

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICS, BIOMECHANICS
AND MECHATRONICS



Master's thesis

Analysis of hip joint cup wear

Bc. Jan Mervart

Supervisor: Prof. RNDr. Matej Daniel, Ph.D.

7th August 2016

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mervart** Jméno: **Jan** Osobní číslo: **382298**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Strojní inženýrství**
Studijní obor: **Biomechanika a lékařské přístroje**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Analýza opotřebenění jamek kyčelního kloubu

Název diplomové práce anglicky:

Analysis of hip joint cup wear

Pokyny pro vypracování:

1. Rešerše literatury
2. Vývoj metody pro měření explantovaných acetabulárních komponent
3. Bezkontaktní měření povrchu jamek
4. Vývoj algoritmu pro určení opotřebenění
5. Pilotní studie

Seznam doporučené literatury:

- 1: Košak R, Kralj-Iglič V, Iglič A, Daniel M. Polyethylene wear is related to patient-specific contact stress in THA. Clin Orthop Relat Res. 2011 Dec;469(12):3415-22. doi: 10.1007/s11999-011-2078-5.
- 2: Allepuz A, Havelin L, Barber T, Sedrakyan A, Graves S, Bordini B, Hoeffel D, Cafri G, Paxton E. Effect of femoral head size on metal-on-HXLPE hip arthroplasty outcome in a combined analysis of six national and regional registries. J Bone Joint Surg Am. 2014 Dec 17;96 Suppl 1:12-8.
- 3: Ivan Landor a kol. Revizní operace totálních náhrad kyčelního kloubu, Maxdorf, 2012

Jméno a pracoviště vedoucí(ho) diplomové práce:

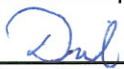
prof. RNDr. Matej Daniel, Ph.D.


Jméno a pracoviště konzultanta(ky) diplomové práce:

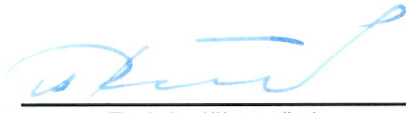
Datum zadání diplomové práce: **12.04.2016**

Termín odevzdání diplomové práce: **12.08.2016**

Platnost zadání diplomové práce: _____


Podpis vedoucí(ho) práce


Podpis vedoucí(ho) ústavu/katedry


Podpis děkana(ky)

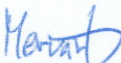
Neodevzdá-li student bakalářskou nebo diplomovou práci v určeném termínu, tuto skutečnost předem písemně zdůvodnil a omluva byla děkanem uznána, stanoví děkan studentovi náhradní termín odevzdání bakalářské nebo diplomové práce. Pokud se však student řádně neomluvil nebo omluva nebyla děkanem uznána, může si student zapsat bakalářskou nebo diplomovou práci podruhé

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

III. PŘEVZETÍ ZADÁNÍ

25.4.2016

Datum převzetí zadání



Podpis studenta(ky)

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Prof. RNDr. Matej Daniel, Ph.D., who has patiently supported me throughout my work on this thesis and provided me with his knowledge.

My co-advisor, Ing. Vlastimil Králík, Ph.D., has been there for me whenever I encountered an obstacle or needed his advice. I am deeply grateful to him for the long discussions that helped me sort out the technical details of my thesis. In addition, I am grateful for his assistance regarding the data measuring in the Laboratory of Biotribology, CTU in Prague.

Finally, I thank my family for supporting me throughout all my studies at Czech Technical University in Prague.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on 7th August 2016

.....


Czech Technical University in Prague
Faculty of Mechanical Engineering

© 2016 Jan Mervart. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Mechanical Engineering. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Mervart, Jan. *Analysis of hip joint cup wear*. Master's thesis. Czech Technical University in Prague, Faculty of Mechanical Engineering, 2016.

Abstrakt

Tato diplomová práce se zabývá určováním otěru explantovaných acetabulárních komponent totálních náhrad kyčelního kloubu (jamek). Jejím cílem je vyvinutí metody, která umožní naskenování povrchu explantovaných jamek, a vyvinutí algoritmu, který z naskenovaných dat vyhodnotí otěr. V rámci pilotní studie bylo naskenováno několik explantovaných jamek a získaná data byla analyzována vyvinutým algoritmem. Algoritmus je interaktivně řízen pomocí GUI programu. Vyvinutá metoda určování otěru bude použita v následném výzkumu zaměřeném na prodloužení životnosti náhrad.

Klíčová slova Lineární otěr, Objemový otěr, Náhrada kyčelního kloubu, Analýza dat, Optický SMS, *Matlab GUI*, UHMWPE, Biotribologie, Artroplastika

Abstract

This thesis deals with wear estimation of explanted acetabular components of total hip joint replacements (cups). Its aim is to develop the method, which is capable of scanning the surface of explanted cups, and to develop the algorithm, which is capable of estimating the wear from scanned data. Within the pilot study, several explanted cups were scanned and reached data was analyzed by the developed algorithm. The algorithm is interactively controlled by GUI program. Developed method of wear estimation is going to be used in upcoming research that will focus on extending the lifetime of replacements.

Keywords Linear wear, Volumetric wear, Hip joint replacement, Data Analysis, Optical CMM, *Matlab GUI*, UHMWPE, Biotribology, Arthroplasty

Contents

| | |
|--|-----------|
| Introduction | 1 |
| 1 Wear estimation - State of the art | 3 |
| 1.1 <i>In vivo</i> methods | 3 |
| 1.2 <i>In vitro</i> methods | 7 |
| 2 Thesis aims | 9 |
| 3 Data measurement | 11 |
| 3.1 Surface 3D scanning | 11 |
| 3.2 Casting method | 14 |
| 4 Data analysis | 19 |
| 4.1 Sphere - Mathematical background | 19 |
| 4.1.1 Coordinate Systems | 19 |
| 4.1.2 Surface Area & Volume | 25 |
| 4.2 Calculation procedure | 26 |
| 4.2.1 Data uploading | 27 |
| 4.2.2 Data transformation | 28 |
| 4.2.3 Data interpolation | 32 |
| 4.2.4 Data smoothing | 34 |
| 4.2.5 Reference sphere estimation | 35 |
| 4.2.6 Positioning of measured surface against reference sphere | 43 |
| 4.2.7 Maximal measured angle α | 44 |
| 4.2.8 Measured area A | 45 |
| 4.2.9 Volumetric wear U | 46 |
| 4.2.10 Average linear wear u_{av} | 47 |
| 4.2.11 Visualization of Linear wear u | 48 |
| 4.2.12 Used <i>Matlab</i> functions | 53 |
| 5 Pilot study | 55 |
| 6 Results | 61 |

| | | |
|----------|---|-----------|
| 6.1 | Verification of casting method | 61 |
| 6.2 | Verification of algorithm for wear estimation | 63 |
| 6.3 | Results of pilot study | 67 |
| 7 | Discussion | 83 |
| | Conclusion | 87 |
| | Bibliography | 89 |
| A | Nomenclature | 91 |
| B | Acronyms | 93 |
| C | Contents of enclosed CD | 95 |

List of Figures

| | | |
|------|--|----|
| 0.1 | Total hip joint replacement (http://goo.gl/8G06eM) | 2 |
| 1.1 | RTG image of total hip joint replacement (http://goo.gl/IKlXeS) | 4 |
| 1.2 | Livermore & Dorr wear measurement technique [4] | 5 |
| 1.3 | Definition of variables for volumetric wear calculation [5] | 6 |
| 1.4 | Error of volumetric wear calculation | 6 |
| 3.1 | Optical 3D CMM by <i>RedLux Ltd.</i> (http://goo.gl/veqx5V) | 12 |
| 3.2 | Workspace of <i>RedLux Artificial Hip Profiler</i> [6] | 12 |
| 3.3 | Chromatically encoded confocal measurement method [6] | 13 |
| 3.4 | Output file of <i>RedLux Artificial Hip Profiler</i> - coordinates of point cloud | 13 |
| 3.5 | The need for cast | 14 |
| 4.1 | Position of the cup in the Cartesian coordinate system [7] | 20 |
| 4.2 | Map of Antarctica (http://goo.gl/Hxc5mM) | 21 |
| 4.3 | Definition of Spherical coordinates in Cartesian and Pole c.s. | 22 |
| 4.4 | Hemisphere visualized at various coordinate systems | 23 |
| 4.5 | Meridians and parallels [7] & Sphere element | 25 |
| 4.6 | Calculation Flow chart | 26 |
| 4.7 | Data uploading | 27 |
| 4.8 | Data transformation | 28 |
| 4.9 | Visualization of data transformation | 30 |
| 4.10 | Data interpolation | 32 |
| 4.11 | Data interpolation - $E \rightarrow F$ | 33 |
| 4.12 | Data smoothing | 34 |
| 4.13 | Data smoothing - $F \rightarrow G$ | 35 |
| 4.14 | Reference sphere estimation | 36 |
| 4.15 | Tolerance boundaries | 37 |
| 4.16 | Positioning of measured surface against reference sphere | 43 |
| 4.17 | Definition of Maximal measured angle α and Measured area A | 44 |
| 4.18 | Maximal measured angle α | 45 |
| 4.19 | Measured area A | 46 |
| 4.20 | Volumetric wear U | 47 |
| 4.21 | Average linear wear u_{av} | 48 |

| | | |
|------|--|----|
| 4.22 | Linear wear u | 49 |
| 4.23 | Visualization of Linear wear u - examples | 50 |
| 5.1 | Sample no. 1 - (a-b) Cup. (c) Cast. (d) <i>Matlab GUI</i> calculation settings | 56 |
| 5.2 | Sample no. 2 - (a-b) Cup. (c) Cast. (d) <i>Matlab GUI</i> calculation settings | 57 |
| 5.3 | Sample no. 3 - (a-b) Cup. (c) Cast. (d) <i>Matlab GUI</i> calculation settings | 58 |
| 5.4 | Sample no. 4 - (a-b) Cup. (c) Cast. (d) <i>Matlab GUI</i> calculation settings | 59 |
| 5.5 | Sample no. 5 - (a-b) Cup. (c) Cast. (d) <i>Matlab GUI</i> calculation settings | 60 |
| 6.1 | New unworn UHMWPE cup and its casts | 61 |
| 6.2 | Evaluation of volumetric change | 62 |
| 6.3 | Evaluation of surface reconstruction | 62 |
| 6.4 | Simulated ellipsoidal wear | 63 |
| 6.5 | Results of case a) $c=14.1\text{mm}$ | 64 |
| 6.6 | Results of case b) $c=16\text{mm}$ | 65 |
| 6.7 | Results of case c) $c=18\text{mm}$ | 66 |
| 6.8 | Sample no. 1 - Outputs table and wear map (top view) | 68 |
| 6.9 | Sample no. 1 - Wear map and wear isolines (side views) | 69 |
| 6.10 | Sample no. 1 - Sample surface inside tolerance boundaries B_{red} | 70 |
| 6.11 | Sample no. 2 - Outputs table and wear map (top view) | 71 |
| 6.12 | Sample no. 2 - Wear map and wear isolines (side views) | 72 |
| 6.13 | Sample no. 2 - Sample surface inside tolerance boundaries B_{red} | 73 |
| 6.14 | Sample no. 3 - Outputs table and wear map (top view) | 74 |
| 6.15 | Sample no. 3 - Wear map and wear isolines (side views) | 75 |
| 6.16 | Sample no. 3 - Sample surface inside tolerance boundaries B_{red} | 76 |
| 6.17 | Sample no. 4 - Outputs table and wear map (top view) | 77 |
| 6.18 | Sample no. 4 - Wear map and wear isolines (side views) | 78 |
| 6.19 | Sample no. 4 - Sample surface inside tolerance boundaries B_{red} | 79 |
| 6.20 | Sample no. 5 - Outputs table and wear map (top view) | 80 |
| 6.21 | Sample no. 5 - Wear map and wear isolines (side views) | 81 |
| 6.22 | Sample no. 5 - Sample surface inside tolerance boundaries B_{red} | 82 |

List of Tables

| | | |
|-----|---|----|
| 5.1 | Explanted cups for pilot study | 55 |
| 6.1 | Comparison between actual results and results calculated by the algorithm . . | 67 |
| 6.2 | Numerical results of pilot study | 67 |

List of Listings

| | | |
|------|--|----|
| 4.1 | <i>Cup_analysis.m</i> - Transformations between coordinate systems | 24 |
| 4.2 | <i>Cup_analysis.m</i> - Data uploading | 27 |
| 4.3 | <i>Cup_analysis.m</i> - Data transformation | 31 |
| 4.4 | <i>Cup_analysis.m</i> - Save of ' <i>B data</i> ' | 31 |
| 4.5 | <i>Cup_analysis.m</i> - Data interpolation - $B \rightarrow C$ | 32 |
| 4.6 | <i>Cup_analysis.m</i> - Data smoothing - $C \rightarrow D$ | 34 |
| 4.7 | <i>Cup_analysis.m</i> - Reference sphere estimation | 38 |
| 4.8 | <i>Cup_analysis.m</i> - Standby method of reference sphere estimation | 42 |
| 4.9 | <i>Cup_analysis.m</i> - Positioning of measured surface against reference sphere | 44 |
| 4.10 | <i>Cup_analysis.m</i> - Maximal measured angle α | 45 |
| 4.11 | <i>Cup_analysis.m</i> - Measured area A | 45 |
| 4.12 | <i>Cup_analysis.m</i> - Volumetric wear U | 47 |
| 4.13 | <i>Cup_analysis.m</i> - Average linear wear u_{av} | 48 |
| 4.14 | <i>Cup_analysis.m</i> - Scale of Linear wear u | 49 |
| 4.15 | <i>Cup_analysis.m</i> - Wear map | 51 |
| 4.16 | <i>Cup_analysis.m</i> - Wear isolines | 51 |

Introduction

Total hip joint replacements manage to improve the quality of patients' lives. However, their lifetime is not unlimited. In the years 2003 to 2009, there was performed 74 987 surgeries of hip joint replacement in the Czech Republic. Almost 19% of these surgeries were re-surgeries. Re-surgeries are expensive and dangerous for the patients. Therefore, it is worth to decrease the number of re-surgeries and extend the lifetime of total hip joint replacements. Currently, the lifetime of total hip joint replacements is approximately 10 to 15 years. Failure of total hip joint replacements can be caused by many reasons. For example, total hip joint replacement can fail because of biological reasons (such as infection), mechanical damage, etc. However, mentioned reasons cause only 27% of all failure cases. Wear is responsible for most of the failures. **Wear causes 73% of all cases of total hip joint replacements failure.** [1]

The failure caused by wear is a mixture of mechanical, biological and chemical issues. Wear particles in joint's neighbourhood cause body reaction which removes not only wear particles but adjacent bone tissue as well. Accordingly, hip joint replacement is slowly released from its stable position. After few years, the hip joint replacement is released as much as it does not fulfill its function anymore. The principle of the wear is not properly known so far. There is a large number of factors influencing wear. Surely, the friction between femoral head and acetabular component is a significant factor. However, the size of released wear particles is important as well because the body reacts differently to wear particles of various size. [2]

Combination of materials of femoral head and acetabular component is crucial factor of wear. There are two types of contacts called *Hard on hard* and *Hard on soft*. *Hard on hard* contact is formed by metal or ceramics. This type of contact is still used but because of its specific unwanted impacts, the number of its usage is quite low. In most of the cases *Hard on soft* contact is used. It is a contact between metal or ceramics (femoral head) and plastics (acetabular liner). UHMWPE or PEEK are the plastics which are used the most often. [3]

Current research is focused on material development which would decrease the wear growth. Accordingly, the lifetime of total hip joint replacements would be increased too. The Laboratory of Biomechanics, CTU in Prague, is involved in this type of research within project called *The study of new materials used as articulation surface of joint replacement (ID No. NV15-31269A)*. For purposes of this project, there is the need for wear quantification. As plastics liner is much softer than the femoral head, significant wear

can be observed just within acetabular liner in cases of *Hard on soft* contact. Therefore, we focus on wear quantification of acetabular components of hip joint replacements. We consider wear of femoral head as negligible within *Hard on soft* contact.

The aim of this thesis is wear quantification of explanted acetabular components of hip joint replacements (cups).

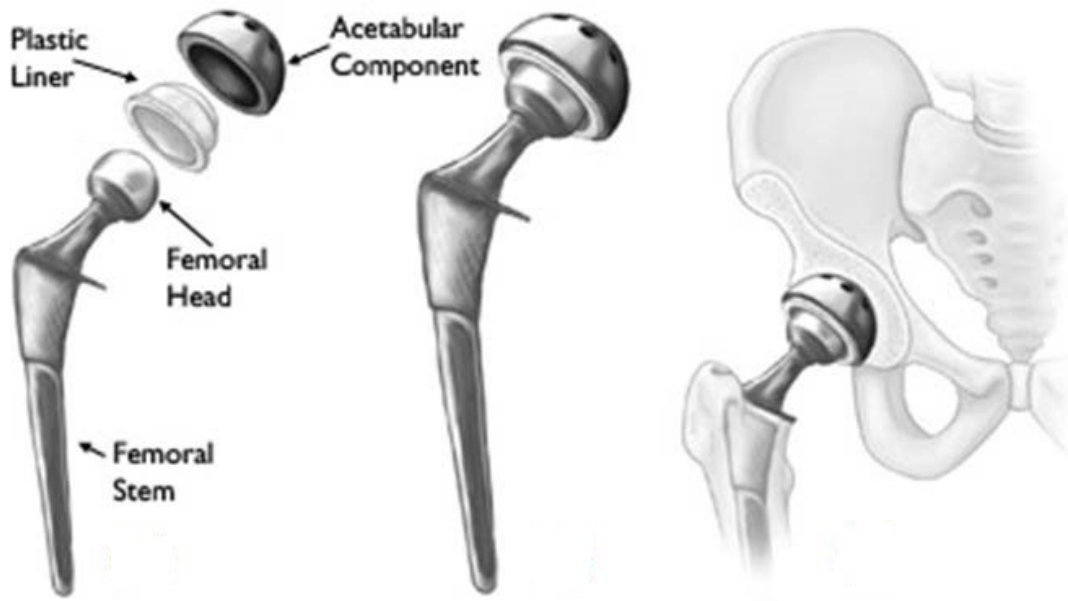


Figure 0.1: Total hip joint replacement (<http://goo.gl/8G06eM>)

Wear estimation - State of the art

Wear estimation contains the investigation of material loss caused by friction. We are interested in:

- Volumetric wear - the number of cubic millimeters of released material
- Linear wear - the number of millimeters of lost material at individual spots of acetabular hip joint replacement; wear distribution
- Concentration of released particles in joint's neighbourhood - the number of pieces of wear particles per 1 cubic millimeter of surrounding space
- Morphology of released wear particles - their size and shape

Volumetric wear and linear wear are worth to know for evaluating the resistance of the materials against wear and logically, the better wear resistance is, the longer lifetime of whole replacement can be expected in wear point of view. Concentration and morphology of released particles are important too because various combinations of materials of femoral head and acetabular component release different sizes and shapes of wear particles. There can be a large concentration of tiny nanoparticles or low concentration of relatively large particles. In that case, the numbers of volumetric and linear wear could be similar but the body's response could be way different. Accordingly, the investigation of concentration and morphology is very important for wear evaluation but it is no longer going to be part of this thesis. We can read more about concentration and morphology of wear particles at e.g. *Ladron, et al., 2012* [3]. Within the thesis, we are going to focus on estimation of volumetric and linear wear. [3]

1.1 *In vivo* methods

The radiographic measurements are used for total hip joint replacements with UHMWPE acetabular components inside humans' bodies. Several methods are used but all of them contain two steps - RTG screening of hip joint replacement; and estimation of penetration of femoral head into acetabular cup. All radiographic techniques are limited by the RTG image resolution so the results gained will never be very accurate. Unfortunately, it is not supposed to evaluate wear distribution from RTG images. Only maximal linear wear

can be estimated and based on its value, calculation of volumetric wear can be performed. Moreover, it is not supposed to distinguish wear and plastic deformation or creep. [3]



Figure 1.1: RTG image of total hip joint replacement (<http://goo.gl/IK1XeS>)

For evaluation of maximal linear wear, the following techniques are used:

Livermore technique (Fig. 1.2) [4]: "On the preresion radiograph, a transparent overlay of concentric circles is used to determine the femoral head center of rotation (FH COR). A compass is used to determine the minimum distance between the FH COR and the acetabular shell outer diameter (OD). The line between the FH COR and the acetabular OD is defined as the line of maximum wear. The postoperative radiograph is measured in the same manner along the line of maximum wear. The difference in the polyethylene thickness between the postoperative and preresion radiographs is the amount of linear wear."

Computer Livermore technique [4]: "This techniques uses the computer/digitizer to determine the FH COR, the acetabular OD, and the distance between the 2 for the postoperative and preresion radiographs. For each radiograph, 3 points are digitized along the FH silouhette and the outer surface of the acetabular shell. The computer calculates the best-fit circles (with their CORs) for FH and the acetabular shell. The changes in the CORs for the FH and the acetabular shell can be used to measure wear, by calculating the minimum distance between 2 circles (w) using the following equation: $w = r_1 - r_2 - [(y_2 - y_1)^2 - (x_2 - x_1)^2]^{1/2}$, where w is the minimum polyethylene thickness, and (x_1, y_1) ; r_1 and (x_2, y_2) ; r_2 are the COR coordinates and

radii for the acetabular shell and FH. The minimum distance is calculated for the postoperative (w_{postop}) and prerevision (w_{prerev}) radiographs. The linear wear is calculated as the change in the polyethylene thickness ($w_{postop} - w_{prerev}$)."

Uniradiographic technique [4]: "This technique requires only a prerevision radiograph, and the measurements do not need to be corrected for radiographic magnification. The narrowest measurement in weight-bearing area between the FH OD and the acetabular OD is subtracted from the widest measurement in the non-weight-bearing zone. The difference between the 2 distances is divided by 2 and gives the calculated linear wear."

Duoradiographic technique [4]: "The narrowest measurement in the weight-bearing zone between the FH OD and the acetabular OD is determined on the prerevision radiograph. The distance between the FH OD and the acetabular OD in the same zone is measured on the postoperative radiograph. The difference in the distances represents the linear wear when corrected for magnification factors."

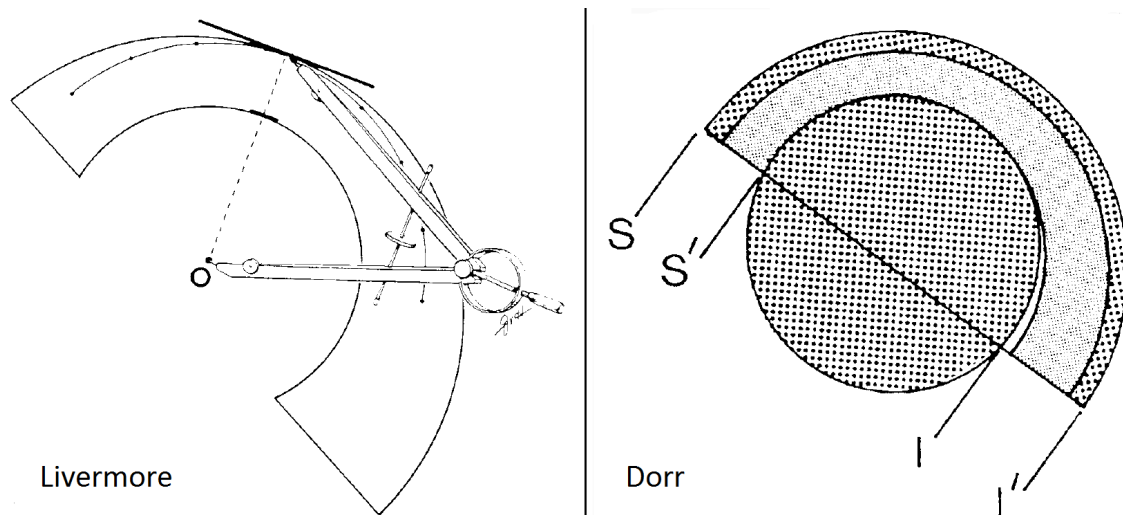


Figure 1.2: Livermore & Dorr wear measurement technique [4]

Tangential technique [4]: "Using the face of the acetabular shell as the reference, a tangential line to the superior aspect of the FH is drawn as well as to the superior aspect of the acetabular shell (ie, each line is a tangent to the respective circle, and it is perpendicular to the line that crosses the face of the acetabulum at its widest margin). The distance between the 2 tangential lines is calculated for the postoperative and the prerevision radiographs. The change in distance between these 2 lines, after correcting for magnification, is calculated as the linear wear."

Dorr technique (Fig. 1.2) [4]: "This technique requires only the prerevision radiograph. The acetabular face reference line is drawn on the radiograph. Along the acetabular reference line, the distance between the superior aspect of the FH and the superior aspect of the acetabulum (SS') is measured. The distance along the line between the inferior aspect of the FH and the inferior aspect of the acetabulum (II') is measured.

Linear wear is calculated based on the formula: $Linear\ wear = (II' - SS')/2$, after distances have been corrected for magnification.”

Since we know maximal linear wear d , its position β and radius of FH R (Fig. 1.3) we are able to calculate volumetric wear V . Sophisticated calculations are presented by *Košak, et al., 2003 [5]*. Standard mathematical model estimates volumetric wear according to the formula: $V = \pi \cdot R^2 \cdot d$. Modified mathematical model derived by *Košak, et al., 2003 [5]* considers position of maximal linear wear β and estimates volumetric wear according to the formula: $V = \pi \cdot R^2 \cdot d \cdot (1 + \sin \beta)/2$.

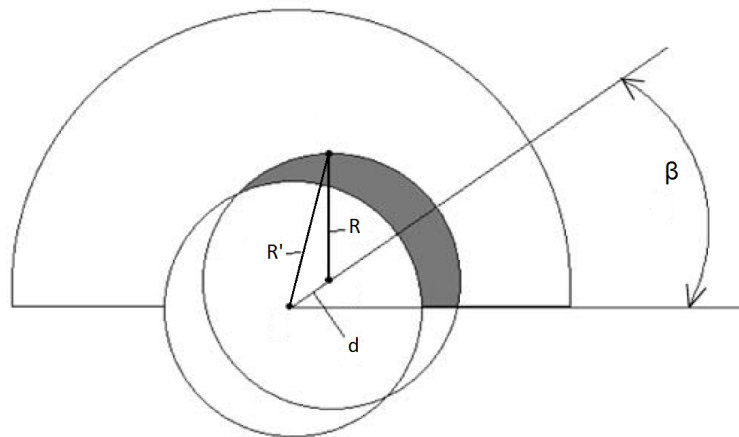


Figure 1.3: Definition of variables for volumetric wear calculation [5]

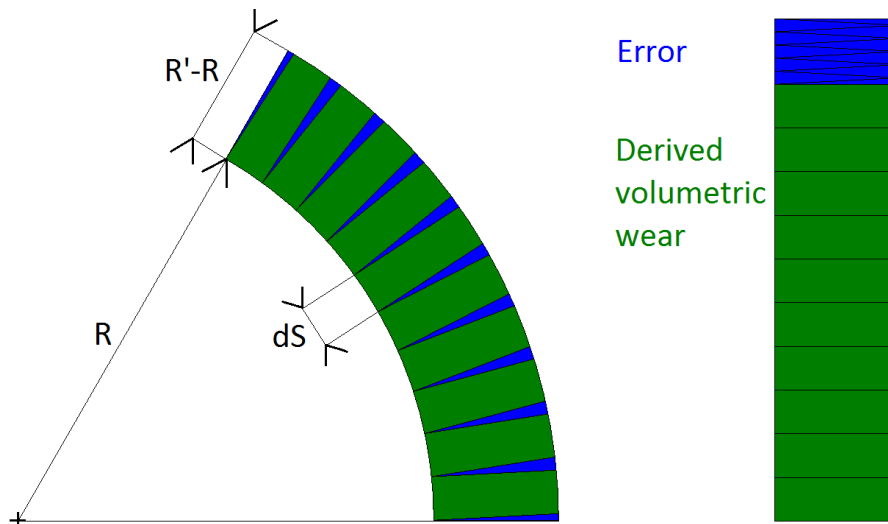


Figure 1.4: Error of volumetric wear calculation

The derivation is based on the formula: $V = \int (R' - R) \cdot dS$ where R' is the distance between center of FH before shift and selected point on worn surface; and dS is the infinitesimal surface area element of FH before shift. However, the derivation neglects a

fact which is significant for the result. This fact is well visible from the Fig 1.4. Let's consider the green-blue area at the left side of Fig. 1.4 as a general gray area from the Fig. 1.3. Derived calculation of volumetric wear neglects blue areas which cause an error. For clearer illustrative purposes, only the 2D problem is visualized. In 3D space, the blue volume would occupy a larger percentage of volume symbolizing real volumetric wear than the percentage which the blue area occupies in shown 2D picture. The larger the maximal linear wear d is, the larger the error arises due to the described neglect. This fact explains one of the sources of the difference between results gained by this type of calculation and results gained by accurate *in vitro* fluid displacement method (mentioned below).

1.2 *In vitro* methods

The worn samples for *in vitro* methods of wear estimation could be obtained from two sources. The first source is constituted by real explanted cups. The second source is constituted by samples worn in simulators of motion. Machines for wear simulations are based on the movement of a softer pin (typically UHMWPE) on a harder disc (typically metal). Therefore, these machines are usually called *pin-on-disc*. *Pin-on-disc* machines are either unidirectional - pin is allowed to move in one direction, or multidirectional - pin is allowed to move in more than one direction. Testing within unidirectional simulations does not correspond with *in vivo* situation and its results are unsatisfied. Multidirectional joint simulators provide wider possibilities for wear simulations. Usually, the kinematics and kinetics are controlled by computers and also real replacements can be used instead of pins and discs. It is supposed to simulate real movement and loading of joint replacements. [3]

For wear estimation of either real explanted cups or laboratory samples, several methods are used:

Gravimetric method [3]: Weight loss of material can be observed. This measurement provides the information of volumetric wear but the information of how much the material soaks up the surrounding liquids must be known. This fact has a significant influence on result accuracy. Linear wear is not supposed to investigate by gravimetric method. On the other hand, this is the only method which is supposed to distinguish wear and plastic deformation or creep. Gravimetric method is used wider for the wear estimation of laboratory samples than for the wear estimation of real cup explants. Usually, real cup explants were not weighted before surgery and moreover their weight is influenced by soaked body fluid.

Fluid displacement method [3]: Volumetric wear can be measured by filling the cup (or sample) by water (or any other fluid), and observing the volume of used fluid. This method is very accurate but it is not supposed to distinguish wear and plastic deformation or creep. Unfortunately, linear wear is not supposed to measure by this method either.

Surface 3D scanning [3]: Another option for wear estimation is the 3D scanning of the surface and following surface analysis in appropriate software. Nowadays, very accurate surface measurement is supposed to reach within modern 3D CMM. Afterward, volumetric wear is calculated but it is not supposed to distinguish wear and

plastic deformation or creep. However, the surface 3D scanning is the only method providing visualization of wear distribution. The accuracy of the results depends on the accuracy of measured data by scanner and correctness of software evaluating procedure.

The Laboratory of Biotribology within Department of Mechanics, Biomechanics and Mechatronics, FME, CTU in Prague, owns very accurate 3D scanner (more about this 3D scanner in section 3.1 on page 11). Accordingly, we are choosing the surface 3D scanning as the wear estimation method we will apply in this thesis. We will measure surface by mentioned 3D scanner and analyze measured data in created evaluating program.

Thesis aims

Acetabular cup wear is the principal factor limiting the lifetime of total hip replacement. Common wear estimation methods rely mostly on the analysis of radiograms. The accuracy of radiographic methods is limited by RTG image resolution. These methods estimate maximum linear wear and do not provide wear distribution. A large number of explanted cups available in clinical practice would allow application of precise methods of surface measurement. **The main aim of this thesis is to develop the method of wear analysis of explanted acetabular cups based on accurate surface measurement.**

Subsequently, specific aims are to:

- Develop casting method of cup explants
- Develop algorithm for wear evaluation
- Create *Matlab GUI* program
- Verify casting method of cup explants
- Verify algorithm for wear evaluation
- Study series of explanted cups in a pilot study

Data measurement

For successful wear estimation of cup explants, we are challenging two main steps - data measurement; and data analysis. This chapter deals with the measurement of worn cup surfaces. We will use optical 3D CMM by *RedLux Ltd.* (Fig. 3.1) which is very accurate 3D scanner placed in The Laboratory of Biotribology within Department of Mechanics, Biomechanics and Mechatronics, FME, CTU in Prague. The aim is to measure coordinates of point cloud which discretely describes the worn surface of explanted cup.

3.1 Surface 3D scanning

Tuke, et al., 2010 [6] describe measuring principle of optical 3D CMM by *RedLux Ltd.* by following sentences: "The *RedLux Artificial Hip Profiler* (*RedLux Ltd.*, Southampton, UK) is optical 3D CMM which combines the high resolution with the high coverage, while using an automated, non-contact sensor for increased speed of measurement. It is capable of scanning the surface of spherical objects. Unlike other optical instruments, it can scan the whole bearing surface of an artificial hip joint in a single measurement taking only a few minutes. The instrument is a 4-axis optical CMM, as shown in the Fig. 3.2. It consists of two linear and two rotary axes, all of which utilise air bearings to achieve superior accuracy of motion. The two rotary stages carry the head or cup and the linear axes carry the sensor. All axes utilise optical encoders and linear or brushless motors. The sensor allows the measurement of artificial heads as well as cups. It is a point sensor, based on the chromatically encoded confocal measurement. With this sensor, the lens error commonly known as chromatic aberration is used to measure the distance to an object. The axial position of the focal point of an uncorrected lens depends on the colour (wavelength) of the light to be focussed. In the visible spectral region, the focal distance for blue light is shortest while it is longest for red light. The focal points of the other colours are located in between. Depending on the distance of the target from the focusing lens, light of just a very small wavelength region λ_1 is focused on the target's surface (Fig. 3.3). The focusing lens is also used to receive the backscattered light from the target's surface and to focus it into an optical fibre and later a spectrometer. The wavelength scale of the spectrometer is calibrated versus the distance to the scattering surface, giving the required output signal. By combining the sensor signal with the knowledge over the exact position of all 4 stages, a 3D representation of the surface can be created in computer's memory. Once these data are available, they can be processed and analysed. For a typical



Figure 3.1: Optical 3D CMM by *RedLux Ltd.* (<http://goo.gl/veqx5V>)

50mm diameter head, 40 000 points are measured, while for the corresponding cup 30 000 points are taken. For detailed investigations, over one million data points are available. To achieve better accuracy, the system is located on a vibration isolation table.”

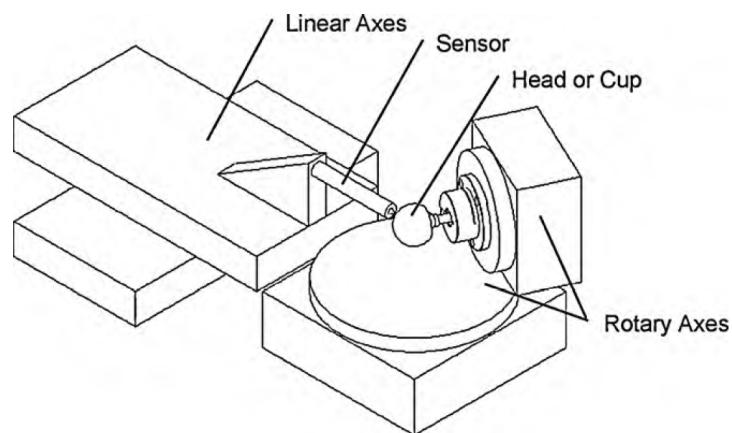


Figure 3.2: Workspace of *RedLux Artificial Hip Profiler* [6]

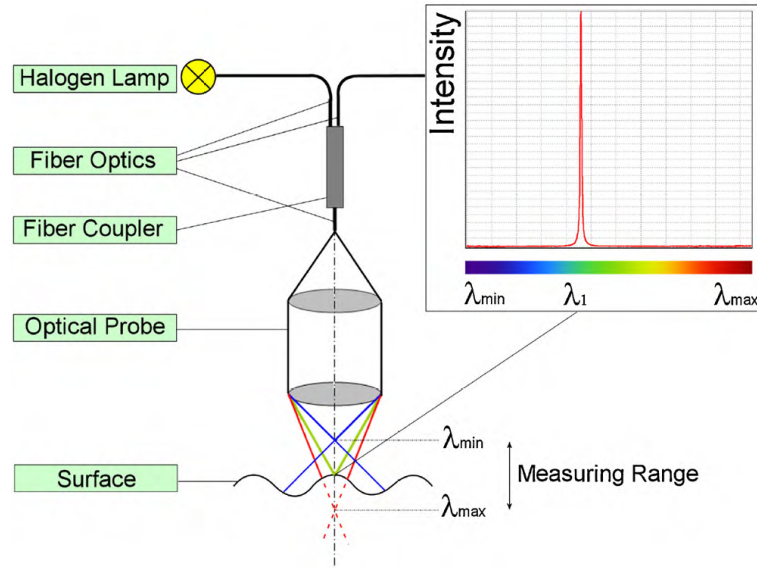


Figure 3.3: Chromatically encoded confocal measurement method [6]

The accuracy of resulting cloud of points is given by resolution of 2 linear axes, resolution of 2 rotary axes and resolution of the probe. The resolution of each linear axis is 100nm, the resolution of each rotary axis is $10''$ and the resolution of the probe is 20nm. Throughout all performed measurements, we have been measuring point cloud with density 720 points per one rotation of the measured sample in vertical plane and within every single rotation the measured sample was turned by 0.5° in horizontal plane.

| | | |
|-------------------|-------------------|------------------|
| -8.82126838749669 | -14.0072559636073 | 8.42190007269006 |
| -8.95514485981797 | -13.9482591917328 | 8.43312999204765 |
| -9.08553085171818 | -13.8831863361474 | 8.44144947703428 |
| -9.21511045566796 | -13.8163688595789 | 8.44943464351792 |
| -9.34509803990566 | -13.7496498060891 | 8.45821857896798 |
| -9.47527558933062 | -13.6826598970492 | 8.46759138624535 |
| -9.60474471896162 | -13.6140876237318 | 8.47609737168355 |
| -9.73497472342428 | -13.5460237981924 | 8.48632955260894 |
| -9.86412746589781 | -13.4758720670353 | 8.49601547211151 |
| -9.9892649912472 | -13.3997414370578 | 8.50269567792881 |
| -10.1127209686231 | -13.3209930098422 | 8.50779387579513 |
| -10.2447139891861 | -13.252991682007 | 8.52117838774841 |
| -10.4044682967031 | -12.9841472914327 | 8.46391429891756 |
| -10.5128611073078 | -12.8871154422258 | 8.45954194274782 |
| -10.6248487753409 | -12.794614840362 | 8.45941317357661 |
| -10.738650449113 | -12.7042613395833 | 8.46136226457175 |
| -10.8532399629373 | -12.6147106779702 | 8.46452749387703 |

Figure 3.4: Output file of *RedLux Artificial Hip Profiler* - coordinates of point cloud

Optical 3D CMM by *RedLux Ltd.* is capable of measuring whole surfaces of unworn cups. However, we cannot say so for worn cup surfaces. Practical experience from the laboratory showed that as the wear exceeds the certain value, the probe is no longer capable of measuring area surrounding the cup edge. It is given by that the measuring of convex surfaces contains certain disadvantages, as shown on the top of Fig. 3.5. The worn cup rotates around its rotary axis and the probe is in its appropriate position - 4.5mm from the measured surface (0.3mm measuring range); and the deviation angle between radiated optic stream from the probe and bounced one from the measured surface stays lower than 30° because the optic stream must be bounced back from the measured surface

to the probe. If the probe is measuring the area surrounding the edge, then the probe has very tiny space for movements. In that case, unexpected fold or relatively large radius deviation could cause the need for additional movements; so there is a risk of the probe crashing with the opposite side of the cup. Another restriction is that the probe is capable of measuring only cups with the diameter larger than 28 mm. Some of the cups are manufactured with the diameter lower than 28mm though. If we measure the cup with such a small diameter close to the edge and the deviation angle of bounced optic stream shall remain lower than 30° , then there is also a risk of the probe crashing with the opposite side of the cup.

Measuring of concave surfaces does not indicate problems mentioned above (bottom of the Fig. 3.5). The probe has enough of space for all necessary extra movements so we are capable of measuring whole worn surfaces or surfaces with small diameter easily. Therefore, we **decided to cast the explanted cups and measure the casts**.

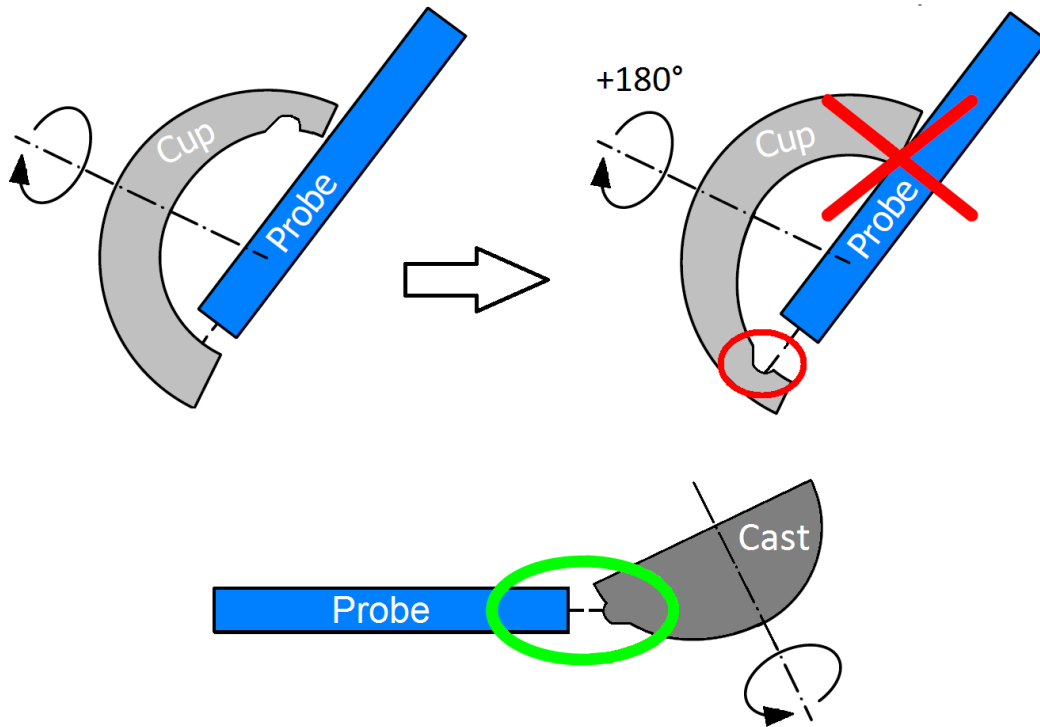


Figure 3.5: The need for cast

3.2 Casting method

Casts of cup explants made for the surface 3D scanning have to fulfill these criterion aligned in the order of their importance:

- Sufficient elasticity - to enable casts easy removal from the cup when the cup surface is worn in such a way that the cast has to be squeezed for removal.

- Negligible volumetric change - to avoid an error of the volumetric and linear wear
- Excellent surface reconstruction - to avoid incorrect visualization of wear distribution
- 'User-friendly' behavior - to avoid toxic compounds, etc.

We were experimenting with four types of casting materials. In the end, we chose *Addition Cure Moulding Rubber MM242* as the material we are going to use because this material fulfilled all our requirements more than sufficiently:

Resin - *Dentacryl* (<http://goo.gl/pJExBo>) ✗

- ✗ Sufficient elasticity - It is not capable of removing resin cast from the cup as worn as mentioned above. Resin is too hard and cannot be squeezed.
- ✗ Negligible volumetric change - According to the Fig. 6.2 on page 62, volumetric shrinkage of resin cast is 1.335%. This value is not good enough for our purposes.
- ✓ Excellent surface reconstruction - According to the Fig. 6.3 on page 62, the Weighted standard deviation of radius sd_w of the resin cast is 0.007168mm which is close enough to the Weighted standard deviation of radius sd_w of the original cup 0.007025mm.
- ✗ 'User-friendly' behavior - During the mixing of two components, a very unpleasant odour is released. This odour can cause a headache to the worker.
- Casting procedure - Two components are mixed together (according to given volume fraction). Afterward, the cup is filled by arisen mixture. In the end, bubbles of the air (caused by mixing) are sucked out of the cast by vacuum machine and the cast is let to harden.

Gypsum ✗

- ✗ Sufficient elasticity - It is not capable of removing gypsum cast from the cup as worn as mentioned above. Gypsum is too hard and cannot be squeezed.
- ✓ Negligible volumetric change - According to the Fig. 6.2 on page 62, volumetric expansion of gypsum cast is 0.028%. This value is good enough for our purposes.
- ✓ Excellent surface reconstruction - According to the Fig. 6.3 on page 62, the Weighted standard deviation of radius sd_w of the gypsum cast is 0.007553mm which is close enough to the Weighted standard deviation of radius sd_w of the original cup 0.007025mm.
- ✓ 'User-friendly' behavior - No harmful effects of gypsum were observed.
- Casting procedure - Gypsum powder is mixed with water. These two components are stirred as long as smooth fluid solution arise. In this moment, the cup is filled by arisen solution and the cast is let to harden.

Wax ✗

- ✗ Sufficient elasticity - It is not capable of removing wax cast from the cup as worn as mentioned above. Wax is too soft and relatively small press avoids plastic deformation.

- ✗ Negligible volumetric change - According to the Fig. 6.2 on page 62, volumetric shrinkage of wax cast is 2.264%. This value is not good enough for our purposes.
- ✗ Excellent surface reconstruction - According to the Fig. 6.3 on page 62, the Weighted standard deviation of radius sd_w of the wax cast is 0.017730mm which is not close enough to the Weighted standard deviation of radius sd_w of the original cup 0.007025mm.
- ✓ 'User-friendly' behavior - No harmful effects of wax were observed.
- Casting procedure - Wax is melt on the cooker. When the wax is in liquid phase the cup is filled by dissolved wax and the cast is let to cool down and harden.

Silicone - *Addition Cure Moulding Rubber MM242* (<https://goo.gl/TZJDZd>) ✓

- ✓ Sufficient elasticity - It is capable of removing silicone cast from the cup as worn as mentioned above because it is easy to squeeze this cast and elastic deformation is not exceeded during the squeezing.
- ✓ Negligible volumetric change - According to the Fig. 6.2 on page 62, volumetric shrinkage of silicone cast is 0.104%. This value is good enough for our purposes.
- ✓ Excellent surface reconstruction - According to the Fig. 6.3 on page 62, the Weighted standard deviation of radius sd_w of the silicone cast is 0.007091mm which is close enough to the Weighted standard deviation of radius sd_w of the original cup 0.007025mm. Besides, every fold on the original surface seems to be preserved on the cast surface too - obvious from the casts of worn surfaces.
- ✓ 'User-friendly' behavior - No harmful effects of silicone were observed.
- Casting procedure - Two components are mixed together (according to given mass fraction). Afterward, the cup is filled by arisen mixture. In the end, bubbles of the air (caused by mixing) are sucked out of the cast by vacuum machine and the cast is let to harden.

There are special casting materials on the market which would fulfill our criterion even better than *Addition Cure Moulding Rubber MM242* but their price exceeds our financial limit. However, features of *Addition Cure Moulding Rubber MM242* are very close to those special casting materials.

Special *Matlab* script was created for **verification of casting methods**. We were interested in how much the final results of wear estimation are influenced by casts. The *Matlab* script is contained on enclosed CD as attachment called *D - Verification of casting method*. The calculation of this *Matlab* script is based on derivations presented in chapter 4 beginning on page 19. Firstly, measured data is transformed (based on subsection 4.2.2 on page 28), then the data is interpolated (based on subsection 4.2.3 on page 32) and afterward the volume V of the cast is calculated (according to Eq. 4.10) as:

$$V_i = \frac{r_i^3}{3} \cdot \Delta \cdot \left[\cos \left(\xi_i - \frac{\Delta}{2} \right) - \cos \left(\xi_i + \frac{\Delta}{2} \right) \right] \quad (3.1)$$

$$V = \sum_{i=1}^n V_i \quad (3.2)$$

where V_i is the volume of pyramid represented by individual interpolated point i , r_i is the distance between the origin of the coordinate system and point i , Δ is the Interpolation step (subsection 4.1.2 on page 25) and ξ_i is the spherical coordinate of point i (subsection 4.1.1 on page 19). Volume change Θ_j is calculated as:

$$\Theta_j = \frac{V_j - V_{cup}}{V_{cup}} \cdot 100 \quad (3.3)$$

where $j = cup, resin, gypsum, wax, silicone$. Weighted average of radius R_w is given as:

$$R_w = \frac{\sum_{i=1}^n r_i \cdot B_i}{\sum_{i=1}^n B_i} \quad (3.4)$$

where B_i is area represented by individual interpolated point i (according to Eg. 4.8):

$$B_i = r_i^2 \cdot \Delta \cdot \left[\cos \left(\xi_i - \frac{\Delta}{2} \right) - \cos \left(\xi_i + \frac{\Delta}{2} \right) \right] \quad (3.5)$$

Weighted standard deviation of radius sd_w is given by formula:

$$sd_w = \sqrt{\frac{\sum_{i=1}^n B_i \cdot (r_i - R_w)^2}{(n'-1) \cdot \frac{\sum_{i=1}^n B_i}{n'}}} \quad (3.6)$$

where n' is the number of non-zero weights.

Results of the verification of casting methods are presented in section 6.1 on page 61. According to those results, we can make the conclusion that the inaccuracy injected into the final results by chosen *Addition Cure Moulding Rubber MM242* cast is acceptable for our purposes.

Data analysis

Measured data gained from the 3D scanner must be analyzed. The aim is to calculate and visualize changes of geometry which arose after implantation. The main properties considered are Volumetric wear U and Linear wear u . Volumetric wear U gives the information of how many cubic millimeters of material has been lost throughout the cup. This output is a scalar value. While Linear wear u defines how many millimeters of material has disappeared at each point of cup surface. It is a matrix (discrete surface description) expressed as a map similar to FEM results (points in 3D space are colored according to fourth variable value). These are the necessary info needed and any additional info acquired is a bonus.

The crucial decision was to choose a suitable software for the realization of the analysis. *Matlab* developed by *The MathWorks, Inc.* seems to be a very good choice. Especially with the utilization of its GUI mode, an interactive program could be written for quick and easy analysis of many cup samples.

For these reasons, the idea is to create ***Matlab GUI* program** which is supposed to upload measured data, to let the user set calculation parameters interactively and to calculate and visualize outputs.

4.1 Sphere - Mathematical background

If we are interested in wear characteristics we have to know how the surface looked before implantation and compare the current surface with the old one. That is quite a difficult issue because explanted cups are usually more than 10 years old and nobody made any measurements of these cups before surgery. However, the active geometry of acetabular replacements (inner surface which is in touch with the femoral head replacement and where the wear occurs) is produced as a hemisphere. In this sense, the calculation can be based on mathematical knowledge of the sphere. This approach makes upcoming work easier. The advantage of a simple mathematical description of original geometry is really wide. If we assume the original (reference) surface was spherical then it is enough to estimate its center and radius.

4.1.1 Coordinate Systems

There are three coordinate systems which are important to define:

- Cartesian coordinate system
- Spherical coordinate system
- Pole coordinate system

Initial data we receive from 3D scanner are **Cartesian coordinates** (x, y, z) of individual surface points (Fig. 3.4 on page 13). Cartesian coordinates generally visualize undistorted 3D geometry. In the Cartesian coordinate system we use (Fig. 4.1), the origin is set at the center point of the hemisphere and the x axis passes through the top (the top means the pole from the Fig. 4.4). Since Cartesian coordinates are not much suitable for some mathematical operations, we define the following two coordinate systems where various calculations will be made.

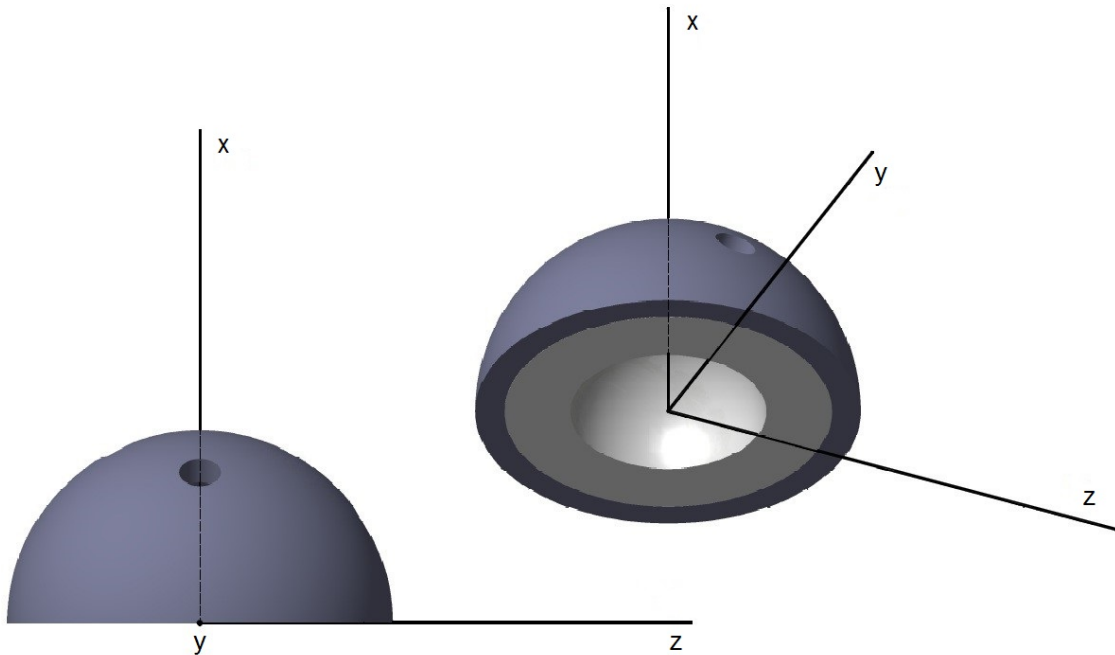


Figure 4.1: Position of the cup in the Cartesian coordinate system [7]

Spherical coordinates (r, ξ, θ) will be needed for the upcoming analysis. Their definition is shown in the Fig. 4.3. We have to determine permitted intervals for angles of Spherical coordinates. Their values were chosen $\xi \in \langle 0, \pi \rangle$, $\theta \in \langle -\pi, \pi \rangle$ which covers whole surface of sphere. The advantage of this system is that one of its axis (or variables) symbolizes sphere radius r . If we want to observe Linear wear u (deviation from original radius) this coordinate system is suitable for us. However it is not enough. There is also disadvantage of this coordinate system which is visible in the Fig. 4.4. The pole is not point anymore at Spherical coordinates system and hemisphere surface was sectioned along one of the parallels. This fact is unacceptable for us because it interrupts continuity of sphere surface. For upcoming interpolating and smoothing processes we need to keep sphere surface continuous (regardless if it is distorted or not).

Therefore, we must make one more step and define **Pole coordinates** (r, s_1, s_2) according to Fig. 4.3. The simplest way how to imagine these coordinates is shown in the

Fig. 4.2 where 2D map of Antarctica is visualized. The South Pole is the only undistorted place of the Southern Hemisphere surface (the distortion increases as we move away from the South Pole) but its area is continuous everywhere. That is a very interesting feature for us since the addition of a third axis (radius r) Linear wear u can be observed and the projected cup surface is continuous (Fig. 4.4).



Figure 4.2: Map of Antarctica (<http://goo.gl/Hxc5mM>)

Transformation relations between chosen coordinate systems must be defined. There is no direct relation between Cartesian and Pole coordinates. The sphere is transformed from Cartesian coordinates to Spherical ones and back. Or it is transformed from Spherical coordinates to Pole ones and back. Accordingly in total we have four relations which manage to express our data in three coordinate systems, regardless which system is the initial one. Transformations are derived according to coordinates definition shown in the Fig. 4.3. As visible in the Eq. 4.3 and Eq. 4.4, the constant R is used, $R \in \mathbf{R}_{>0}$, and its value, a representative radius of the sphere, is chosen.

A hemisphere expressed at Cartesian coordinate system is transformed into rectangle at Spherical coordinates system while it creates circle at Pole coordinate system. This is well visible in the Fig. 4.4.

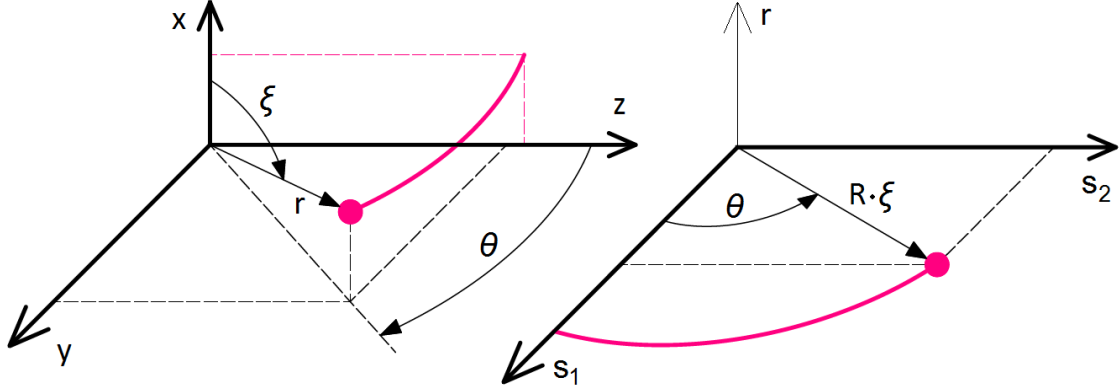


Figure 4.3: Definition of Spherical coordinates in Cartesian and Pole c.s.

T1. Cartesian c.s. $(x, y, z) \rightarrow$ Spherical c.s. (r, ξ, θ)

$$\begin{aligned}
 r &= \sqrt{x^2 + y^2 + z^2} \\
 \xi &= \arccos\left(\frac{x}{r}\right) \\
 \theta &= \arctan\left(\frac{y}{z}\right) \begin{cases} \text{if } z \geq 0 \\ +\pi & \text{if } z < 0 \wedge y \geq 0 \\ -\pi & \text{if } z < 0 \wedge y < 0 \end{cases}
 \end{aligned} \tag{4.1}$$

T2. Spherical c.s. $(r, \xi, \theta) \rightarrow$ Cartesian c.s. (x, y, z)

$$\begin{aligned}
 x &= r \cdot \cos \xi \\
 y &= r \cdot \sin \xi \cdot \sin \theta \\
 z &= r \cdot \sin \xi \cdot \cos \theta
 \end{aligned} \tag{4.2}$$

T3. Spherical c.s. $(r, \xi, \theta) \rightarrow$ Pole c.s. (r, s_1, s_2)

$$\begin{aligned}
 r &= r \\
 s_1 &= R \cdot \xi \cdot \cos \theta \\
 s_2 &= R \cdot \xi \cdot \sin \theta
 \end{aligned} \tag{4.3}$$

T4. Pole c.s. $(r, s_1, s_2) \rightarrow$ Spherical c.s. (r, ξ, θ)

$$\begin{aligned}
 r &= r \\
 \xi &= \frac{\sqrt{s_1^2 + s_2^2}}{R} \\
 \theta &= \arctan\left(\frac{s_2}{s_1}\right) \begin{cases} \text{if } s_1 \geq 0 \\ +\pi & \text{if } s_1 < 0 \wedge s_2 \geq 0 \\ -\pi & \text{if } s_1 < 0 \wedge s_2 < 0 \end{cases}
 \end{aligned} \tag{4.4}$$

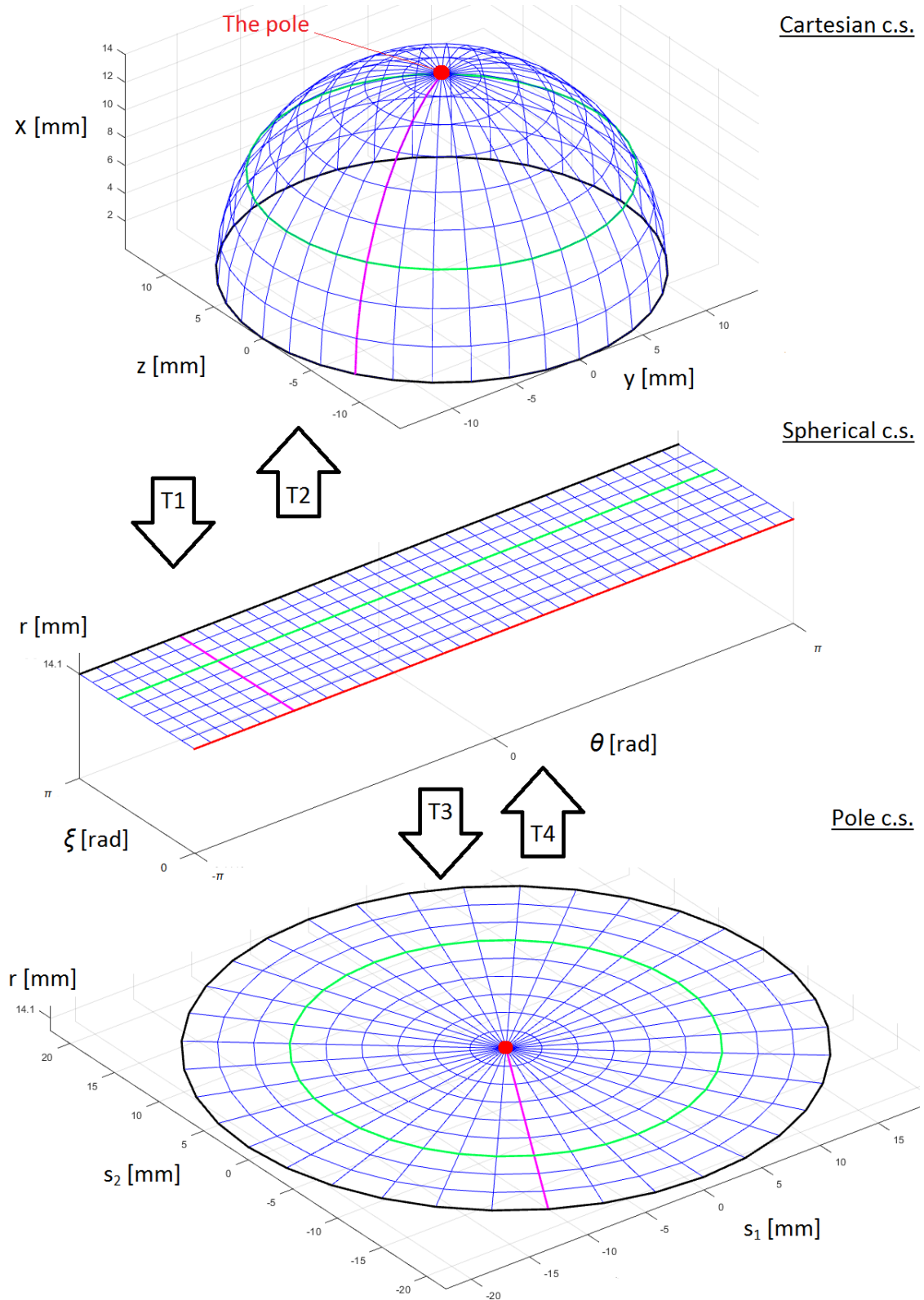


Figure 4.4: Hemisphere visualized at various coordinate systems

Listing 4.1: *Cup_analysis.m* - Transformations between coordinate systems

```
937 % TRANSFORMS X,Y,Z DATA TO SPHERICAL COORDINATES
938 function [r,theta,xi] = x_z_y__r_theta_xi(x,y,z)
939
940 r = sqrt (x.^2+y.^2+z.^2);
941
942 xi = acos(x./r);
943
944 size_y = size(y); theta(1:size_y(1,1),1:size_y(1,2)) = NaN;
945 log = (z >= 0);
946     theta(log) = atan(y(log)./z(log));
947 log = (y >= 0) & (z < 0);
948     theta(log) = atan(y(log)./z(log)) + pi;
949 log = (y < 0) & (z < 0);
950     theta(log) = atan(y(log)./z(log)) - pi;
951 theta(isnan(theta)) = 0; % for cases when atan(0/0)
952
953
954 % TRANSFORMS SPHERICAL COORDINATES TO X,Y,Z DATA
955 function [x,y,z] = r_theta_xi__x_y_z(r,theta,xi)
956
957 x = r.*cos(xi);
958 y = r.*sin(xi).*sin(theta);
959 z = r.*sin(xi).*cos(theta);
960
961
962 % TRANSFORMS SPHERICAL COORDINATES TO POLE COORDINATES
963 function [s1,s2] = Radius_theta_xi__s1_s2(Radius,theta,xi)
964
965 s1 = xi*Radius.*cos(theta);
966 s2 = xi*Radius.*sin(theta);
967
968
969 % TRANSFORMS POLE COORDINATES TO SPHERICAL COORDINATES
970 function [theta,xi] = Radius_s1_s2__theta_xi(Radius,s1,s2)
971
972 size_s1 = size(s1); theta(1:size_s1(1,1),1:size_s1(1,2)) = NaN;
973 log = (s1 >= 0);
974     theta(log) = atan(s2(log)./s1(log));
975 log = (s1 < 0) & (s2 >= 0);
976     theta(log) = atan(s2(log)./s1(log)) + pi;
977 log = (s1 < 0) & (s2 < 0);
978     theta(log) = atan(s2(log)./s1(log)) - pi;
979 theta(isnan(theta)) = 0; % for cases when atan(0/0)
980
981 xi = sqrt(s1.^2 + s2.^2)/Radius;
```

4.1.2 Surface Area & Volume

Point cloud, that we receive from 3D scanner, provides discrete description of the cup surface. Scattered data have to be interpolated. We need an aligned net of points where each point represents certain local area of sphere surface. We shall calculate value of this local area for the following wear calculations. Therefore, sphere surface is divided by meridians and parallels (given by θ and ξ - Spherical coordinates) as shown in Fig. 4.5, where also infinitesimal pyramid element is depicted for illustrative purposes.

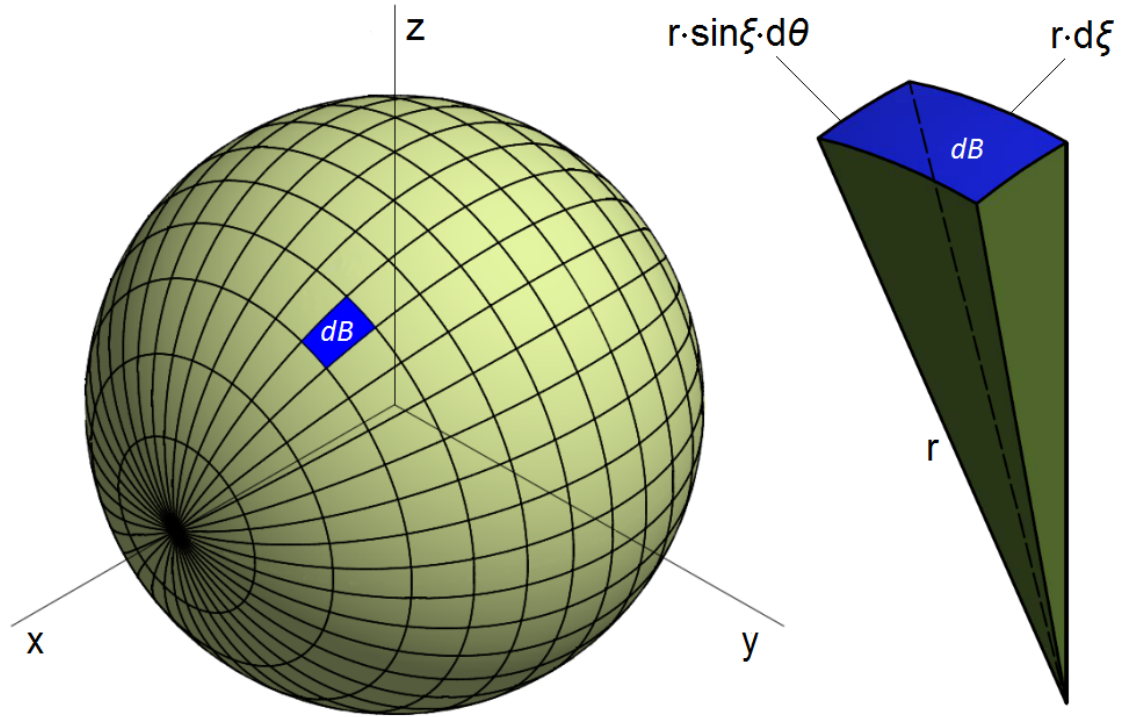


Figure 4.5: Meridians and parallels [7] & Sphere element

Infinitesimal surface area element dB is derived as

$$dB = r^2 \cdot \sin \xi \cdot d\xi \cdot d\theta \quad (4.5)$$

Local surface area B_l represented by certain interpolated point is finite area element where interpolated point occurs in the middle of this area and area borders are given by **Interpolation step** Δ . Interpolation step Δ defines density of the net. In our case, Interpolation step Δ was chosen same for both directions. It means $\xi = 0 : \Delta : \xi_{max}$ and $\theta = -\pi + \Delta : \Delta : \pi$ is the sense of interpolation we are going to use. Then local surface area is determined as

$$B_l = \iint_{B_l} dB \quad (4.6)$$

$$B_l = r^2 \cdot \int_{\xi - \frac{\Delta}{2}}^{\xi + \frac{\Delta}{2}} \sin \xi \cdot d\xi \cdot \int_{\theta - \frac{\Delta}{2}}^{\theta + \frac{\Delta}{2}} d\theta \quad (4.7)$$

$$B_l = r^2 \cdot \Delta \cdot \left[\cos \left(\xi - \frac{\Delta}{2} \right) - \cos \left(\xi + \frac{\Delta}{2} \right) \right] \quad (4.8)$$

For calculation of the **Local volume** V_l given by the pyramid, whose base is defined by the Local surface area B_l and apex is defined by sphere center point, we are going to use the following formula for volume of a pyramid $V = A_B \cdot \frac{h_B}{3}$ (where A_B is base area and h_B is base height) applied onto our case

$$V_l = B_l \cdot \frac{r}{3} \quad (4.9)$$

$$V_l = \frac{r^3}{3} \cdot \Delta \cdot \left[\cos \left(\xi - \frac{\Delta}{2} \right) - \cos \left(\xi + \frac{\Delta}{2} \right) \right] \quad (4.10)$$

4.2 Calculation procedure

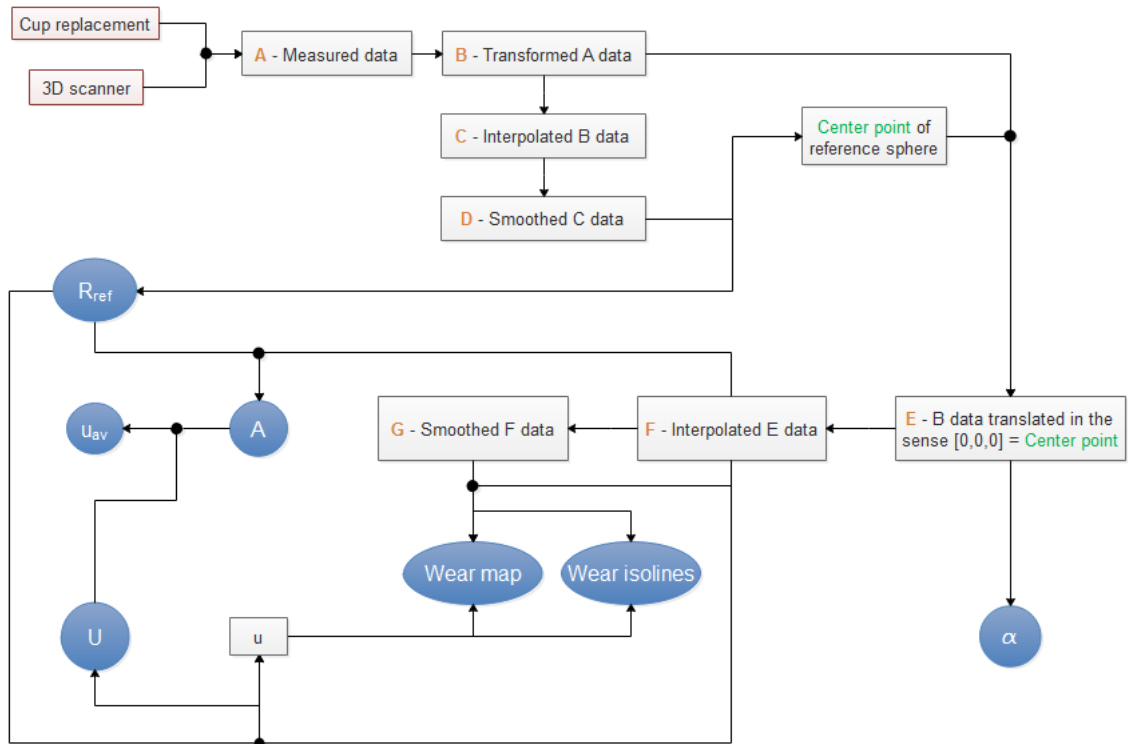


Figure 4.6: Calculation Flow chart

Data Analysis realized in created *Matlab GUI* program is presented in this section. First step of the calculation procedure is uploading the measured data into *Matlab*. Afterward, the data is adjusted to the correct position as required for following calculations. Before estimation of reference geometry starts, the data have to be interpolated and smoothed. When measured data and reference sphere are in correct position against each other, the data are re-interpolated and re-smoothed again. In this situation program has all required inputs to calculate and visualize outputs. Let's see how calculation flows step by step in details.

4.2.1 Data uploading

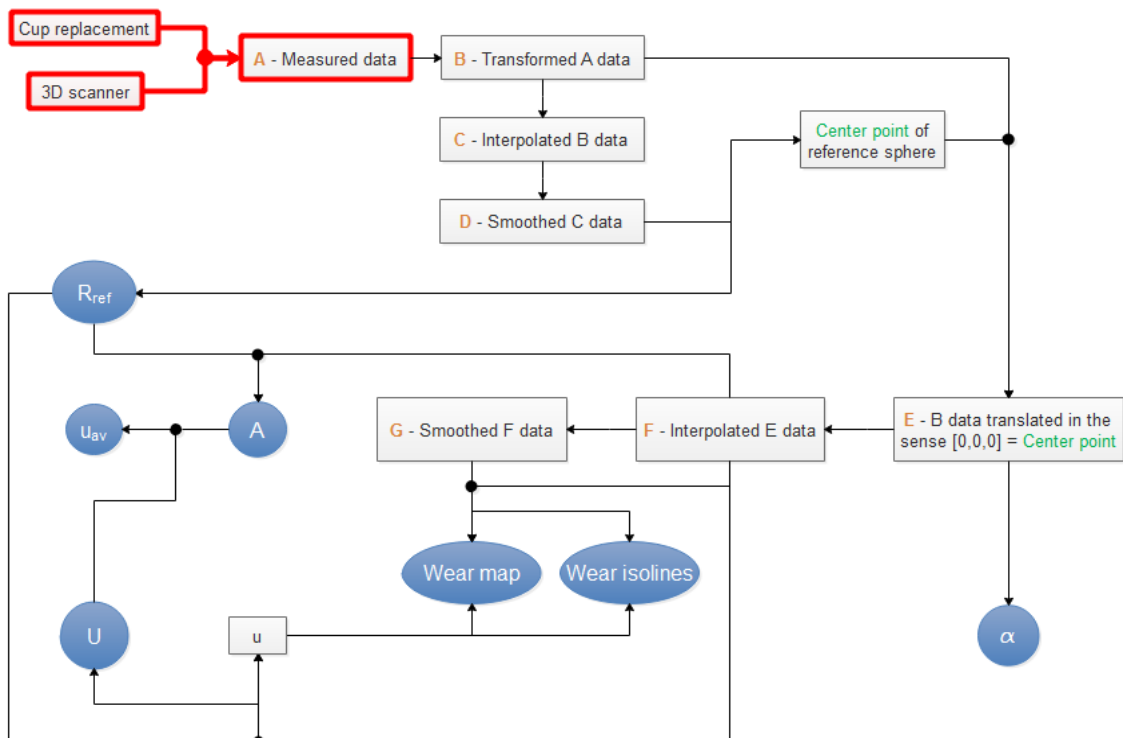


Figure 4.7: Data uploading

Measured data are obtained in a form of text file (*.xyz* format) from 3D scanner. This file contains Cartesian coordinates of point cloud describing the cup surface, as shown in the Fig 3.4 on page 13. The file is imported into *Matlab* and saved as a variable. Measured data are marked as 'A data' in the Flow chart (Fig. 4.7).

Listing 4.2: *Cup_analysis.m* - Data uploading

```

90 % FILE UPLOADING
91 [filename, pathname] = uigetfile({'*.xyz'}, 'File Selector');
92 fullpathname = strcat(pathname, filename);
93 X = importdata(fullpathname); % data loading
  
```

4.2.2 Data transformation

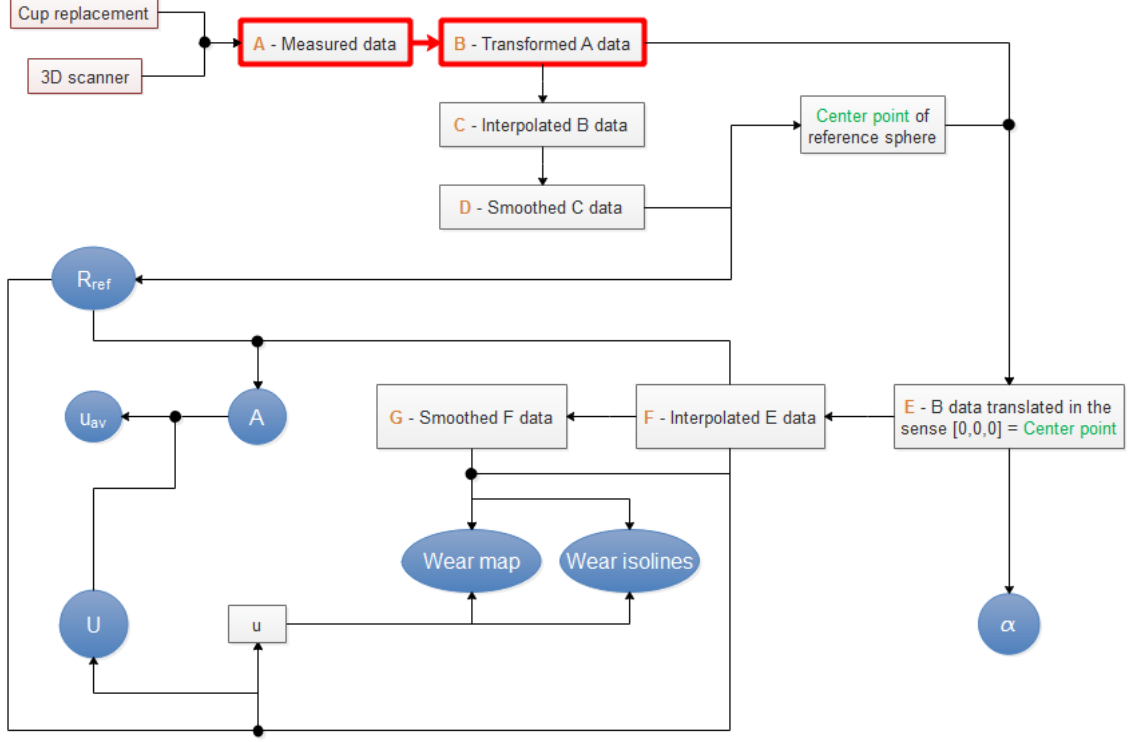


Figure 4.8: Data transformation

'A data' are measured in a Cartesian coordinate system determined by a 3D scanner. We need to have them in correct position according to the definition in the Fig. 4.1 on page 20. Therefore, data transformation is included in the calculation.

We have to find the center point of fitted sphere into point cloud and move the origin of the coordinate system to that center point. Therefore, we use *Matlab* function *sphereFit*, based on Least Square Method (LSQ). More about this function is mentioned in subsection 4.2.12 on page 53. Its outputs are center point coordinates and value of radius of fitted sphere. *Matlab* realization is shown on lines from 95 to 101 in the Lis. 4.3.

Data rotation where the top of the cup (the pole from the Fig. 4.4) is directed along the x axis follows. The following derivation is inspired by Valášek, et al., 2004 [8]. Let's denote current coordinate system by number '1' and the new one, in which we want to express the data, by number '2'. Coordinate system '1' is characterized by its unit vectors

$$\mathbf{i}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{j}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{k}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.11)$$

Position of coordinate system '2' against coordinate system '1' is expressed by unit vectors

of coordinate system '2' expressed in coordinate system '1'

$${}^1\mathbf{i}_2 = \begin{bmatrix} \cos \alpha_x \\ \cos \beta_x \\ \cos \gamma_x \end{bmatrix}, \quad {}^1\mathbf{j}_2 = \begin{bmatrix} \cos \alpha_y \\ \cos \beta_y \\ \cos \gamma_y \end{bmatrix}, \quad {}^1\mathbf{k}_2 = \begin{bmatrix} \cos \alpha_z \\ \cos \beta_z \\ \cos \gamma_z \end{bmatrix} \quad (4.12)$$

where we used directional angles α, β, γ which occurs between these unit vectors and axes of coordinate system '1'. There is angle α_x between vector \mathbf{i}_2 and axis x_1 , β_x between vector \mathbf{i}_2 and axis y_1 and γ_x between vector \mathbf{i}_2 and axis z_1 . In the same sense angles $\alpha_y, \beta_y, \gamma_y$ are defined against \mathbf{j}_2 and angles $\alpha_z, \beta_z, \gamma_z$ are defined against \mathbf{k}_2 . We are able to transform point P from coordinate system '2' to coordinate system '1' as

$$\begin{bmatrix} x_{P1} \\ y_{P1} \\ z_{P1} \end{bmatrix} = \begin{bmatrix} \cos \alpha_x & \cos \alpha_y & \cos \alpha_z \\ \cos \beta_x & \cos \beta_y & \cos \beta_z \\ \cos \gamma_x & \cos \gamma_y & \cos \gamma_z \end{bmatrix} \begin{bmatrix} x_{P2} \\ y_{P2} \\ z_{P2} \end{bmatrix} \quad (4.13)$$

and as

$$\mathbf{P}_1 = \mathbf{S}_{12}\mathbf{P}_2 \quad (4.14)$$

where \mathbf{S}_{12} is matrix of directional cosines

$$\mathbf{S}_{12} = \begin{bmatrix} \cos \alpha_x & \cos \alpha_y & \cos \alpha_z \\ \cos \beta_x & \cos \beta_y & \cos \beta_z \\ \cos \gamma_x & \cos \gamma_y & \cos \gamma_z \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{i}_2 & {}^1\mathbf{j}_2 & {}^1\mathbf{k}_2 \end{bmatrix} \quad (4.15)$$

However, our case is that we need to transform points from coordinate system '1' to coordinate system '2'

$$\mathbf{P}_2 = \mathbf{S}_{12}^{-1}\mathbf{P}_1 \quad (4.16)$$

We can also write

$$\mathbf{P}_2 = \mathbf{S}_{12}^T\mathbf{P}_1 \quad (4.17)$$

because $\mathbf{S}_{12}^T = \mathbf{S}_{12}^{-1}$, proven as

$$\begin{aligned} \mathbf{S}_{12}^T\mathbf{S}_{12} &= \begin{bmatrix} {}^1\mathbf{i}_2^T \\ {}^1\mathbf{j}_2^T \\ {}^1\mathbf{k}_2^T \end{bmatrix} \begin{bmatrix} {}^1\mathbf{i}_2 & {}^1\mathbf{j}_2 & {}^1\mathbf{k}_2 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{i}_2^T\mathbf{i}_2 & {}^1\mathbf{i}_2^T\mathbf{j}_2 & {}^1\mathbf{i}_2^T\mathbf{k}_2 \\ {}^1\mathbf{j}_2^T\mathbf{i}_2 & {}^1\mathbf{j}_2^T\mathbf{j}_2 & {}^1\mathbf{j}_2^T\mathbf{k}_2 \\ {}^1\mathbf{k}_2^T\mathbf{i}_2 & {}^1\mathbf{k}_2^T\mathbf{j}_2 & {}^1\mathbf{k}_2^T\mathbf{k}_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{S}_{12}^{-1}\mathbf{S}_{12} \end{aligned} \quad (4.18)$$

Equation 4.17 provides easy rotation of coordinate systems. To realize the rotation we have to determine unit vectors ${}^1\mathbf{i}_2$, ${}^1\mathbf{j}_2$ and ${}^1\mathbf{k}_2$ to fill matrix \mathbf{S}_{12} , according to Eq. 4.15. Required orientation of coordinate system '2' is described in the subsection 4.1.1 beginning on page 19. We are going to use already calculated fitted sphere and estimate vector \mathbf{m} oriented from its center point (origin of c.s.) to its pole (Fig. 4.4 on page 23). Direction of vector \mathbf{m} will become x axis of coordinate system '2'. To do so, measured data are aligned into fitted sphere surface. This is realized from line 104 to line 107 in Lis. 4.3. Then, vector \mathbf{m} is estimated as

$$\mathbf{m} = \begin{bmatrix} \frac{x_{max}+x_{min}}{2} & \frac{y_{max}+y_{min}}{2} & \frac{z_{max}+z_{min}}{2} \end{bmatrix} \quad (4.19)$$

Length of vector \mathbf{m} is

$$l_m = \sqrt{x_m^2 + y_m^2 + z_m^2} \quad (4.20)$$

Afterward, desired unit vector ${}^1\mathbf{i}_2$ is calculated as

$${}^1\mathbf{i}_2 = \frac{\mathbf{m}}{l_m} \quad (4.21)$$

Alignment of measured data into fitted sphere is necessary in order to avoid deviations from fitted radius that would cause deflection of the desired direction. The direction of the remaining axes is not important as long as all unit vectors create orthonormal basis. Cross product of two vectors gives a perpendicular vector against both initial vectors. We use this and calculate unit vector ${}^1\mathbf{j}_2$ with only condition that it must be perpendicular to unit vector ${}^1\mathbf{i}_2$

$${}^1\mathbf{j}_2 = \frac{\mathbf{i}_1 \times {}^1\mathbf{i}_2}{\|\mathbf{i}_1 \times {}^1\mathbf{i}_2\|} \quad (4.22)$$

Then, unit vectors ${}^1\mathbf{i}_2, {}^1\mathbf{j}_2, {}^1\mathbf{k}_2$ creates orthonormal basis of coordinate system '2' if

$${}^1\mathbf{k}_2 = {}^1\mathbf{i}_2 \times {}^1\mathbf{j}_2 \quad (4.23)$$

Data rotation is realized from line 109 to line 126 in Lis. 4.3.

Transformed data are marked as 'B data' in calculation Flow chart (Fig. 4.8). Their Cartesian coordinates are transformed to Spherical and Pole coordinates, according to Eq. 4.1 and Eq. 4.3. This process is shown in Lis. 4.4 and Lis. 4.1.

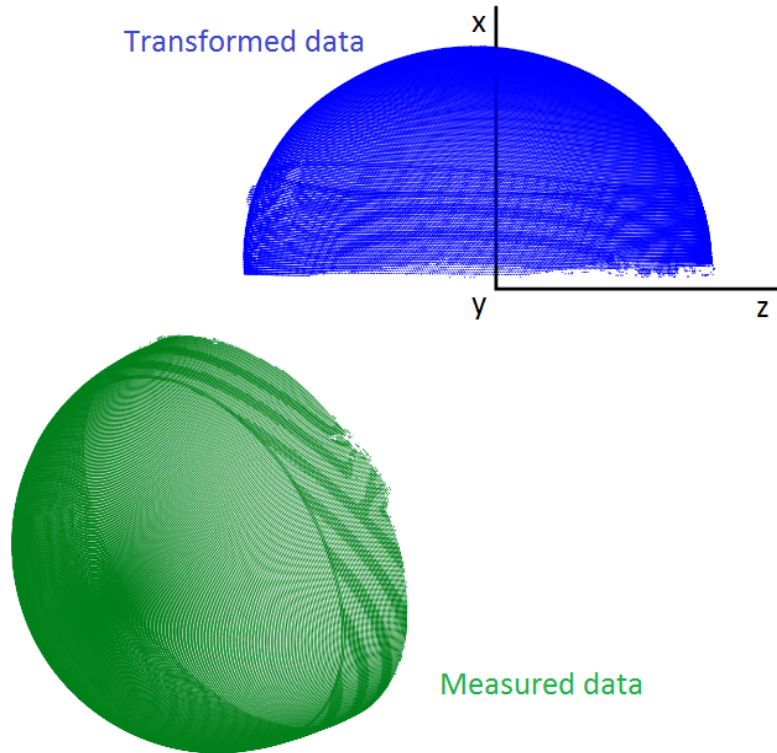


Figure 4.9: Visualization of data transformation

Listing 4.3: *Cup_analysis.m* - Data transformation

```

95 % DATA SPHERE FITTING
96 [Center,Radius] = sphereFit(X);
97
98 % DATA TRANSLATION
99 x = X(:,1)-Center(1);
100 y = X(:,2)-Center(2);
101 z = X(:,3)-Center(3); clear X
102
103 xa=x; ya=y; za=z; % save of data a
104 [r,theta,xi] = x_z_y__r_theta_xi(x,y,z); % spherical coordinates
105 % align fitted radius to all data
106 r(:, :) = Radius;
107 [x,y,z] = r_theta_xi__x_y_z(r,theta,xi);
108
109 % DATA ROTATION ALGORITHM
110
111 % x axis vector of new coordinate system
112 middle = [(max(x)+min(x))/2 (max(y)+min(y))/2 (max(z)+min(z))/2];
113 l = sqrt(middle(1)^2+middle(2)^2+middle(3)^2);
114
115 % unit vectors of new coordinate system axes
116 i2 = middle/l;
117 j2 = cross([1 0 0],i2); l = sqrt(j2(1)^2+j2(2)^2+j2(3)^2); j2 = j2/l;
118 k2 = cross(i2,j2);
119
120 S = [i2(1) j2(1) k2(1) % transformation matrix
121      i2(2) j2(2) k2(2)
122      i2(3) j2(3) k2(3)];
123
124 xb = S(1,1)*xa + S(2,1)*ya + S(3,1)*za; % [x y z]' = S'*[x1 y1 z1]'
125 yb = S(1,2)*xa + S(2,2)*ya + S(3,2)*za;
126 zb = S(1,3)*xa + S(2,3)*ya + S(3,3)*za;

```

Listing 4.4: *Cup_analysis.m* - Save of 'B data'

```

406 x=xb; y=yb; z=zb; % save of data b
407
408 [r,theta,xi] = x_z_y__r_theta_xi(x,y,z);
409 theta_b = theta; xi_b = xi; r_b = r; % save of data b (spherical coordinates)
410
411 [s1,s2] = Radius_theta_xi__s1_s2(Radius,theta,xi);
412 s1b=s1; s2b=s2; % save of data b (pole coordinates)

```

4.2.3 Data interpolation

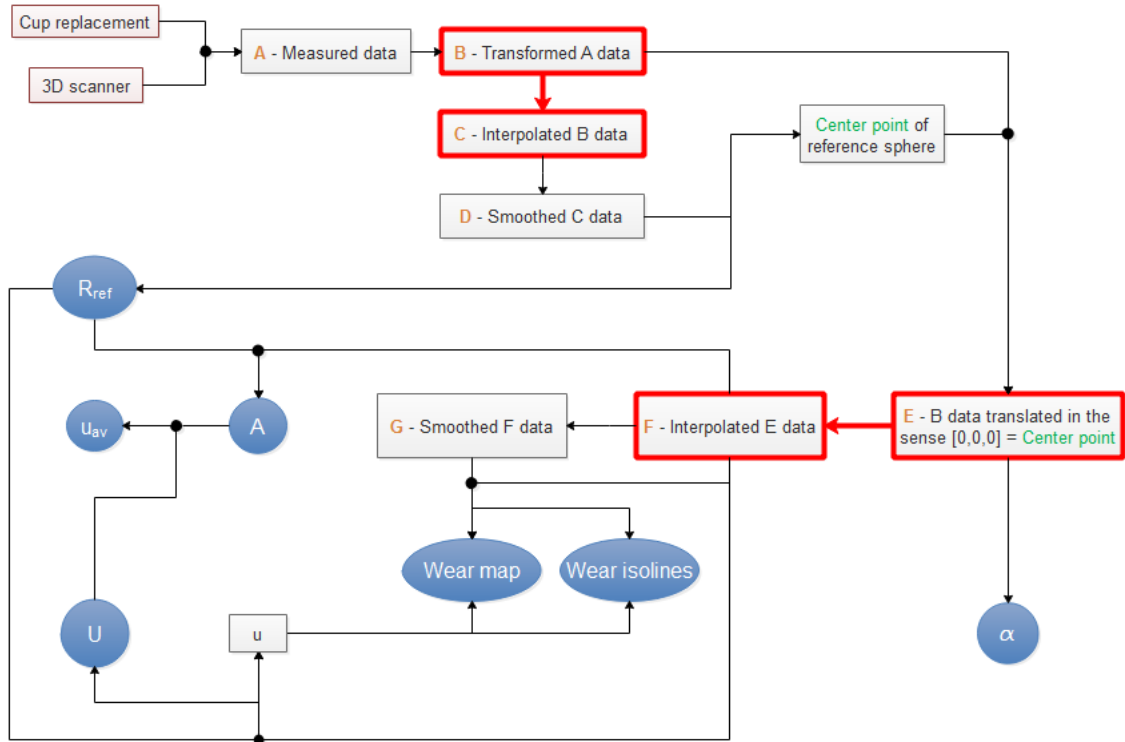


Figure 4.10: Data interpolation

Point cloud of measured cup shall be interpolated by meridians and parallels as described in subsection 4.1.2 on page 25. However it is not possible to do so in Cartesian coordinate system because from top view (observation from infinite point of x axis) some point could be hidden behind another. Deviations from reference radius could cause this problem. We use Pole coordinate system to interpolate measured surface where every measured point is always visible from top view.

scatteredInterpolant function was used for realization of interpolation in *Matlab*. More about this function is mentioned in subsection 4.2.12 on page 53. Interpolation is made four times within the whole calculation. 'B data' are interpolated to 'C data' (Lis. 4.5). 'E data' are interpolated to 'F data' (Fig. 4.11). Also interpolation procedure is part of smoothing algorithm ($C \rightarrow D$ and $F \rightarrow G$), as obvious from Lis. 4.6.

Listing 4.5: *Cup_analysis.m* - Data interpolation - $B \rightarrow C$

```

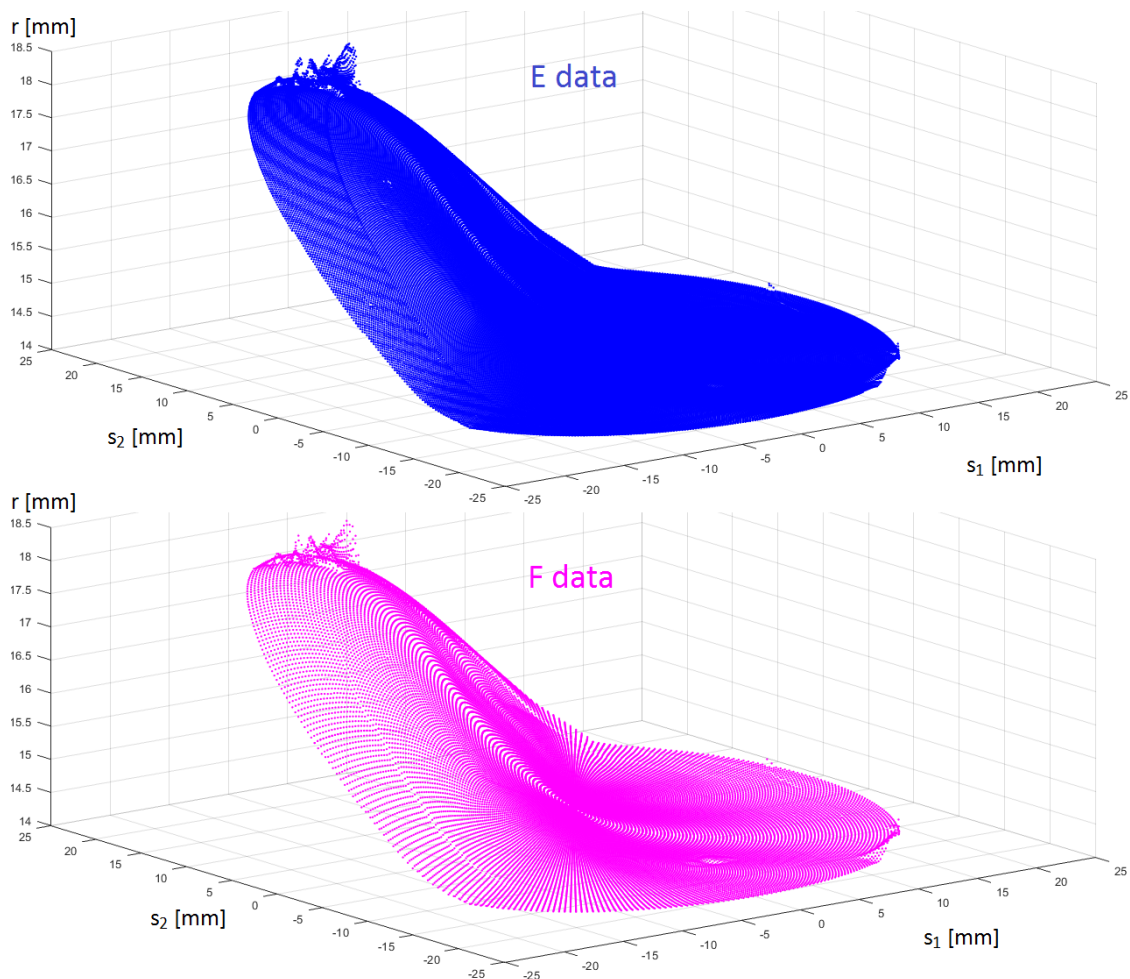
418 % INTERPOLATION OF b DATA VIA scatteredInterpolant FUNCTION
419 F = scatteredInterpolant(s1b,s2b,r_b,'natural','nearest');
420 xii = 0:step:max(xi_b);
421 thetai = -pi+step:step:pi;
422 [xi,theta]=meshgrid(xii,thetai);

```

```

423 [s1,s2] = Radius_theta_xi__s1_s2(Radius,theta,xi);
424 % save of data c (pole coordinates)
425 s1cd = s1; s2cd = s2;
426 r_c = F(s1cd,s2cd); r=r_c;
427
428 % save of data c (spherical coordinates)
429 [theta,xi] = Radius_s1_s2__theta_xi(Radius,s1,s2);
430 theta_cd = theta; xi_cd =xi;
431
432 % save of data c
433 [x,y,z] = r_theta_xi__x_y_z(r,theta,xi);
434 xc=x; yc=y; zc=z;

```

Figure 4.11: Data interpolation - $E \rightarrow F$

4.2.4 Data smoothing

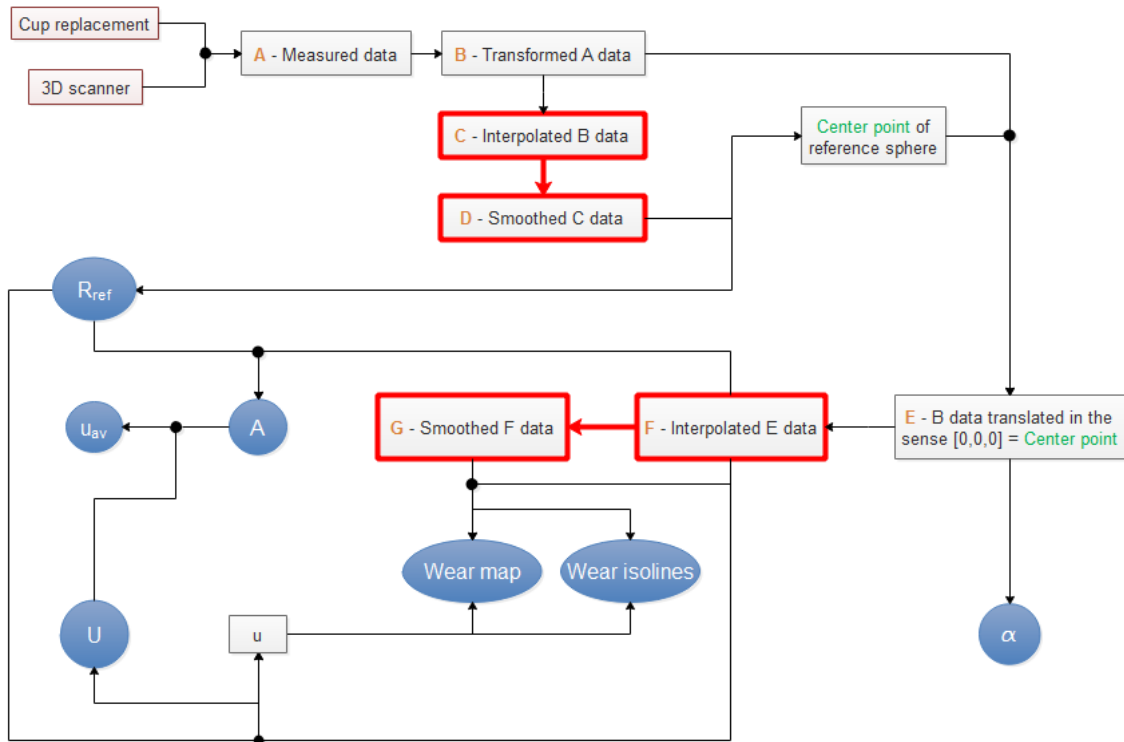


Figure 4.12: Data smoothing

Interpolated data have to be smoothed to avoid impurities that appeared on cup surface during measurement or to smooth scratches and other deviations not caused by wear. However, we cannot use the Cartesian coordinate system due to the same reason as data interpolation must use Pole coordinate system (previous subsection 4.2.3). Therefore, the Pole coordinate system is used to smooth data.

gridfit function was used for realization of smoothing in *Matlab*. More about this function is mentioned in subsection 4.2.12 on page 53. Smoothing is made twice within the whole calculation. 'C data' are smoothed to 'D data' (Lis. 4.6). 'F data' are smoothed to 'G data' (Fig. 4.13).

Listing 4.6: *Cup_analysis.m* - Data smoothing - $C \rightarrow D$

```

436 % SMOOTHING OF DATA c TO DATA d
437 xnodes = min(s1cd(:)):0.5*step*Radius:max(s1cd(:)); ynodes = min(s2cd(:))
      :0.5*step*Radius:max(s2cd(:));
438 x = s1cd; y = s2cd; z = r_c;
439 subst = str2double(get(handles.edit6,'String')); % sets chosen smoothing
      parameter
440 [zgrid,xgrid,ygrid] = gridfit(x,y,z,xnodes,ynodes);
441 G = scatteredInterpolant(xgrid(:),ygrid(:),zgrid(:),'natural','nearest');
442
443 % save of data d

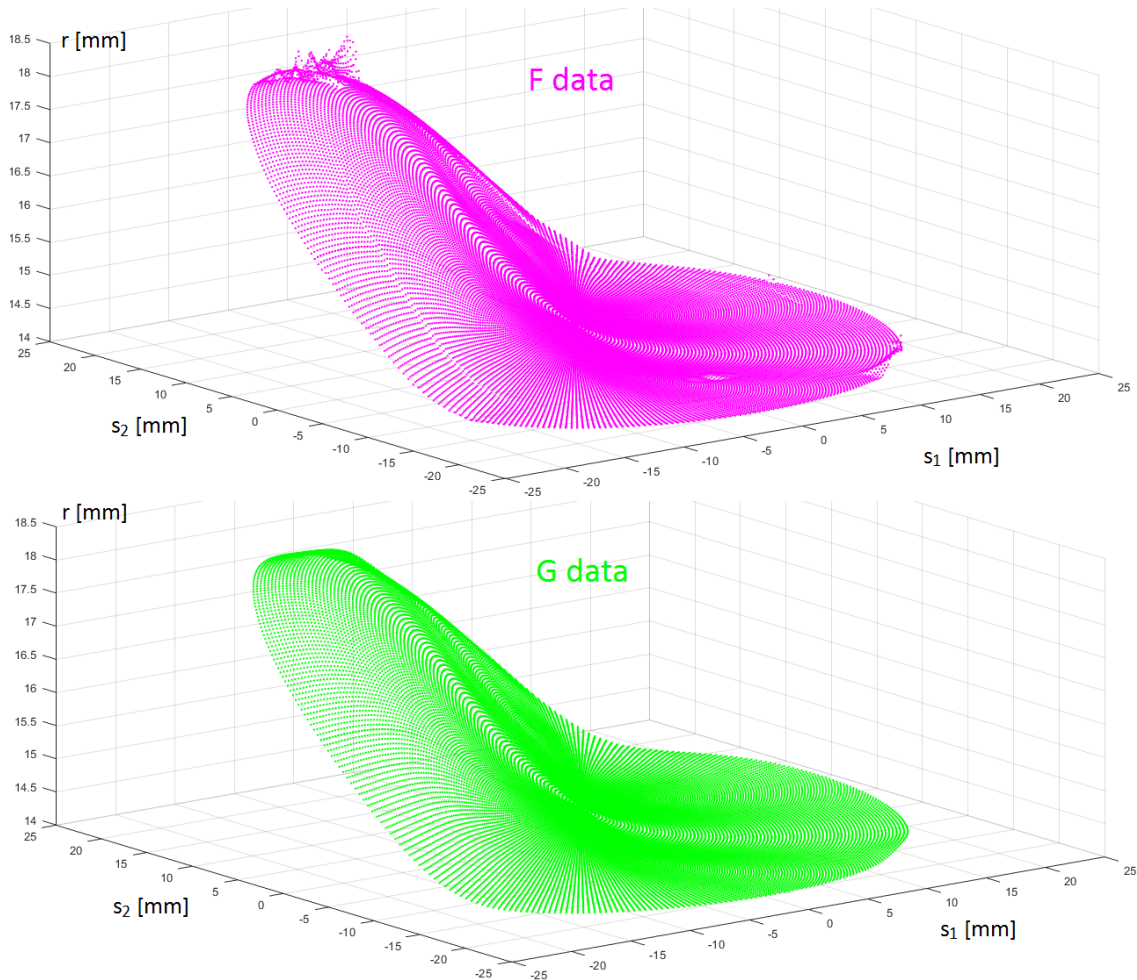
```



```

444 r_d = G(s1cd,s2cd);
445 r = r_d;
446 [x,y,z] = r_theta_xi_x_y_z(r,theta,xi);
447 xd=x; yd=y; zd=z;

```

Figure 4.13: Data smoothing - $F \rightarrow G$

4.2.5 Reference sphere estimation

Estimation of reference sphere is a crucial part of the whole calculation because wear parameters are derived by reference sphere. We are looking for Reference radius R_{ref} and coordinates of reference sphere center $[x_s, y_s, z_s]$.

Position of acetabular component of hip joint replacement (cup) in body and dominant loading result in wear just in a certain area of the cup surface while the rest of the cup surface is in its original state. We are going to focus on this part of original surface. We will try to identify it and align reference sphere on it. That is the major idea of reference geometry estimation. Identification of original surface uses international standard *ISO*

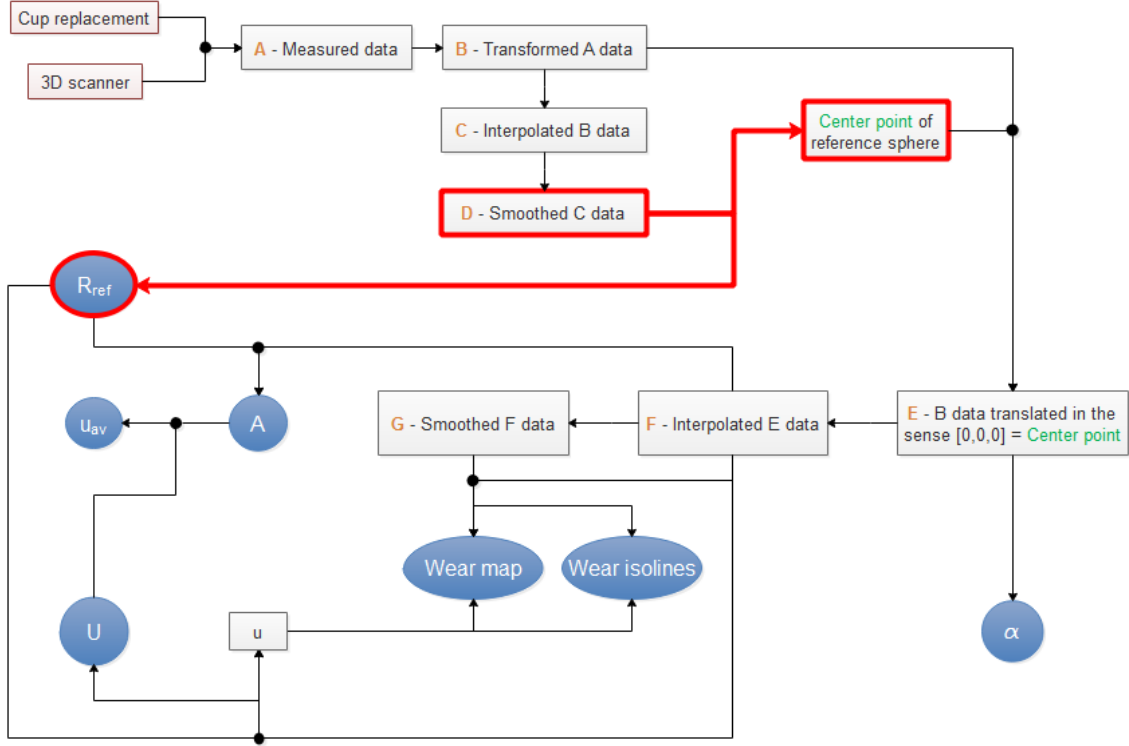


Figure 4.14: Reference sphere estimation

7206-2:1996(E) called *Implants for surgery - Partial and total hip joint prostheses* [9]. In the part dealing with plastics acetabular components, the standard says that “The spherical socket shall have a diameter equal to the nominal diameter within a tolerance of $+0.1$ mm to $+0.3$ mm at the temperature of $20\text{ }^{\circ}\text{C} \pm 2\text{ }^{\circ}\text{C}$ (i.e. it shall be oversized within the given tolerance).” Accordingly, we know values of tolerance boundaries R_{min} and R_{max} from the standard. The task now is to find a center point of nominal sphere whose tolerance boundaries contain the largest value of original surface in between each other, as shown in Fig. 4.15.

A local method of static optimization is used for the estimation of center point. Optimization parameters are Cartesian coordinates of center point $[x_s, y_s, z_s]$. The aim of optimization process is maximization of objective function B_{red_i} which is the value of Sample (cup) surface inside tolerance boundaries (given by Eq 4.26 and Eq. 4.27).

$$\max(B_{red}(x_s, y_s, z_s)) \quad (4.24)$$

We are able to roughly identify the original unworn area on the cup explant by simple eye look. Therefore, GUI enables the user to constrain the unworn area - to reduce cup area which is going to be included in this part of the calculation. It is worth to use this constriction to avoid wrong convergence of local method of static optimization and to reduce calculation time.

Firstly, we have to find a suitable initial center point for static optimization. To do so, we calculate local radius of every point (‘D data’) of surface specified in GUI. Locality is given by *Angle of locality*, which is specified by the user in the created GUI program.

Angle of locality defines the surrounding points of the investigated point involved in the calculation of the local radius of investigated point. It is the maximal permitted angle between two vectors (first vector - given by coordinates of investigated point, second vector - given by coordinates of some other certain point). If the angle between these two vectors is lower than *Angle of locality*, the certain point is involved into the local radius calculation of the investigated point. This angle is calculated by scalar product of these two vectors (lines 1015 to 1017 in Lis. 4.7). When we know all the points included in the local radius calculation of investigated point, we apply *sphereFit* LSQ method for fitting sphere into selected '*D data*' (more about *sphereFit* in subsection 4.2.12 on page 53). If the value of fitted radius is between tolerance boundaries R_{min} and R_{max} we investigate the value of Sample surface inside tolerance boundaries B_{red} (center point of nominal sphere moved to center point calculated by *sphereFit*). This value is stored and compared with values of Sample surface inside tolerance boundaries B_{red} of other points whose local radius fulfilled tolerance criterion. Center point (calculated by *sphereFit*) of the point whose Sample surface inside tolerance boundaries B_{red} is the largest one, becomes initial point for static optimization. Calculation of initial point is visible in lines 1011 to 1045 in Lis. 4.7.

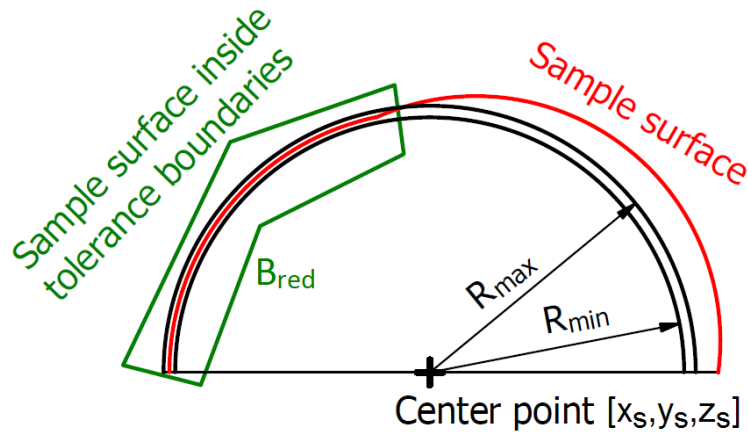


Figure 4.15: Tolerance boundaries

Static optimization is visible in lines 1052 to 1140 in Lis. 4.7. It is a local method of static optimization, so once it finds a maximum of objective function the calculation is terminated. The idea of this optimization method is to move center point of nominal sphere and observe how the value of Sample surface inside tolerance boundaries B_{red} is changing. Local method of static optimization applied here works very simply. In each iteration it takes center point of nominal sphere and moves it by the value of optimization step (specified in created GUI program by the user) to six directions ($+x, -x, +y, -y, +z, -z$, according to coordinate system). Then it chooses the largest value of Sample surfaces inside tolerance boundaries B_{red} , gained by the movements, and compares it with current value of Sample surface inside tolerance boundaries B_{red} . Current Sample surface inside tolerance boundaries B_{red} is replaced by the new one if the new one is larger than the current one. In that case, current center point of nominal sphere is also replaced by the new one (current center point of nominal sphere is moved by value of optimization

step in the direction where the largest Sample surface inside tolerance boundaries B_{red} appeared). Within the first iteration where the value of Sample surface inside tolerance boundaries B_{red} does not increase, calculation is terminated and the last center point of nominal sphere becomes reference sphere center point $[x_s, y_s, z_s]$.

When center point $[x_s, y_s, z_s]$ and Sample surface inside tolerance boundaries B_{red} are known, we can calculate **Reference radius R_{ref} as weighted average**. We consider only Sample surface inside tolerance boundaries B_{red} given by points i ('D data') where $i = 1, 2, \dots, n$. Weighted average is then given as

$$R_{ref} = \frac{\sum_{i=1}^n r_i \cdot B_{red_i}}{B_{red}} \quad (4.25)$$

where r_i is distance between center point and point i . Area represented by point i is calculated according to Eq. 4.8 as

$$B_{red_i} = R_{LSQ}^2 \cdot \Delta \cdot \left[\cos\left(\xi_{D_i} - \frac{\Delta}{2}\right) - \cos\left(\xi_{D_i} + \frac{\Delta}{2}\right) \right] \quad (4.26)$$

where radius R_{LSQ} is calculated by LSQ method within data transformation (subsection 4.2.2 on page 28) and Δ is Interpolation step (subsection 4.1.2 on page 25). Sample surface inside tolerance boundaries B_{red} is then given as

$$B_{red} = \sum_{i=1}^n B_{red_i} \quad (4.27)$$

Listing 4.7: *Cup_analysis.m* - Reference sphere estimation

```

984 % ESTIMATION OF REFERENCE GEOMETRY
985 function [xs,ys,zs,R_ref,announcement] = LocalRadius_fcn_Optimition(xd,yd,zd,
    r_d,step,Radius,xi_cd,locality_angle,r_max,r_min,krok,xi_min,xi_max,
    theta_min,theta_max,theta_cd)
986
987 xCut = xd; yCut = yd; zCut = zd; % data stored for eventual Cut off method
988
989 B = (Radius^2)*step.*(cos(xi_cd-step/2)-cos(xi_cd+step/2)); % [mm^2] area
    matrix
990
991 % DECIDES WHICH DATA ARE APPROPRIATE FOR CALCULATION ACCORDING TO THETA_MIN,
    THETA_MAX; XI_MIN, XI_MAX
992 if theta_min < theta_max
993     log_cropped_theta = (theta_cd >= theta_min*pi/180) & (theta_cd <=
        theta_max*pi/180);
994 else
995     log_cropped_theta = (theta_cd >= theta_min*pi/180) | (theta_cd <=
        theta_max*pi/180);
996 end
997 xd = xd(log_cropped_theta);
998 yd = yd(log_cropped_theta);
999 zd = zd(log_cropped_theta);

```

```

1000     r_d = r_d(log_cropped_theta);
1001     B = B(log_cropped_theta);
1002     xi_cd = xi_cd(log_cropped_theta);
1003
1004     log_cropped_xi = (xi_cd >= xi_min*pi/180) & (xi_cd <= xi_max*pi/180);
1005     xd = xd(log_cropped_xi);
1006     yd = yd(log_cropped_xi);
1007     zd = zd(log_cropped_xi);
1008     r_d = r_d(log_cropped_xi);
1009     B = B(log_cropped_xi);
1010
1011     B_red = -1; % initial indicator
1012     i_end = size(r_d);
1013     for i = 1:i_end(1)
1014
1015         % CHOOSE DATA FOR SPHERE FIT LSQ METHOD ACCORDING TO ANGLE OF LOCALITY
1016         dot_uv = xd*xd(i) + yd*yd(i) + zd*zd(i);
1017         alpha = acos(dot_uv./(r_d*r_d(i)));
1018         log = locality_angle>=alpha;
1019         X(:,1) = xd(log); X(:,2) = yd(log); X(:,3) = zd(log);
1020
1021         [Center,Radius] = sphereFit(X);
1022         clear X
1023
1024         % IS REALIZED IF CALCULATED RADIUS SUITS INTO TOLERANCE
1025         if (Radius>=r_min) && (Radius<=r_max)
1026
1027             xn = xd - Center(1); yn = yd - Center(2); zn = zd - Center(3);
1028             r = (xn.^2 + yn.^2 + zn.^2).^0.5;
1029
1030             log2 = (r >= r_min) & (r <= r_max);
1031             compare = sum(B(log2));
1032
1033             % IS THE AREA CONTAINED BETWEEN TWO SPHERES (R_min AND R_MAX
1034             % RADIUSSES) LARGER THAN THE PREVIOUS ONE?
1035             if compare > B_red
1036
1037                 % IF YES...LAST VALUES ARE OVERWRITTEN BY THE CURRENT ONES
1038                 B_red = compare;
1039                 R_ref = sum(r(log2).*B(log2))/B_red;
1040                 xs = Center(1);
1041                 ys = Center(2);
1042                 zs = Center(3);
1043             end
1044         end
1045     end

```

```
1046
1047 % DECIDES IF ANY INVESTIGATED POINT FULFILLED CONDITIONS
1048 if B_red == -1 % if not, then CutOff fcn is applied
1049     disp('No original surface found => CutOff method applied');
1050     announcement= 'CutOff method applied.';
1051     [xs,ys,zs,R_ref] = CutOff_fcn(xCut,yCut,zCut,r_max,r_min);
1052 else % if yes, then static optimization is applied
1053     disp('LocalRadius method and upcoming Optimization applied');
1054     announcement= 'LocalRadius method and Optimization applied.';
1055     i = 0;
1056     while 1 % moves center point of reference geometry in 6 directions and
           investigates if the situation is better somewhere
1057
1058         % +x
1059         x_xpos = xd-xs -krok;    y_xpos = yd-ys;        z_xpos = zd-zs;
1060         r_xpos = (x_xpos.^2 + y_xpos.^2 + z_xpos.^2).^0.5;
1061
1062         log1 = (r_xpos >= r_min) & (r_xpos <= r_max);
1063         B_xpos = sum(B(log1));
1064
1065         % -x
1066         x_xneg = xd-xs +krok;    y_xneg = yd-ys;        z_xneg = zd-zs;
1067         r_xneg = (x_xneg.^2 + y_xneg.^2 + z_xneg.^2).^0.5;
1068
1069         log2 = (r_xneg >= r_min) & (r_xneg <= r_max);
1070         B_xneg = sum(B(log2));
1071
1072         % +y
1073         x_ypos = xd-xs;        y_ypos = yd-ys -krok;    z_ypos = zd-zs;
1074         r_ypos = (x_ypos.^2 + y_ypos.^2 + z_ypos.^2).^0.5;
1075
1076         log3 = (r_ypos >= r_min) & (r_ypos <= r_max);
1077         B_ypos = sum(B(log3));
1078
1079         % -y
1080         x_yneg = xd-xs;        y_yneg = yd-ys +krok;    z_yneg = zd-zs;
1081         r_yneg = (x_yneg.^2 + y_yneg.^2 + z_yneg.^2).^0.5;
1082
1083         log4 = (r_yneg >= r_min) & (r_yneg <= r_max);
1084         B_yneg = sum(B(log4));
1085
1086         % +z
1087         x_zpos = xd-xs;        y_zpos = yd-ys;        z_zpos = zd-zs -krok
           ;
1088         r_zpos = (x_zpos.^2 + y_zpos.^2 + z_zpos.^2).^0.5;
1089
1090         log5 = (r_zpos >= r_min) & (r_zpos <= r_max);
```

```

1091     B_zpos = sum(B(log5));
1092
1093     % -z
1094     x_zneg = xd-xs;          y_zneg = yd-ys;          z_zneg = zd-zs +krok
        ;
1095     r_zneg = (x_zneg.^2 + y_zneg.^2 + z_zneg.^2).^0.5;
1096
1097     log6 = (r_zneg >= r_min) & (r_zneg <= r_max);
1098     B_zneg = sum(B(log6));
1099
1100
1101     A1 = [B_red,          B_xneg,B_ypos,B_yneg,B_zpos,B_zneg];
1102     A2 = [B_red,B_xpos,          B_ypos,B_yneg,B_zpos,B_zneg];
1103     A3 = [B_red,B_xpos,B_xneg,          B_yneg,B_zpos,B_zneg];
1104     A4 = [B_red,B_xpos,B_xneg,B_ypos,          B_zpos,B_zneg];
1105     A5 = [B_red,B_xpos,B_xneg,B_ypos,B_yneg,          B_zneg];
1106     A6 = [B_red,B_xpos,B_xneg,B_ypos,B_yneg,B_zpos          ];
1107
1108
1109     % if it is better somewhere, then last center point is overwritten by
        the new one and new reference radius is calculated
1110     if B_xpos > max(A1)
1111         xs = xs + krok;
1112         B_red = B_xpos;
1113         R_ref = sum(r_xpos(log1).*B(log1))/B_red;
1114     elseif B_xneg > max(A2)
1115         xs = xs - krok;
1116         B_red = B_xneg;
1117         R_ref = sum(r_xneg(log2).*B(log2))/B_red;
1118     elseif B_ypos > max(A3)
1119         ys = ys + krok;
1120         B_red = B_ypos;
1121         R_ref = sum(r_ypos(log3).*B(log3))/B_red;
1122     elseif B_yneg > max(A4)
1123         ys = ys - krok;
1124         B_red = B_yneg;
1125         R_ref = sum(r_yneg(log4).*B(log4))/B_red;
1126     elseif B_zpos > max(A5)
1127         zs = zs + krok;
1128         B_red = B_zpos;
1129         R_ref = sum(r_zpos(log5).*B(log5))/B_red;
1130     elseif B_zneg > max(A6)
1131         zs = zs - krok;
1132         B_red = B_zneg;
1133         R_ref = sum(r_zneg(log6).*B(log6))/B_red;
1134     else % if not, static optimization is terminated
1135         break

```

```

1136     end
1137     i = i+1;
1138 end
1139 NumberOfIterations = i
1140 end

```

Eventually, it is possible that no original surface is identified on cup surface (no local radius fulfills the tolerance criterion). In that case, calculation switches to **standby method of reference geometry estimation**.

This method does not focus on original cup surface. It is based on cutting off the maximal values of deviations from fitted sphere calculated by *sphereFit* (more about *sphereFit* in subsection 4.2.12 on page 53). This method consists of 7 iterations. In each iteration *sphereFit* fits sphere into Cartesian coordinates of the 'D data' left and translates these 'D data' by the value of gained center points' coordinates of fitted sphere. Afterwards, 20% of points with largest distance from the origin are deleted. That is the end of the iteration. After 7 cycles there is about 21% of all 'D data' left. Values of center points coordinates calculated by *sphereFit* are stored. After all the cycles they are summarized and the summarization represents center point of reference sphere $[x_s, y_s, z_s]$.

Reference radius R_{ref} is then given as

$$R_{ref} = \frac{r_{max} + r_{min}}{2} \quad (4.28)$$

where r mean distances between calculated center point and individual 'D data' points left after 7 iterations.

However this method does not provide as accurate results as we would like to obtain. It is recommended to modify calculation parameters in created GUI program and repeat the calculation.

Listing 4.8: *Cup_analysis.m* - Standby method of reference sphere estimation

```

1144 % ESTIMATION OF REFERENCE GEOMETRY (IN CASE WHEN LOCAL RADIUS METHOD DID NOT
      SUCCEED)
1145 function [xs,ys,zs,R_ref] = CutOff_fcn(xCut,yCut,zCut,r_max,r_min)
1146
1147 Cx = reshape(xCut,[],1); Cy = reshape(yCut,[],1); Cz = reshape(zCut,[],1);
1148 X(:,1) = Cx; X(:,2) = Cy; X(:,3) = Cz;
1149
1150 xs = 0; ys = 0; zs = 0;
1151
1152 for i=1:7 % 7 cycles ... from 100% data to approximately 21%
1153
1154     [Center] = sphereFit(X);
1155
1156     X(:,1) = X(:,1)-Center(1);
1157     X(:,2) = X(:,2)-Center(2);
1158     X(:,3) = X(:,3)-Center(3);
1159
1160     r = (X(:,1).^2 + X(:,2).^2 + X(:,3).^2).^0.5;

```



```

1161     r_sort = sort(r, 'descend'); size_r_sort = size(r_sort);
1162     r_crit = r_sort(round(0.2*size_r_sort(1,1)),1); % EACH CYCLE CUTS OFF
           20% OF MAXIMAL VALUES
1163
1164     x = X(:,1); y = X(:,2); z = X(:,3); clear X
1165
1166     log = r > r_crit;
1167     x(log) = NaN; y(log) = NaN; z(log) = NaN; r(log) = NaN;
1168
1169     x = x(~isnan(x)); X(:,1) = x;
1170     y = y(~isnan(y)); X(:,2) = y;
1171     z = z(~isnan(z)); X(:,3) = z;
1172
1173     xs = xs + Center(1); ys = ys + Center(2); zs = zs + Center(3);
1174 end
1175 R_ref = (r_max+r_min)/2;

```

4.2.6 Positioning of measured surface against reference sphere

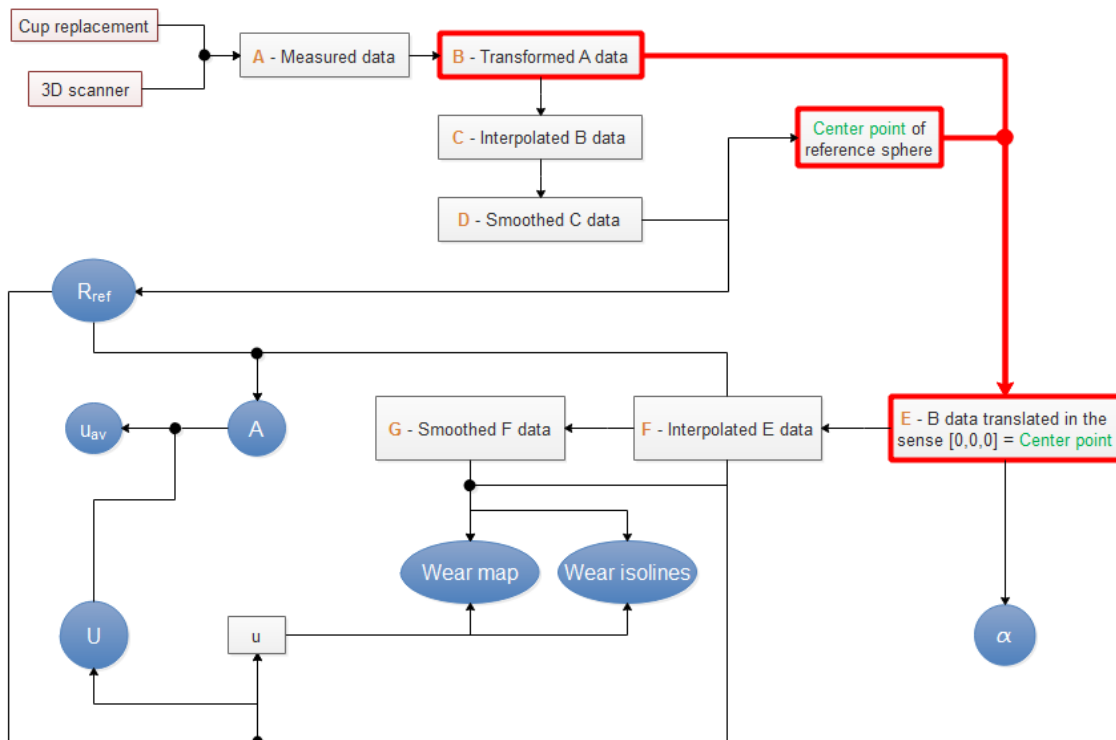


Figure 4.16: Positioning of measured surface against reference sphere

'B data' are going to be positioned against reference sphere whose center point is given by $[x_s, y_s, z_s]$ in Cartesian coordinate system in which 'B data' are expressed. We are going to translate origin of coordinate system into center point $[x_s, y_s, z_s]$. 'B data'

are collectively translated by the value of $[x_s, y_s, z_s]$. We reach 'E data' of each individual point i as

$$x_{E_i} = x_{B_i} - x_s, \quad y_{E_i} = y_{B_i} - y_s, \quad z_{E_i} = z_{B_i} - z_s \quad (4.29)$$

'E data' are re-interpolated and re-smoothed again as described in subsection 4.2.3 on page 32 and in subsection 4.2.4 on page 34. Thereafter we have all the required data to calculate and visualize outputs.

Listing 4.9: *Cup_analysis.m* - Positioning of measured surface against reference sphere

```

461 % save of data e
462 xe = xb - xs; x = xe;
463 ye = yb - ys; y = ye;
464 ze = zb - zs; z = ze;
465
466 % save of data e (spherical coordinates)
467 [r,theta,xi] = x_z_y__r_theta_xi(x,y,z);
468 theta_e = theta; xi_e = xi; r_e = r;
469
470 % save of data e (pole coordinates)
471 Radius = R_ref;
472 [s1,s2] = Radius_theta_xi__s1_s2(Radius,theta,xi);
473 s1e=s1; s2e=s2;

```

4.2.7 Maximal measured angle α

Maximal measured angle α is the angle between x axis and hem of measured surface (Fig. 4.17). Value of angle α is very useful for evaluating the results. The larger the Maximal measured angle α is, the more valuable the results we obtain from the whole calculation because we are interested in wear parameters of whole explanted cup, not just a certain section of it. Therefore, this angle was involved in final outputs of whole calculation. Maximal measured angle α is chosen as ξ_{max} of all 'E data' (Spherical coordinates).

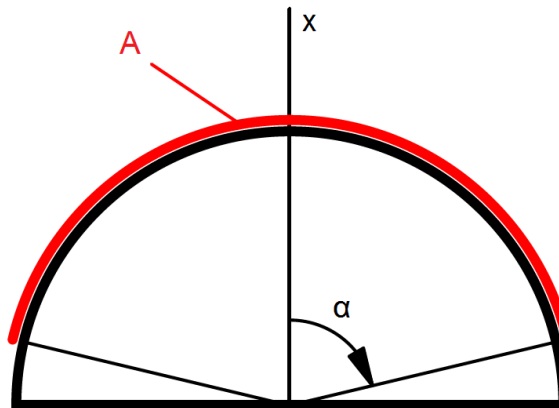
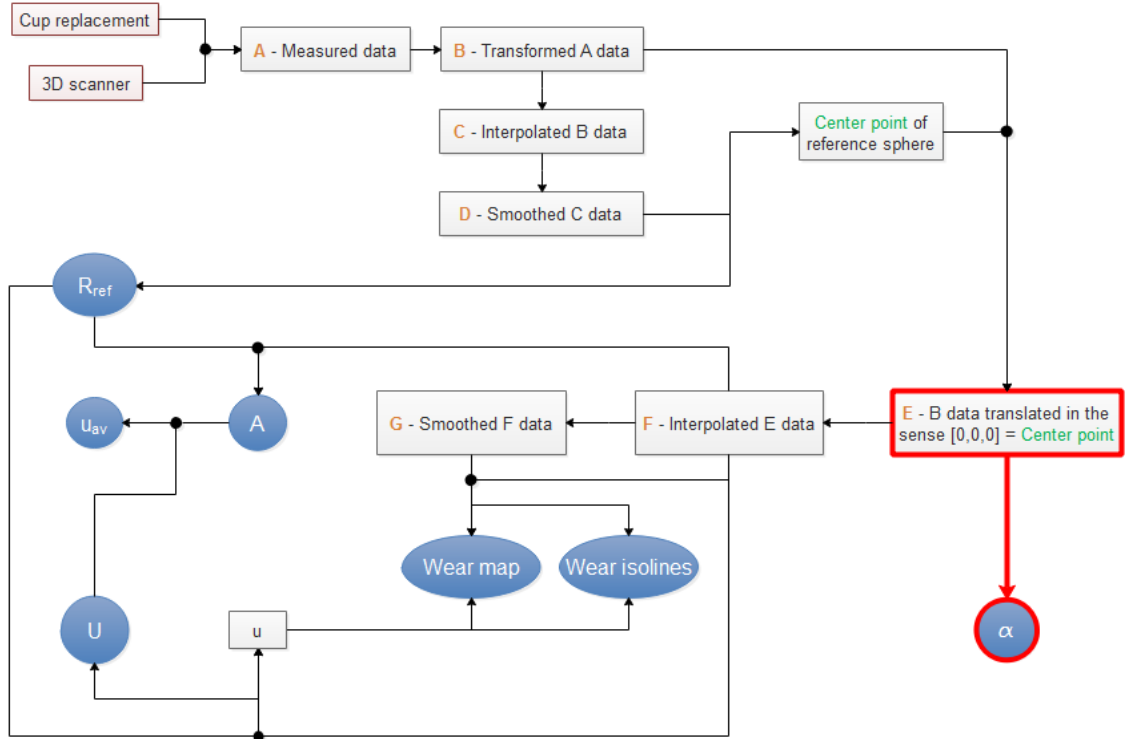


Figure 4.17: Definition of Maximal measured angle α and Measured area A

Figure 4.18: Maximal measured angle α Listing 4.10: *Cup_analysis.m* - Maximal measured angle α

```
531 alfa = max(xi_e)*180/pi; assignin('base','alfa',alfa) % [deg]
```

4.2.8 Measured area A

Measured area A is part of the **reference** sphere area which was measured by 3D scanner, as shown in the Fig. 4.17 on page 44. We are able to calculate its value due to interpolation of measured point cloud. We know the value of local area B_i represented by each individual point i of 'F data'. Within knowledge of Eq. 4.8 we can write

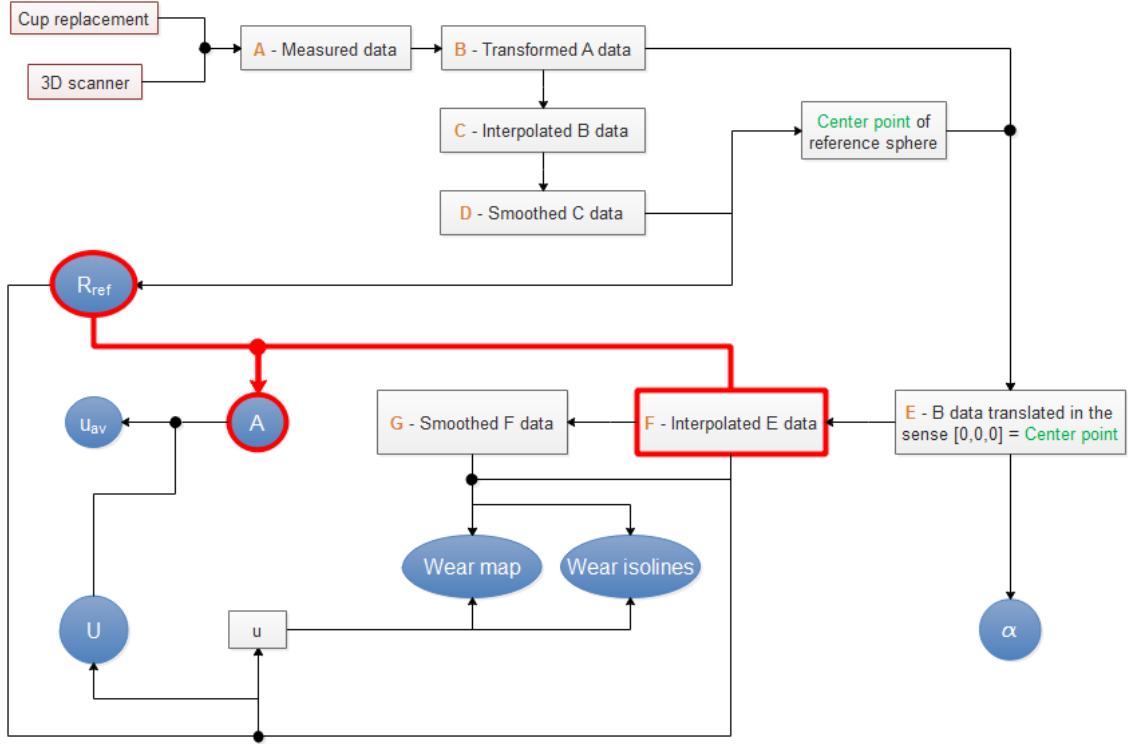
$$B_i = R_{ref}^2 \cdot \Delta \cdot \left[\cos \left(\xi_{F_i} - \frac{\Delta}{2} \right) - \cos \left(\xi_{F_i} + \frac{\Delta}{2} \right) \right] \quad (4.30)$$

where R_{ref} is Reference radius (gained from Eq. 4.25 or Eq. 4.28), Δ is Interpolation step (subsection 4.1.2 on page 25). Measured area A is then calculated as

$$A = \sum_{i=1}^n B_i \quad (4.31)$$

Listing 4.11: *Cup_analysis.m* - Measured area A

```
522 B = (R_ref^2)*step.*(cos(xi_fg-step/2)-cos(xi_fg+step/2)); A = sum(B(:));
    assignin('base','A',A) % [mm^2] area matrix
```


 Figure 4.19: Measured area A

4.2.9 Volumetric wear U

Volumetric wear U is the volume of material which has been lost during the years. According to Eq. 4.10, we calculate Volumetric wear U by following sequence of equations

$$V_{ref_i} = \frac{R_{ref}^3}{3} \cdot \Delta \cdot \left[\cos \left(\xi_{F_i} - \frac{\Delta}{2} \right) - \cos \left(\xi_{F_i} + \frac{\Delta}{2} \right) \right] \quad (4.32)$$

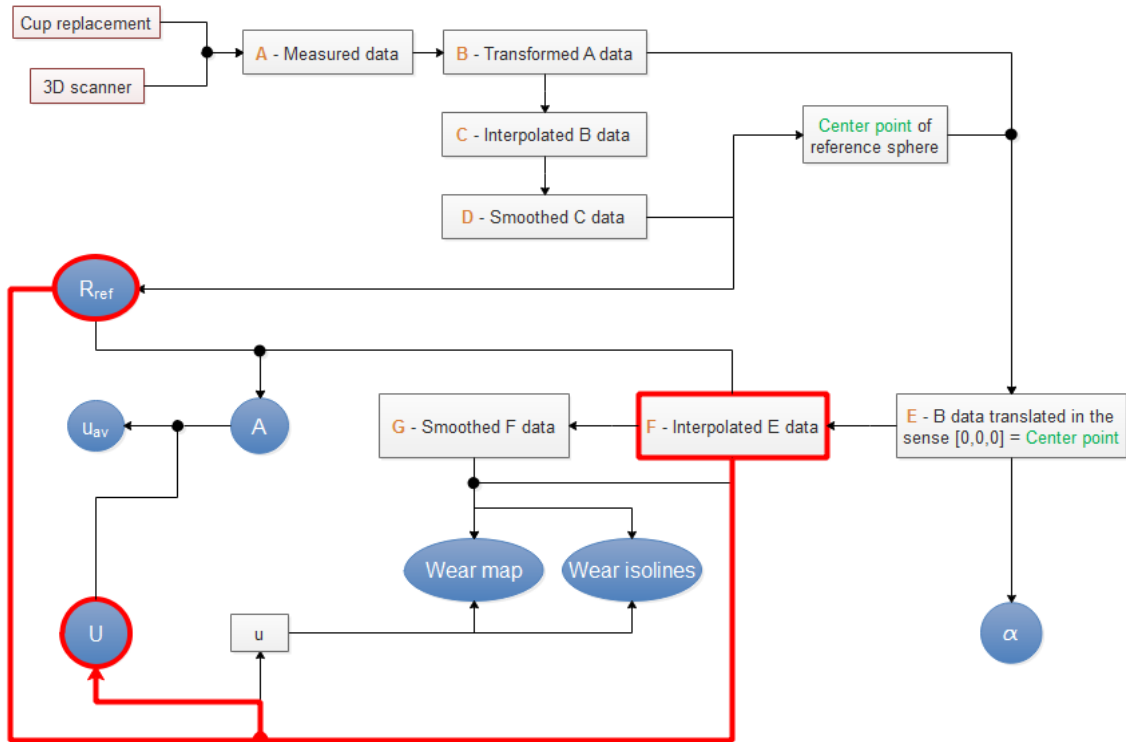
$$V_{F_i} = \frac{r_{F_i}^3}{3} \cdot \Delta \cdot \left[\cos \left(\xi_{F_i} - \frac{\Delta}{2} \right) - \cos \left(\xi_{F_i} + \frac{\Delta}{2} \right) \right] \quad (4.33)$$

$$U_i = V_{F_i} - V_{ref_i}, \quad \forall U_i < 0: U_i = 0 \quad (4.34)$$

$$U = \sum_{i=1}^n U_i \quad (4.35)$$

where R_{ref} is Reference radius (gained from Eq. 4.25 or Eq. 4.28), Δ is Interpolation step (subsection 4.1.2 on page 25). There is also condition that any Local volumetric wear U_i of individual point i (' F data') cannot be lower than 0 because we assume no plastic deformation nor creep.

Volumetric wear U is (together with visualization of wear) a major output of the whole calculation. Its value is important to know because the amount of released material into joint neighbourhood has huge influence on the lifetime of the total hip replacement.

Figure 4.20: Volumetric wear U Listing 4.12: *Cup_analysis.m* - Volumetric wear U

```

524 % VOLUMETRIC WEAR CALCULATION
525 V_ref = (1/3)*(R_ref^3)*step.*(cos(xi_fg-step/2)-cos(xi_fg+step/2));
526 V_f   = (1/3)*(r_f.^3)*step.*(cos(xi_fg-step/2)-cos(xi_fg+step/2));
527 logV = V_f<V_ref; V_ref(logV) = 0; V_f(logV) = 0; % assumed no plastic
    deformation nor creep
528 U_matrix = V_f - V_ref; U = sum(U_matrix(:)); assignin('base','U',U); % [mm
    ^3] volumetric wear

```

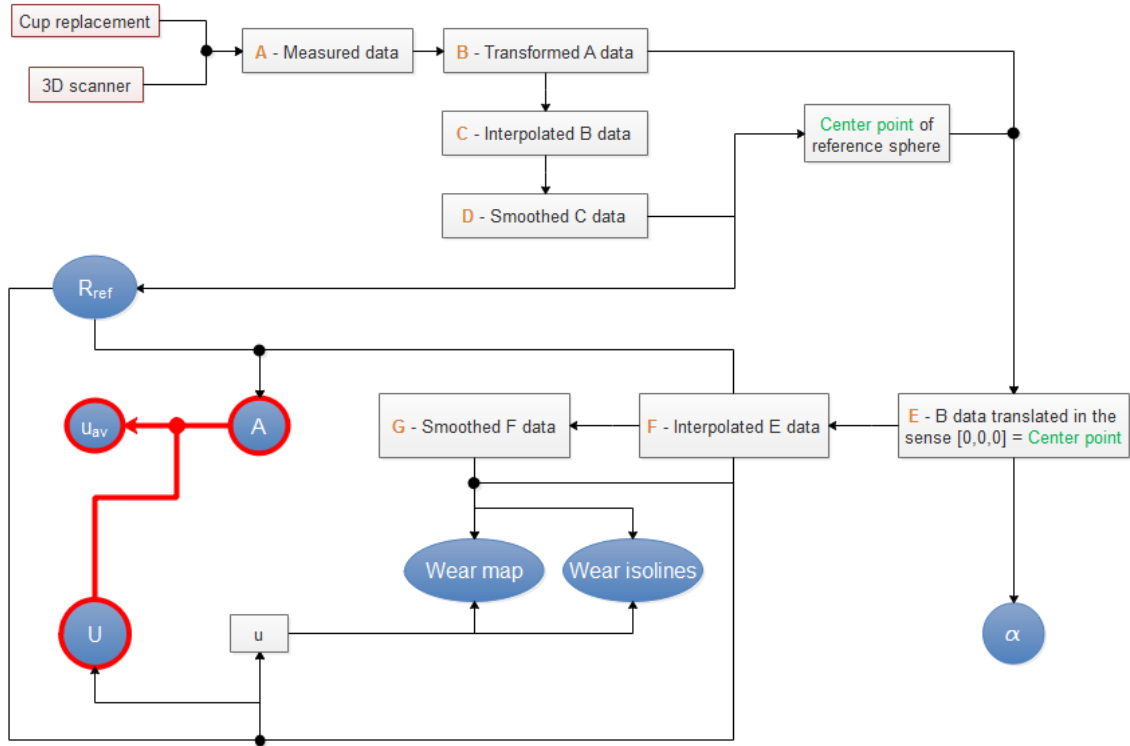
4.2.10 Average linear wear u_{av}

Another output we decided to calculate, is Average linear wear u_{av} . It is given by

$$u_{av} = \frac{U}{A} \quad (4.36)$$

where U is Volumetric wear and A is Measured area.

Not all cups were produced by the same nominal sphere diameter and not all explanted cups were measured by the same Maximal measured angle α and the same Measured area A . In that case, we cannot compare values of volumetric wear U between each other. Advantage of Average linear wear u_{av} is that its value is comparable with other Average linear wear u_{av} values of all measured and analyzed cups. Basically Average linear wear

Figure 4.21: Average linear wear u_{av}

u_{av} means how many cubic millimeters of material has been lost per 1 square millimeter of reference surface.

Listing 4.13: *Cup_analysis.m* - Average linear wear u_{av}

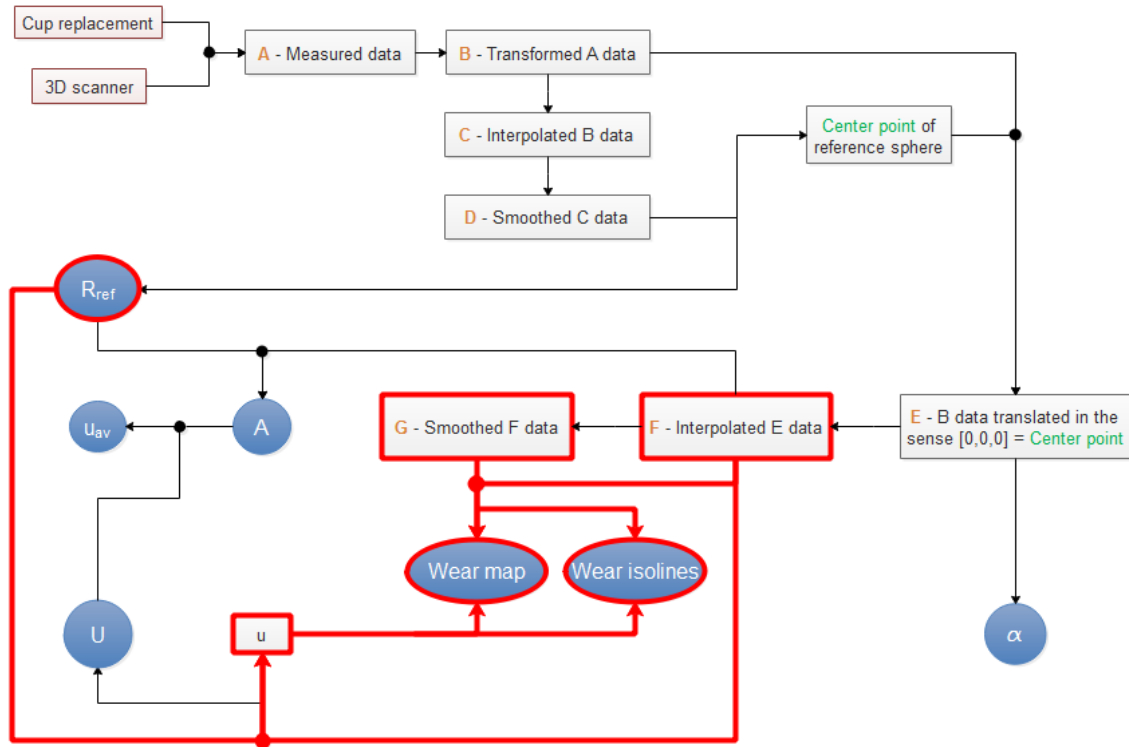
```
530 u_av = U/A; assignin('base', 'u_av', u_av) % [mm = mm^3/mm^2]
```

4.2.11 Visualization of Linear wear u

Visualization of Linear wear u is a very interesting tool for evaluating the results. It visualizes a scatter of the wear on explanted cup surface in details. All individual points i of the surface are colored according to their value of Linear wear u_i .

$$u_i = r_i - R_{ref} \quad (4.37)$$

where R_{ref} is Reference radius (gained from Eq. 4.25 or Eq. 4.28) and r_i is a distance between the origin of coordinate system and individual point i . Distance r_i is changeable from the value r_{G_i} (smoothed surface) to the value r_{F_i} (rough surface). User can change this value in the created GUI program (slider called *Smoothing ratio*) thus control the smoothness of the surface. There can be some peaks of Linear wear u on the surface caused by plastic deformation or impurities instead of wear and they can extend scale range as much as wear issues important for evaluating could be barely visible. Other possibility provided by created GUI program is the setting of scale range. User can set

Figure 4.22: Linear wear u

maximum of Linear wear u_{max} and minimum of Linear wear u_{min} thus more details can be observed at focused area. Described visualization of Linear wear u is called Wear map. Another interesting way of visualization is called Wear isolines. The difference of these two ways of visualization is that Wear isolines do not color all the points i but the surface is represented by colored contours (isolines) where every single contour represents one certain value of Linear wear u . Wear isolines could be also smoothed or their scale range could be changed in the same way as it is possible to do so with Wear map. Every possible cases of the visualization of Linear wear u are presented in the Fig. 4.23.

Listing 4.14: *Cup_analysis.m* - Scale of Linear wear u

```

1208 % DRAWS THE SCALE
1209 function scale(R,R_ref,n)
1210
1211 delete(get(gca,'Children'))
1212 hold off
1213 for i = 1:n
1214     b = rectangle('Position',[0,max(R(:))-R_ref-(max(R(:)) - min(R(:)))*i/n
1215         ,2*(max(R(:)) - min(R(:)))/n,(max(R(:)) - min(R(:)))/n], 'FaceColor'
1216         ,[cos(pi*0.5*i/n)^2 sin(pi*i/n) sin(pi*0.5*i/n)^2]);
1215     hold on
1216 end
1217

```

4. DATA ANALYSIS

```

1218 axis equal
1219 xlim([0 (max(R(:)) - min(R(:))*2/n)]; ylim([min(R(:))-R_ref max(R(:))-R_ref
    ]); xlabel('u [mm]')
1220 set(get(b, 'parent'), 'ytick', [min(R(:))-R_ref, max(R(:))-R_ref-(max(R(:)) -
    min(R(:))*0.9, max(R(:))-R_ref-(max(R(:)) - min(R(:))*0.8, ...
1221 max(R(:))-R_ref-(max(R(:)) - min(R(:))*0.7, max(R(:))-R_ref-(max(R(:)) - min
    (R(:))*0.6, max(R(:))-R_ref-(max(R(:)) - min(R(:))*0.5, ...
1222 max(R(:))-R_ref-(max(R(:)) - min(R(:))*0.4, max(R(:))-R_ref-(max(R(:)) - min
    (R(:))*0.3, max(R(:))-R_ref-(max(R(:)) - min(R(:))*0.2, ...
1223 max(R(:))-R_ref-(max(R(:)) - min(R(:))*0.1, max(R(:))-R_ref], 'XTick', []);

```

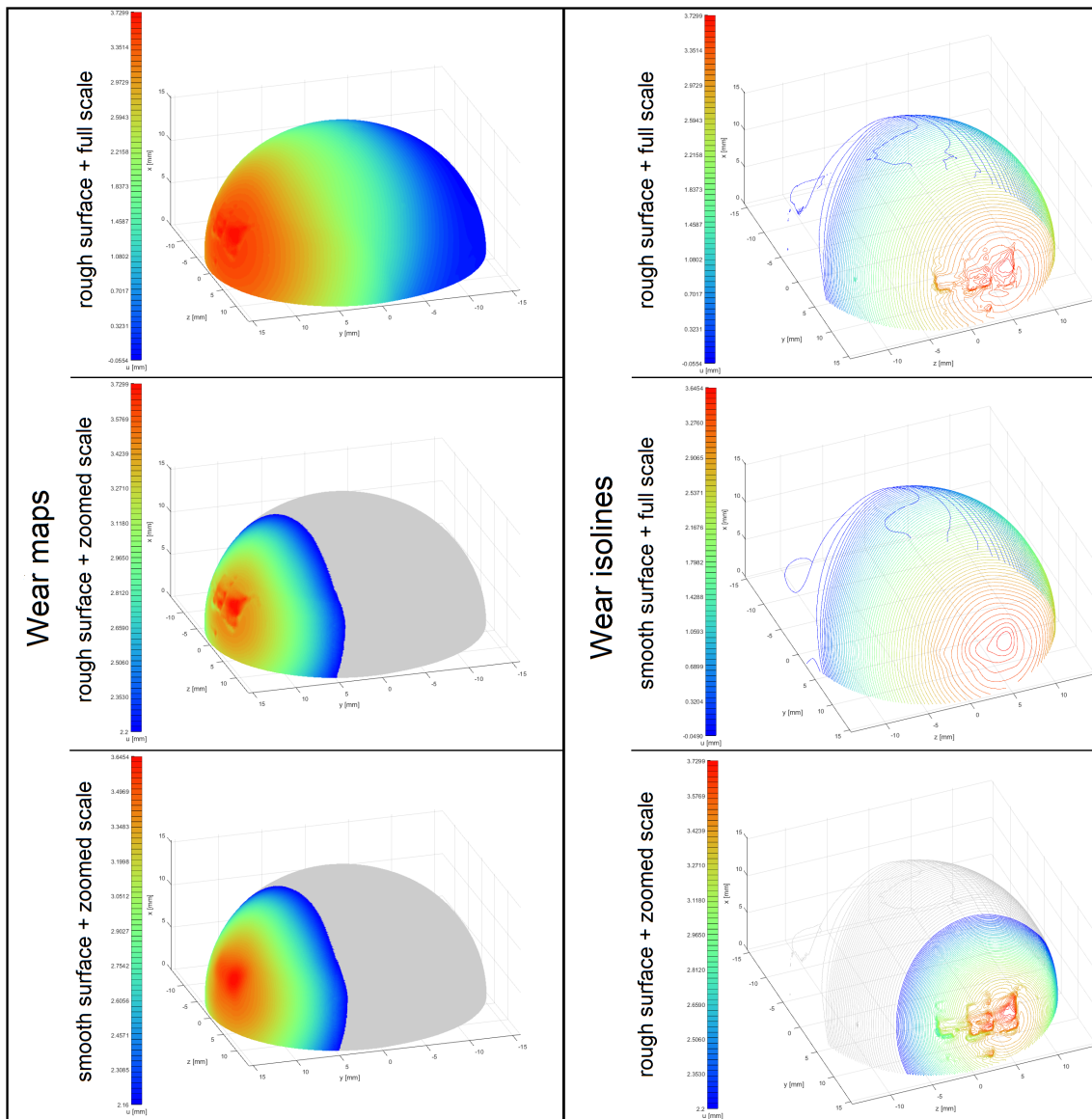


Figure 4.23: Visualization of Linear wear u - examples

Listing 4.15: *Cup_analysis.m* - Wear map

```

1178 % COLORS THE SURFACE ACCORDING TO ITS REFERENCE RADIUS DEVIATION
1179 function wear_map(R,xg,yg,zg,n,xs,ys,zs,look1,Rw)
1180
1181 delete(get(gca,'Children'))
1182 xw = reshape(xg,[],1); yw = reshape(yg,[],1); zw = reshape(zg,[],1);Rw =
    reshape(Rw,[],1);
1183
1184 % FIRSTLY IT SHOWS POINTS OUT OF SCALE RANGE
1185 log1 = Rw > max(R(:));
1186 log2 = Rw < min(R(:));
1187 plot3(yw(log1)+ys,zw(log1)+zs,xw(log1)+xs,'MarkerSize',15,'Marker','.',
    'LineStyle','none','Color',[0.9 0.9 0.9])
1188 hold on
1189 plot3(yw(log2)+ys,zw(log2)+zs,xw(log2)+xs,'MarkerSize',15,'Marker','.',
    'LineStyle','none','Color',[0.8 0.8 0.8])
1190
1191 % COLOURS POINTS
1192 for i = 1:n
1193     log = (Rw <= max(R(:)) - (max(R(:)) - min(R(:)))*(i-1)/n) & (Rw >= max(R
        (:)) - (max(R(:)) - min(R(:)))*i/n) ;
1194     plot3(yw(log)+ys,zw(log)+zs,xw(log)+xs,'MarkerSize',15,'Marker','.',
        'LineStyle','none','Color',[cos(pi*0.5*i/n)^2 sin(pi*i/n) sin(pi
            *0.5*i/n)^2])
1195
1196     Rw(log) = NaN; Rw = Rw(~isnan(Rw)); % removes already colored points
        to reduce time of calculation
1197     xw(log) = NaN; xw = xw(~isnan(xw));
1198     yw(log) = NaN; yw = yw(~isnan(yw));
1199     zw(log) = NaN; zw = zw(~isnan(zw));
1200 end
1201
1202 plot3(0,0,0,'Marker','none','LineStyle','none')
1203 axis equal, grid on; view(look1)
1204 xlabel('y [mm]');ylabel('z [mm]');zlabel('x [mm]');
1205 hold off

```

Listing 4.16: *Cup_analysis.m* - Wear isolines

```

1226 % DRAWS THE ISOLINES ON THE SURFACE ACCORDING TO ITS REFERENCE RADIUS
    DEVIATION
1227 function wear_isolines(R, theta_fg, xi_fg,n,xs,ys,zs,look2,Rw)
1228
1229 C = contour3(xi_fg,theta_fg,Rw,n); hold off % isolines out of scale range
1230 Rw(Rw<min(R(:))) = min(R(:)) - (max(R(:))-min(R(:)))/n; Rw(Rw>max(R(:))) =
    max(R(:)) + (max(R(:))-min(R(:)))/n;
1231

```

```
1232 D = contour3(xi_fg,theta_fg,Rw,n+2);hold off % isolines into scale range
1233 delete(get(gca,'Children'))
1234
1235 sizeC = size(C);
1236 c = 1; % counter
1237
1238 while c < sizeC(1,2) % isolines out of scale range
1239
1240     xi = C(1,c+1:c+C(2,c))';
1241     theta = C(2,c+1:c+C(2,c))';
1242     r = ones(C(2,c),1)*C(1,c);
1243
1244     [x,y,z] = r_theta_xi__x_y_z(r,theta,xi);
1245
1246     if r(1,1) > max(R(:))
1247         plot3(y+ys,z+zs,x+xs,'Color',[0.9 0.9 0.9])
1248         hold on
1249     elseif r(1,1) < min(R(:))
1250         plot3(y+ys,z+zs,x+xs,'Color',[0.8 0.8 0.8])
1251         hold on
1252     end
1253
1254     c = c + C(2,c) + 1;
1255
1256 end
1257
1258 sizeC = size(D);
1259 c = 1; % counter
1260 while c < sizeC(1,2) % isolines into scale range
1261
1262     xi = D(1,c+1:c+D(2,c))';
1263     theta = D(2,c+1:c+D(2,c))';
1264     r = ones(D(2,c),1)*D(1,c);
1265
1266     [x,y,z] = r_theta_xi__x_y_z(r,theta,xi);
1267
1268     if r(1,1) > max(R(:))
1269
1270     elseif r(1,1) < min(R(:))
1271
1272     else
1273         i=1;
1274         log = (r(1,1) <= max(R(:)) - (max(R(:)) - min(R(:)))*(i-1)/n) & (r
            (1,1) >= max(R(:)) - (max(R(:)) - min(R(:)))*i/n) ;
1275         while log == 0
1276             log = (r(1,1) <= max(R(:)) - (max(R(:)) - min(R(:)))*(i-1)/n) &
                (r(1,1) >= max(R(:)) - (max(R(:)) - min(R(:)))*i/n) ;
```

```

1277         i=i+1;
1278     end
1279
1280     plot3(y+ys,z+zs,x+xs,'Color',[cos(pi*0.5*(i-1)/n)^2 sin(pi*(i-1)/n)
1281         sin(pi*0.5*(i-1)/n)^2])
1282     hold on
1283 end
1284 c = c + D(2,c) + 1;
1285
1286 end
1287 plot3(0,0,0,'Marker','none','LineStyle','none')
1288 axis equal, grid on,view(look2)
1289 xlabel('y [mm]');ylabel('z [mm]');zlabel('x [mm]'); hold off

```

4.2.12 Used *Matlab* functions

In this subsection we can find clarification of four *Matlab* functions that were used throughout the calculation and their principle has not been explained yet. They are significant for the whole calculation because they directly influence or convert data.

sphereFit This function is used within Data transformation (subsection 4.2.2 on page 28) and Estimation of reference geometry (subsection 4.2.5 on page 35). *sphereFit* is not included in a basic *Matlab* package but it was downloaded from File Exchange section on official website of *MathWorks, Inc.* Its author is Alan Jennings (University of Dayton). *sphereFit* fits a sphere to a data point cloud (x_i, y_i, z_i) . *sphereFit* requires inputs in the form of $n \times 3$ matrix of Cartesian data and its outputs are center point's Cartesian coordinates of the fitted sphere $[x_c, y_c, z_c]$ and radius of the fitted sphere r . *sphereFit* minimizes

$$\sum_{i=1}^n \left[(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 - r^2 \right]^2 \quad (4.38)$$

The least squared equations are used to re-create this problem into matrix equation $[3 \times 3] \cdot [1 \times 3] = [1 \times 3]$. *sphereFit* does not require a large arc or many data points. It assumes points are not singular (co-planar).

Source: <http://goo.gl/PfJ7mN>

scatteredInterpolant This function is used within Data interpolation (subsection 4.2.3 on page 32). *scatteredInterpolant* is included in a basic *Matlab* package. *scatteredInterpolant* performs interpolation on a 2-D or 3-D set of points that have no structure among their relative locations (scattered data set). For example, we can pass a set of s_{1B}, s_{2B} points and values r_B ('*B data*' Pole coordinates) to *scatteredInterpolant*, and it returns a surface of the form $r_B = F(s_{1B}, s_{2B})$. This surface always passes through the sample values at the point locations. We can evaluate this surface at query point s_{1C}, s_{2C} , to produce an interpolated value r_C ('*C data*' Pole coordinates). *Matlab* provides several interpolation methods for *scatteredInterpolant* function. In

our case 'natural' method was used. This method provides C^1 continuity all around the interpolated surface except at sample points. Since required net of points can overlap the scattered data set, an extrapolation method was specified to 'nearest'. This method evaluates to the value of the nearest neighbor on the boundary.

Source: <http://goo.gl/yimtCG>

gridfit This function is used within Data smoothing (subsection 4.2.4 on page 34). *gridfit* is not included in a basic *Matlab* package but it was downloaded from File Exchange section on official website of *MathWorks, Inc.* Its author is John D'Errico. *gridfit* is supposed to replace a scattered data by a smooth lattice of points. It does not always interpolate all supplied data. Its goal is a smooth surface that approximates the initial data, and allows to control the amount of smoothing. Required inputs are scattered data x, y, z and x_{nodes}, y_{nodes} - vectors defining the nodes in the grid in the independent variables x, y . The outputs are x_{grid}, y_{grid} - matrices defining the lattice of points; and z_{grid} - matrix assigning third dimension to the lattice points. *gridfit* principle is explained in the text documents attached to *gridfitdir.zip* file available to download from File Exchange section on official website of *MathWorks, Inc.* These text documents describes the ideas behind *gridfit*.

Source: <http://goo.gl/sAogyE>

contour3 This function is used within Wear isolines plotting (Lis. 4.16). *contour3* is included in a basic *Matlab* package. *contour3* creates a 3-D contour plot of a surface defined on a rectangular grid. Our inputs are r_F, ξ_F, θ_F ('*F data*' Spherical coordinates) and outputs are the points which create individual contours (isolines). Afterward, these isolines are tranformed to Cartesian coordinates and visualized in the 3D plot.

Source: <http://goo.gl/gKzZiY>

Pilot study

Developed wear estimation method was tested within pilot study. We received several real cup explants from Motol University Hospital (Prague, Czech Republic). All of those cups failed due to wear. Photos of those cup explants and their casts are visible in this chapter but the same photos with higher resolution are saved in the attachment *C - Pilot study*. This attachment is part of enclosed CD. The results of pilot study are shown in section 6.3, page 67.

Bottom right part (d) of each Figure 5.1-5.5 shows the settings of the calculation in *Matlab GUI* program. These settings directly influence the results so it is always necessary to present the results together with the calculation settings. More info about calculation settings is presented on enclosed CD (attachment *B - Cup Analysis GUIDELINE.pdf*).

Table 5.1: Explanted cups for pilot study

| Sample no. | 1 | 2 | 3 | 4 | 5 |
|-----------------------------|------------|------------|------------|------------|------------|
| ID code | 2015/7/125 | 2015/7/139 | 2015/7/141 | 2015/7/151 | 2015/7/152 |
| Patient | Name 1 | Name 2 | Name 3 | Name 4 | Name 5 |
| Sex | female | female | male | female | female |
| Year of birth | 1955 | 1948 | 1944 | 1978 | 1967 |
| Cup material | UHMWPE | UHMWPE | UHMWPE | UHMWPE | UHMWPE |
| Nominal radius [mm] | 14 | 14 | 14 | 14 | 14 |
| Date of implantation | 1/1997 | 1/2002 | 1/2000 | 1/2001 | 1/2000 |
| Date of explantation | 7/2015 | 12/2014 | 12/2014 | 10/2015 | 9/2015 |
| Lifetime [years] | 18.6 | 13.0 | 15.0 | 14.8 | 15.8 |
| Figure | 5.1 | 5.2 | 5.3 | 5.4 | 5.5 |

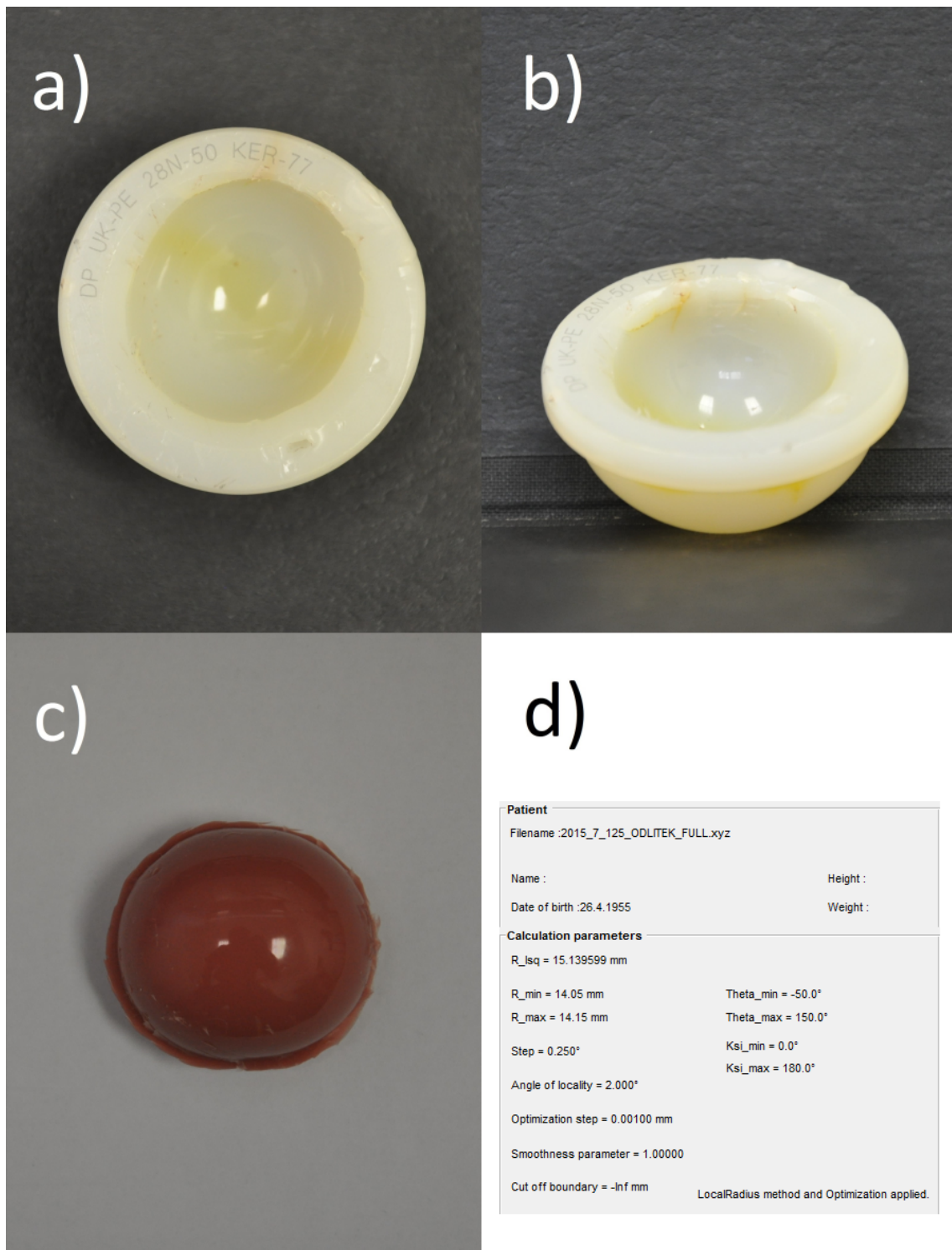


Figure 5.1: Sample no. 1 - (a-b) Cup. (c) Cast. (d) *Matlab GUI* calculation settings

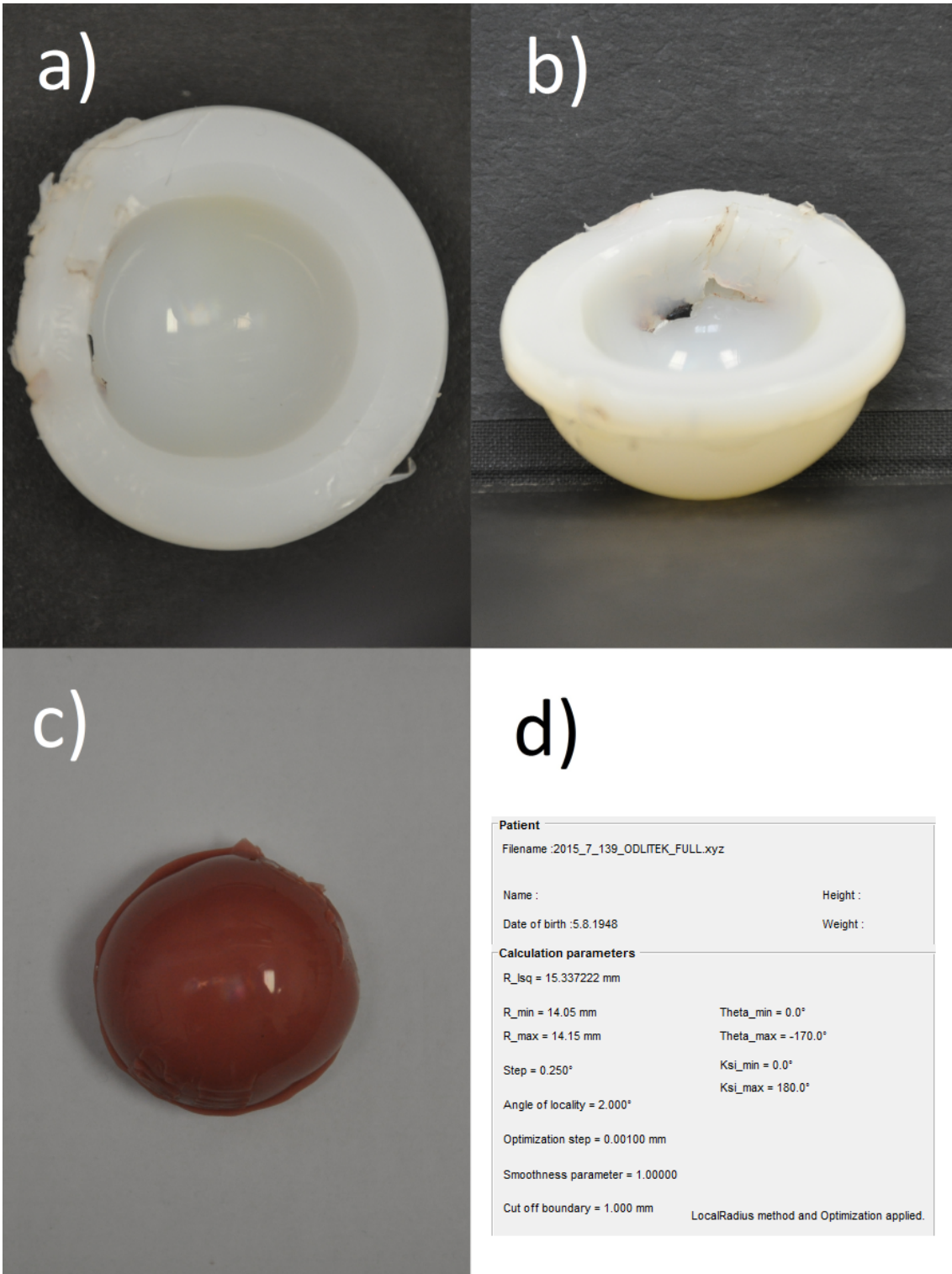


Figure 5.2: Sample no. 2 - (a-b) Cup. (c) Cast. (d) *Matlab GUI* calculation settings

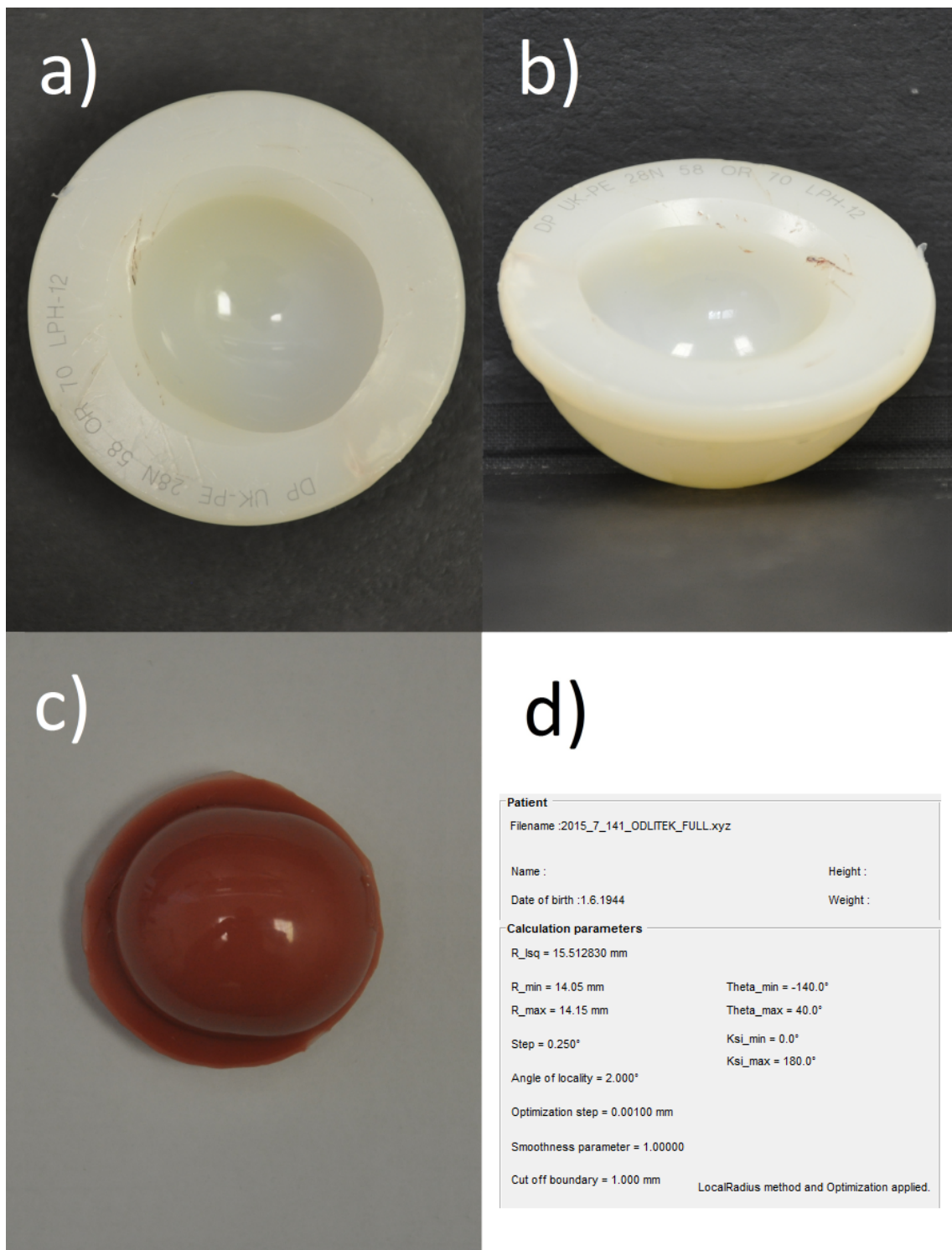


Figure 5.3: Sample no. 3 - (a-b) Cup. (c) Cast. (d) *Matlab GUI* calculation settings

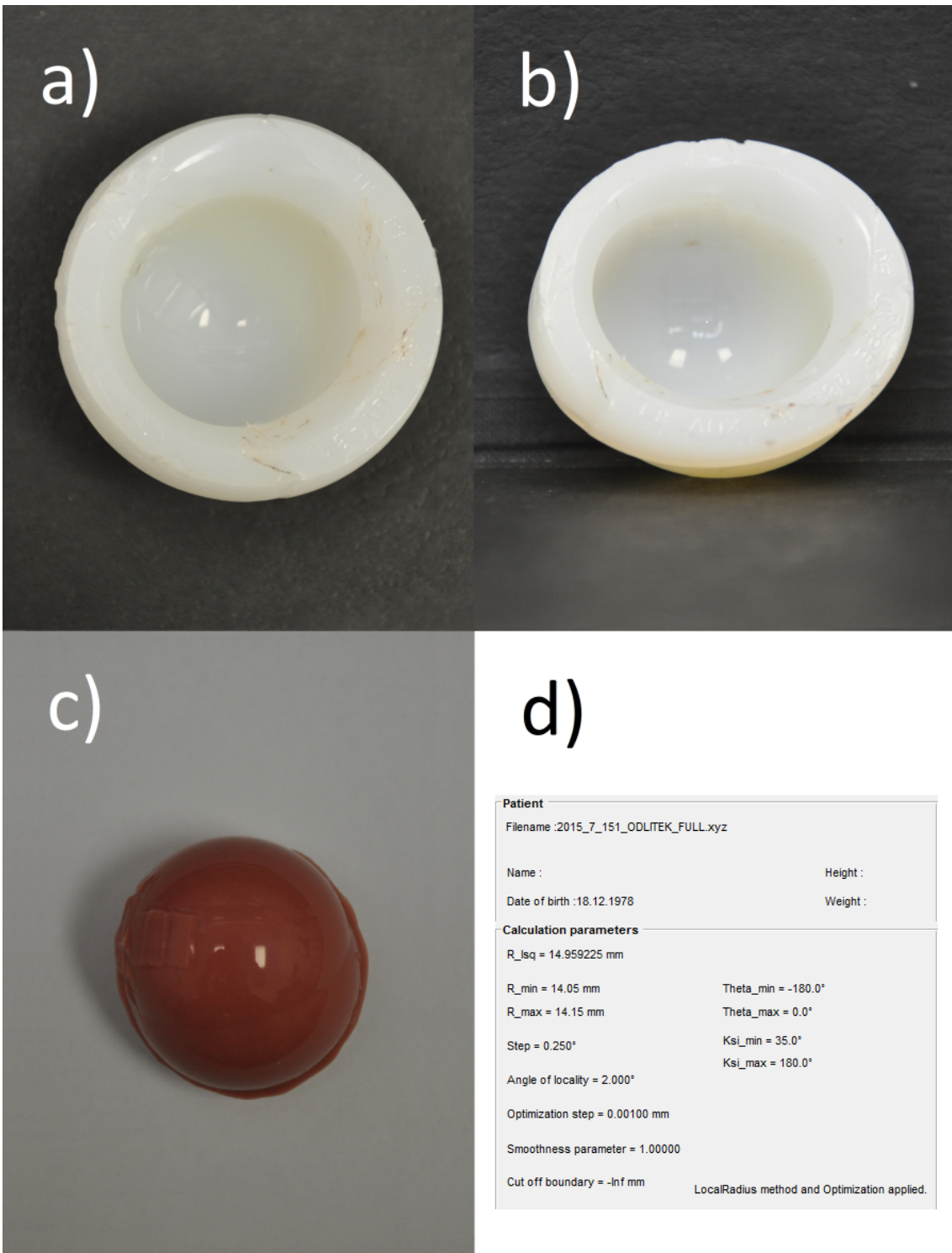


Figure 5.4: Sample no. 4 - (a-b) Cup. (c) Cast. (d) *Matlab GUI* calculation settings

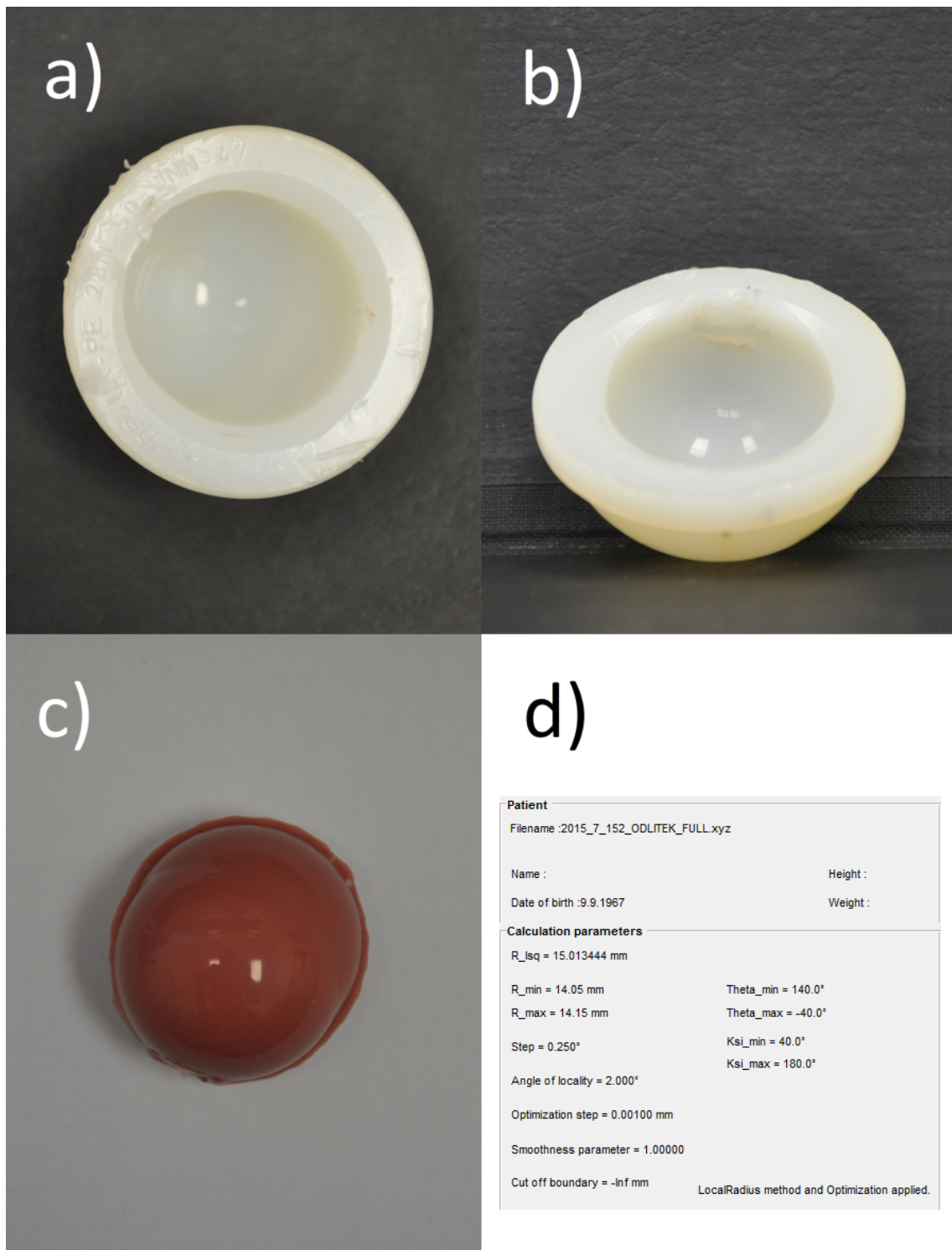


Figure 5.5: Sample no. 5 - (a-b) Cup. (c) Cast. (d) *Matlab GUI* calculation settings

Results

This chapter consists of results of casting method verification, results of algorithm verification and results of pilot study. The verification of casting method is described in section 3.2 beginning on page 14. This verification is realized by *Matlab* script which is contained on enclosed CD as attachment called *D - Verification of casting method*. Attachment *A - Cup Analysis (Matlab GUI)* from enclosed CD contains created *Matlab GUI* program - the desired program described within chapter 4 beginning on page 19. Verification of the algorithm applied in this program is realized by calculation of worn surface whose wear we are able to calculate analytically. The results calculated by the program (attachment *E - Verificatiion of algorithm* from enclosed CD) are compared against the actual results calculated analytically. The results of the pilot study are presented afterward. For full understanding of the results displayed by *Matlab GUI* program, read attachment *B - Cup Analysis GUIDELINE.pdf* from enclosed CD.

6.1 Verification of casting method

The cup we used for verification of casting method was a new unworn cup with nominal radius 14mm, made of UHMWPE. We chose unworn cup because optical 3D CMM by *RedLux Ltd.* is capable of measuring directly the surface of this cup so this surface can be compared with measured surfaces of casts. For the verification, all the data was cropped to the value of Maximal measured angle $\alpha = 70^\circ$ in *Matlab* script.

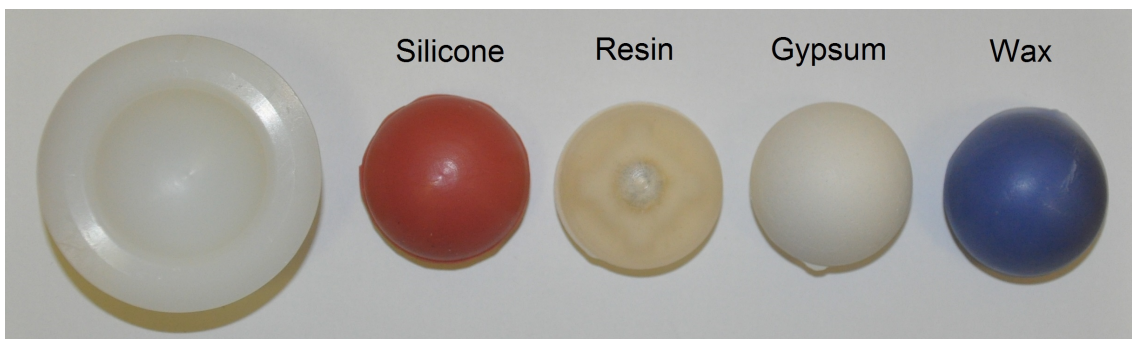


Figure 6.1: New unworn UHMWPE cup and its casts

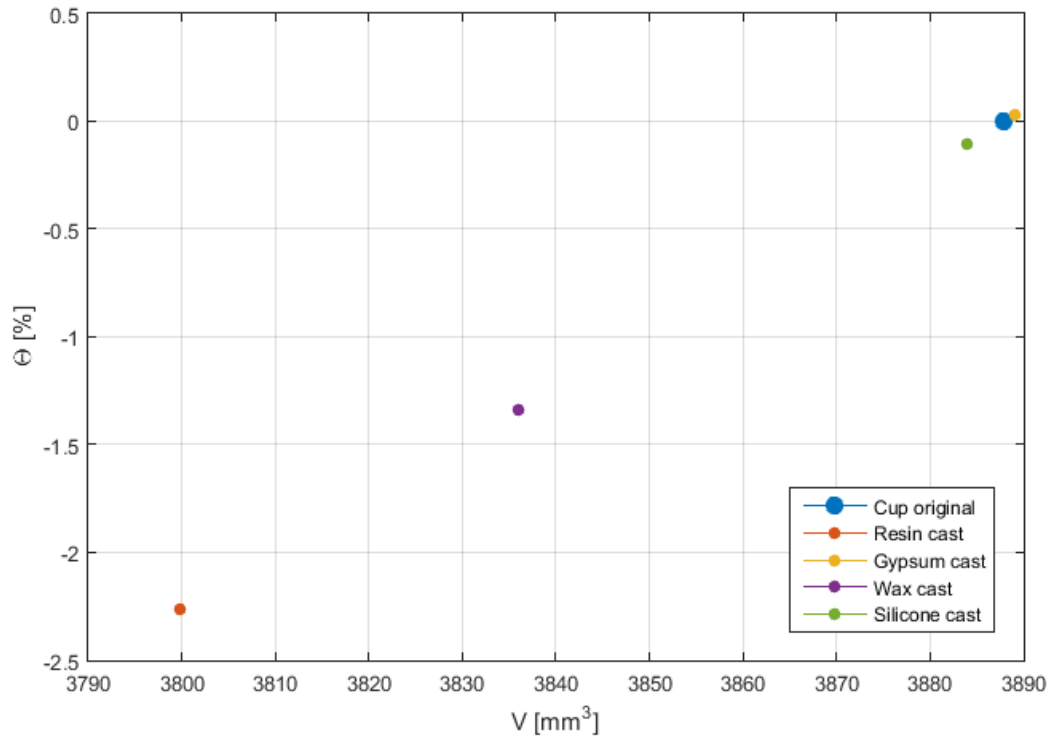


Figure 6.2: Evaluation of volumetric change

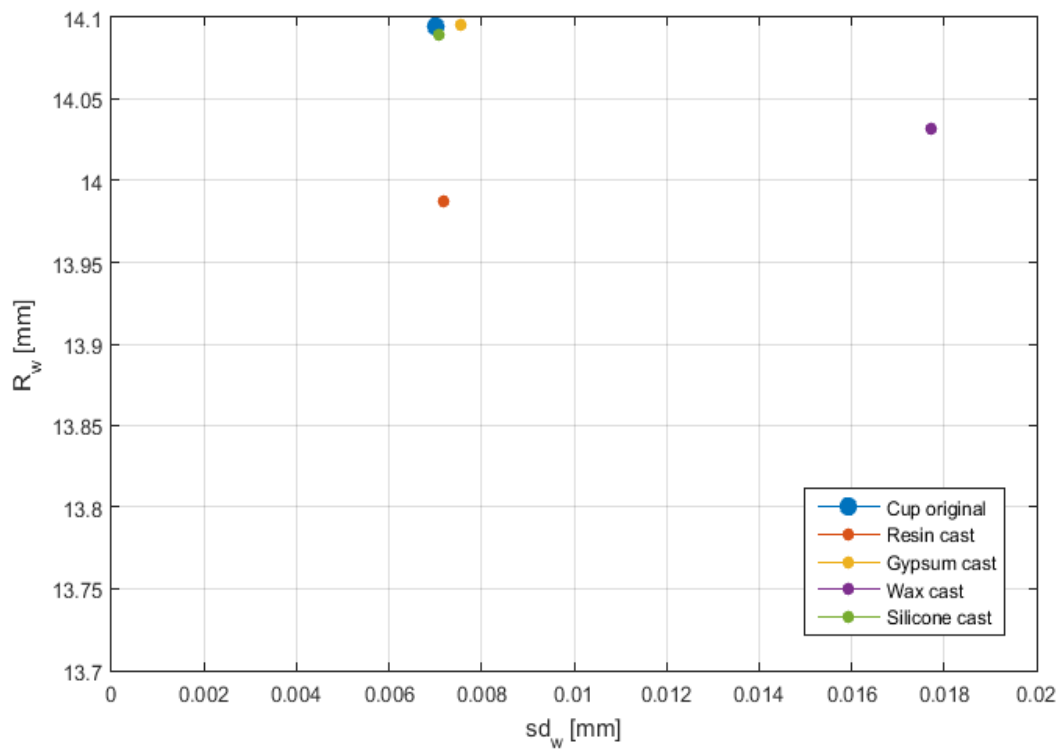


Figure 6.3: Evaluation of surface reconstruction

6.2 Verification of algorithm for wear estimation

For verification of the created algorithm (realized in *Matlab GUI* program), we simulate ellipsoidal wear, according to Fig. 6.4. As an original unworn cups, we consider plastic cups with nominal radius 14mm, made in accordance with international standard *ISO 7206-2:1996(E)* [9].

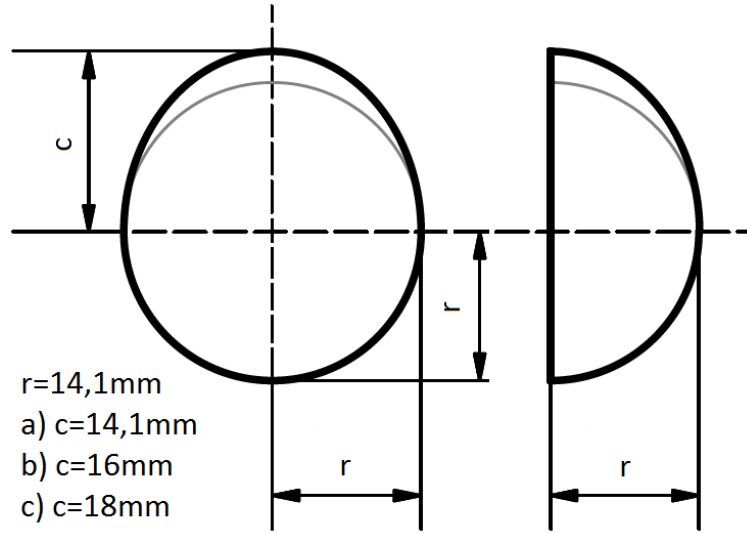


Figure 6.4: Simulated ellipsoidal wear

Imaginary measured data of the cups worn as mentioned above was generated in *Matlab*. Afterward, that data was saved into text file (.xyz format). This text file was uploaded into created *Matlab GUI* program and calculation was executed. In this section, we are denoting the results of algorithm calculation as $k_{algorithm}$ where $k = U, A, \alpha, R_{ref}, u_{max}, u_{min}$. Unlike the actual results calculated analytically are denoted as k_{actual} . The actual results calculated analytically are given as:

- $U_{actual} = \frac{1}{3} \cdot \pi \cdot r^2 \cdot (c - r)$
- $A_{actual} = 2 \cdot \pi \cdot r^2 \cdot (1 - \cos \alpha_{actual})$
- $\alpha_{actual} = 90^\circ$
- $R_{ref_{actual}} = 14.1mm$
- $u_{max} = c - r$
- $u_{min} = 0mm$

The results of the algorithm are presented in the Fig. 6.5, Fig. 6.6 and Fig. 6.6. For comparison of the results calculated by the algorithm and analytically, we assess Percent error δ_k as:

$$\delta_k = 100 \cdot \left| 1 - \frac{k_{algorithm}}{k_{actual}} \right| \quad (6.1)$$

6. RESULTS

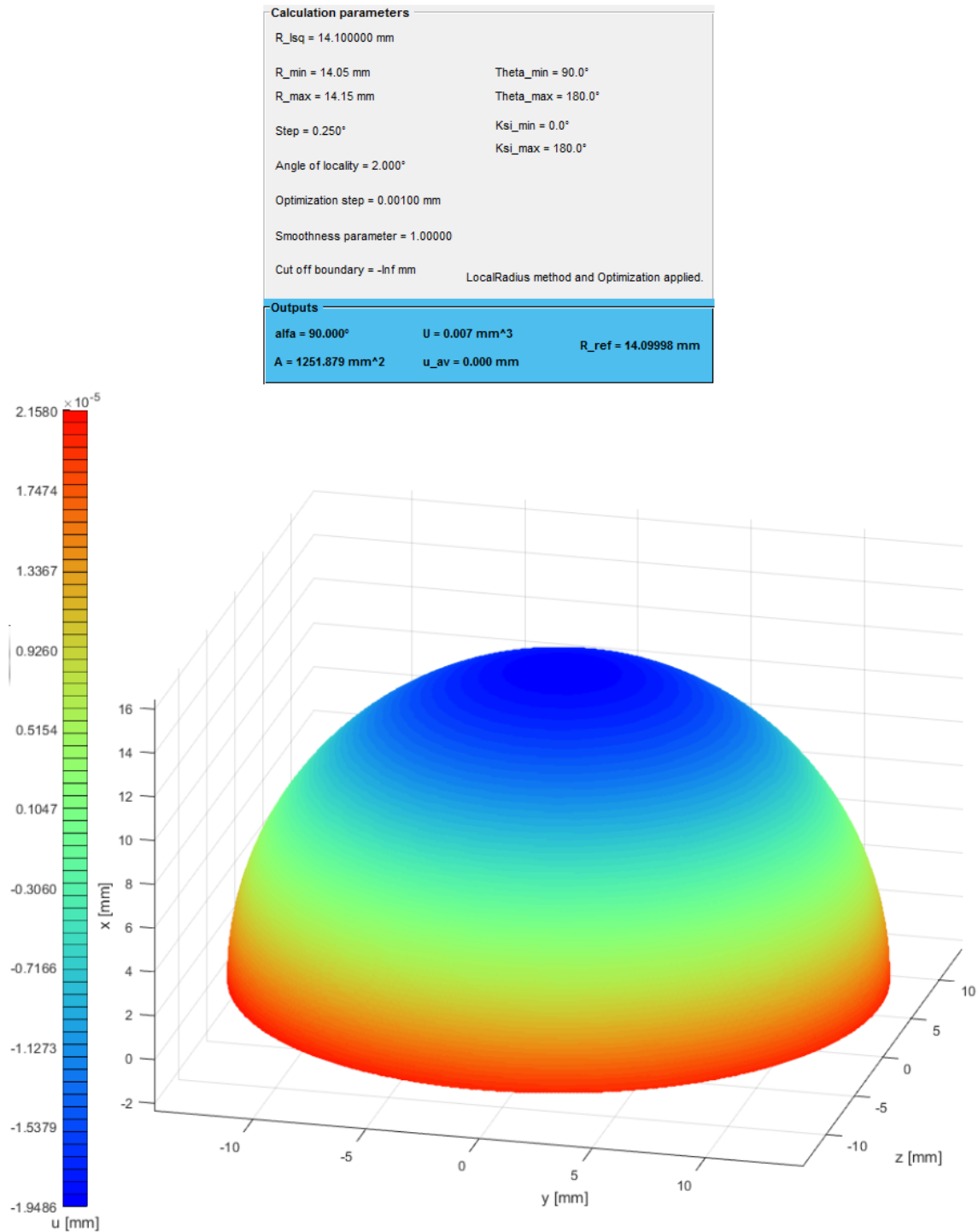
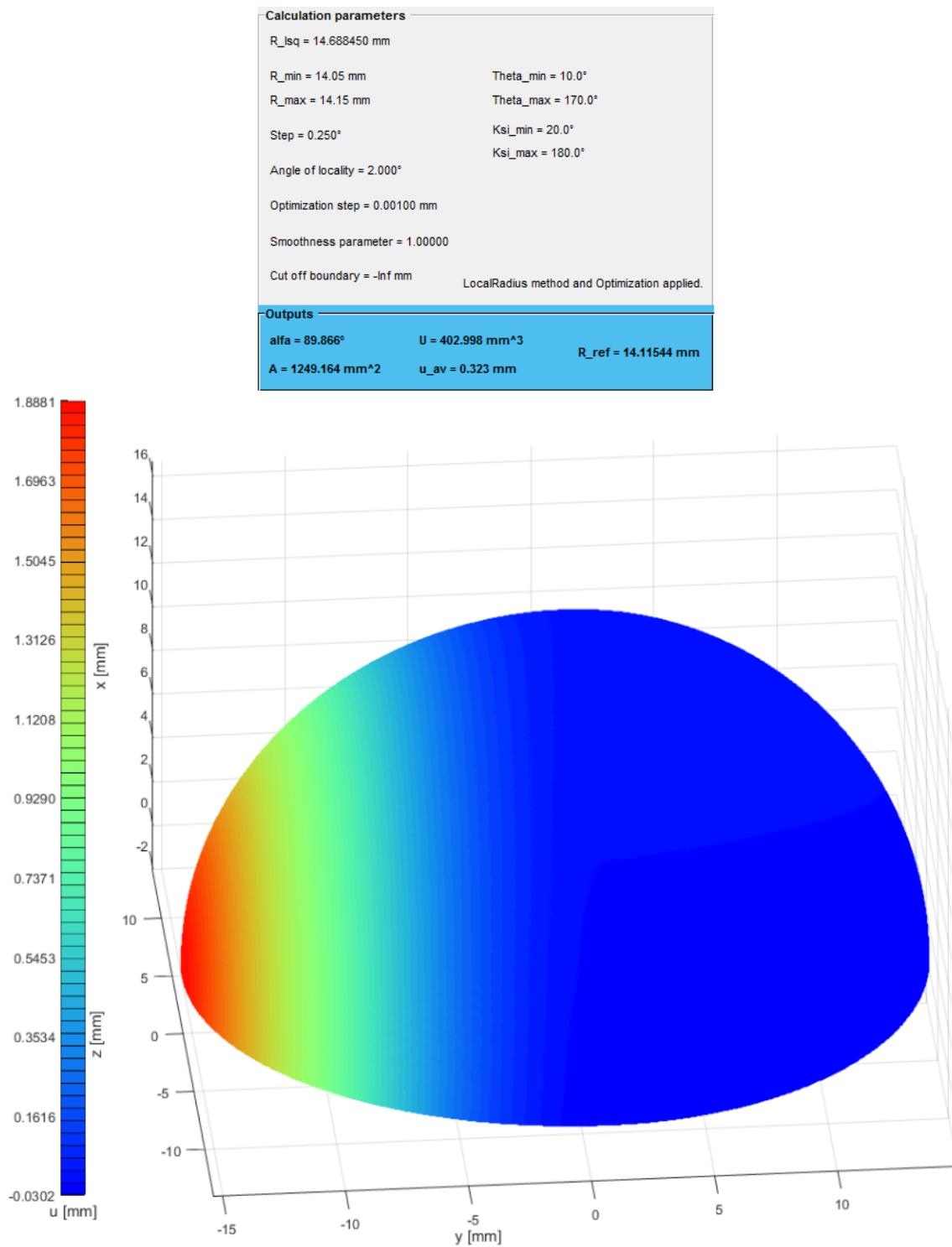


Figure 6.5: Results of case a) $c=14.1\text{mm}$

6.2. Verification of algorithm for wear estimation



6. RESULTS

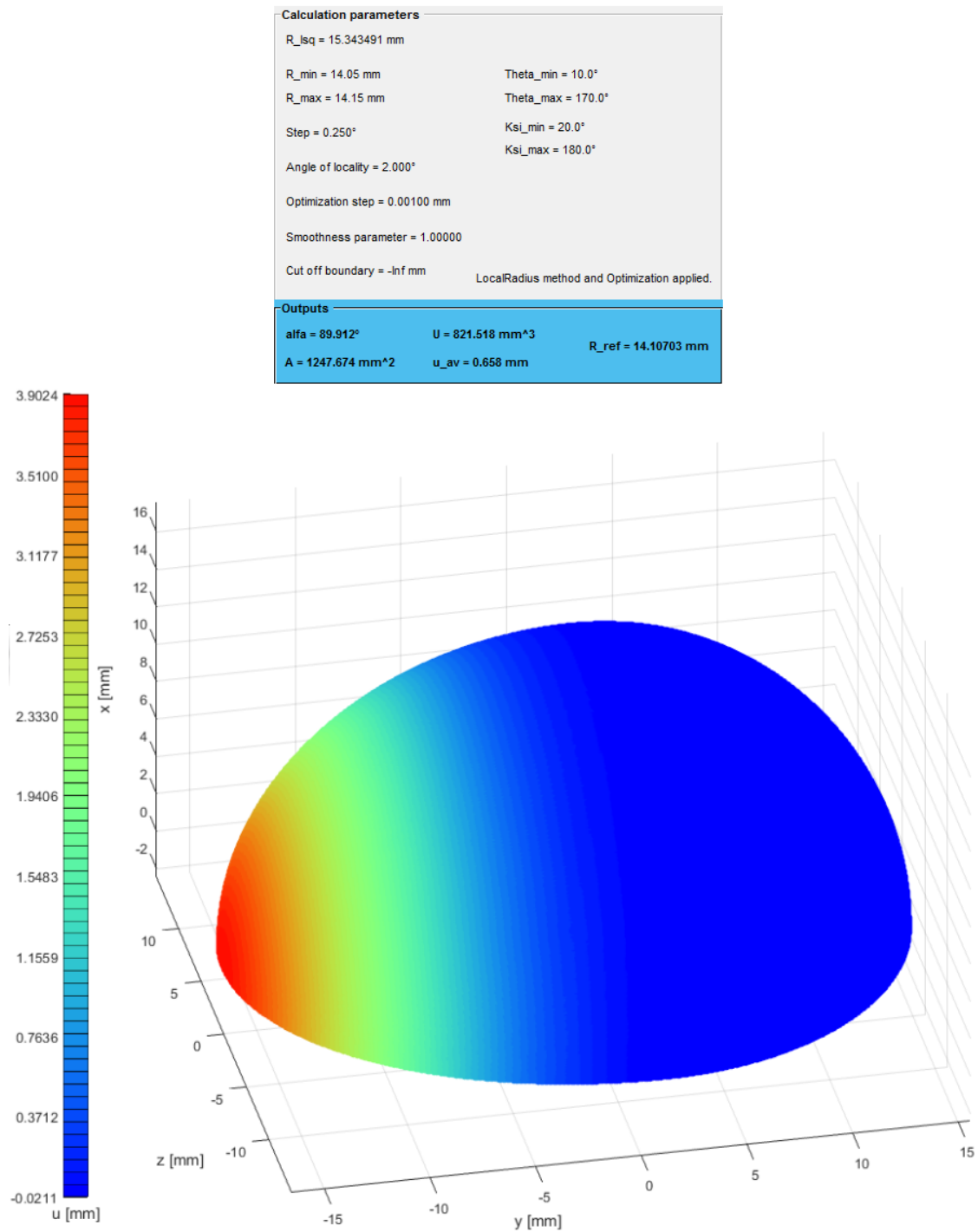


Figure 6.7: Results of case c) $c=18\text{mm}$

Table 6.1: Comparison between actual results and results calculated by the algorithm

| a) c=14.1mm | $U[mm^3]$ | $A[mm^2]$ | $\alpha[^\circ]$ | $R_{ref}[mm]$ | $u_{max}[mm]$ | $u_{min}[mm]$ |
|--------------------|--------------|--------------|------------------|---------------|---------------|---------------|
| Actual | 0 | 1249.160 | 90 | 14.1 | 0 | 0 |
| Algorithm | 0.007 | 1251.879 | 90.000 | 14.100 | 0.000 | 0.000 |
| δ [%] | NaN | 0.218 | 0.000 | 0.000 | NaN | NaN |
| b) c=16mm | $U[mm^3]$ | $A[mm^2]$ | $\alpha[^\circ]$ | $R_{ref}[mm]$ | $u_{max}[mm]$ | $u_{min}[mm]$ |
| Actual | 395.567 | 1249.160 | 90 | 14.1 | 1.9 | 0 |
| Algorithm | 402.998 | 1249.164 | 89.866 | 14.115 | 1.888 | -0.030 |
| δ [%] | 1.879 | 0.000 | 0.149 | 0.106 | 0.632 | NaN |
| c) c=18mm | $U[mm^3]$ | $A[mm^2]$ | $\alpha[^\circ]$ | $R_{ref}[mm]$ | $u_{max}[mm]$ | $u_{min}[mm]$ |
| Actual | 811.954 | 1249.160 | 90 | 14.1 | 3.9 | 0 |
| Algorithm | 821.518 | 1247.674 | 89.912 | 14.107 | 3.902 | -0.021 |
| δ [%] | 1.178 | 0.119 | 0.098 | 0.050 | 0.051 | NaN |

6.3 Results of pilot study

The results of the pilot study, which is presented in chapter 5 on page 55, are shown in this section. The casts of explanted cups within the pilot study were measured by 3D scanner and resulting *.xyz* files with measured data were analyzed by created *Matlab GUI* program (attachment A - *Cup Analysis (Matlab GUI)* on enclosed CD). The outputs of these analyses were saved into *.mat* files. All the files (*.xyz* files with the measured data and *.mat* files with the results) are saved within attachment C - *Pilot study* on enclosed CD.

Table 6.2: Numerical results of pilot study

| Sample no. | 1 | 2 | 3 | 4 | 5 |
|-------------------|----------|----------|----------|----------|----------|
| $U[mm^3]$ | 899.9 | 1142.9 | 1394.5 | 1814.3 | 2148.5 |
| $A[mm^2]$ | 1223.1 | 1229.5 | 1229.1 | 1207.7 | 1195.7 |
| $u_{av}[mm]$ | 0.736 | 0.929 | 1.135 | 1.502 | 1.797 |
| $\alpha[^\circ]$ | 88.849 | 89.220 | 89.233 | 88.141 | 87.697 |
| $R_{ref}[mm]$ | 14.091 | 14.097 | 14.094 | 14.097 | 14.090 |
| $u_{max}[mm]$ | 3.317 | 3.453 | 4.183 | 3.674 | 4.152 |
| $u_{min}[mm]$ | -0.774 | -0.112 | -0.056 | -0.064 | -0.043 |

6. RESULTS

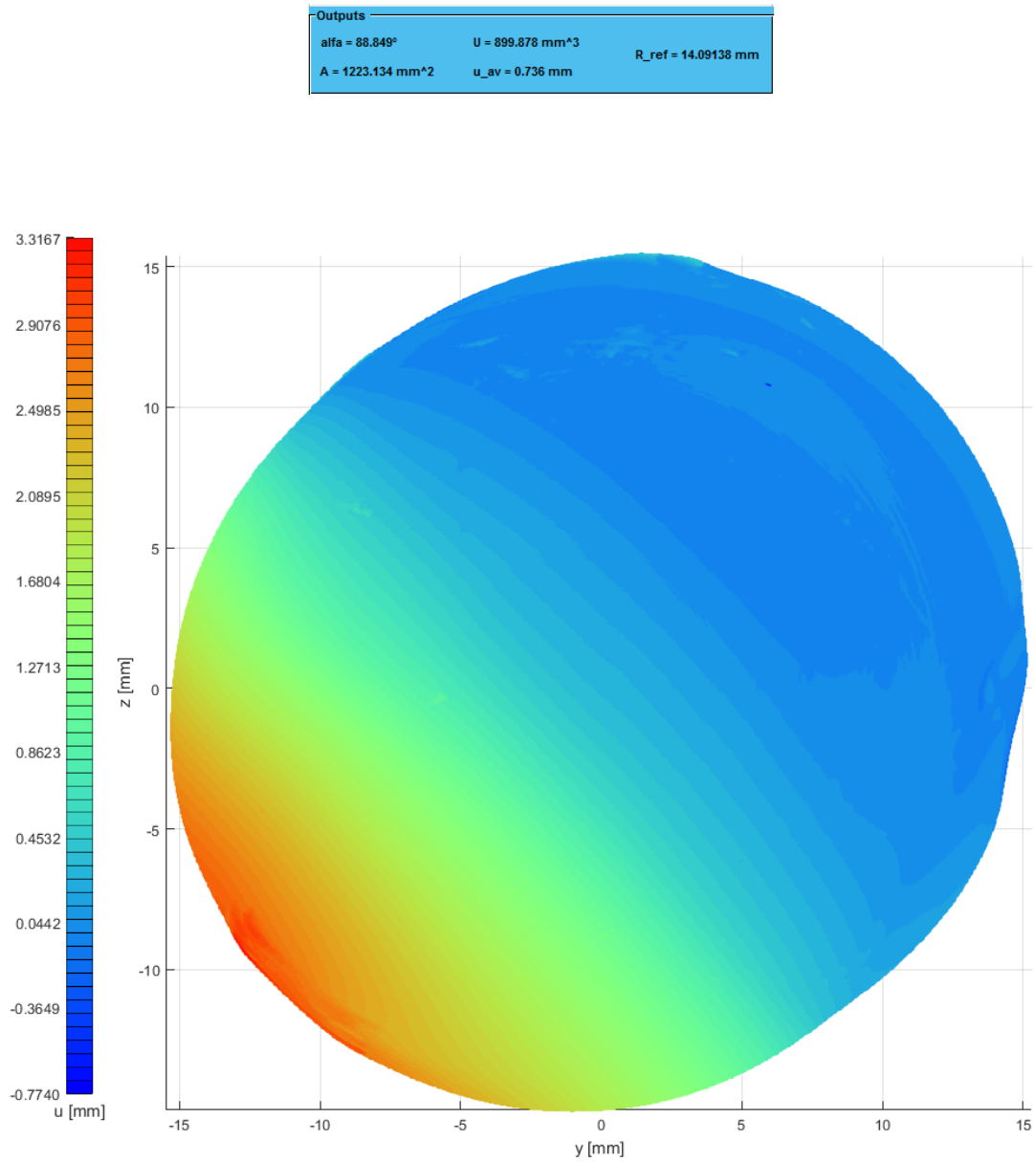


Figure 6.8: **Sample no. 1** - Outputs table and wear map (top view)

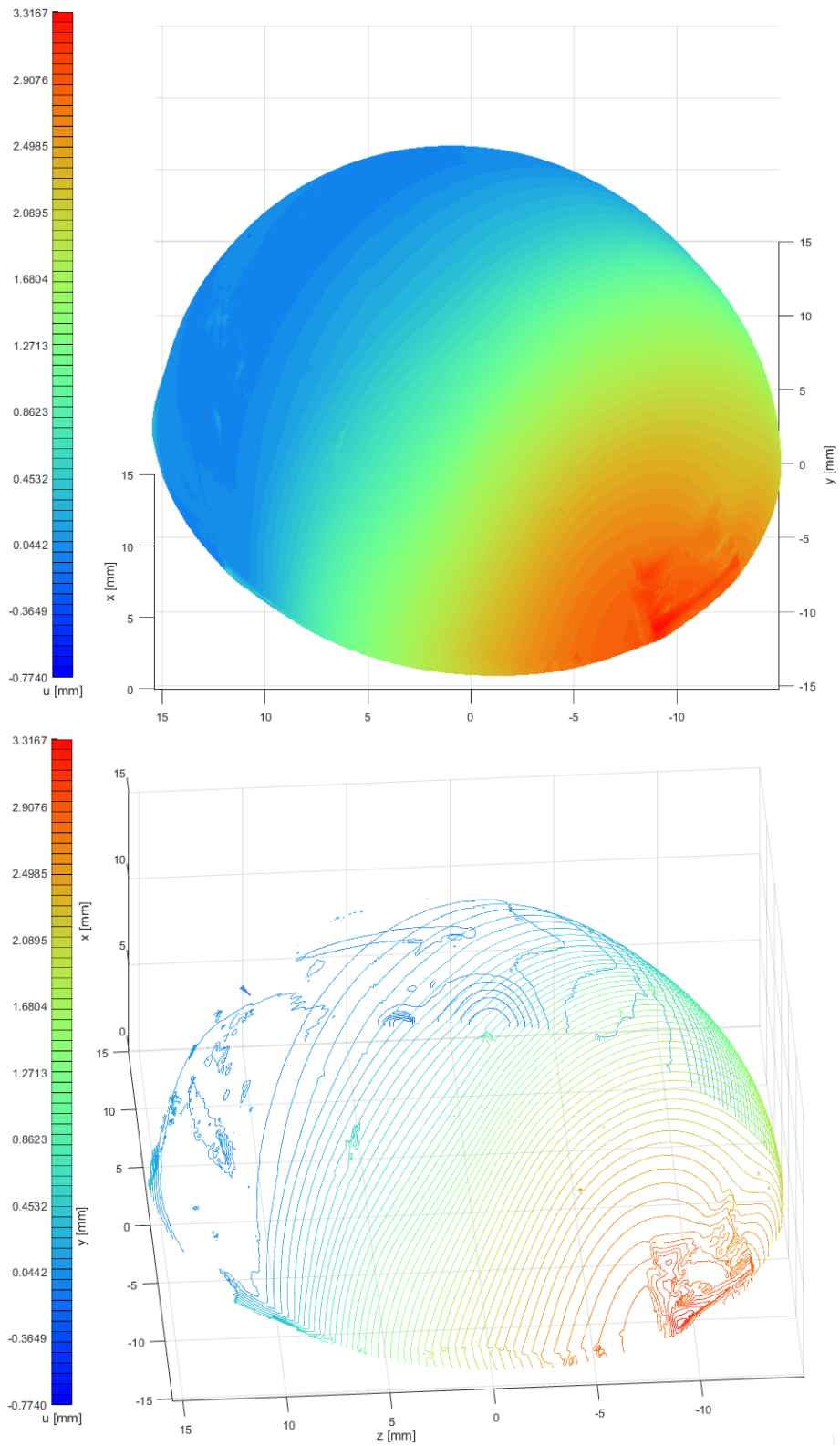


Figure 6.9: **Sample no. 1** - Wear map and wear isolines (side views)

6. RESULTS

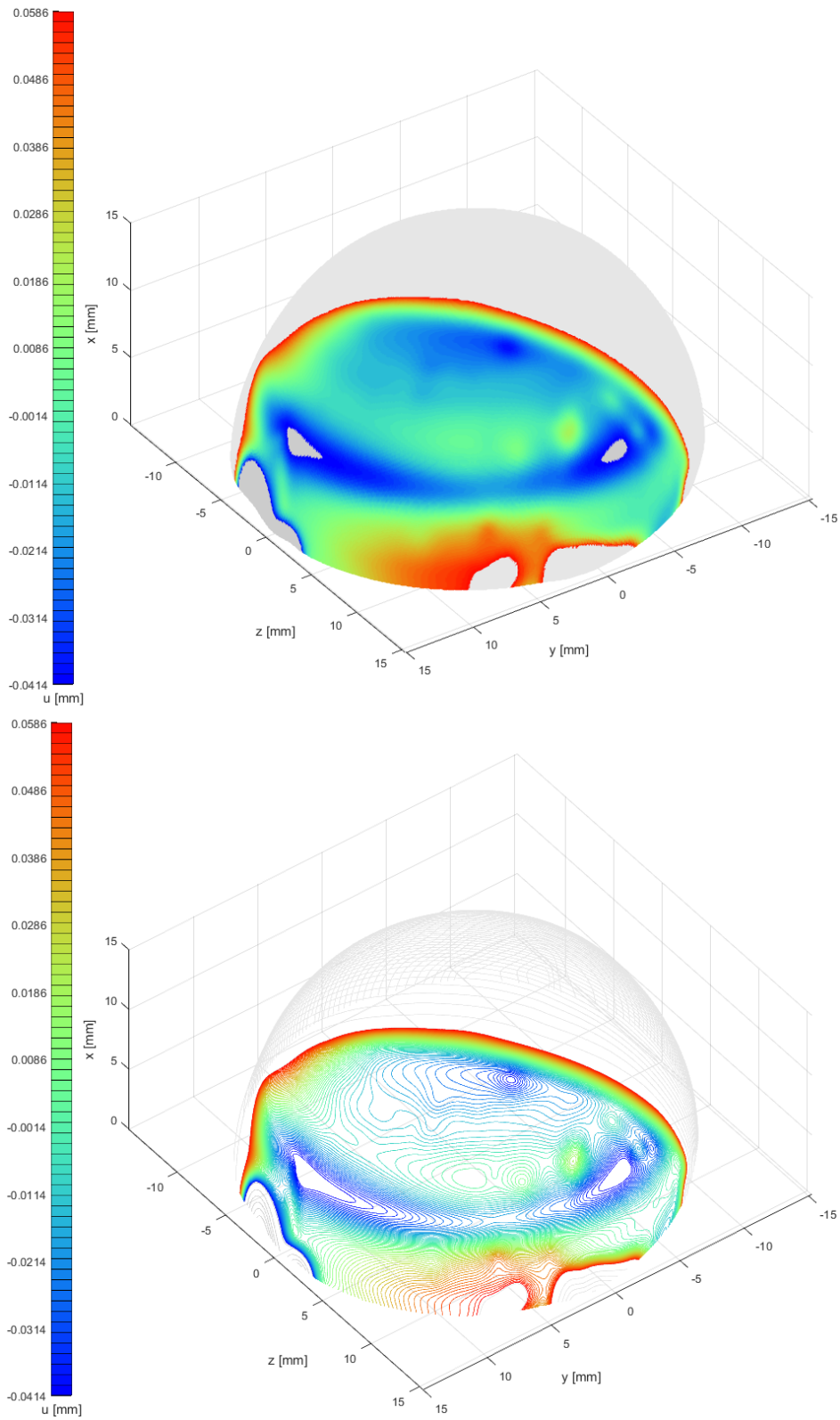
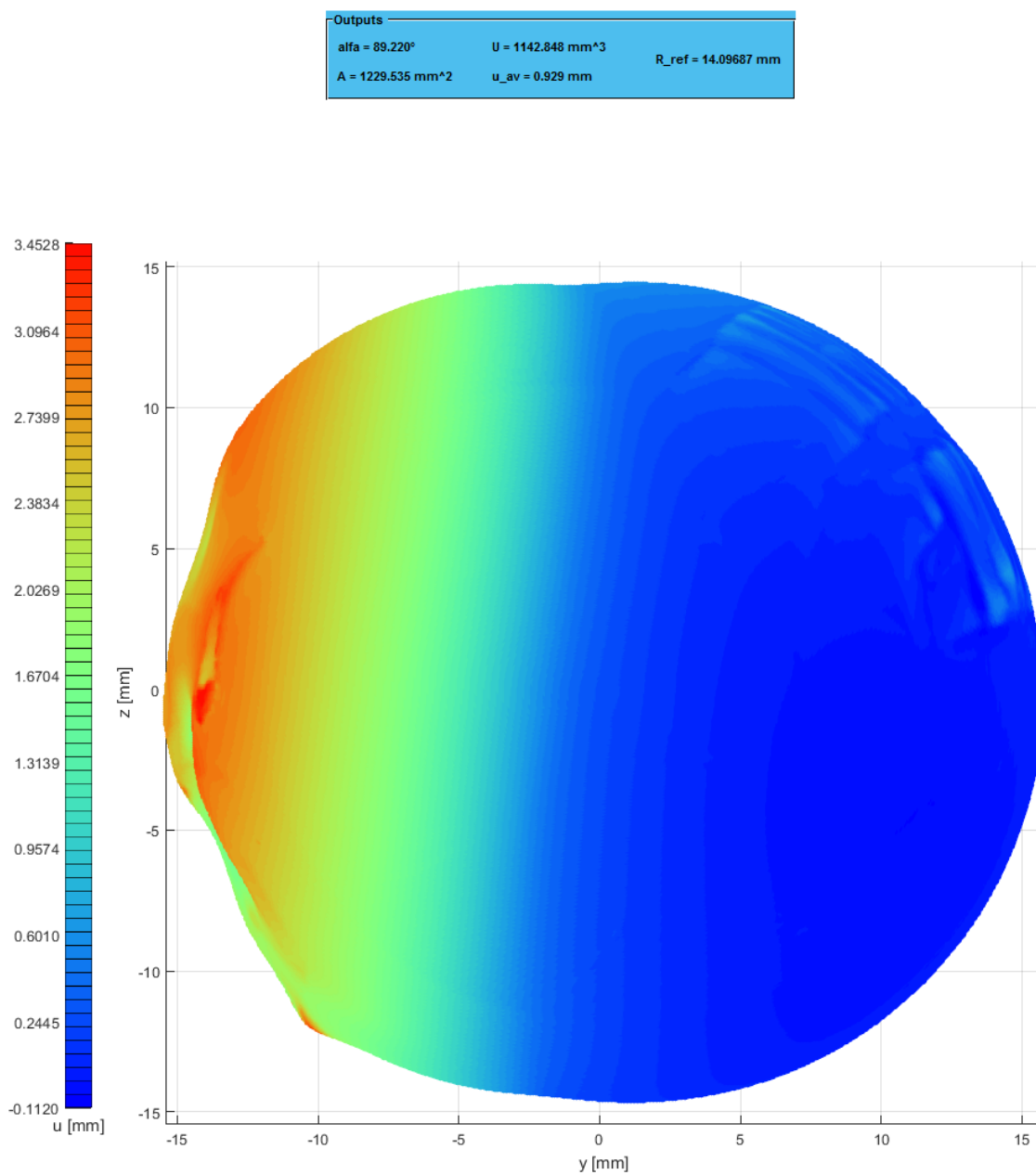


Figure 6.10: **Sample no. 1** - Sample surface inside tolerance boundaries B_{red}

Figure 6.11: **Sample no. 2** - Outputs table and wear map (top view)

6. RESULTS

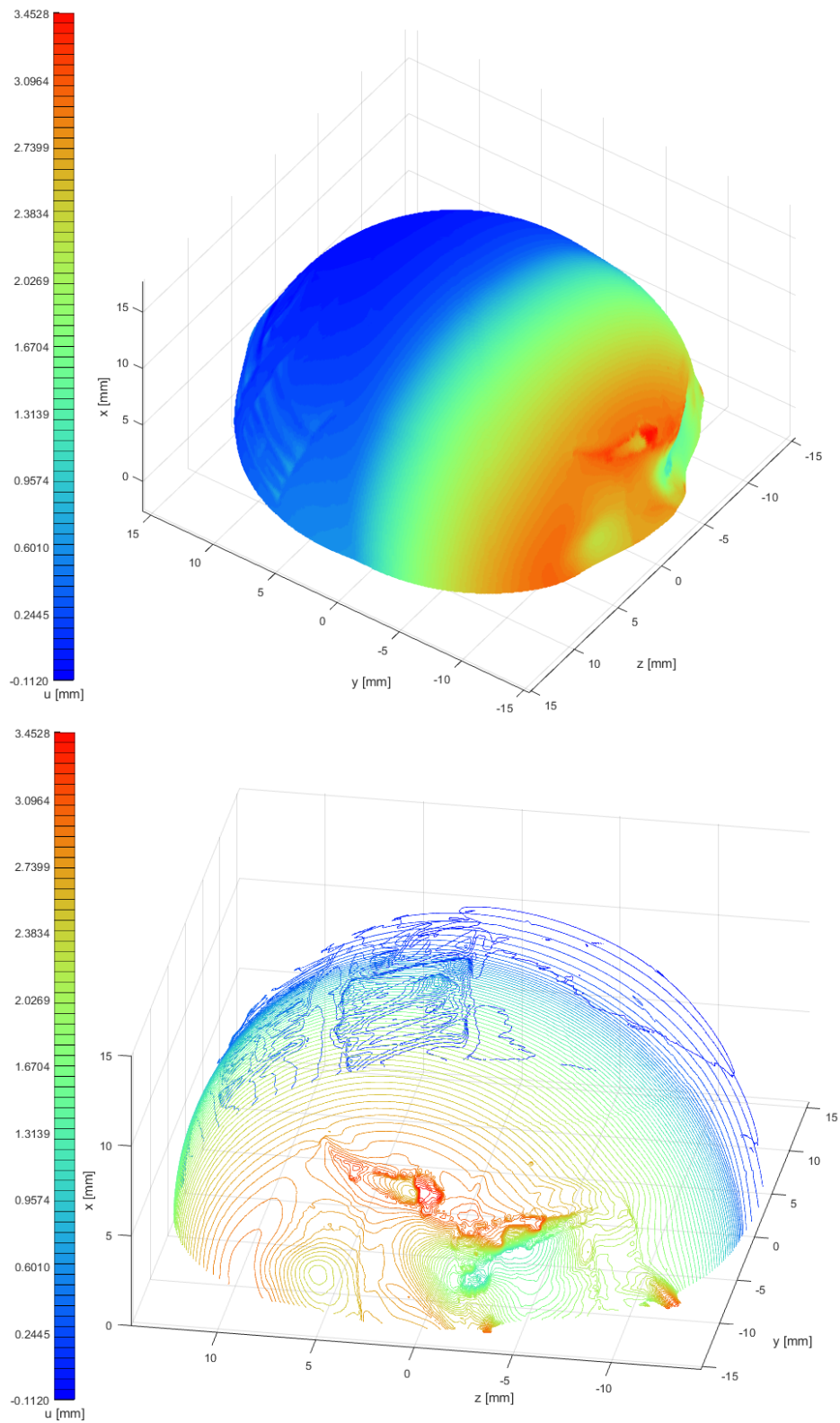


Figure 6.12: **Sample no. 2** - Wear map and wear isolines (side views)

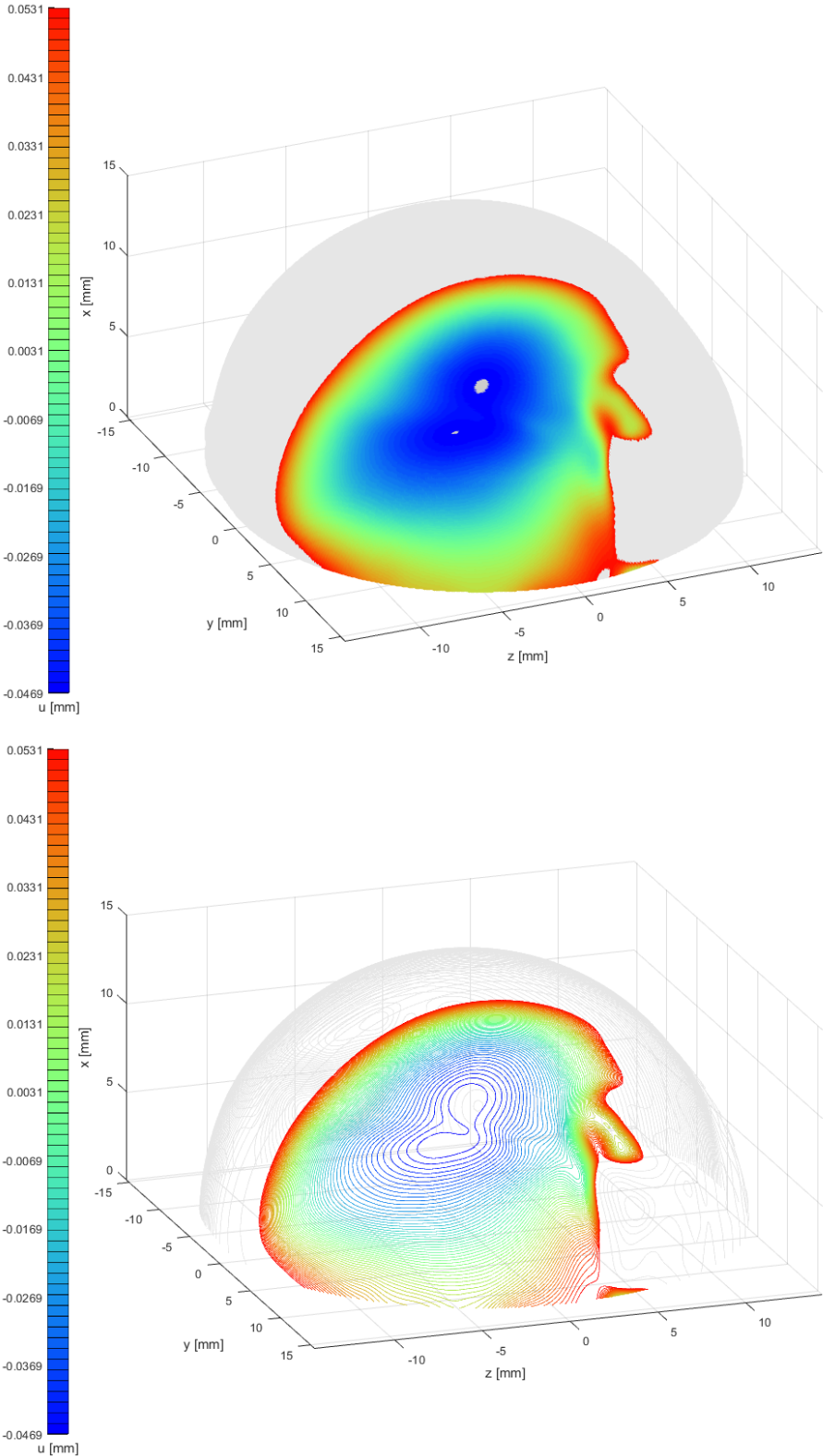


Figure 6.13: **Sample no. 2** - Sample surface inside tolerance boundaries B_{red}

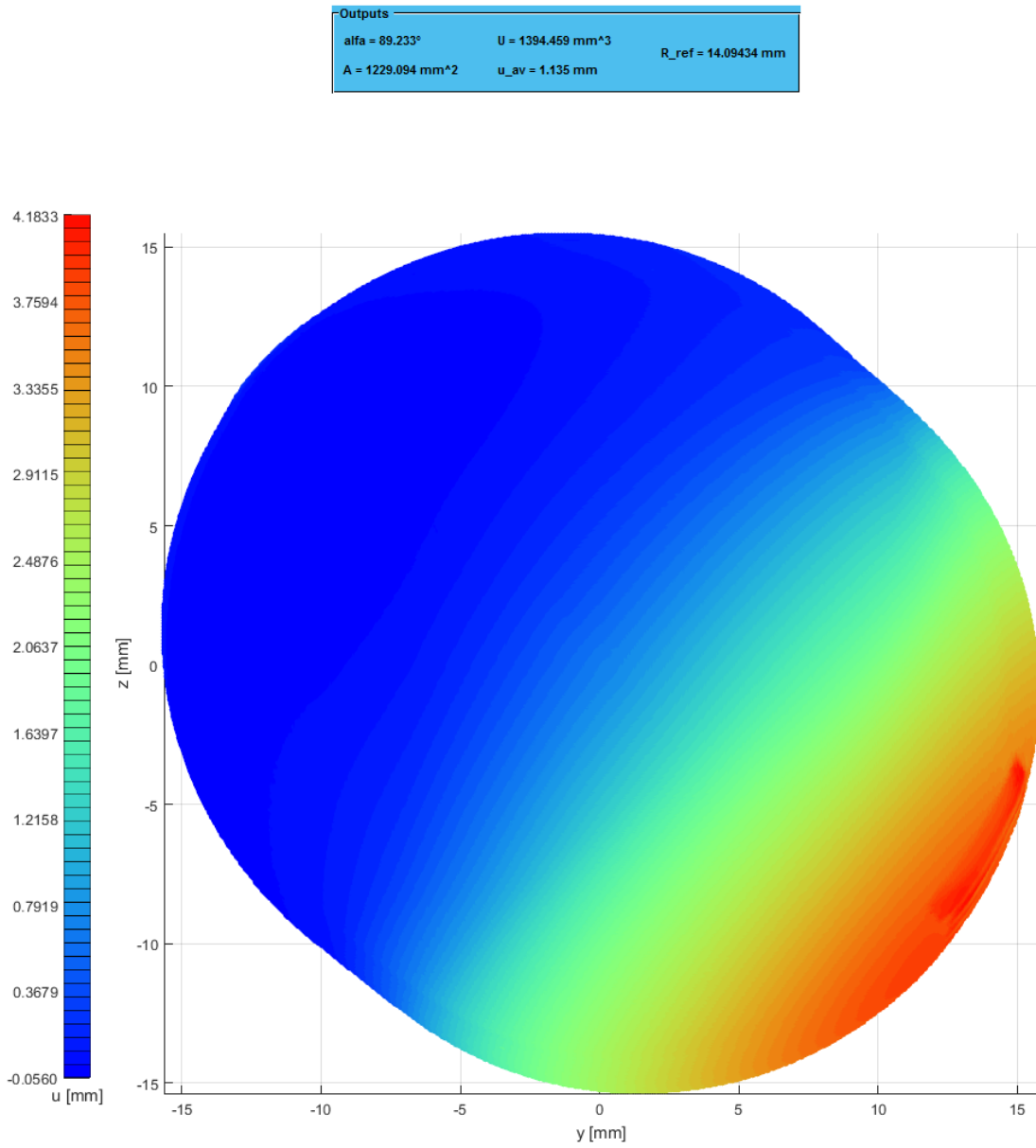
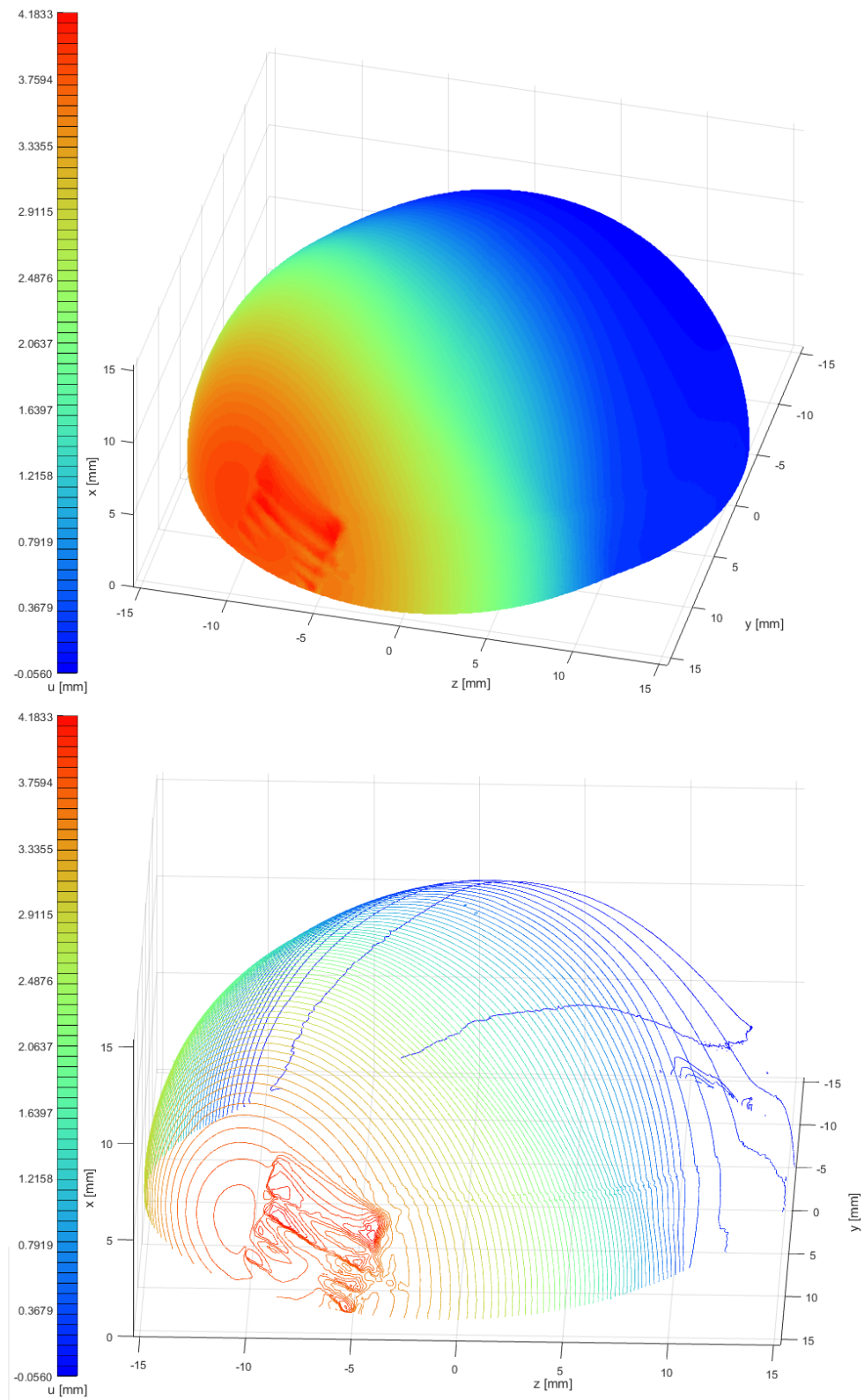


Figure 6.14: **Sample no. 3** - Outputs table and wear map (top view)

Figure 6.15: **Sample no. 3** - Wear map and wear isolines (side views)

6. RESULTS

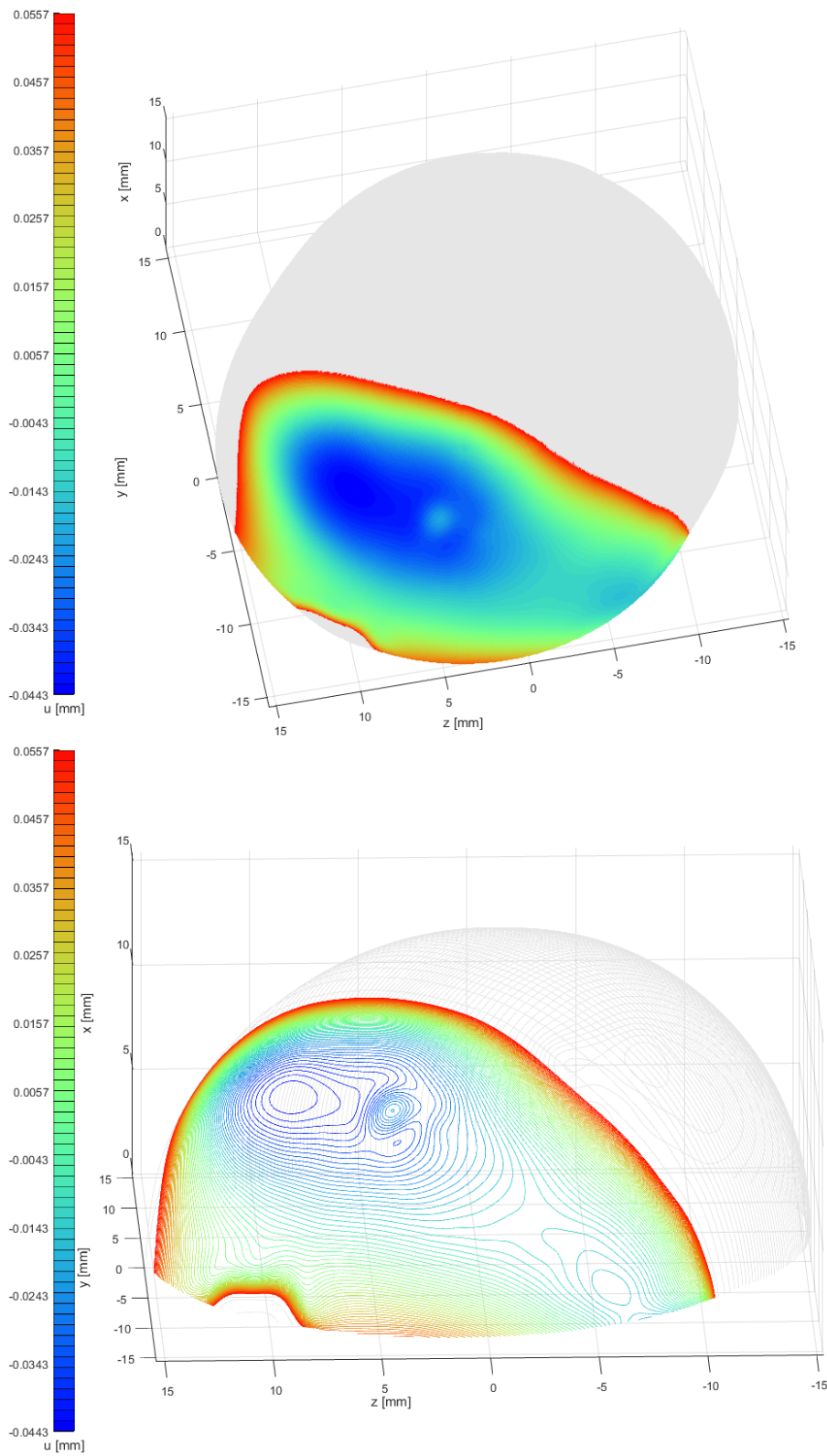
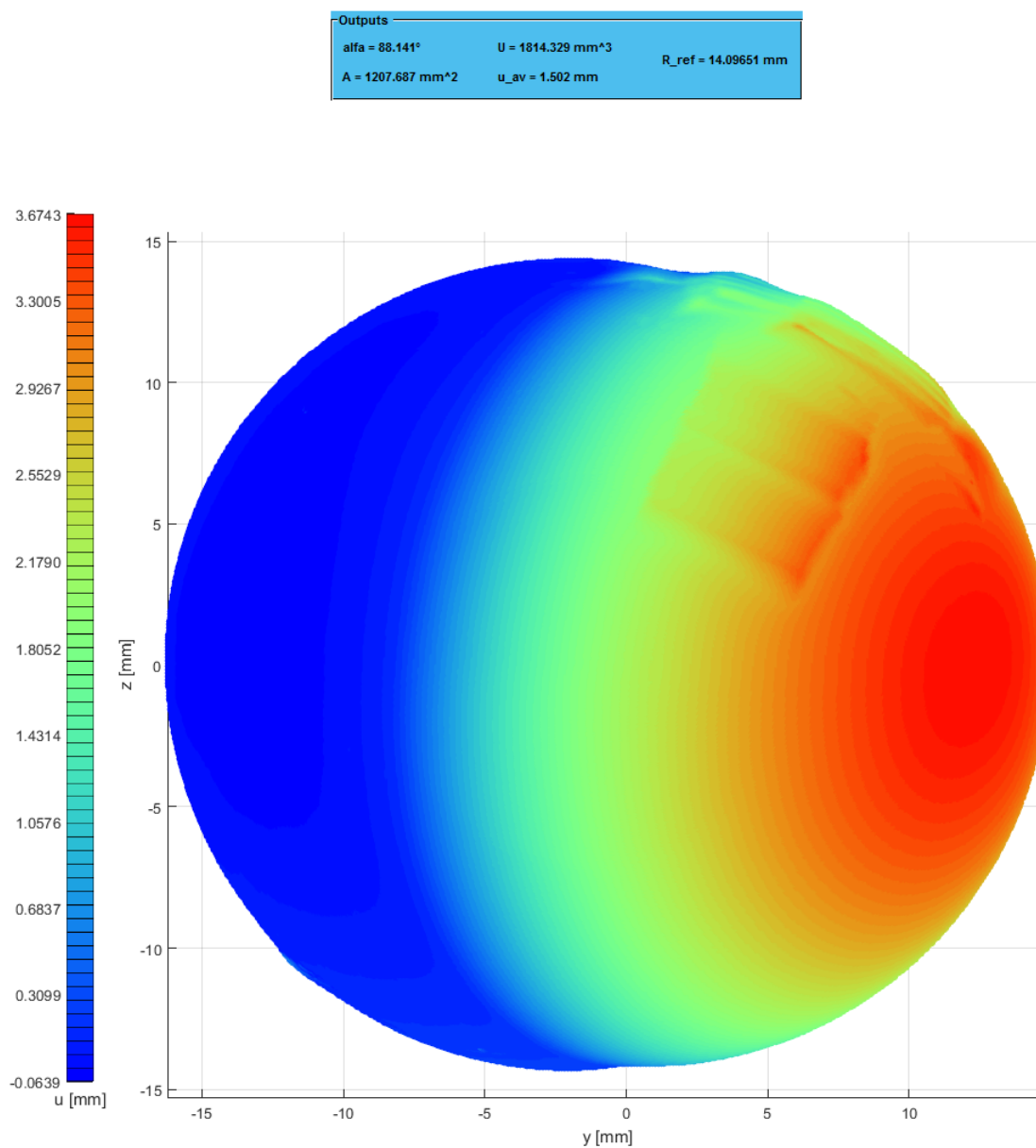


Figure 6.16: **Sample no. 3** - Sample surface inside tolerance boundaries B_{red}

Figure 6.17: **Sample no. 4** - Outputs table and wear map (top view)

6. RESULTS

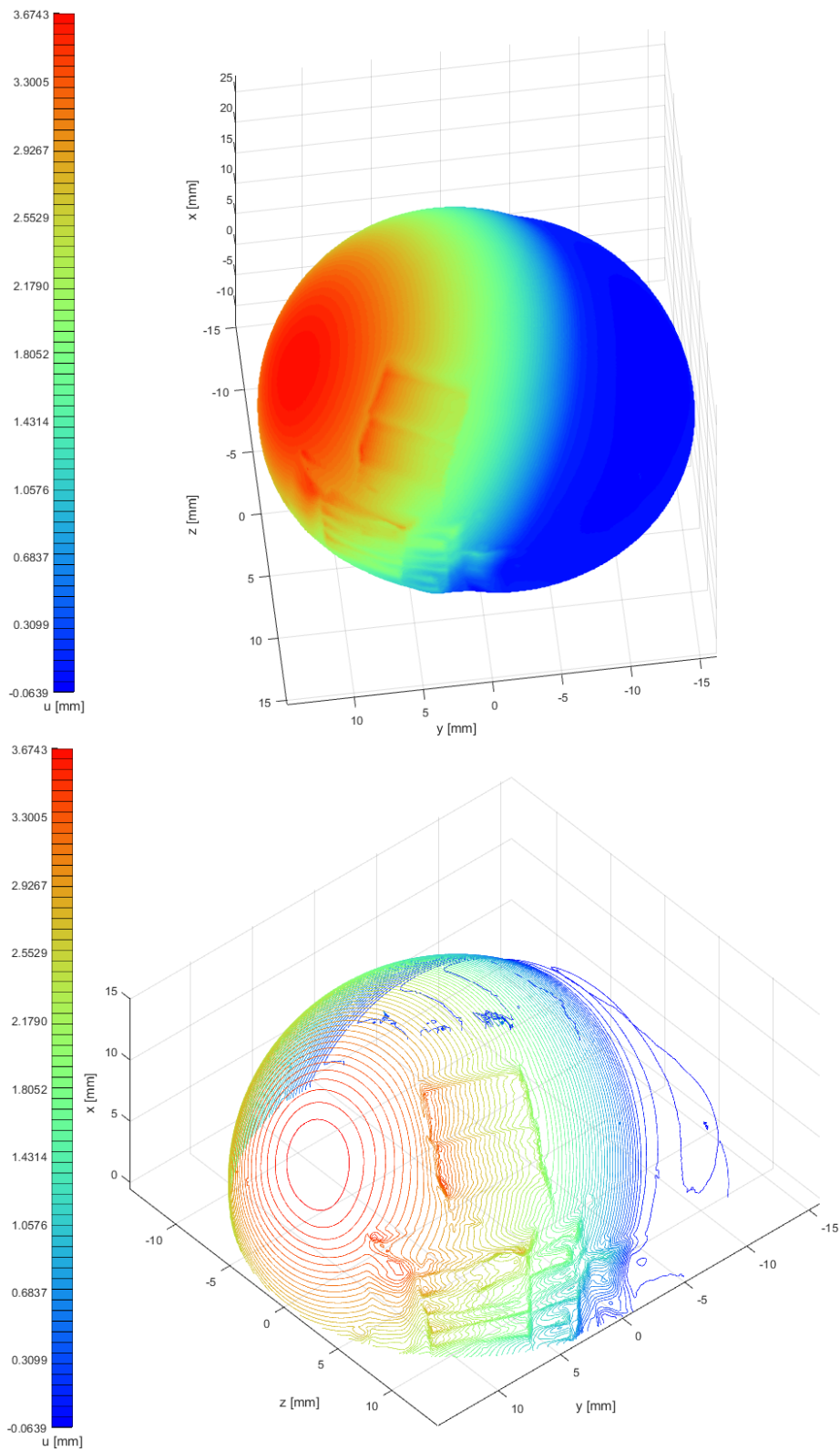


Figure 6.18: **Sample no. 4** - Wear map and wear isolines (side views)

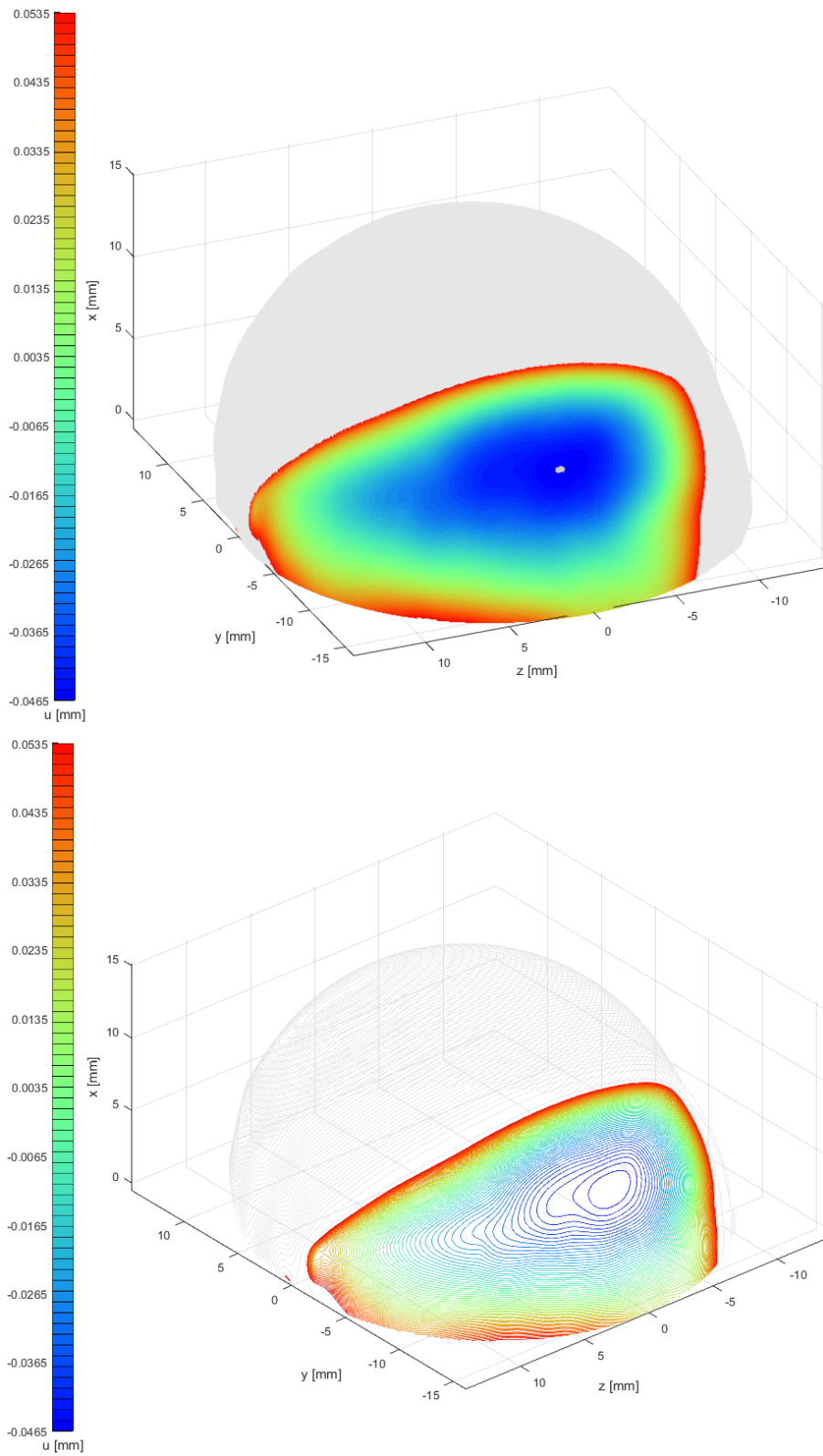


Figure 6.19: **Sample no. 4** - Sample surface inside tolerance boundaries B_{red}

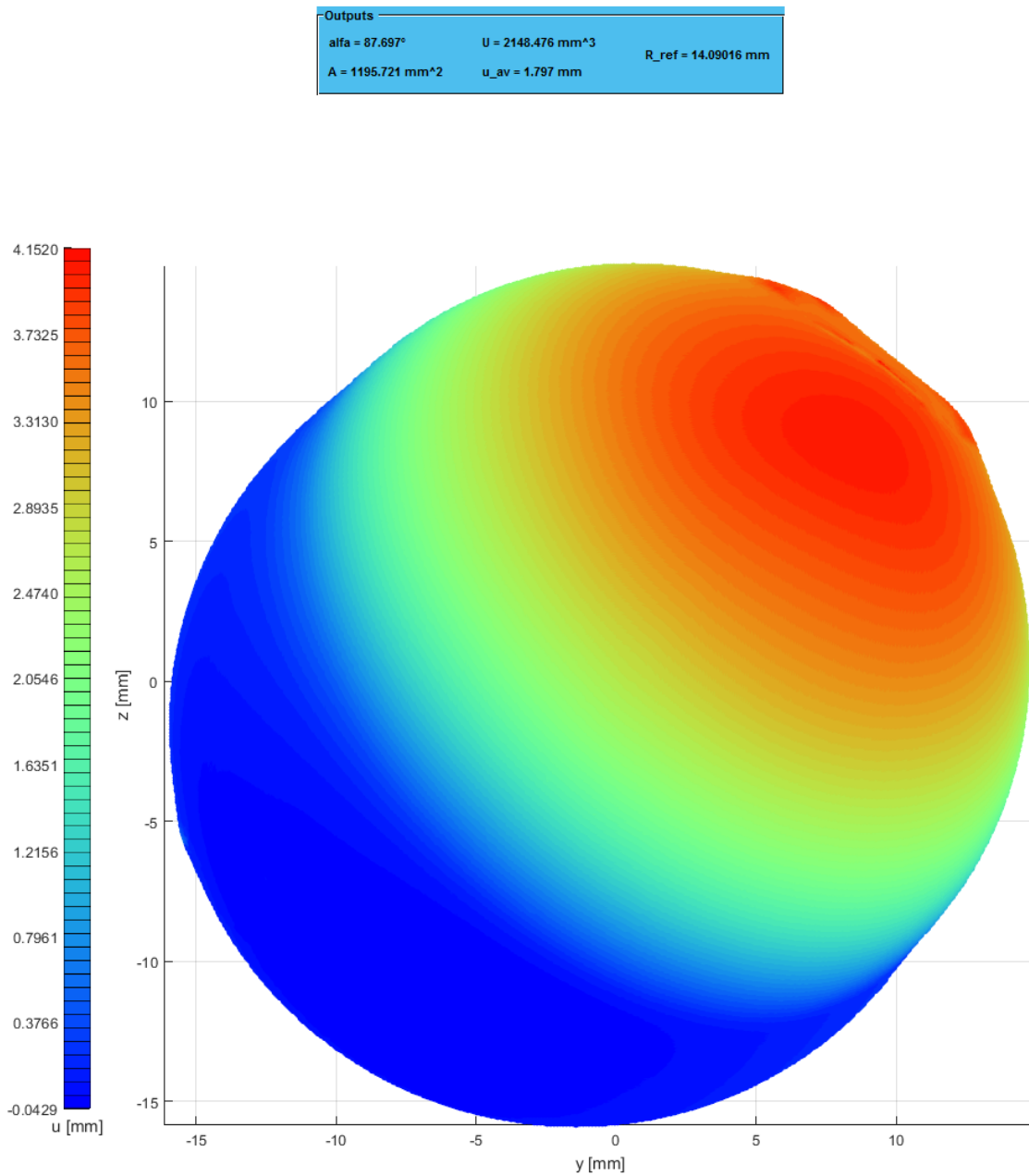
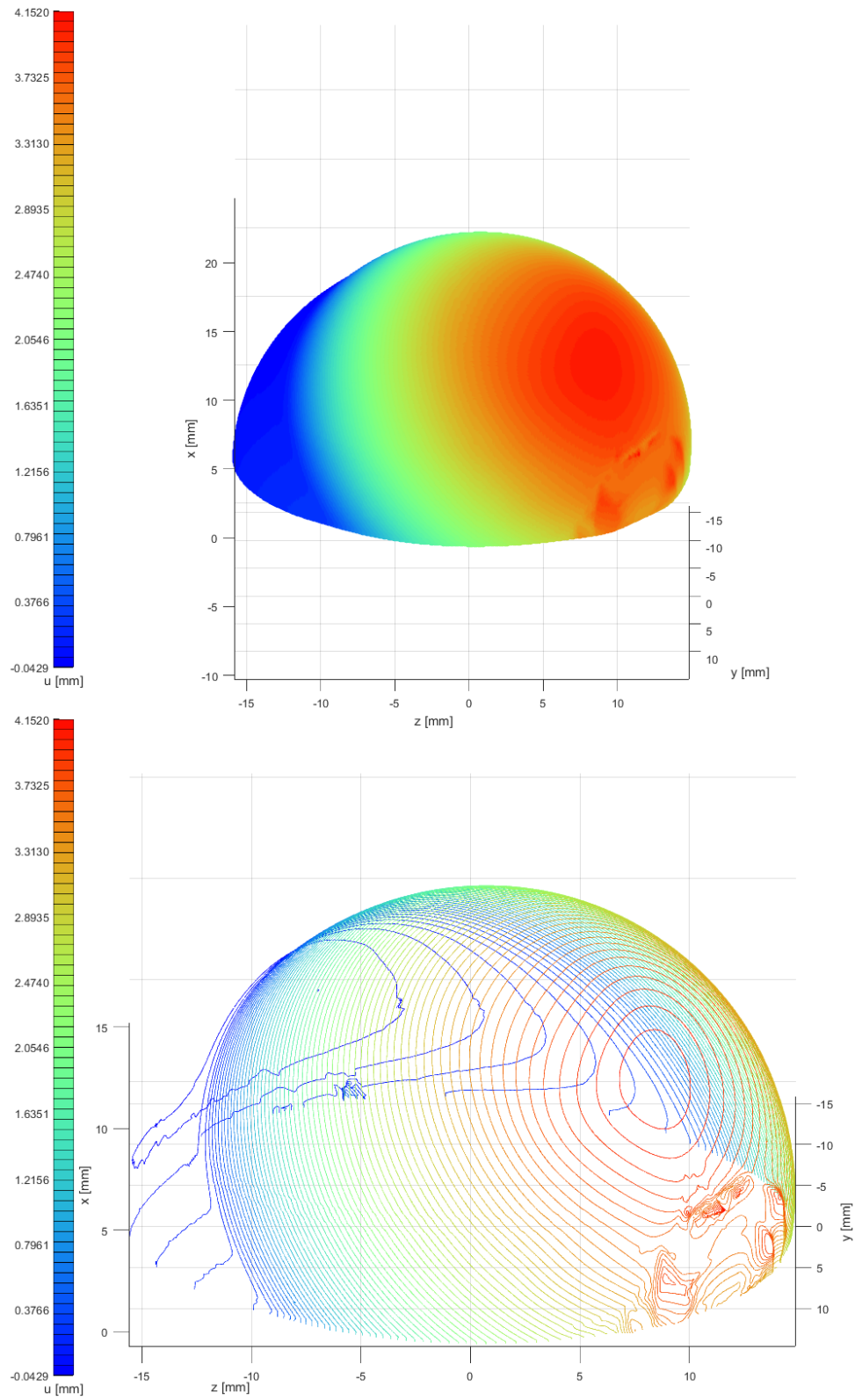


Figure 6.20: **Sample no. 5** - Outputs table and wear map (top view)

Figure 6.21: **Sample no. 5** - Wear map and wear isolines (side views)

6. RESULTS

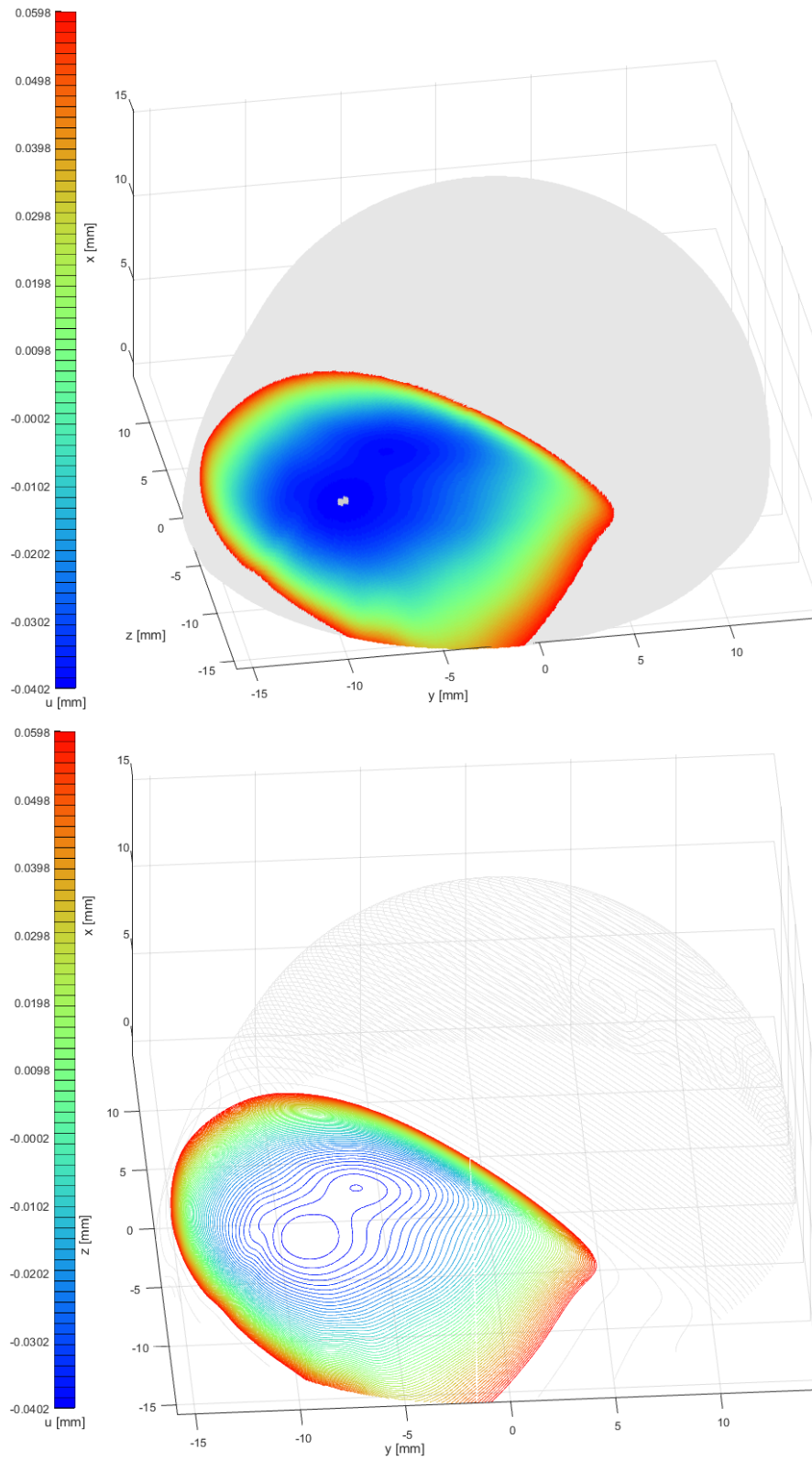


Figure 6.22: **Sample no. 5** - Sample surface inside tolerance boundaries B_{red}

Discussion

Optical 3D CMM by *RedLux Ltd.* offers very auspicious opportunity of pretty accurate *in vitro* wear estimation of cup explants. In accordance with this knowledge, we were facing the task to develop the transformation of the raw measured data into final valuable wear results. In addition, the experience from the laboratory showed that mentioned 3D scanner is not capable of measuring directly the worn surfaces of explanted cups so we were forced to develop a casting method and measure the surfaces of the casts. *Matlab GUI* program was created to analyze the raw data of measured surfaces. The accuracy of the results is influenced by several factors:

- Data measuring - surface 3D scanning
 - *Sample surface is measured discretely.* We measure coordinates of just a certain number of an infinite number of points which occur on measured surface. Therefore, we have to estimate the coordinates of the unmeasured points which logically brings an error. The error is as lower as denser the cloud of measured points is. We have been measuring point cloud with density 720 points per one rotation of the measured sample in the vertical plane and within every single rotation, the measured sample was turned by 0.5° in the horizontal plane. This density is enough to describe the measured surface sufficiently. Accordingly, the error caused by this factor is not significant for our results.
 - *Probe resolution is not absolutely accurate.* The resolution of the probe is 20nm which is more than sufficient for our purposes.
 - *Movements of axes of the 3D scanner are not absolutely accurate.* The resolution of each linear axis is 100nm, the resolution of each rotary axis is $10''$. These numbers are also more than sufficient for our purposes.
 - ***Casts geometry is not identical with the geometry of cup the explants.*** This factor could potentially be a significant source of inaccuracies of the results so it was very important to develop appropriate casting method. In the end, we chose silicone - *Addition Cure Moulding Rubber MM242* as a casting material. The difference between the measured data of the cast and measured data of the cup surface was evaluated in section 6.1 on page 61. According to Fig. 6.2, the Volume change Θ of the silicone cast is 0.104%. According to Fig. 6.3, the Weighted standard deviation of radius sd_w of the silicone cast is 0.007091mm;

the Weighted standard deviation of radius sd_w of the original cup surface is 0.007025mm. After this evaluation, we found silicone - *Addition Cure Moulding Rubber MM242* as sufficient casting material for our purposes. Nevertheless, there is a certain potential to eliminate this factor more. Either to find a better casting material or to take into account correction of this factor within data analysis.

- Data analysis - *Matlab GUI* program
 - *Measured surface is interpolated.* Due to a discrete description of measured surface, we have to interpolate measured data. *Matlab* function *scatteredInterpolant* is used for the realization of the interpolation. Since the cloud of measured points is pretty dense, *scatteredInterpolant* works without any problems. This factor is not significant for the results if Interpolation step Δ is small enough. The value of Interpolation step Δ is set by the user within GUI and its suggested value is 0.25° . Of course, lower value is better for the results but the calculation time increases enormously in that case. Besides, the density of interpolated net shall be similar to the density of measured data so this aspect must be taken into account during setting of Interpolation step Δ in GUI.
 - *Original unworn cup surface is assumed as a perfect sphere.* We do not take into account the inaccuracies given by manufacturing technology. We could do it if the cups were measured before implantation. However, it did not happen so we consider sphere as a reference geometry. The significance of this factor is complicated to evaluate. Nevertheless, since the volumetric wear reaches the values about thousand of cubic millimeters, which explanted cups after 15 years normally do, the significance of this factor is very low.
 - *Reference sphere estimation.* This factor is the crucial one. The position of the reference geometry has a significant influence on the results. Our way of estimation of reference sphere is based on identification of part of original unworn geometry. We try to find the original geometry manufactured according to international standard *ISO 7206-2:1996(E)* [9] and to set the reference geometry in accordance with this original part. This method seems to work well. The evaluation of correctness of reference geometry estimation must be made by the user though. GUI enables to specify the area which is going to be used for identification of reference geometry to avoid bad converge of the estimating method so the user can influence the estimation of reference geometry directly.
 - *Volumetric wear calculation simplifies surface of investigated worn cup.* This factor is related to interpolation problem. The surface is represented by a certain number of interpolated points and each point represents a certain local area of measured surface given by Interpolation step Δ . Within the calculation of volumetric wear, we consider the linear wear has the same value within the whole local area - the value of linear wear right in the interpolated point. Since the values of linear wear of surrounding points in the local area are different, this simplification is a source of inaccuracy. However, when the Interpolation step Δ is small enough, this factor is not significant for the results.

-
- Results provide the information about the change of geometry. They do not distinguish wear from plastic deformation or creep though.
 - The evaluator has to determine the influence of the plastic deformation or creep for the results. The influence can vary case by case.

Verification of the algorithm for wear estimation was published in section 6.2 beginning on page 63. According to Tab. 6.1, the results are very satisfying. Three simulated wear cases were analyzed by created *Matlab GUI* program and the results provided by the program were compared to the actual results calculated analytically. The highest Percent error δ of Volumetric wear U was found 1.879%; the highest Percent error δ of Measured area A was found 0.218%; the highest Percent error δ of Maximal measured angle α was found 0.149%; the highest Percent error δ of Reference radius R_{ref} was found 0.106%; and the highest Percent δ error of maximal Linear wear u_{max} was found 0.622%. According to those numbers, we can notice that Volumetric wear U is more sensitive for the mentioned sources if inaccuracies than the rest of calculated parameters.

Furthermore, the developed wear estimation method was tested within the pilot study. Five real cup explants were cast, their casts were measured by optical 3D CMM by *RedLux Ltd.* and measured data was analyzed by created *Matlab GUI* program. The results of the pilot study are published in section 6.3 beginning on page 67. Tab. 6.2 shows the numerical results of the pilot study, while Fig. 6.8 - 6.22 visualize wear distribution. The calculated values of Volumetric wear U varies between 899.9 - 2148.5mm³; the calculated values of Average linear wear u_{av} varies between 0.736 - 1.797mm. Unfortunately, we have no info about patients' weights so it is tough to make some conclusions. However, it is interesting that the lifetime of sample no. 1, whose value of Average linear wear u_{av} is the lowest one, is the longest one - 18.6 years. The calculated values of maximal Linear wear u_{max} varies between 3.317 - 4.152mm; and the calculated values of minimal Linear wear u_{min} varies between -0.744 - -0.043mm. The values of maximal and minimal Linear wear u_{max} and u_{min} are usually influenced by various issues which are not related to wear. Those issues can be scratches arisen during explantation and damages of similar character. We can easily notice a pliers prints on each of the samples of the pilot study. It is caused by that the doctors use pliers for removal the cup from human body so it has nothing to do with wear but the wear results are obviously influenced by those pliers prints. The evaluator has to take it into account and evaluate each sample individually with his knowledge and experience. Nevertheless, created *Matlab GUI* enables certain possibility to distinguish wear and issues like these. It enables to smooth the surface so the user can visualize the wear distribution eliminated by sharp peaks, which usually are caused by the issues mentioned above.

Conclusion

Since The Laboratory of Biomechanics, CTU in Prague, is involved in the project called *The study of new materials used as articulation surface of joint replacement (ID No. NV15-31269A)*, the need for wear quantification arose. On that account, **the method of wear analysis of explanted acetabular cups based on accurate surface measurement was developed.**

For successful surface measurement, it was necessary to develop casting method of cup explants. Several materials were tested as a casting material. The differences between casting surfaces and the surface of the original cup were evaluated. In the end, Silicone - *Addition Cure Moulding Rubber MM242* was chosen as the casting material, which fulfills all our requirements.

The measured data is analyzed at created *Matlab GUI* program, which is able to calculate and visualize wear parameters. This program was verified by data of worn cup surfaces, whose wear parameters we are able to calculate analytically. The comparison between the results by *Matlab GUI* program and the actual results calculated analytically showed very satisfying similarity. Afterward, the samples of real cup explants were analyzed by created *Matlab GUI program* within the pilot study.

This thesis introduces a very powerful tool for wear quantification and visualization. Within the created *Matlab GUI* program, we are capable of analyzing a relatively large number of measured surfaces very easily and quickly.

Bibliography

- [1] *Acta chirurgiae orthopaedicae et traumatologiae Čechoslovaca*. Prague: Česká společnost pro ortopedii a traumatologii, 1926-, y. 2010. ISSN 0001-5415.
- [2] YH, Zhu, Chiu KY a Tang WM. Polyethylene wear and osteolysis in total hip arthroplasty. *Journal of Orthopaedic Surgery*. , volume: 9. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/12468851>
- [3] LANDOR, Ivan. Revizní operace totálních náhrad kyčelního kloubu. Prague: Maxdorf, c2012. Jessenius. ISBN 978-80-7345-254-4.
- [4] BARRACK, Robert L., Carlos LAVERNIA, Edward S. SZUSZCZEWICZ a Jaswin SAWHNEY. Radiographic wear measurements in a cementless metal-backed modular cobalt-chromium acetabular component. *The Journal of Arthroplasty*. 2001, 16(7), 820-828. DOI: 10.1054/arth.2001.26589. ISSN 08835403. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0883540301641457>
- [5] KOŠAK, R., V. ANTOLIČ, V. PAVLOVČIČ, V. KRALJ-IGLIČ, I. MILOŠEV, G. VIDMAR a A. IGLIČ. *Polyethylene wear in total hip prostheses: the influence of direction of linear wear on volumetric wear determined from radiographic data*. DOI: 10.1007/s00256-003-0685-2. ISBN 10.1007/s00256-003-0685-2. Available from: <http://link.springer.com/10.1007/s00256-003-0685-2>
- [6] TUKE, Mike, Andy TAYLOR, Anne ROQUES a Christian MAUL. 3D linear and volumetric wear measurement on artificial hip joints—Validation of a new methodology. *Precision Engineering*. 2010, 34(4), 777-783. DOI: 10.1016/j.precisioneng.2010.06.001. ISSN 01416359. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0141635910000905>
- [7] MERVART, Jan. *Modelování otěru kloubních náhrad*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Mechanical Engineering, 2013
- [8] VALÁŠEK, Michael, Jiří BŘEZINA a Vladimír STEJSKAL. *Mechanika A*. Prague: Vydavatelství ČVUT, 2004. ISBN 80-01-02890-9.

- [9] *Formální úprava vědeckých a technických zpráv: ČSN ISO 5966 (01 0173)*. Prague: Český normalizační institut, 1995. Dokumentace, 5966 (01 0173).
- [10] KOŠAK, Robert, Veronika KRALJ-IGLIČ, Aleš IGLIČ a Matej DANIEL. Polyethylene Wear is Related to Patient-specific Contact Stress in THA. *Clinical Orthopaedics and Related Research*. 2011, 469(12), 3415-3422. DOI: 10.1007/s11999-011-2078-5. ISSN 0009-921x. Available from: <http://link.springer.com/10.1007/s11999-011-2078-5>
- [11] ALLEPUZ, A., L. HAVELIN, T. BARBER, et al. Effect of Femoral Head Size on Metal-on-HXLPE Hip Arthroplasty Outcome in a Combined Analysis of Six National and Regional Registries. *The Journal of Bone*. 2014, 96(Supplement_1), 12-18. DOI: 10.2106/JBJS.N.00461. ISSN 0021-9355. Available from: <http://jbjs.org/cgi/doi/10.2106/JBJS.N.00461>

Nomenclature

note: Listed symbols can be indexed in various ways throughout whole thesis. Indexes specify local utilization of the symbols, whereas general definition of the symbols is mentioned here.

| Symbol | Units | Description | Definition |
|--|----------------|---|-------------------------------|
| <i>Coordinate systems</i> | | | |
| (x, y, z) | m,m,m | Cartesian coordinates | 4.1.1 Figure 4.1 |
| (r, ξ, θ) | m,rad,rad | Spherical coordinates | Figure 4.3 |
| (r, s_1, s_2) | m,m,m | Polar coordinates | Figure 4.3 |
| <i>Surface area & Volume</i> | | | |
| Δ | rad | Interpolation step | 4.1.2 |
| B | m ² | Surface area | |
| V | m ³ | Volume | |
| <i>Data transformation</i> | | | |
| S | | Matrix of directional cosines | 4.2.2 |
| i, j, k | m | Unit vectors of the axes of a Cartesian c. s. | |
| <i>Reference sphere</i> | | | |
| B_{red} | m ² | Sample surface inside tolerance boundaries | 4.2.5 Figure 4.15 |
| R_{max}, R_{min} | m | Tolerance boundaries | Figure 4.15 |
| R_{ref} | m | Reference radius | |
| $[x_s, y_s, z_s]$ | m | Coordinates of reference sphere center point | Figure 4.15 |
| <i>Calculation outputs</i> | | | |
| α | ° | Maximal measured angle | 4.2.7 - 4.2.11 Figure 4.17 |
| A | m ² | Measured area | Figure 4.17 |
| U | m ³ | Volumetric wear | |
| u | m | Linear wear | |
| u_{av} | m | Average linear wear | |
| <i>Verification of casting method</i> | | | |
| Θ | % | Volume change | 3.2 |
| R_w | m | Weighted average of radius | |
| sd_w | m | Weighted standard deviation of radius | |
| <i>Verification of algorithm</i> | | | |
| δ | % | Percent error | 6.2 |

Acronyms

CD Compact disc

CMM Coordinate measuring machine

CTU Czech Technical University

FEM Finite element method

FME Faculty of Mechanical Engineering

GUI Graphical user interface

NaN Not-a-Number

PEEK Polyether ether ketone

RTG Radioisotope thermoelectric generator

UHMWPE Ultra-high-molecular-weight polyethylene

Contents of enclosed CD

```
MT_Mervart_2016.pdf
├── Attachments
│   ├── A - Cup Analysis (Matlab GUI)
│   ├── B - Cup Analysis GUIDELINE.pdf
│   ├── C - Pilot study
│   ├── D - Verification of casting method
│   ├── E - Verification of algorithm
│   └── F - Downloaded Matlab functions
```