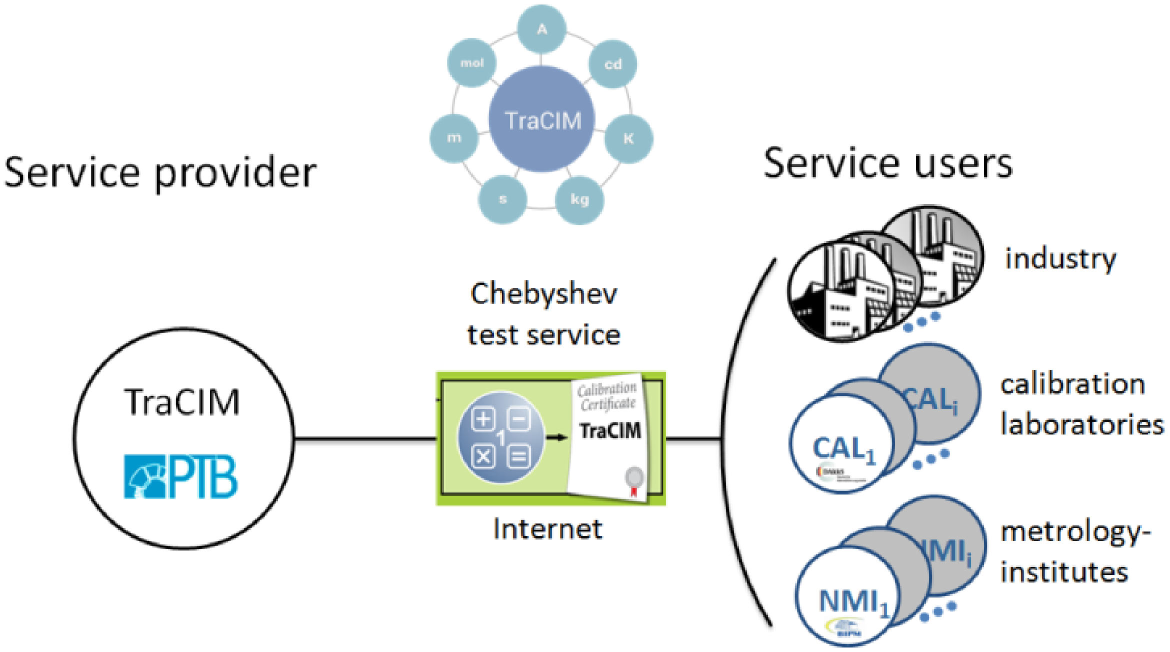


Figure 1. TraCIM Service for Validation of Chebyshev Algorithms

2. Mathematical Background

Clear specification of computational aims for the calculation of Chebyshev geometrical elements is fundamental for a traceable validation of algorithms. This requires a unambiguous definition of input data, appropriate output parameters and the mathematical model which links the input with the output for correct calculation. Figure 2 outlines these properties for fitting a 2D circle with the MZ criterion.

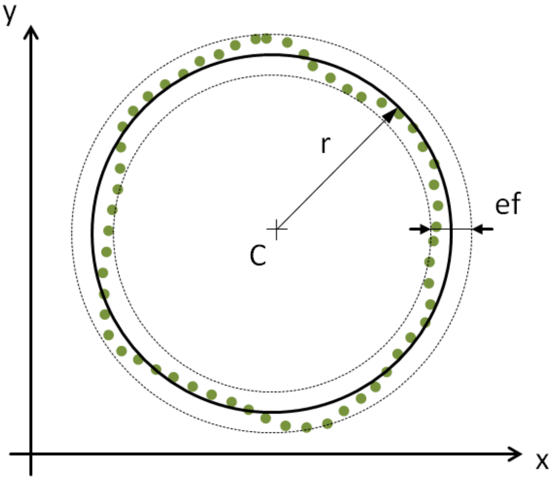


Figure 2. Outline of a 2D circle associated with a point cloud by MZ Criterion

The input data is a finite set of point coordinates which represent the extracted contour of a real circular feature. In the sequel this set is called a point cloud. The output parameters give a 2D circle with ideal geometrical shape and a quantity for the form deviation. The parameters of the circle are the center point C and the radius r. The form deviation is two times the maximum orthogonal distance between all points of the point cloud and the circle. It is denoted by ef. Under assumption of the MZ criterion the objective is to find output parameter values with the smallest possible form deviation. In Fig. 2 this property is depicted by two concentric circles that are adjacent to the point cloud. The area between both circles contains all points and has minimum width. The associated Chebyshev circle is the median circle of the adjacent elements.

In accordance to the previous example general definitions for Chebyshev computational aims are made in the sequel. After giving detailed specifications for each geometrical element of interest the section is concluded with remarks on approaches for calculation of the Chebyshev elements.

2.1. Chebyshev Computational Aims

For $N \in \mathbb{N}$ the point cloud is a set of Cartesian coordinates $P_1,\ldots,P_N$ with $P_i \in \mathbb{R}^k$ for all indices i in index set $I := \{1,\ldots\ ,\ N\}$. In case of 2D elements ($k=2$) there is $P_i=(x_i,y_i)^T$. Else for 3D elements ($k=3$) there is $P_i=(x_i,y_i,z_i)^T$.

Further let $b \in G \subset \mathbb{R}^M$ ($M \in \mathbb{N}$) denote a vector that gives the output parameters of the associated geometrical element. The amount of parameters depends on the individual element and values are restricted to a specific permissible range G.

The orthogonal distance between points P_i and the lateral surface of the geometrical element with parameters b is calculated by the distance function $f: \mathbb{R}^k \times \mathbb{R}^M$, $(P_i,b) \mapsto f(P_i,b)$. The terms $f_i(b) := f(P_i,b)$ are frequently used in further descriptions.

Then the computational aim for calculation of Chebyshev geometrical elements with the MZ criterion is to solve

$$\min_{b \in G} \max_{i \in I} |f_i(b)|. \qquad \text{(P)}$$

b parameter vector
G permissible parameter value range
i index of point P_i
I index set of point cloud
$f_i(b)$ orthogonal distance function

In (P) the term $F(b) := \max_{i \in I} |f_i(b)|$ is minimized. This meets the requirement of geometrical element fitting with the smallest possible form deviation which computes as $ef = 2F(b)$. Individual definitions of parameters and distance functions for the geometrical elements 2D line, 2D circle, plane, sphere and cylinder are listed below. Note, the chosen parameterization agrees with recommendations from ISO 10360 part 6. Moreover the definition of some parameters depends on the centroid $\overline{P}$ of the point cloud. It is defined by equation (1).

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i \qquad (1)$$

$\bar{P}$ centroid of point cloud
N amount of points in point cloud
P_i location vector of point i

2.1.1. Specification for 2D Line

The geometrical element is defined by two parameters. The first are Cartesian coordinates $C = (C_x, C_y)^T$ of a point on the line. For a unique definition this point may be calculated as the projected centroid that is the point on the line with shortest distance to the point cloud centroid. The second parameter is the orientation vector $v = (v_x, v_y)^T$ giving the direction of the line. Its length is constraint to $\langle v,v \rangle = 1$. In total there are four different parameter values. These are assigned to the parameter vector $b = (C_x, C_y, v_x, v_y)^T$ with value range $G = \{b \in\ \mathbb{R}^2 \times S^2 : \langle \overline{P} - C, v \rangle = 0\}$. Term $S^2 := \{v \in\ \mathbb{R}^2\ : \langle v,v \rangle = 1\}$ denotes the unit sphere in $\mathbb{R}^2$. Moreover $\langle \overline{P} - C, v \rangle = 0$ is a constraint for the definition of the projected centroid. Equation (2) gives the orthogonal distance function for 2D line.

$$f_i(b) := \langle P_i - C, n\rangle \tag{2}$$

i index of point P_i
b parameter vector
P_i location vector of point i
C projected centroid on 2D line
n vector orthogonal to 2D line

Vector $n := \left(-v_y, v_x\right)^T$ is orthogonal to v. Note, that the scalar product used for 2D line is the Euclidian product $\langle u, v\rangle := u_x w_x + u_y w_y$ in $\mathbb{R}^2$.

The solution of (P) for the 2D line may be depicted by a pair of parallel lines. All points of the point cloud are enclosed between these lines. The orthogonal distance of the lines is the smallest possible to enclose all points. Then the Chebyshev 2D line is the median line of the pair of lines with minimum distance. The form deviation is the orthogonal distance of the outer adjacent lines.

2.1.2. Specification for 2D Circle

The parameters of the associated 2D circle are the center point $C = \left(C_x, C_y\right)^T$ and the radius $r > 0$. There are no additional constraints on the parameter definition. The parameter vector is $b = \left(C_x, C_y, r\right)^T$. Its permissible value range is $G = \mathbb{R}^2 \times \mathbb{R}_+$. Equation (3) gives the orthogonal distance function for the 2D circle.

$$f_i(b) := \|P_i - C\|_2 - r \tag{3}$$

i index of point P_i
b parameter vector
P_i location vector of point i
C center point of 2D circle
r radius of 2D circle

For an arbitrary $w \in$ ℝ2 the norm calculates as $\|w\|_2 = \sqrt{w_x^2 + w_y^2}$. These definitions and (P) conclude the basic description of the Chebyshev 2D circle from Figure 2.

2.1.3. Specification for Plane

Geometrical element plane has two parameters. Its position is defined by the Cartesian coordinates $C = \left(C_x, C_y, C_z\right)^T$ of a point on the plane. For a unique definition of the plane it may be calculated as the projected centroid that is the point on the plane with shortest distance to the point cloud centroid. The second parameter is the orientation vector $n = \left(n_x, n_y, n_z\right)^T$. It gives the normal of the plane. The length of the vector is constrained to $\langle n, n\rangle = 1$. In total there are six different parameter values defining the plane. These are assigned to the parameter vector $b = \left(C_x, C_y, C_z, n_x, n_y, n_z\right)^T$. Valid values b are within the parameter range $G = \left\{b \in \mathbb{R}^3 \times S^3 : \left(\overline{P} - C\right) \times n = 0\right\}$. Term $S^3 := \left\{v \in \mathbb{R}^3 : \langle v, v\rangle = 1\right\}$ denotes the unit sphere in $\mathbb{R}^3$. Moreover $\left(\overline{P} - C\right) \times n = 0$ is a constraint for the definition of the projected centroid. The operator × in the previous term is the cross product in $\mathbb{R}^{3.}$ It is defined by

$$u \times v = \begin{pmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{pmatrix}$$

u, v vectors in $\mathbb{R}^3$

The orthogonal distance between the plane and the points from the point cloud is calculated by (4).

$$f_i(b) = \langle P_i - C, n\rangle \tag{4}$$

i index of point P_i
b parameter vector
P_i location vector of point i
C projected centroid on plane
n normal vector of plane

Note, that the scalar product used for the plane is the Euclidian product $\langle u, v\rangle := u_x w_x + u_y w_y + u_z v_z$ in $\mathbb{R}^3$.

The solution of (P) for the plane could be depicted by a pair of parallel planes. All points of the point cloud are enclosed between these planes. The orthogonal distance of the planes is the smallest possible to enclose all points. Then the Chebyshev plane is the median plane of this pair of planes. The form deviation is the orthogonal distance of the outer adjacent planes.

2.1.4. Specification for Sphere

The parameters of the associated sphere are the center point $C = \left(C_x, C_y, C_z\right)^T$ and the radius $r > 0$. There are no additional constraints on the parameter definition. The parameter vector is $b = \left(C_x, C_y, C_z, r\right)^T$. Its permissible value range is $G = \mathbb{R}^3 \times \mathbb{R}_+$. Equation (5) gives the orthogonal distance function for the sphere.

$$f_i(b) := \|P_i - C\|_2 - r \tag{5}$$

i index of point P_i
b parameter vector
P_i location vector of point i
C center point of sphere
r radius of sphere

For an arbitrary $w \in \mathbb{R}^3$ the norm calculates as $\|w\|_2 = \sqrt{w_x^2 + w_y^2 + w_z^2}$.

The solution of (P) for the sphere may be depicted by a pair of concentric spheres. All points of the point cloud are enclosed between the lateral surface of both spheres. The radius difference between the spheres is the smallest possible distance to enclose all points. Then the Chebyshev sphere is the median sphere between the pair of concentric spheres. Further the form deviation is the difference of radii between the outer adjacent spheres with minimum distance.

2.1.5. Specification for Cylinder

The geometrical element for cylinder is defined by three parameters. The first parameter gives the position of the cylinder axis. It is denoted by $C = \left(C_x, C_y, C_z\right)^T$ and may be calculated as the point on the cylinder axis with shortest distance to the point cloud centroid (projected centroid). The second parameter gives the orientation vector $v = \left(v_x, v_y, v_z\right)^T$ of the cylinder axis. It is constraint to $\langle v, v\rangle = 1$. Finally the third parameter is the radius $r > 0$ In total there are seven different parameter values assigned to the parameter vector $b = \left(C_x, C_y, C_z, v_x, v_y, v_z, r\right)^T$. The parameter range is $G := \left\{b \in \mathbb{R}^3 \times S^3 \times \mathbb{R}_+ : \left\langle \overline{P} - C, v\right\rangle = 0\right\}$. Condition $\left\langle \overline{P} - C, v\right\rangle = 0$ implies that C is the projected centoid.

Equation (6) gives the orthogonal distance function between the lateral surface of the cylinder and points from the point cloud.

$$f_i(b) = \|(P_i - C) \times v\|_2 - r \tag{6}$$

i index of point P_i
b parameter vector
P_i location vector of point i
C projected centroid on cylinder axis
n direction vector of cylinder axis
r radius of cylinder

The norm is the same as defined in (5). Further the cross product in $\mathbb{R}^3$ is used for calculation of the distance between P_i and the cylinder axis.

The solution of (P) for the cylinder can be depicted by a pair of coaxial cylinders. All points of the point cloud are enclosed between the lateral surface of both cylinders. The radius difference between the cylinders is the smallest possible distance to enclose all points. Then the Chebyshev cylinder is the median cylinder between the pair of concentric cylinders. Further the form deviation is the difference of radii between the outer adjacent cylinders with minimum distance.

2.2. Outline of Chebyshev Element Calculation

Problem (P) is a non linear minimax program (MP) in mathematical optimization theory [6], [7]. The difficulty for a direct solution with an appropriate numerical algorithm for MP is the insufficient smooth objective. E. g. the terms $|f_i(b)|$ are not continuous differentiable in G. Hence other approaches need to be applied. Frequently this requires the transformation of (P) into an ordinary non linear constraint program. Let $e > 0$ denote a new parameter that is an upper bound for $F(b)$. Then $F(b) \le e$ is equivalent to $f_i(b) \le e$ and $-f_i(b) \le e$ for all $i \in I$. The parameter vector is extended to $a := \left(b, e\right)^T$ with the new permissible value range $H := G \times \mathbb{R}_+$. Further there is $g_i^+(a) := f_i(b) - e$ and $g_i^-(b) := -f_i(b) - e$ for all $i \in I$. With these preliminary definitions the transformed computational aim for Chebyshev elements is given by (P').

$$\min_{\mathrm{a} \in \mathrm{H}} e \tag{P'}$$

$$\text{s.t.} \quad g_i^+(a) \le 0 \text{ and } g_i^-(a) \le 0 \ \forall\, i \in I$$

a extended parameter vector
H permissible value range of a
e half form deviation value
i index of point P_i
I index set of point cloud
$g_i^+(a)$ constraint function of point i
$g_i^-(a)$ constraint function of point i

General methods for solving problems of type (P') are presented in [8]. These are for example penalty-barrier-methods or sequential-quadratic-programs. More specific calculation methods applicable for the MZ criterion with distance functions (2) – (6) are presented in [9], [10], [11], [12] and [13].

Finally the existence and uniqueness of a solution for (P) and likewise (P') majorly depends on the structure of the point cloud. Basic recommendations for feasible input data are listed below.

- For element 2D line the point cloud should at least have three pairwise different points.
- For element 2D circle the point cloud should at least have four pairwise different points that are not collinear.
- For element plane the point cloud should at least have four pairwise different points that are not collinear.
- For element sphere the point cloud should at least have five pairwise different points that are not coplanar.
- For element cylinder the point cloud should at least have six pairwise different points that are not coplanar.

Moreover, in the case of symmetric point clouds there may exist multiple equivalent solutions for the Chebyshev computational aim (P).

3. TraCIM Quality Rules

The correctness of the results, the liability of the service as well as the unambiguous rules and descriptions for using of test procedure are essential demands which guarantee an unproblematic and user-friendly service. Therefore, the TraCIM e. V. specified quality rules which have to be followed by all institutes offering the TraCIM test. Below the most important are listed:

§1. Each institute providing tests is liable for the correctness of the reference results.
§2. Input values and its associated uncertainty values are defined as error free.
§3. Reference parameters and their associated uncertainty must be provided.
§4. The test data have to be verified successfully by at least three independent software applications.
§5. Tests shall provide only one unambiguous result.
§6. Test cases shall reflect common practical situations.
§7. Clear description of the tests procedure, parameters to be calculated and validation criterion have to be provided.
§8. It is recommended to provide public test data and reference parameter.

§9. All data sets sent and received as well as all test reports are subject to archiving.

In opposite to an academic view all of the quality rules are targeting pragmatic handling of software tests.

4. IT-Architecture of TraCIM System

PTB provides the TraCIM service by running a server with the TraCIM system application for software testing. It is a worldwide accessible, long-term usable and easy to maintain online application for testing metrological algorithms. TraCIMs' IT architecture consists of four central modules (Figure 3). The web shop is the user interface [5]. Similar to online shopping, interested service users have to get registered via the Internet before they are able to order individual tests. Internally, the web shop is directly connected with the server core application for automated processing of incoming requests such as orders, requests for test data and evaluation of calculated test results.

The core application is a management module, which is operated by a competent metrology institute. It manages all of the operating data and controls the data flow to the other modules.

The expert modules are developed by experts and provide individual tests. Each expert module operates basically autonomously and deals with all logical operations in connection with a test. It makes the test data sets available on request, compares the results computed by the service user (customer) with its own reference results and, finally, issues the test report. Since the individual tests may vary significantly from one application to the other, only few input parameters have been defined by TraCIM for the data traffic between the expert modules and the core application. This applies, for instance, to the support of a software interface in JAVA which allows the expert system to be logged into the server system. Basic processing of data such as unique keys identifying each test and each request are transmitted via this interface. The formatting of the test data may be freely selected, the expert is, to a large extent, free to generate the test data according to his needs. Therefore, new tests and special data structures such as for implementing test cases for Chebyshev geometrical element software can easily be integrated into the TraCIM system.

The formal specification of the TraCIM client-server interface is more restrictive in contrast. The communication runs via a REST interface that is a special type of Https connection. Hereby, the data are being embedded into an XML structure. Then again, within this structure, free formats of test data (such as binary formats or established test data structures) can be defined, depending on the application. The expert module is solely responsible for defining the test data format, generating the test data and analysing the calculated results.

5. Test Design and Processing Data

Figure 4 shows the detailed flow chart for the test of Chebyshev algorithms starting from user registration at the web shop site until issuing the test report.

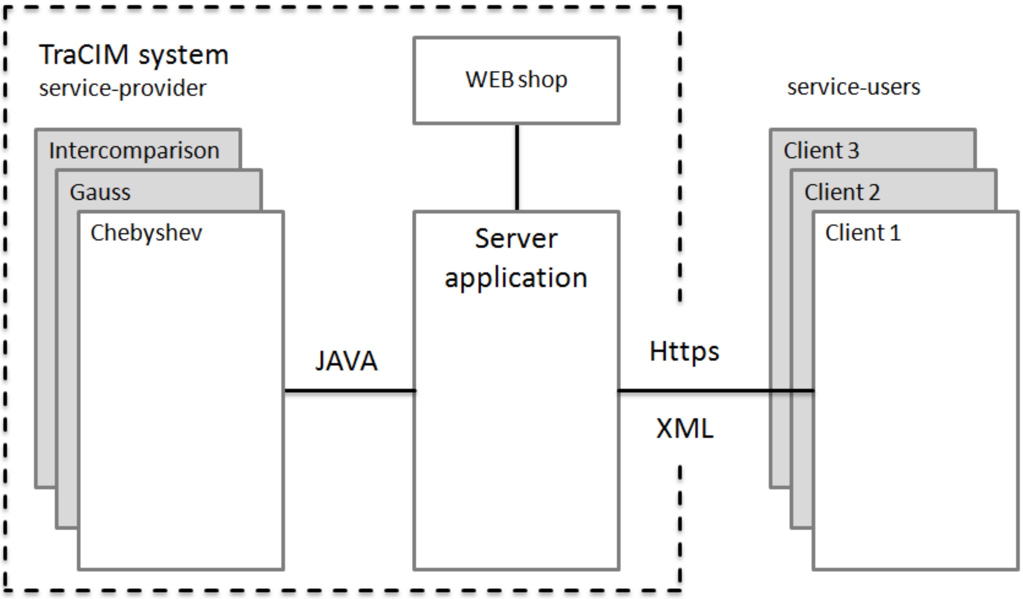


Figure 3. Modular IT-Architecture of TraCIM System

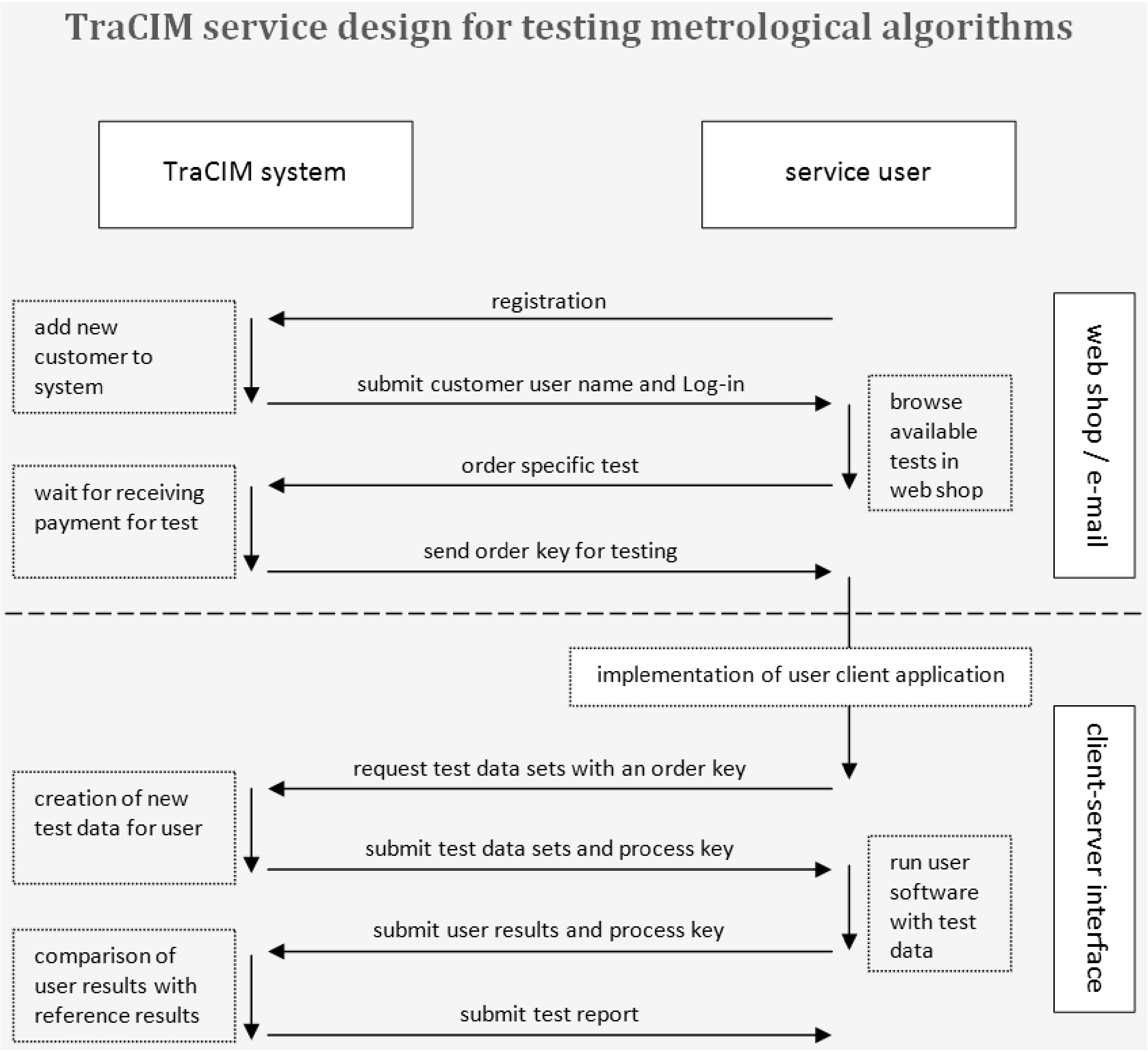

TraCIM service design for testing metrological algorithms
TraCIM system
service user
registration
add new customer to system
submit customer user name and Log-in
browse available tests in web shop
order specific test
wait for receiving payment for test
send order key for testing
web shop / e-mail
implementation of user client application
request test data sets with an order key
creation of new test data for user
submit test data sets and process key
run user software with test data
submit user results and process key
comparison of user results with reference results
submit test report
client-server interface

Figure 4. Flowchart for performing an software test offered by the TraCIM System

5.1. Registration

In order to get access to the TraCIM service any customer must at first register at TraCIM system. PTB's TraCIM web shop (http://tracim.ptb.de) provides a fully automated online registration form. A service user must provide valid contact information like company name, address, phone number and e-mail address. It is essential for PTB to have this information in order to carry out any business activities associated with issuing official test reports and to provide appropriate support. After registration the account data including a unique customer ID, user name and password are generated automatically by the TraCIM system and submitted to the service user by e-mail. These are necessary for further communication with the server and for ordering tests. Registration is free of any charge for all service users.

5.2. Order of PTBs' Chebyshev Test

A valid TraCIM account allows the service users to order individual tests for validating their software. Login at the TraCIM web shop site gives access to a test ordering area. Here the service user may select the test for Chebyshev algorithms. There are two different offers. One is the test with public data at no charge. It should be used preferably to verify the correct implementation of the client-server communication before ordering test data subject to charge. The service user may order this test at any time. The test can be repeated an unlimited number of times. No fees are claimed. The second offer is directed at customers who want to receive an official test report signed by PTB. This test must bepurchased. Therefore, TraCIM office sends a payment request to the service user by e-mail. After payment is received the validation test will be enabled and an order key is automatically generated and send to the service user by e-mail. The order key allows the service user to identify all tests he ordered in the customer area of the web shop.

5.3. Implementation of a Test Client

The technical requirement for performing a TraCIM test is that a service user provides a client application for requesting test data from and submitting calculated results to PTB's TraCIM server. For communication with the TraCIM server a client application must use an Https (Hypertext Transfer Protocol Secure = encrypted Https) connection that allows to send and receive content in the form of character strings, which in case of Chebyshev algorithm tests are messages in XML format. Each Https connection is generated from a specific URL (Uniform Resource Locator), i.e. for requests of test data or requests of the test result. After opening the connection according to URL the following configurations must be done:

- Enable output and input operations for the connection.
- Set the request method "GET" in case of test data request.
- Set the request method "POST" in case of test result transmission.
- Set connection property "content-Type" to "application/xml".
- Set connection property "accept" to "application/xml".
- Set connection property "content-length" to the amount of characters of the content that is send to the TraCIM system.

Service users are advised to use the free-of-charge public test data for checking the correct communication between their client application and the TraCIM server before performing validation tests subjected to charge. Software packages for creation and configuration of Https connections are available for different programming languages such as for example

- Java: java.net API (HttpURLConnection)
- C/C++: Microsoft C++ REST SDK ("Casablanca") or similar
- C#: System.Net (.NET 4.5: System.Net.Http) Assembly

5.4. Requesting Test Data Sets

For test data requests the service user has to contact the TraCIM server using the client application. The necessary Https connection must be configured for a 'GET' request and is opened with the following predefined URL that contains the order key.

https://tracim.ptb.de/tracim/api/order/
<ORDER_KEY>/test

After receiving the request the server application verifies the order key. If the order key is valid the server returns a message containing the test data in a special XML format. A detailed description of the data structure is given in the section on Chebyshev test data. Test data sets are not send by the server, if the test has already been send to the service user, if the identification is unknown or if the validity of the test has expired.

For each test data set the TraCIM system generates a corresponding 'process key' in order to clearly identify the data set. It also allows to assign messages transmitted via the internet connection to individual test data and to a particular testing process. The process key is submitted together with the test data sets to the service user.

5.5. Send Calculation Results for Evaluation

The service user evaluates the test data sets and calculates parameter values of the associates Chebyshev geometrical elements by his software. For validation the results have to be written in a predefined XML structured message which is in detail described in the upcoming section on user results.

The service user should submit the message with his results to TraCIM server with the client application. The necessary Https connection must be configured for a 'POST' request and is opened with the following predefined URL.

https://www.tracim.ptb.de/tracim/api/test/
<PROCESS_KEY>

The address must contain the process key which the service user has received together with the test data sets.

5.6. Receiving the Test Report

After receiving results of the software under test the TraCIM system verifies the validity of the process key. If the key is accepted, the correctness of the results is tested and a report on the outcome is created. It states, whether the application software of the user has computed correct results for the underlying test data sets or not. The report is send back to the service users' client application within a XML message whose content is schematically shown below.

```
<?xml version="1.0" encoding="UTF-8"
                                   standalone="true"?>
<tracim:tracim
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-
                                   instance
  xmlns:chebyshev=http://tracim.ptb.de/chebyshev/test
  xmlns:tracim="http://tracim.ptb.de/tracim">
  <tracim:validation>
     <tracim:reportPDF>JKD5iuDUD098IHh[…]
                                   </tracim:reportPDF>
  </tracim:validation>
</tracim:tracim>
```

The element 'tracim:reportPDF' contains the test report, which is a 'base64' encoded character string. The service user has to decode it into an easy readable pdf-file.

User results are not accepted by the server if they were already send, if the process key is unknown or if the validity of the test has expired.

5.7. Costs

The validation of mathematical software algorithms is subject to charges. This has several reasons: firstly, the service user receives an official report on the evaluation, secondly, the management and maintenance of the TraCIM system generates costs, and, last but not least, the development costs for the system need to be refunded in the long run. Figure 5 shows the prices for testing of Chebyshev algorithms.

There are different offers. Besides a single test, also test packages of 10, 20 and 50 tests will be available.

Purchasing test packages makes sense, for example, in case of a software-developing company willing to have its updates or upgrades validated regularly.

The test is provided by PTB since March 2015. Strictly speaking, a validation is valid only for the software used in the test. Only the software manufacturer can appraise when a modification in his complex analysis software could have impact on the result of the computation. It is therefore his responsibility vis-à-vis his customers to assess whether the validation is still valid or whether the validation must be repeated.

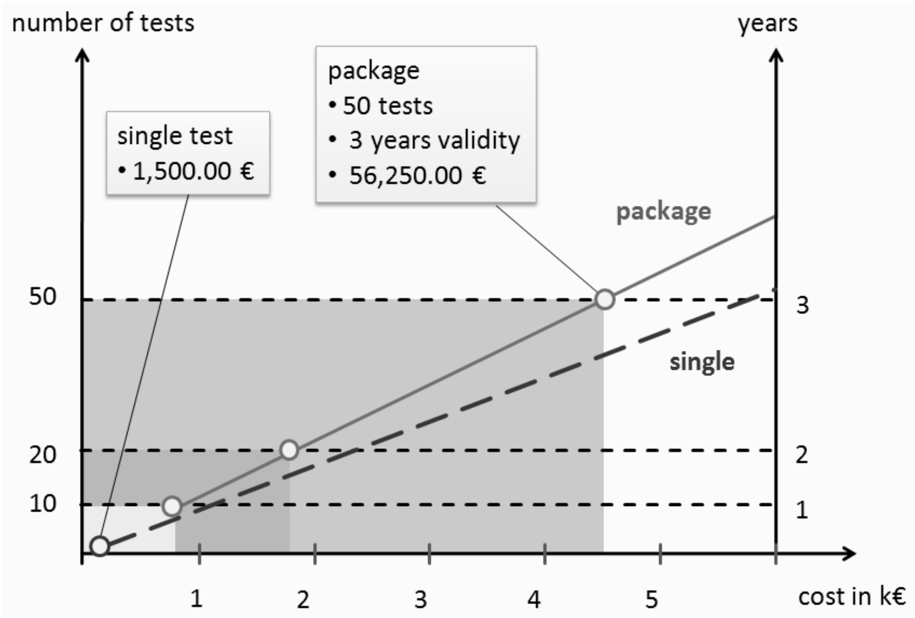
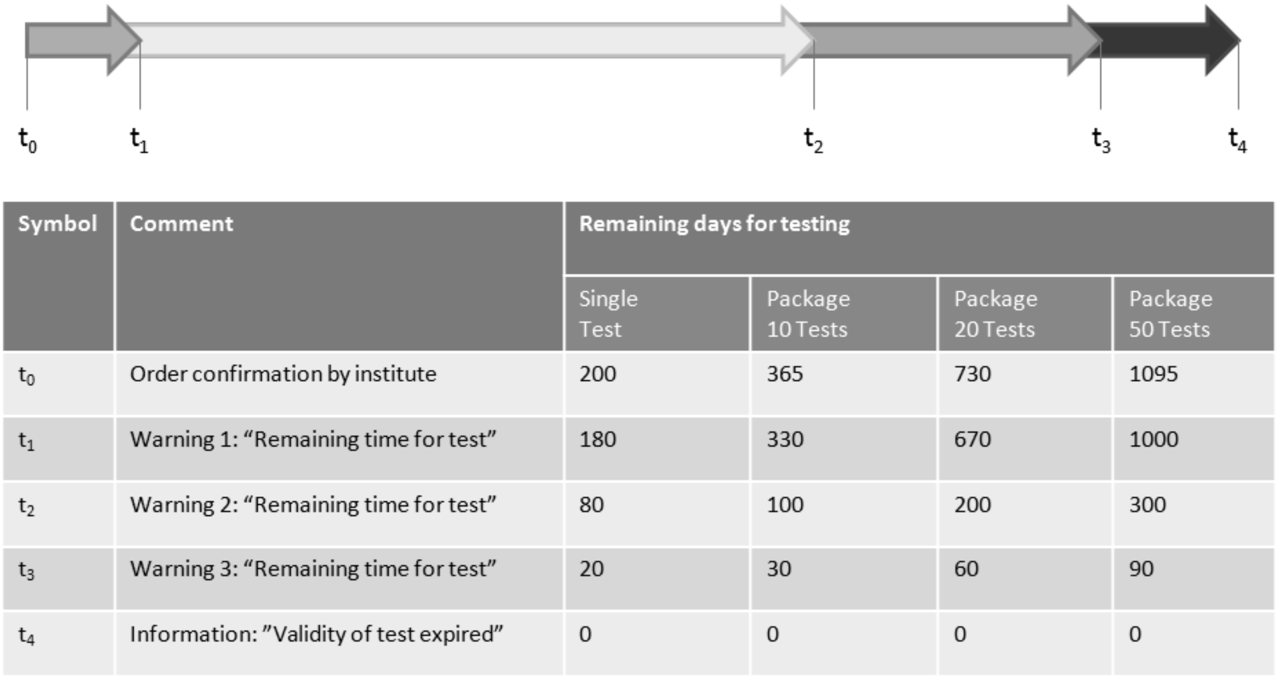


Figure 5. PTB cost model for testing of Chebyshev algorithms


t_0
t_1
t_2
t_3
t_4

Symbol	Comment	Remaining days for testing			
		Single Test	Package 10 Tests	Package 20 Tests	Package 50 Tests
t_0	Order confirmation by institute	200	365	730	1095
t_1	Warning 1: "Remaining time for test"	180	330	670	1000
t_2	Warning 2: "Remaining time for test"	80	100	200	300
t_3	Warning 3: "Remaining time for test"	20	30	60	90
t_4	Information: "Validity of test expired"	0	0	0	0

Figure 6. Outline on regular TraCIM system information on remaining days for using TraCIM service before expiration of order key

5.8. Availability of the Service

Order keys for single tests and test packages have time-limited validity for testing which is shown in Figure 6. A single test may only be performed once. For additional tests a service user must purchase new order keys. In opposite order keys for packages are applicable multiple times according to the amount of tests within the package. If all tests are consumed the order key is not valid anymore.

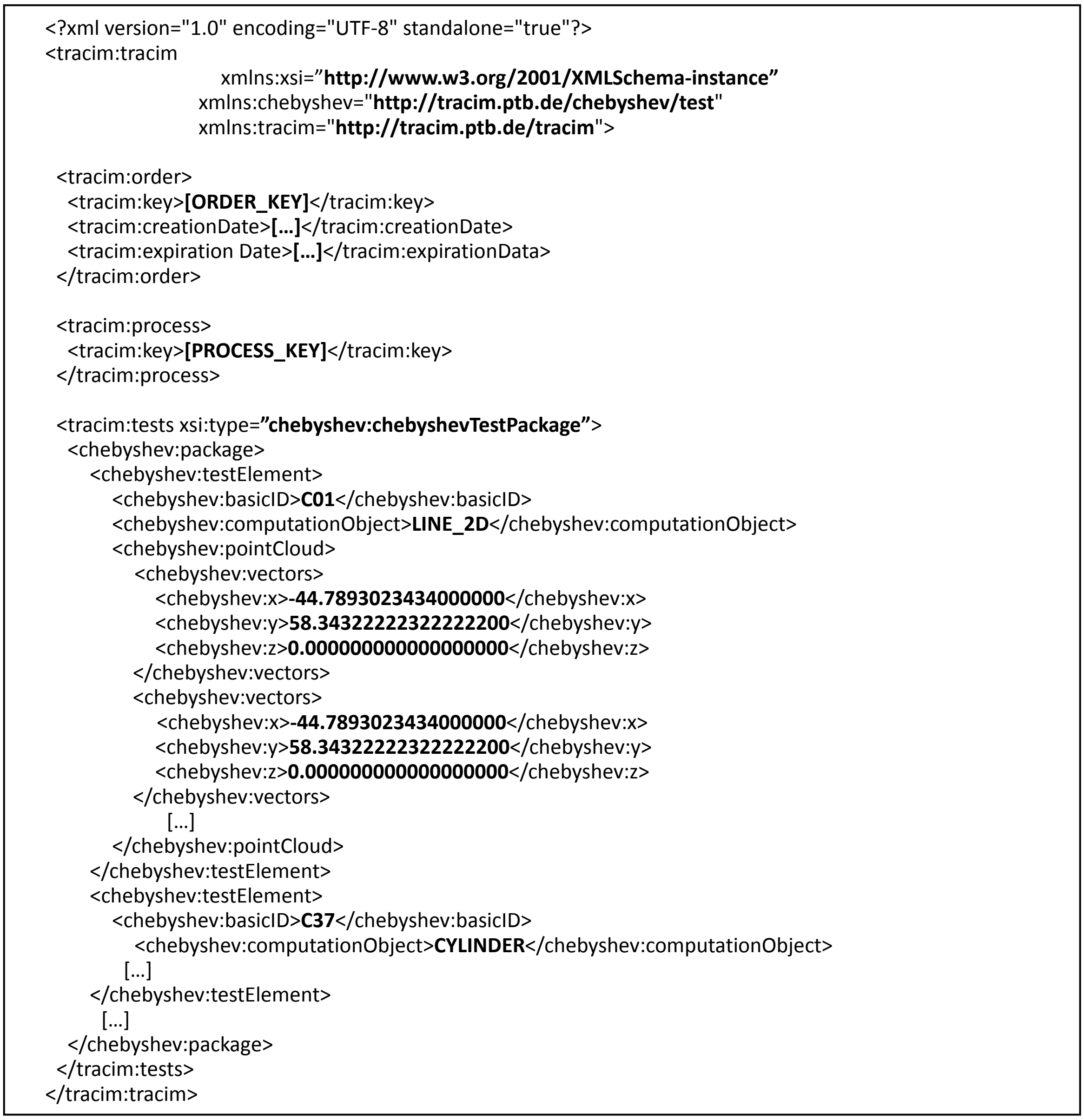
The TraCIM system regularly informs a service user on the remaining time for performing tests by e-mail. An initial message is sent when a service user gets the order key at day t_0. Interim messages follow giving information on the remaining time for testing and in case of test packages the remaining number of available test runs. If the service user does not test his software by the expiration date a final message is send that the order key will lose its validity at day t_4.

5.9. Test Data

5.9.1. General XML Data Structure

Test data denotes the input values for algorithms to be tested, which are automatically transmitted by the TraCIM system to the service user. For PTB’s Chebyshev test this comprises 50 different points clouds. The number of points per data set varies between 10 and 631. Figure 7 shows the general XML data structure of the test data.

The test data that is returned by the TraCIM system is composed according to an XML scheme with three major elements for order identification, process identification and the test data sets. The order element contains a copy of the order key, a date of the creation of the test data and a date for the expiration of the test data (deadline for sending test results for evaluation).



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<tracim:tracim
                  xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
                xmlns:chebyshev="http://tracim.ptb.de/chebyshev/test"
                xmlns:tracim="http://tracim.ptb.de/tracim">

 <tracim:order>
  <tracim:key>[ORDER_KEY]</tracim:key>
  <tracim:creationDate>[…]</tracim:creationDate>
  <tracim:expiration Date>[…]</tracim:expirationData>
 </tracim:order>

 <tracim:process>
  <tracim:key>[PROCESS_KEY]</tracim:key>
 </tracim:process>

 <tracim:tests xsi:type=”chebyshev:chebyshevTestPackage”>
  <chebyshev:package>
    <chebyshev:testElement>
      <chebyshev:basicID>C01</chebyshev:basicID>
      <chebyshev:computationObject>LINE_2D</chebyshev:computationObject>
      <chebyshev:pointCloud>
        <chebyshev:vectors>
          <chebyshev:x>-44.7893023434000000</chebyshev:x>
          <chebyshev:y>58.34322222322222200</chebyshev:y>
          <chebyshev:z>0.000000000000000000</chebyshev:z>
        </chebyshev:vectors>
        <chebyshev:vectors>
          <chebyshev:x>-44.7893023434000000</chebyshev:x>
          <chebyshev:y>58.34322222322222200</chebyshev:y>
          <chebyshev:z>0.000000000000000000</chebyshev:z>
        </chebyshev:vectors>
             […]
      </chebyshev:pointCloud>
    </chebyshev:testElement>
    <chebyshev:testElement>
      <chebyshev:basicID>C37</chebyshev:basicID>
        <chebyshev:computationObject>CYLINDER</chebyshev:computationObject>
       […]
    </chebyshev:testElement>
     […]
  </chebyshev:package>
 </tracim:tests>
</tracim:tracim>
```

Figure 7. General XML Data Structure for Test Data

The process element contains the process key associated with the test data request. The test element is of type ‘chebyshevTestPackage’.It contains a list with fifty different ‘testElement’ entries. Each of these elements gives a unique data set ID (‘basicID’), the point cloud and the geometrical element that must be fitted to the point cloud (‘computationObject’). Table 1 provides a list of all entries for the data set ID element and the computation object element.

Table 1. Data Set IDs and Associated Geometric Elements

basicID	computationObject
C01 – C08	LINE_2D
C09 – C16	CIRCLE_2D
C17 – C26	PLANE
C27 – C36	SPHERE
C37 – C50	CYLINDER

There are eight data sets for testing Chebyshev 2D line, eight data sets for Chebyshev 2D circle, ten data sets for Chebyshev plane, ten data sets for Chebyshev sphere and fourteen data sets for Chebyshev cylinder. The points of a point cloud are given by ‘vector’ elements. Each vector gives the x, y and z coordinate values of one point. Properties of the values are

- Each point coordinate value is delivered as decimal number with 20 digits and scientific e-format. The values refer to the unit mm (millimeter).
- All point coordinates are within the value range [-500 mm, +500 mm].
- For LINE_2D and CIRCLE_2D the z-component of each point is set zero (x, y, 0) as these are 2D elements.

Note, that the restriction to the unit mm and value range is subject to future change. Newer versions of the TraCIM test may accept customer defined unit and range.

The test data sets are stored within a database that is connected with PTB’s expert module for testing Chebyshev algorithms. Whenever a service user requests test data fifty sets of test data are randomly selected from that data base and send to the service user.

5.9.2. Test Data Design

Point clouds for Chebyshev algorithm testing were created with a generator software which was developed at University of Applied Science Zwickau (https://www.fh-zwickau.de). The tests cover data sets with points that are both randomly and systematically distributed on fragments of the geometrical elements mainly representing full features with different random and systematic form deviation components. A few datasets represent partial features. The data sets are designed to have only one unique solution when calculating the associated geometrical element with the MZ criterion according to formula (P). The functionality of the data generator for the Chebyshev test is shown in Figure 8.

Input for the test data creation are values for geometrical parameters of the Chebyshev elements. In the first step the generator software calculates a point cloud from these input parameters. Basic algorithmically approaches for the creation are presented in [14] and [15].

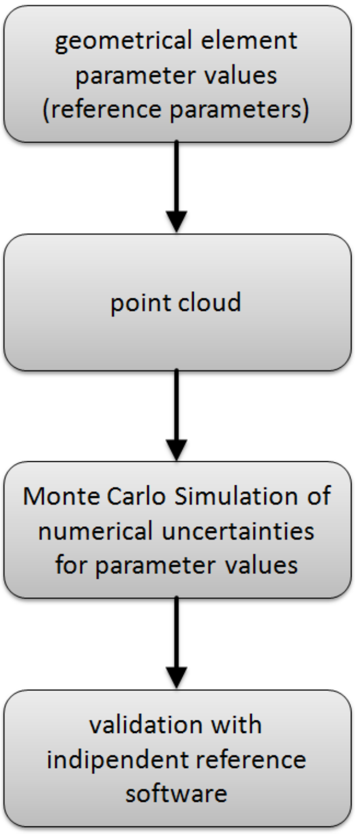

geometrical element parameter values (reference parameters)
point cloud
Monte Carlo Simulation of numerical uncertainties for parameter values
validation with indipendent reference software

Figure 8. Outline of Chebyshev Test Data generation procedure

The point data set has the property that the preset parameter values give a sufficiently accurate reference solution for the fitting problem (P) with the generated point cloud. Hence the input parameters are used as reference parameters for PTBs’ Chebyshev algorithm test. The second step of the generator procedure evaluates numerical uncertainties of the reference parameter values. Here an Monte Carlo Simulation method is applied. It evaluates the change in parameter values for the solution of (P) due to small perturbation within the input point cloud. Thereby a random noise with relative magnitude $1.e^{-15}$ is added to each coordinate value of all points in a data set. Later the reference parameters and their associated uncertainties are used at the test evaluation. Finally the last step of the generator software is a validation of parameter values and point clouds. The accuracy of the reference result is simply checked by comparison with parameter values calculated by a reference software for the Chebyshev element fitting.

5.10. User Results

For submitting computation results, which have been obtained by use of the software under test, to the TraCIM system the data structure in Figure 9 applies.

The root element of the XML structured content is the ‘chebyshev Result Package’. It contains information on the software under test, a ‘resultPackage’ element with the geometry parameter values calculated by this software and MPE (maximum permissible error) values for the test of the accuracy of the software results. A user has to give the following information on the test process and tested software:

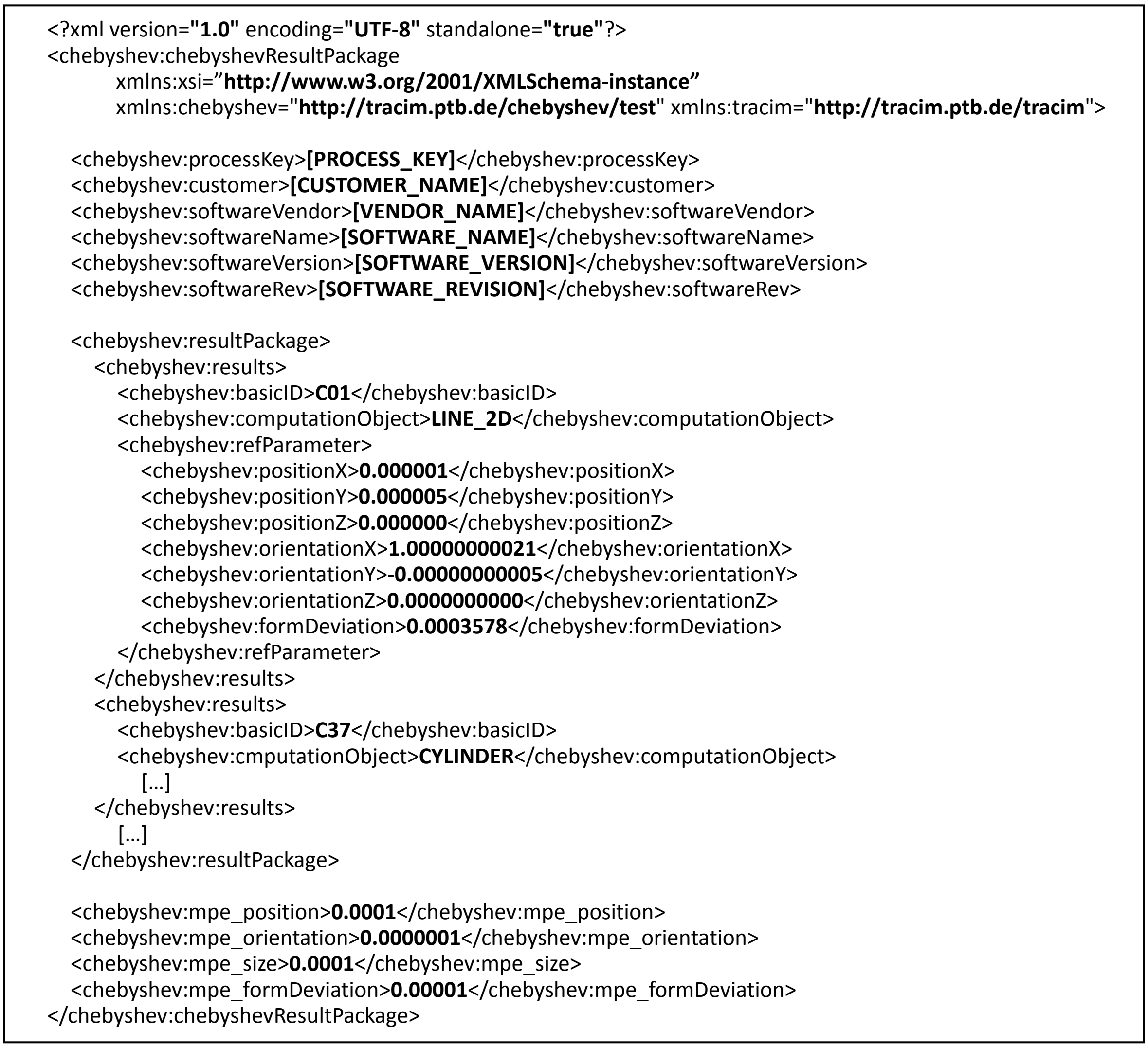
- [PROCESS_KEY]: the key that was send with the test data (Fig. 7)
- [CUSTOMER_NAME]: user name from TraCIM system registration
- [VENDOR_NAME]: name of the software vendor
- [SOFTWARE_NAME]: name of software that computed the test results
- [SOFTWARE_VERSION]: version of software that computed the test results
- [SOFTWARE_REVISION] revision of software that computed the test results

The ‘resultPackage’ element must contain a list of fifty ‘results’ elements. For each point cloud of the test data there has to be one ‘result’. The different elements are identified by the basic IDs and computational objects according to Table 1. The parameter values of the tested software are written to the ‘refParameter’ element. Table 2 shows which parameter values a user has to write into the XML structure for each geometrical element belonging to the test.

Table 2. Specification of User Result Parameters

XML Element	**Geometrical Parameter**[a]				
	L2	*C2*	*P*	*S*	*C*
positionX	C_x	C_x	C_x	C_x	C_x
positionY	C_y	C_y	C_y	C_y	C_y
positionZ	0	0	C_z	C_z	C_z
orientationX	v_x	$-$	n_x	$-$	v_x
orientationY	v_y	$-$	n_y	$-$	v_y
orientationZ	0	$-$	n_z	$-$	v_z
radius	$-$	r	$-$	r	r
formDeviation	ef	ef	ef	ef	ef

a. abbreviations: L2 – LINE_2D, C2-CIRCLE_2D, P-PLANE, S-SPHERE, C-Cylinder)



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<chebyshev:chebyshevResultPackage
        xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
        xmlns:chebyshev="http://tracim.ptb.de/chebyshev/test" xmlns:tracim="http://tracim.ptb.de/tracim">

  <chebyshev:processKey>[PROCESS_KEY]</chebyshev:processKey>
  <chebyshev:customer>[CUSTOMER_NAME]</chebyshev:customer>
  <chebyshev:softwareVendor>[VENDOR_NAME]</chebyshev:softwareVendor>
  <chebyshev:softwareName>[SOFTWARE_NAME]</chebyshev:softwareName>
  <chebyshev:softwareVersion>[SOFTWARE_VERSION]</chebyshev:softwareVersion>
  <chebyshev:softwareRev>[SOFTWARE_REVISION]</chebyshev:softwareRev>

  <chebyshev:resultPackage>
    <chebyshev:results>
      <chebyshev:basicID>C01</chebyshev:basicID>
      <chebyshev:computationObject>LINE_2D</chebyshev:computationObject>
      <chebyshev:refParameter>
        <chebyshev:positionX>0.000001</chebyshev:positionX>
        <chebyshev:positionY>0.000005</chebyshev:positionY>
        <chebyshev:positionZ>0.000000</chebyshev:positionZ>
        <chebyshev:orientationX>1.00000000021</chebyshev:orientationX>
        <chebyshev:orientationY>-0.00000000005</chebyshev:orientationY>
        <chebyshev:orientationZ>0.0000000000</chebyshev:orientationZ>
        <chebyshev:formDeviation>0.0003578</chebyshev:formDeviation>
      </chebyshev:refParameter>
    </chebyshev:results>
    <chebyshev:results>
      <chebyshev:basicID>C37</chebyshev:basicID>
      <chebyshev:cmputationObject>CYLINDER</chebyshev:computationObject>
        […]
    </chebyshev:results>
      […]
  </chebyshev:resultPackage>

  <chebyshev:mpe_position>0.0001</chebyshev:mpe_position>
  <chebyshev:mpe_orientation>0.0000001</chebyshev:mpe_orientation>
  <chebyshev:mpe_size>0.0001</chebyshev:mpe_size>
  <chebyshev:mpe_formDeviation>0.00001</chebyshev:mpe_formDeviation>
</chebyshev:chebyshevResultPackage>
```

Figure 9. General XML Data Structure for submitting User Results

The geometrical element parameters are those from section 2. In case of 2D elements line and circle the z-component of the position parameter has to be zero.

Moreover the user must write zero for the z-component of the orientation in case of the 2D line. For 2D circle and sphere the orientation elements must not be defined in the XML data structure. Similar the radius element must not be written for 2D line and plane.

Further the values provided by the user have to fulfil the following requirements:

- All parameter values must be in decimal number format with the maximum amount of decimal digits the tested software provides (fixed point or floating point e-format are applicable).
- Only a dot is allowed as separator of decimal places for all values.
- The usage of a comma is forbidden within any decimal value.
- The values for position, radius and form deviation refer to the unit mm (millimeter).
- Radius and form deviation must be positive values.

Finally the client has to specify the MPE values for position, orientation, size and form deviation that should be used for test evaluation. User MPE values are subjected to unit millimetre for length and rad for orientation. The default values for PTBs‘ Chebyshev test are presented in Table 3. User specified values for the MPEs must be greater or equal to the default values. Details on the application of the MPE values at test evaluation are presented in the following section.

Table 3. Default MPE Values for Chebyshev Test

MPE Element	Default Value	
mpe_position	0.0001 mm	(0.1 µm)
mpe_orientation	0.0000001 rad	(0.1 µrad)
mpe_size	0.0001 mm	(0.1 µm)
mpe_formDeviation	0.00001 mm	(0.01 µm)

All values that are entered into the schema must meet the previous specifications. The XML schemata for the Chebyshev test is downloadable from the following URLs:

Test data schema:
https://tracim.ptb.de/tracim/api/schema/PTBWHZ_MATH_CHEBYSHEV_v1_test.xsd;
User result schema:
https://tracim.ptb.de/tracim/api/schema/PTBWHZ_MATH_CHEBYSHEV_v1_result.xsd;
Report schema:
https://tracim.ptb.de/tracim/api/schema/tracim.xsd.

Service users are advised to utilize these schemata during client development in order to check a proper functioning with the client.

5.11. Test Evaluation

Accessing the accuracy of software is carried out by comparison of calculated user *parameter* values with the reference results of PTB for each geometrical element. Thereby deviations between user results and reference results are computed by four different types of test parameters. These are

- ΔC Deviation of position
- Δv Deviation of orientation
- Δr Deviation of radius
- Δef Deviation of form deviation

The calculation of ΔC is individual for each geometrical element. In the case of 2D line the position deviation is the orthogonal distance between the line position of the user and the reference line. For circle and sphere it is the distance of user center point and reference center point. For element plane the test parameter computes as the orthogonal distance of the user plane position and the reference plane. Finally at evaluation of cylinder results the position deviation is the orthogonal distance between the user position of the cylinder axis and the reference cylinder axis.

The test parameter for the orientation deviation is applied for elements 2D line, plane and cylinder. It calculates the sine of the angle between user orientation vector and reference orientation vector. The orientation vectors are the direction vector of 2D line and cylinder axis as well as the normal vector for element plane.

For elements 2D circle, sphere and cylinder the radius deviation is the amount of the difference between user radius value and reference radius value.

For all elements Δef is the amount of the difference between user form deviation value and reference form deviation value.

For test evaluation these test parameters are compared with the user specified MPE bounds for maximum permissible deviations. The evaluation takes the numerical uncertainties of the reference parameter values into account. Equations (7) – (8) show the principal test criteria.

$$\Delta C \le mpe\_position + U_C \tag{7}$$

$$\Delta v \le mpe\_orientation + U_v \tag{8}$$

$$\Delta r \le mpe\_size + U_r \tag{9}$$

$$\Delta ef \le mpe\_formDeviation + U_{ef} \tag{10}$$

U_C, U_v, U_r, U_{ef} numerical uncertainties

U_C is the numerical uncertainty for the position deviation ΔC due to the uncertainty of the reference position parameter value. U_v is the numerical uncertainty for the orientation deviation Δv due to the uncertainty of the reference orientation parameter value. U_r is the numerical uncertainty of the reference radius value and U_{ef} is the numerical uncertainty of the reference form deviation value. All uncertainty values are separately calculated for each of the 50 tested elements by a Monte Carlo simulation with the test data and reference results. The values represent extended measurement uncertainties with covering factor $k = 2$ [16].

A customer passes the Chebyshev algorithm test if for each of the fifty test data sets conditions (7) – (10) hold.

If for any user result one of the previous defined conditions is not satisfied then the test report will give a table with individual evaluation results for each test parameter and each test data set.

6. Public Test Data

Public test data are available for checking the client-server communication. They comprise a full test with fifty data set in order to detect and correct software errors within the client application. For any registered customer the public data are free of charge. In comparison to a test with purchased test data the report issued to the service user is marked as draft.

All registered customers can get an order key for public data on demand. Please consult the TraCIM website http://tracim.ptb.de.

For each geometrical element one public test data set is presented in tables 4 – 8 in the appendix.

7. Summary

Testing of evaluation software is an indispensable and cost-efficient supplement to the conventional calibration of CMMs. In the long run it will improve the quality of calculation results for many standard applications within the area of product inspection by ISO 1101 [17] and ASME Y14.5 [18]. The result is an increased trust in measurements values and better comparability of different software solutions.

Tests for Chebyshev geometrical element software are strongly required by industry. These elements are the basis for work piece form assessment and position tolerance inspection with a CMM. By the development of the Chebyshev test for elements 2D line, 2D circle, plane, sphere and cylinder subjected to the MZ criterion PTB established a fundamental validation capability for five key algorithms of evaluation software. The test with its fifty individual test data sets covers a large area of cases with high practical relevance in industrial applications.

By using PTBs TraCIM service for Chebyshev tests manufacturers and users of geometrical element fitting software could easily validate their algorithms and hence provide correct calculation for different measurands of work piece features. An automated client-server system allows to perform a whole test with minimum afford. The test service is of high quality according to the strict rules which are postulated by TraCIM association. These comprise the clear description of evaluation procedures, the correctness of the reference results as well as the competence of the responsible and liable NMI.

ACKNOWLEDGEMENTS

This work has been undertaken as part of the EMRP Joint Research Project "NEW06-Traceability for computationally-intensive metrology (TraCIM)". The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union.

The authors thank Thomas Fricke and David Witte of PTB for the implementation of the TraCIM client server system with high professional skills. Further thanks go to Professor Bernd Müller for his major work in developing the TraCIM server application. Finally we thank our colleague Matthias Franke for the support at running the TraCIM system and Manfred Gahrens for his work in handling legal and business aspects.

Appendix – Public Test Data Sets

Table 4. Public Test Data Set for Cylinder C50

Reference Parameter

C_x	50.0
C_y	-50.0
C_z	50.0
v_x	-0.9630868246861
v_y	-0.1203858530857
v_z	0.2407717061715
r	1.65
ef	0.002

Point Cloud

x_i	y_i	z_i
51.716214493664440	-50.721387046525980	48.195819502413924
51.489768757120120	-48.796975230998310	48.252430936550000
47.636335959057640	-49.277421094037670	49.214121304805264
51.525445940887066	-49.759265967772250	47.916826667804550
48.474189176320806	-50.240779642576776	52.083264552893490
48.381040095947490	-51.799535022319290	50.930628757956620
50.400211898614020	-49.999992257448690	51.600584347711380
52.326654174252744	-49.759194818215044	51.119240516322384
48.284077915426600	-49.278311468241810	51.804496659765240
50.210901317343390	-49.036880799454030	51.323639680870210
52.136753967350470	-48.795896617149710	50.842547662030830
48.014071547417290	-48.682408641716990	51.022237264708130
49.940258813699610	-48.442789142239190	50.540303003418550
51.866295765818480	-48.201684079864825	50.058911432771275
47.767082864319820	-48.682215818993686	50.032428155761664
49.693242418492240	-48.442375806471370	49.551194460169440
51.619270816424985	-48.201412681715620	49.069618355200560
49.563314989848645	-49.037025491146110	48.733021339655360
47.673392336817730	-50.240807398457690	48.881020894832280
49.599877808145360	-49.999985607948050	48.399021902815115
47.863346035244720	-51.203333392756590	49.158540307845960
49.789622915938720	-50.962696148278270	48.676754565172330
48.133119286490700	-51.799182080212680	49.940968526795616
50.059829798104014	-51.557842339513860	49.459459093659090
51.985614929442940	-51.318015998262080	48.977711694751840
50.306815842085066	-51.558541487362580	50.449095404315976
52.233268251152425	-51.316187713928665	49.966972857606320
48.510800188328240	-51.203983616694310	51.748819754540924
50.436525935792280	-50.962449675659600	51.266281752152955
52.363133627487420	-50.722282255824230	50.785520441951390

Table 5. Public Test Data Set for Plane C21

<table>
<tr><th colspan="3">Reference Parameter</th></tr>
<tr><td>C_x</td><td colspan="2">310.0</td></tr>
<tr><td>C_y</td><td colspan="2">-95.0</td></tr>
<tr><td>C_z</td><td colspan="2">210.0</td></tr>
<tr><td>n_x</td><td colspan="2">0.7071067811865</td></tr>
<tr><td>n_y</td><td colspan="2">0.0</td></tr>
<tr><td>n_z</td><td colspan="2">0.7071067811865</td></tr>
<tr><td>ef</td><td colspan="2">0.003</td></tr>
<tr><th colspan="3">Point Cloud</th></tr>
<tr><td>x_i</td><td>y_i</td><td>z_i</td></tr>
<tr><td>289.62419312562590</td><td>-104.50369478453633</td><td>230.37792819471770</td></tr>
<tr><td>322.48678825131590</td><td>-76.623813907553860</td><td>197.51533306902763</td></tr>
<tr><td>318.70402942056080</td><td>-116.62387645001311</td><td>201.29809189978273</td></tr>
<tr><td>307.29945055671076</td><td>-96.624108805150450</td><td>212.69842812294568</td></tr>
<tr><td>297.61307054686660</td><td>-92.904194495931460</td><td>222.38630466691914</td></tr>
<tr><td>317.81926558176715</td><td>-79.424317164765950</td><td>202.18174978640926</td></tr>
<tr><td>308.83776249998680</td><td>-96.644032317338230</td><td>211.16057432229648</td></tr>
<tr><td>316.01567757523186</td><td>-82.543956480257250</td><td>203.98483473041676</td></tr>
<tr><td>315.60885882520900</td><td>-92.044127387619600</td><td>204.39055196969266</td></tr>
<tr><td>305.99237882269120</td><td>-112.06387820683375</td><td>214.00767844375776</td></tr>
</table>

Table 6. Public Test Data Set for Sphere C29

<table>
<tr><th colspan="3">Reference Parameter</th></tr>
<tr><td>C_x</td><td colspan="2">150.0</td></tr>
<tr><td>C_y</td><td colspan="2">-120.0</td></tr>
<tr><td>C_z</td><td colspan="2">-90.0</td></tr>
<tr><td>r</td><td colspan="2">25.0</td></tr>
<tr><td>ef</td><td colspan="2">0.005</td></tr>
<tr><th colspan="3">Point Cloud</th></tr>
<tr><td>x_i</td><td>y_i</td><td>z_i</td></tr>
<tr><td>130.359505460562240</td><td>-115.28904904680536</td><td>-75.26287268111895</td></tr>
<tr><td>171.54584085113347</td><td>-127.80387993037534</td><td>-79.99993968518623</td></tr>
<tr><td>145.17443794897067</td><td>-114.43999305633276</td><td>-66.11118923160807</td></tr>
<tr><td>147.36573274021578</td><td>-98.831121082982120</td><td>-76.95913317104015</td></tr>
<tr><td>162.02249857757944</td><td>-101.16353490274310</td><td>-101.2036653306819</td></tr>
<tr><td>145.96277458767368</td><td>-105.53571917795280</td><td>-70.01382784336302</td></tr>
<tr><td>155.64767849217975</td><td>-106.63626849484785</td><td>-69.64021485342760</td></tr>
<tr><td>136.89115781147748</td><td>-106.63517074106765</td><td>-73.42747079474103</td></tr>
<tr><td>163.92905283905554</td><td>-104.25404910923444</td><td>-76.46988116797096</td></tr>
<tr><td>155.64683789212780</td><td>-96.452695762561620</td><td>-83.79059101022951</td></tr>
<tr><td>142.38484543055271</td><td>-96.450166624914670</td><td>-86.46898143722326</td></tr>
<tr><td>131.91012461903523</td><td>-104.25309423461546</td><td>-82.93708465331273</td></tr>
<tr><td>138.64274346300018</td><td>-123.09272294633286</td><td>-67.94349270135353</td></tr>
<tr><td>151.90522754774805</td><td>-123.09273618889179</td><td>-65.26435428340416</td></tr>
<tr><td>162.37885358616683</td><td>-115.28951866632468</td><td>-68.79679909625150</td></tr>
<tr><td>166.33622156575242</td><td>-101.14209453118567</td><td>-88.37991766857577</td></tr>
<tr><td>134.31802815311597</td><td>-101.14473255322506</td><td>-94.84738760097490</td></tr>
</table>

130.57354504099472	-127.78600102071838	-76.32150038475277
162.59363446655356	-127.78592894029095	-69.85357542281248
173.73750692973042	-112.19631532082533	-90.85264053120103
143.26825214455297	-101.16136694704748	-104.9936095788746
128.45510924145576	-112.19646419211868	-99.99961934881653
126.26464441515686	-127.80297742642120	-89.14743674413596
137.97527903741583	-138.83994705878958	-78.79426364741960
156.73218303726230	-138.83985089870451	-75.00542114147109

Table 7. Public Test Data Set for 2D Circle C15

Reference Parameter

C_x	100.0
C_y	120.0
r	25.0
ef	0.001

Point Cloud

x_i	y_i
75.784936680203656	113.782628475935042
110.430609225187300	142.719537659059826
103.211171530292466	95.207594355265980
124.175520694804758	126.369395527447396
78.514279506896740	132.780713309044765
76.814598322861626	129.349871591809554
110.262955537163500	142.796458626203095
75.717112874512044	114.053635613619338
124.763984297964226	123.430513019579665
118.667540687125260	103.371271870198016
75.388731830050250	124.390361426207232
124.627730609605197	124.295540916993733
93.628351748348526	144.174710578764226
78.162942522723800	107.828935713726168
87.689885999027836	141.759725380016283
113.989460694132405	113.989460694132405
81.056313173318907	103.686049998788790
119.320561167715920	104.134815685973270
112.203528886040510	141.818899108820810
113.244502923135984	141.202787567628823
123.565215185593433	111.653226667273121
111.381076341333965	97.740904706500990
98.442714147673559	95.048354847219165
93.487593162102000	95.863310371194316
113.805516219046968	140.842335254469730
111.943542365529594	141.962719008154520
118.300088911844967	102.967048854175275
101.128113355363332	95.025197609602167
90.744017071011080	96.776233859819100
80.069049001074010	135.092185924396264

Table 8. Public Test Data Set for 2D Line C05

Reference Parameter	
C_x	-95.0
C_y	-103.0
v_x	0.8944271909999
v_y	0.447235954999
ef	0.002

Point Cloud	
x_i	y_i
-93.0363622787920	-102.017063105407
-98.0780341149522	-104.540135091464
-90.6574378801770	-100.829836974077
-97.0650500176381	-104.032368347807
-98.2314436701957	-104.614878594266
-91.7810693184072	-101.391151879591
-96.8472276844060	-103.922760586615
-91.8929436437996	-101.445868951205
-96.0914986483551	-103.546249029032
-96.3192505810570	-103.659051764970

REFERENCES

[1] DIN EN ISO 10360 part 6, "Geometrical Product Specifications (GPS) – Acceptance and reverification tests for coordinate measuring machines (CMM) – Part 6: Estimation of errors in computing Gaussian associated features (ISO 10360-6:2001+Cor. 1:2007)," German version EN ISO 10360-6:2001+AC, 2007.

[2] VDI/VDE 2617-Blatt 2.3, "Genauigkeit von Koordinatenmessgeräten Kenngrößen und deren Prüfung – Annahme- und Bestätigungsprüfung vonKoordinatenmessgeräten großer Bauart," 2009.

[3] European Commission, "Testing of three coordinate measuring machine algorithms Phase II," BCR Report 13417 EN, 1991.

[4] TraCIM (Traceability for Computationally Intensive Metrology), EMRP JRP NEW06-TraCIM, *http://www.tracim.eu/* (accessed: April 2015).

[5] TraCIM PTB, Homepage of TraCIM Service at PTB, *http://tracim.ptb.de/* (accessed: June 2015).

[6] W. F. Demjanov, W. N. Malozemov, "Einführung in Minimax-Probleme," 1. Auflage, Akademische Verlagsgesellschaft Geest & Portig K.-G., Leipzig, 1975.

[7] C. Großmann, H. Kleinmichel, K. Vetters, "Minimaxprobleme und nichtlineare Optimierung," Physica-Verlag, Würzburg, Zeitschrift für Operations Research, Band 20, pp 23-38, 1976.

[8] C. Geiger, C. Kanzow, "Theorie und Numerik restringierter Optimierungsaufgaben," Springer, 2002.

[9] G.T. Anthony, B. Bittner, R. Drieschner, J. Kok, et. al., "Chebyshev reference software for the evaluation of coordinate measuring machine data," Commission of the European Communities, Programme for Applied Metrology and Chemical Analysis (BCR), Report EUR 15304 EN, October 1993.

[10] G.T. Anthony, H.M. Anthony, B. Bittner, et. Al, "Reference software for finding Chebyshev best-fit geometric elements," Precision Engineering 19, pp 28-36, 1996.

[11] R. Drieschner, "Chebyshev approximation to data by geometric elements," Numerical Algorithms 5, pp 509-522, 1993.

[12] W. Lotze, U. Lunze, "Tolerance-fit of Geometric Elements and Sculptured Surfaces and Profiles," XVI IMEKO World Congress, Vienna, AUSTRIA, 2000.

[13] A. Gläser, C. Grossmann, U. Lunze, "The Finite Termination Property of an Algorithm for Solving the Minimum Circumscribed Ball Problem," Schedae Informaticae, 2012.

[14] M.G. Cox, M.P. Dainton, A.B. Forbes, P.M. Harris, "Validation of CMM form and tolerance assessment software," Laser Metr. a. Mach. Performance V, G. N. Peggs, ed., WIT Press, Southampton, pp 367-376, 2001.

[15] A.B. Forbes, H.D. Minh, "Generation of numerical artefacts for geometric form and tolerance assessment," Int. J. Metrol. Qual. Eng. 3, pp 145-150, 2012.

[16] JCGM, "Evaluation of measurement data – Guide to the expression of uncertainty in measurement (GUM)," JCGM 100:2008, First edition, September 2008.

[17] DIN EN ISO 1101, "Geometrical Product Specification (GPS) – Geometrical tolerancing – Tolerances of form, orientation, location and run-out (ISO 1101:2004)," German version EN ISO 1101:2005; August 2008.

[18] ASME Y14.5, "Dimensioning and Tolerancing; Engineering Drawing and Related Documentation Practices," The American Society of Mechanical Engineering, 2009.