FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGICAL AND SCIENTIFIC CENTER
CONTROL AND AUTOMATION ENGINEERING UNDERGRADUATE

Helena Borges Daré

Implementation of Methods for Automated Processing of Gear Topographies

Aachen
2023

Helena Borges Daré

Implementation of Methods for Automated Processing of Gear Topographies

Final Course Project submit to the Control and Automation Engineering Undergraduate course from the Federal University of Santa Catarina for obtaining the title of bachelor of Control and Automation Engineering.
Supervisor:: Prof. Hector Bessa Silveira, Dr.
Co-supervisor:: Marius Willecke, Msc.

Aachen
2023

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

http://portalbu.ufsc.br/ficha

Helena Borges Daré

Implementation of Methods for Automated Processing of Gear Topographies

This work in bachelor level was evaluated and approved by the examining board composed by the following members:

Advisor: Prof. Hector Bessa Silveira, Dr.
Federal University of Santa Catarina

Co-adviser: Marius Willecke, M.Sc.
Rheinisch-Westfälische Technische Hochschule

Reviewer: Prof. Gabriel Thaler, M.Sc.
Federal University of Santa Catarina

President of the committee: Prof. Eduardo Camponogara, Dr.
Federal University of Santa Catarina

We certify that this is the **final and original version** of the final course project which was considering appropriate for obtainig the title of bachelor of Control and Automation Engineering.

To my dear parents, Elenice and Gilberto.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to God, who has been my source of strength and protection during these challenging times in my life.

I would also like to express my sincere gratitude to the WZL for providing me with a unique and enriching experience while working at the Institute. Through this opportunity I was able to live in Germany, learn more about this beautiful language and improve my technical skills.

A special acknowledgment goes to my supervisor, Marius Willecke, M.Sc. for his invaluable guidance and teachings throughout this project. I would also like to thank my advisor at UFSC, Prof. Hector Bessa Silveira, Dr., for his motivation, dedication, and commitment to producing high quality work.

I owe a debt of gratitude to my parents, Elenice and Gilberto, and my family for their prayers and unfailing support that have allowed me to be where I am today.

I would also like to thank Thiago Raulino Dal Pont, M.Sc. for his generous support, constructive feedback, and unwavering confidence in me throughout this project.

I would also like to thank all my Brazilian colleagues who accompanied me during this year-long internship. I learned a lot from each of their experiences and their presence made my time at the WZL more enjoyable. Last but not least, I would like to thank all my friends at the University who always believed in me and supported me.

" Ancora imparo (still, I am learning)"
(Michelangelo, 87)

ABSTRACT

There are various file formats, such as MKA, PKT, and GDE, for transferring dimensional and topological data of gears between machines, software, and users, but interoperability is often lacking. Additionally, errors in gear measurement using measurement machines make it difficult to perform accurate simulations. To solve this problem, automatic processing methods for gear topographies were developed to increase the effectiveness of tooth contact simulations. The project includes three main stages: measurement, data reading and processing, and simulation. Data reading from GDE files is performed using algorithms written in C and Fortran programming languages, including a sorting routine that organizes input data for the ZaKo3D simulation software. The data processing stage uses Python's Scipy libraries to correct incomplete or badly measured data from gear measurement machines. GDE files of fully measured spur and helical gears were generated with irregularities in the topography, and various interpolation methods were investigated, including "Griddata", "RBFInterpolator", and "BSpline", using heatmaps to compare original and predicted values. In addition, hyperparameter optimization of the interpolator was conducted using performance metrics such as Root Mean Square Error (RMSE) and Maximum Absolute Error (MaxAE). The best interpolation results were obtained with the Radial Basis Function (RBF), which smoothed irregularities and proved that partially measured surfaces can be reconstructed by extrapolation. Furthermore, reading GDE files is efficient, allowing for the correct transfer of gear data to the coding structure of the Werkzeugmaschinenlabor (WZL) Gear Technology Team. The results clearly demonstrate that the implemented data reading and processing methods increase process automation, minimizing user errors. Additionally, reconstructing surfaces from partially measured originals can reduce measurement time and promote more effective simulations.

Keywords: Gear Topography. GDE file format. RBF Interpolator.

RESUMO

Há diferentes formatos de arquivos, como MKA, PKT e GDE, para transferência de dados dimensionais e topológicos de engrenagens entre máquinas, softwares e usuários, mas frequentemente há falta de interoperabilidade entre eles. Além disso, as falhas na medição de engrenagens em máquinas de medição dificultam a realização de simulações precisas. Para resolver esse problema, foram desenvolvidos métodos de processamento automático de topografias de engrenagens para aumentar a eficácia das simulações de contato entre dentes de engrenagens. O projeto contempla três fases principais: medição, leitura e processamento de dados e simulação. A etapa de leitura de dados em arquivos GDE é feita com algoritmos escritos em linguagens de programação C e Fortran, incluindo uma rotina de classificação que organiza os dados de entrada para o software de simulação de ZaKo3D. Enquanto o estágio de processamento de dados utiliza bibliotecas Scipy do Python para corrigir dados incompletos ou mal medidos pelas máquinas de medição de engrenagens. Para isso, foram gerados arquivos GDE de engrenagens helicoidais e de dentes retos totalmente medidos, mas com irregularidades na topografia. Thês métodos de interpolação foram investigados: “Griddata”, “RBFInterpolador” e “BSpline”, usando heatmaps para comparar os valores originais e previstos. Além da otimização de hiperparâmetros do interpolador através de métricas de desempenho, como a Raiz Quadrática Média (RQM) e o Máximo Erro Absoluto (MaxEA). Os melhores resultados de interpolação foram obtidos com a Função de Base Radial, que suavizou as irregularidades e comprova que as superfícies parcialmente medidas podem ser reconstruídas por meio de extrapolação. Além disso, a leitura dos arquivos GDE é eficiente, permitindo a transferência correta dos dados de engrenagens para a estrutura de códigos do Time de Tecnologia de Engrenagens do Werkzeugmaschinenlabor (WZL). Assim, os resultados demonstram claramente que os métodos de leitura e processamento de dados implementados aumentam a automação do processo, minimizando erros de usuário. Ademais, a reconstrução de superfícies a partir das originais parcialmente medidas, pode reduzir o tempo de medição e promover simulações mais eficazes.

Palavras-Chave: Topografia de Engrenagem. Formato de arquivo GDE. Interpolador RBF.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AE	Absolute Error
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CMM	Coordinate Measuring Machine
CNC	Computer Numerical Control
DIN	German institute for Standardization
DTD	Document Type Definition
FE	Finite Element
FW	Face Width
GDE	Gear Data Exchange
HPC	High-Performance Computing
HTML	HyperText Markup Language
MaxAE	Maximum Absolute Error
MIT	Massachusetts Institute of Technology
NTCA	Numerical Tooth Contact Analysis
NURBS	Non-Uniform Rational B-spline
PKT	General Exchange Format
PTB	Physikalisch-Technische Bundesanstalt
RBF	Radial Basis Function
RD	Root Form Diameter
RMSE	Root Mean Square Error
SG	Spur Gear
TCA	Tooth Contact Analysis
TP	Tip Diameter
UFSC	Federal University of Santa Cataria
VDI	Association of German Engineers
WZL	Machine Tool Laboratory
XML	Extensible Markup Language

LIST OF SYMBOLS

Symbol	Description	Unit
z	Teeth number	
m_n	Normal module	$[mm]$
α_n	Pressure angle	$[\circ]$
β	Helix angle	$[\circ]$
β_b	Base circle diameter	$[mm]$
q	Grinding allowance	$[mm]$
d	Pitch diameter	$[mm]$
d_{Ca}	Tip relief diameter	$[mm]$
d_{Cf}	Root relief diameter	$[mm]$
d_a	Tip diameter	$[mm]$
d_b	Face width	$[mm]$
u	Transmition ration	
d_f	Root diameter	$[mm]$
d_{Fa}	Tip form diameter	$[mm]$
d_{Ff}	Root form circle diameter	$[mm]$
f_w	Face width (code)	$[mm]$
$f_{H\alpha}$	Profile line angle deviation	$[\mu m]$
$f_{H\beta}$	Flank line angle deviation	$[\mu m]$
h_{ap}	Tooth tip height	$[mm]$
h_{fp}	Tooth root height	$[mm]$
k	Addendum modification factor	$[mm]$
p_t	Pitch face	$[mm]$
R	Right flank	
L	Left flank	
t_d	Tip Diameter	$[mm]$
r_d	Root form diameter (code)	$[mm]$
$\Phi(x)$	Radial Basis Function over input x	
γ_i	Weight for iteration i in Radial Basis Function summation	
p_k	Polinomial degree	
p_{bt}	Pitch base circle	$[mm]$
b	Face width	$[mm]$
d_y	Reference diameter	$[mm]$

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

Gearboxes have a long history dating back to ancient times. The most extensive use of gearbox began with the Industrial Revolution of the 18th and 19th centuries, developing more advanced geared transmissions for use in machinery such as textile mills, steam engines, and locomotives. In addition, gears and gearboxes are used in various applications, including automobiles, airplanes, and industrial equipment (BANODIYA; KARMA, 2017).

However, during gear operation, it is common for contact problems to occur between gear surfaces. Transmission errors include problems such as rotational delay between the driving and driven gears. This is caused by the disturbance of random noise factors such as manufacturing errors, and misalignment during assembly (BANODIYA; KARMA, 2017).

In order to represent the actual application behavior of gears in as much detail as possible, the Gear Technology Team from the Chair of Machine Tools, Laboratory for Machine Tools and Production Engineering Machine Tool Laboratory (WZL) at RWTH Aachen University, developed the program ZaKo3D. The ZaKo3D software is a program within the institute's product range for performance Tooth Contact Analysis (TCA). However, the software has as its main drawback the fact that the point clouds used in the simulation are obtained from a gear measuring center, and sometimes the measurement is inaccurate.

Besides the gear manufacturing problems mentioned by Xu et al. (2015), some issues arise from the measurement process itself on Coordinate Measuring Machine (CMM). When a CMM is used to measure the gear topography, measurement errors are practically inevitable. Many factors are causing these measurement errors, including the CMM itself, the measurement sampling techniques used in the probe radius and probe approach, and inadequate cleaning of the piece (WANG et al., 2022).

Furthermore, different measuring machines operate with different file formats for saving gear data. However, the lack of standardization of gear data has made this process complex. In light of this, the Gear Data Exchange (GDE) file format, defined by the Association of German Engineers (VDI), was developed as a flexible data format based on the Extensible Markup Language (XML) format.

As a result, two major problems hinder the correct and accurate inspection of gears. The first was the influence of CMM errors on 3D gear measurements, resulting in incorrectly measured gear geometry data (LIN, H. et al., 2020). The second problem was the difficulty of transferring gear data between CMM and simulation software, or between different simulation software.

Thus, in the context of the first issue, many proposals have been developed to

digitally recreate the tooth surface and smooth out flaws. For instance, Chen-Hsiang Lin and Fong (2015) proposed the Numerical Tooth Contact Analysis (NTCA), in which the tooth surface is measured from a real gear and reconstructed as a B-spline free-form surface or using a Non-Uniform Rational B-spline (NURBS) from Lin Yang et al. (2011) for beveled gears. In addition, Mo et al. (2022) suggests using Radial Basis Function (RBF) under usual constraints. RBF has become one of the leading methods for point cloud fitting because it has a solid mathematical basis, the advantage of the simplicity of computation, and the ability to process non-uniform points.

At last, in order to deal with the second issue, a structured process for handling gear data has been designed by the Gear Technology Team in WZL. This data structure is part of the WZL library and allows files in formats such as MKA and PKT to be read/written and then converted between different formats. However, file formats such as GDE were not yet fully structured, so GDE reading routines were developed.

1.2 OBJECTIVES

1.2.1 General Objective

Perform gear-tooth contact simulations with partially measured or mismeasured gear data from the spur and helical gears and develop reliable software that communicates efficiently with the other code structures of the WZL to create a connected end-to-end pipeline (measurement to simulation).

1.2.2 Specific objectives

- Implement a code capable of numerically recreating gear surfaces using interpolation techniques;
- Develop a structured test routine in Fortran and Python, including automated reading/writing GDE files;
- Validate the interpolation algorithm by running TCA free of load on the ZaKo3D simulation software.

1.3 OVERVIEW OF THE PROPOSED SOLUTION AND OBTAINED RESULTS

After analyzing the questions, motivations, and objectives raised in section 1.2, this project is structured into three primary stages: measurement, data reading and processing, and simulation, with particular emphasis on the data processing step. The measurement tests were conducted using Kingelnberg p16 with spur, helical, and bevel gears.

The primary objective of this approach is to implement high-performance data reading routines for GDE files and organize the data in a readable manner for the ZaKo3D software. To achieve this, the project utilized programming languages such as C and

Fortran and the LibXML2 library, which was specifically developed for navigating and reading files in XML format. It's worth mentioning that the result verification process involved comparing the input and output files rewritten in GDE.

The gear topography data processing step is essential to overcome any ill-measured or partially measured gear surfaces. This step involves implementing experiments with three interpolators from Python's `scipy.interpolator` library: `RBFInterpolator`, `Griddata`, and `BSpline`. Furthermore, a hyperparameter optimization strategy was added to select the best interpolator. Visual methods such as heatmaps were used to identify divergence points between measured and predicted surfaces, which in turn, aided in identifying the RBF interpolator with optimized parameters as the best solution for the problem.

Finally, the simulation of contact between teeth on ZaKo3D was conducted, and the recreated gear flank surfaces exhibited the desired characteristics of the original surfaces but failed to remove all distortions on the face. With the methods implemented to process gear topographies, simulations of gear behavior can now be performed with less user input error. This is because the dimensional and topographic data contained in GDE files are read into the WZL data structure automatically. Additionally, imperfect CMM measurements can be disregarded, as the interpolation routines can recreate the surface and then perform more realistic simulations.

1.4 DOCUMENT STRUCTURE

This work is organized as follows.

Chapter 2 presents the technical background on gears, gear measurement, gear/contact simulation and interpolation techniques.

Chapter 3 describes in greater detail the problem treated in this work along with the proposed solution.

Chapter 4 presents the implementation of the proposed solution and an analysis of the obtained results.

Chapter 5 presents the concluding remarks as well as suggestions for future works

2 THEORETICAL FOUNDATION

In this chapter we introduce the reader to the main concepts and theories necessary to understand in detail the problem, the experiments and the results obtained for this work. It starts with Section 2.1 presenting a general view of gears, their basic shapes and mechanical functioning. Section 2.2 discusses the process of gear measurements. Then, Section 2.3 details the XML language used for describing and storing gear data and the library used for reading and manipulating such data format. Section 2.4 presents the steps involved in gear measurements. Finally, Section 2.5 discuss the interpolation techniques relevant for this work.

2.1 GEAR

A gear is a toothed component that transfers rotation and forces from the driveshaft to the driven shaft. They usually operate in pairs, engaging one to transmit or receive motion from another toothed member. The transmission of torque is by the pressure of the teeth of the pinion on those of the wheel (KLOCKE; BRECHER, C., 2017). There is many appliances of gears in different areas, as machine tools, gears for control, vehicle gears, transportation gears, aerospace or marine gears (RADZEVICH, 2016).

The position of the two wheel axes in relation to each other results in different basic forms of gearbox, which are shown in Figure 1.

In agreement with Klocke and Christian Brecher (2017), the construction and operation of a gear can be differentiated according to three basic forms:

- Gearbox with parallel axes
- Gearbox with intersecting axes
- Gearbox with crossing axes

In the following sections, we discuss in further details parallel axes gears, considering the scope of this thesis. The reader can find more details on intersecting and crossing axes gears in Klocke and Christian Brecher (2017).

2.1.1 Parallel Axis Gears

The simplest types of gear are those that the shafts gear are parallel to each other. Parallel axis gears transmit power and motion with greater efficiency than any other type or form of gearing and they are generally easy to manufacture. The most common type of parallel shaft gear is the spur gear, but there are helical gears, herringbone gears, and others (RADZEVICH, 2016).

Figure 1 – Types of gears.

Source: Learn Engineering (2021).

2.1.1.1 Spur Involute Gears

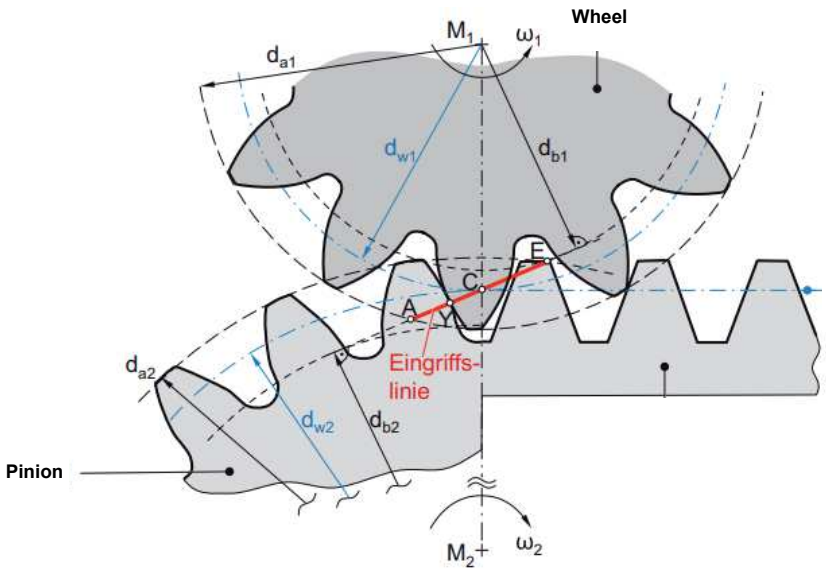
The involute gearing, first presented by Euler, has numerous advantages according to Litvin and Fuentes (2004):

- The tools for developing the involute gears can be built with high precision;
- It is insensitive to changes in the center distance and it is simple to adjust the settings of a tool (such as a machine or device) in order to use it with different types of gears and provide a different center distance;
- Nonstandard involute gears can be generated by using standardized tools applied for standard gears;
- It presents a good compromise between application behavior and manufacturability due to the linear line of action.

From Figure 2, it can be seen the involute curve arises by rolling a straight line on a determined circle, the so-called base circle $\mathrm{d_b}$. Also, it is possible to see the line of action, or *Eingriffslinie* in German, composed of all points of contact in a straight line. Any contact point Y runs between points A and E during meshing. Additionally, the normal of each point on the involute is tangential to the *base form* circle of the respective wheel.

The involute spur gears have straight teeth that are formed by the involute curve of a circle, i.e, the teeth are formed by this curve and are arranged in a straight line along the gear's circumference (KLOCKE; BRECHER, C., 2017). Furthermore, conforming Radzevich (2016), the most common pressure angles used for spur gears are 14.5, 20, and 25.

Figure 2 – Involute gear meshing line.



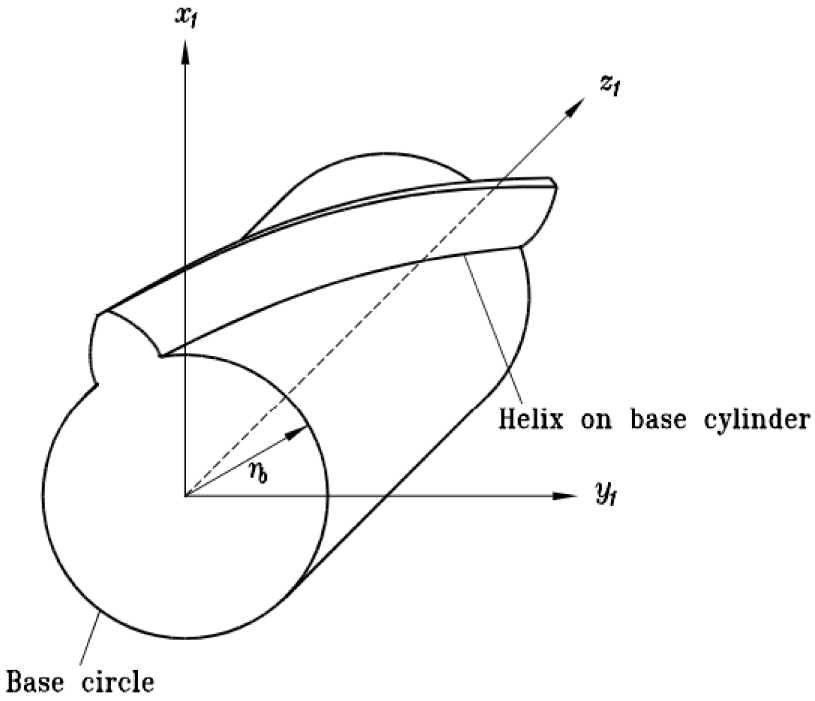
Source: Adapted from Klocke and Christian Brecher (2017).

2.1.1.2 Involute Helical Gears

Helical gears have a helix angle b that affects their parameters, and their teeth are incorporated into the base cylinder at an angle. (KLOCKE; BRECHER, C., 2017). Besides, the helical gears have a smoother and consequently, quiet action when compared to spur gear. This is due to the gradual entry of the teeth into the area of meshing. Still, in contrast, to spur gears, a pair of helical gears have a larger transmitted load, or the life of the gears may be greater for the same loading. Therefore, they are commonly used in applications that require high speeds, large power transmission, or noise reduction (JELASKA, 2012).

Helix angles from only a few degrees up to about 45 are practical. This abatement of the noise and the growth of load capacity is in general going when the helix angle increases from zero. Although, at angles, much above 15 to 20, as the transverse tooth thickness rapidly reduces, the tooth bending power normally starts to decline (RADZEVICH, 2016). In Figure 3 there is a representation of a base cylinder containing a tooth with a helix angle.

Figure 3 – Right-hand helical gear.



Source: Litvin and Fuentes (2004).

2.1.2 Geometric values of involute gearing

This subsection presents crucial names, parameters, and standardizations to understand the gearing process. First, the number of teeth in a gear is specified per z. By definition, the number of teeth is positive for external gears and negative for internal gears. This is related to the directions of rotation, so the direction where the power flows in gear trains can be considered with the correct sign. Additionally, conforming to a transmission ratio, the number of teeth of a gear pair is chosen. Then, the ratio (u) is defined for the number of teeth of the large gear and the number of teeth of the little one,

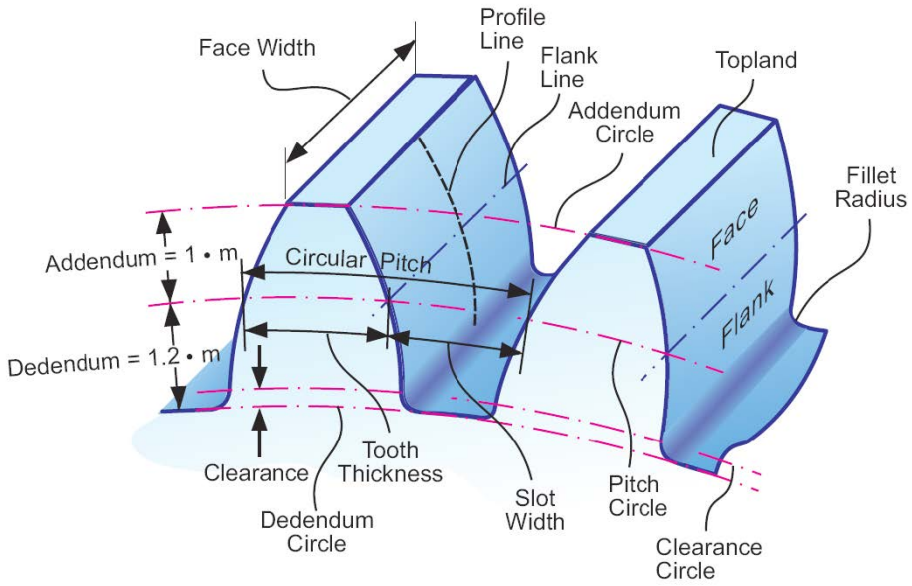
$$\mathrm{u} = \frac{\mathrm{z}_{\mathrm{larger}}}{\mathrm{z}_{\mathrm{little}}}. \tag{1}$$

Another element fundamental in the gear structure is the pitch. The pitch is defined as the distance between two consecutive equal flanks. In other words, the pitch *p* corresponds to the distance between two right flanks or two left flanks on the pitch circle, as seen Figure 4.

Figure 4 also brings some proportions present in the tooth and the location and name of each part, such as b, $\mathrm{p_{bt}}$ and $\mathrm{d_f}$. The profile lines are the lines that cross the tooth vertically, while the flank lines are the lines that cross the tooth longitudinally.

Figure 5, exhibits the interrelationships in the face section. The requirement for

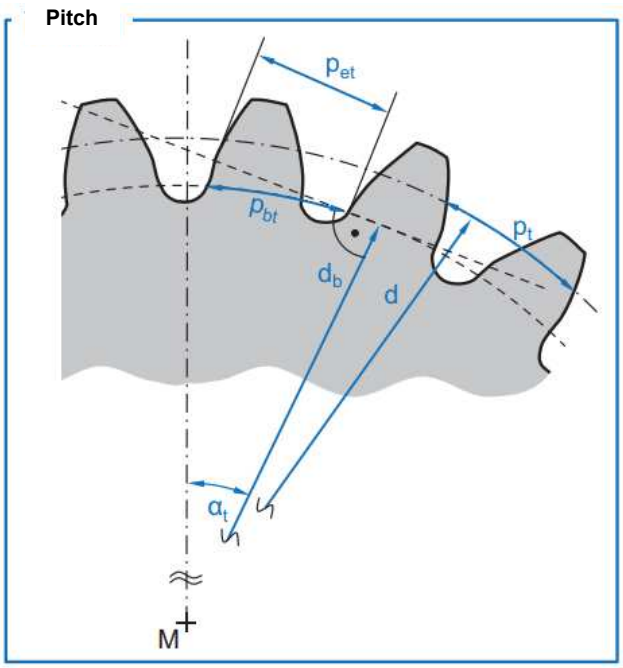
Figure 4 – Tooth geometric quantities.



Source: Stadtfeld (2015).

an equal tooth space pitch means that the gears in a pair must have the same module in order to be able to mesh with one another (KLOCKE; BRECHER, C., 2017).

Figure 5 – Pitch variables on the gear wheel.



Source: Klocke and Christian Brecher (2017).

For a gear pair to be able to roll, the pitches of the two gears must be identical: $p_{1t} = p_{2t} = p_t$. In the case of pitch deviations, for example, due to pitch errors in discontinuous manufacturing processes, the gears can run unbalanced or even jam. Therefore, the pitch face is described as:

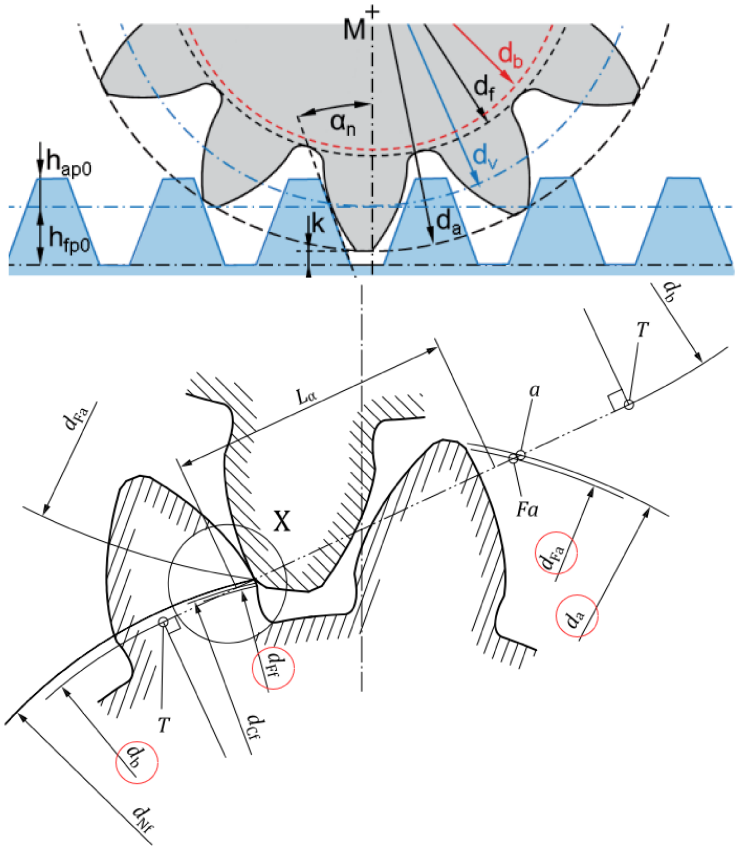
$$p_t = \frac{d \cdot \pi}{z} = \frac{m_n}{\cos \beta} \cdot \pi. \tag{2}$$

In complement to the pitch circle diameter, the pitch can be determined on any other diameter by specifying the reference diameter d_y. There are different pitch values for each reference diameter. The pitch base circle in the face section p_{bt} is the distance between two adjacent teeth on the base circle and corresponds to the face mesh pitch p_{et} along the meshing distance (KLOCKE; BRECHER, C., 2017). The p_{bt} is defined as:

$$p_{bt} = \frac{d_b \cdot \pi}{z} = p_t \cdot \cos \alpha_t. \tag{3}$$

The relationships between the basic sizes of the gearing and the various diameters are described using the illustration in Figure 6.

Figure 6 – Diameters and roll path length for an external gear pair.



Source: Adapted from Normung (2014).

The pitch circle diameter d is a reference value and cannot be measured. This refers to the angle at which the tangent at the involute intersects with the connecting

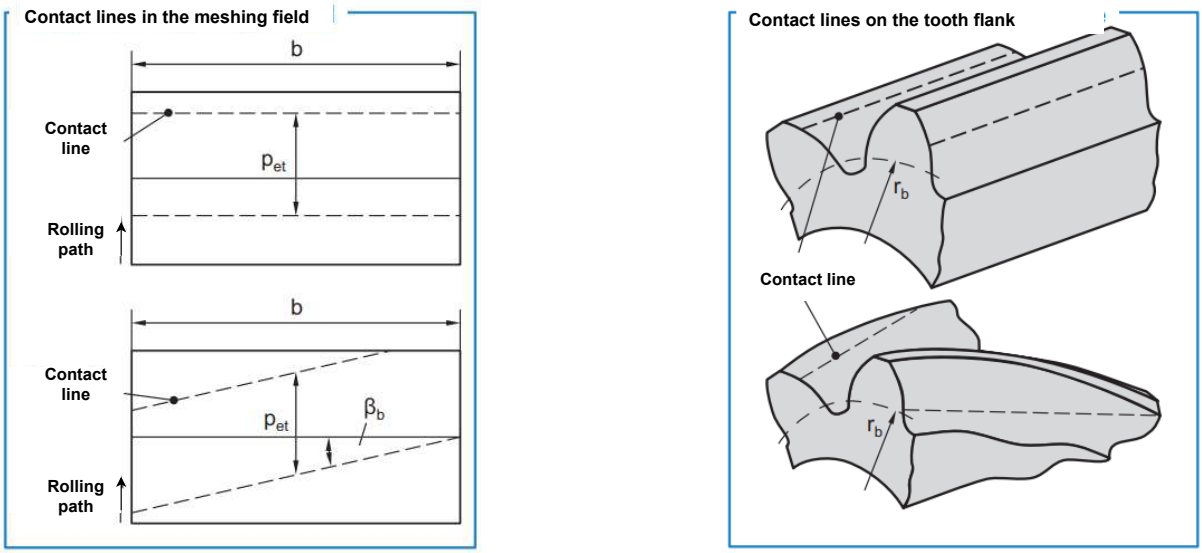
line to the center point, with the angle being determined by the profile angle α_{n}. The profile angle is derived from the reference profile and helps to define the shape of the gear tooth. It corresponds to the diameter of the gear where the tooth space curve and the tooth thickness line have the same magnitude.

Other important definitions are the $\mathrm{d_a}$, $\mathrm{d_b}$, $\mathrm{d_{Ff}}$ and $\mathrm{d_{Fa}}$ diameters, in highlight in Figure 6. There is a difference between the root diameter and the root form diameters. The root diameter of a gear refers to the diameter of the gear's root, which is the bottom-most point of the tooth's profile where it meets the gear's hub. The root form diameter, on the other hand, refers to the diameter of the reference circle used to generate the gear's tooth profile. The root form diameter is typically slightly larger than the root diameter and is used as a reference when designing and manufacturing gears. As a rule, the tip pitch circle $\mathrm{d_{Fa}}$ is identical to the tip diameter $\mathrm{d_a}$.

2.1.2.1 Contact conditions of cylindrical gears

The meshing of two gears can be described in the meshing field. The meshing field is the area defined by the tooth width and the pitch or roll angle between the pitch circle diameters of the two gears. Thus, the meshing field can be regarded as the spatial extent of the meshing line. For cylindrical spur gears, the meshing field is rectangular. In the meshing field, the course of the contact lines of several teeth can be traced over a pitch. In a spatial view, the contact point expands to the contact line, described in Figure 7.

Figure 7 – Contact lines for cylindrical gears in the meshing field and on the tooth flank.



Source: Adapted from Klocke and Christian Brecher (2017).

In the left part of Figure 7, the contact lines in the meshing field and the right

area are the same contact lines on the tooth flank. For spur gears, the contact lines run parallel to the gear axis. In the case of helical gearing (lower image area), the contact lines are inclined at the helix angle on the base circle diameter β_b,

$$\beta_b = \arcsin(\sin(\beta) \cdot \cos(\alpha_n)). \tag{4}$$

2.2 GEAR MEASUREMENT

This section introduces the gear measurement process, with a focus on tooth flank deviations and topographies, and the file formats used to store information about the measurements performed.

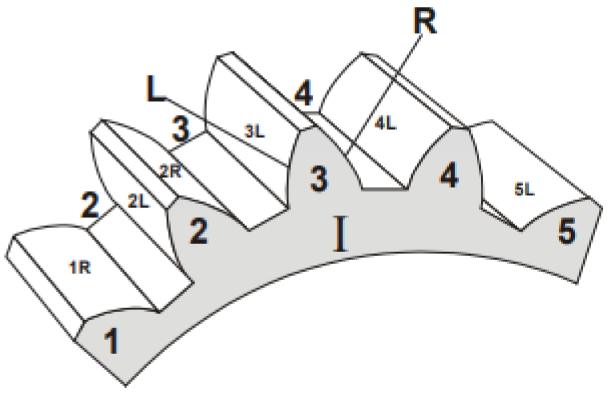
2.2.1 Measurement of tooth flank deviations

This following section [1] is mostly taken from VDI/VDE|2612 (2018) and aims to explain the profile and helix measurements to identify tooth flank deviations.

Some definitions must be considered in the gear measurement process for the exhibition of a measurement reading pattern. One of them is that cylindrical gears must be specified for one of the faces as the reference face. Thus, it is preferably marked with I according to VDI/VDE|2612 (2018).

For an observer looking at the I, in Figure 8, such that the tip of the tooth points upwards, the right flank (letter R) of the tooth is on the right, and the left flank of the tooth is on the left side (letter L).

Figure 8 – Characterization of right and left flanks for external gears.

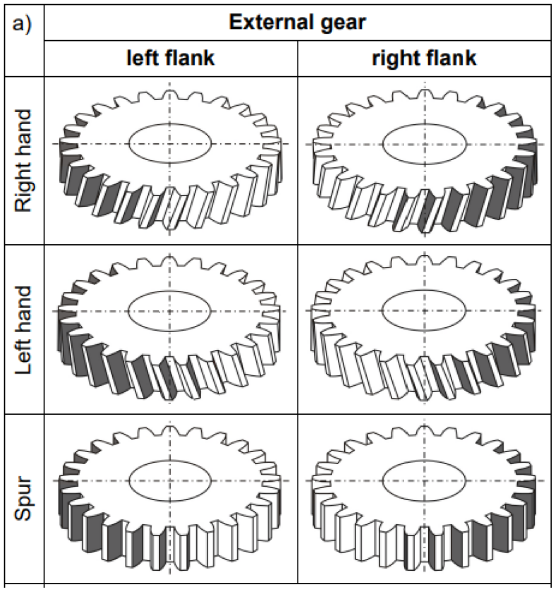


Source: VDI/VDE|2612 (2018).

[1] This section was partially documented in the technical report associated with the discipline Obligatory Internship (DAS5501) of the Control and Automation Engineering course at the Federal University of Santa Cataria (UFSC). The purpose of adding this information to the PFC was requested by the student's supervisor at the WZL institute in order to make the current document a self-contained text.

In the helical gears case, the definition of right and left flanks depend if the helix direction is right-hand, i.e., the helix corresponds to a right-hand screw, or if it is left-hand when the helix corresponds to a left-hand screw as seen in Figure 9. Furthermore, it is important to mention all specifications also apply to internal gears.

Figure 9 – Definition of helix direction and flank assignment.



Source: VDI/VDE|2612 (2018).

The measuring machine automatically converts the profile and flank lines during measurement of the gear geometry into flank test images, which can quantify the profile and flank line deviations according to German institute for Standardization (DIN) 21772 (BEUTH, 2023).

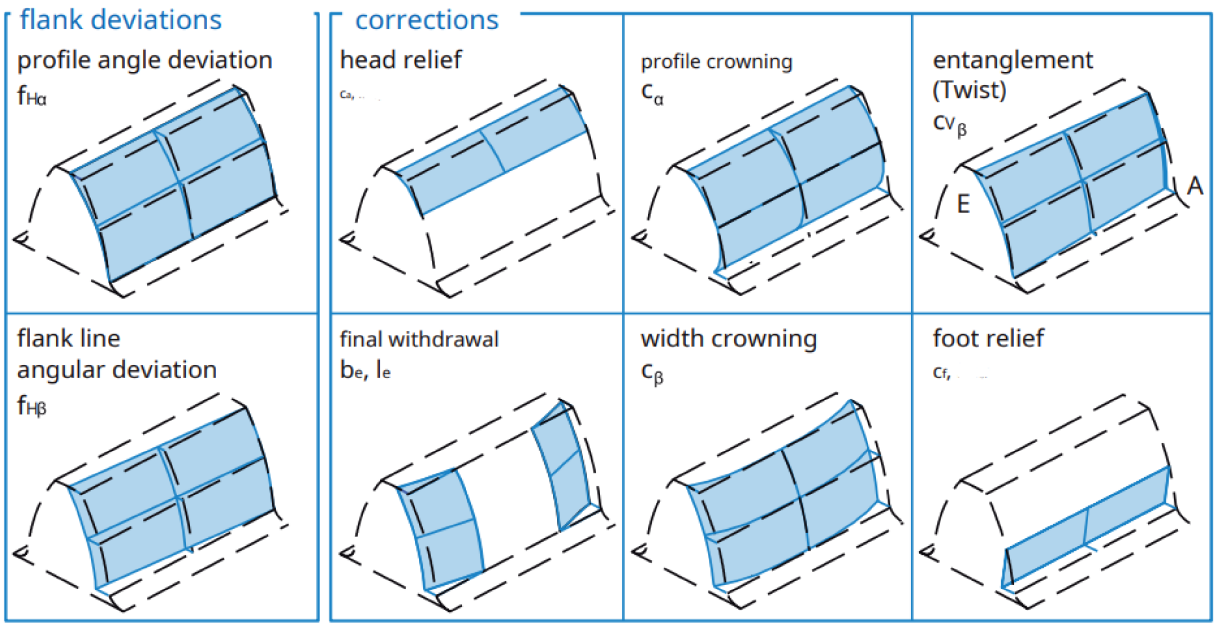
Figure 10 demonstrates the most typically used gear modifications, the deviations and corrections. These modifications must be taken into account when evaluating the geometry measurement. Each imperfection requires different compensation methods (KLOCKE; BRECHER, C., 2017).

In the case of crowning, a quadratic compensation function is normally used. Therefore, when the target geometry is programmed in the measuring program of the coordinate measuring center, the modifications are automatically borne in mind by the evaluation software.

The **profile** is measured in a transverse plane approximately in the middle of the face width (1), as shown in Figure 11. Particular attention might be paid to the measuring plane position in the case of modified tooth flanks. Extra planes can be selected to achieve a more comprehensive evaluation of the entire flank. For a twist measurement, additional measurements are carried near the datum face and the non-datum face (2).

The **helix** or **frank line measurement** is performed on the diameter of the V-

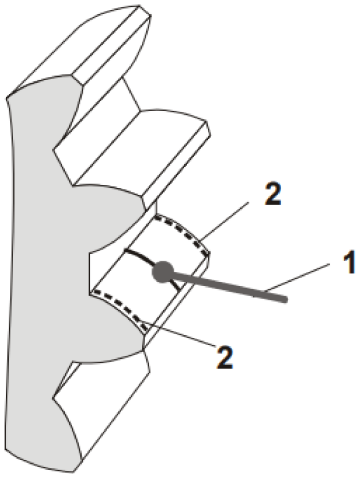
Figure 10 – Deviations and corrections.



Source: Klocke and Christian Brecher (2017).

.

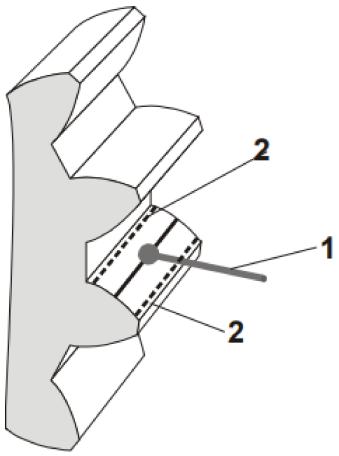
Figure 11 – Measurement of profile.



Source: VDI/VDE|2612 (2018).

cylinder (1), as shown in Figure 12. Again, special attention shall be paid to the measuring circle diameter position in case of modified tooth flanks. More measuring circle diameters can be selected to evaluate the entire flank more extensively. For a twist measurement, further measurements are taken near the active root diameter and the active tip diameter (2).

Figure 12 – Measurement of helix.



Source: VDI/VDE|2612 (2018)

.

2.2.2 Klingelnberg P 16

In this section [2], we present in details the Computer Numerical Control (CNC) machine for this work.

The fully automatic CNC-controlled P 16 precision measuring center (Figure 13), also known as P 16, is provided by the Klingelnberg company. It is a compact unit for the work-piece diameter range up to $160\mathrm{mm}$. The machine inspects cylindrical gears, pinion-type cutters, worms and wheels, hobs, and bevel gears (KLINGELNBERG, 2022b). Moreover, it performs many measurements, suitable for coordinate, shape, gear tooth, and roughness measurements. Besides, it has a scanning 3D tracer head with digital measurement acquisition in all coordinate directions, allowing accurate data acquisition of all gear geometry.

The machine also takes into account industry-compatible calibration to recognized standards (KLINGELNBERG, 2022b). One of the highlights is the test of all measurement centers with normals for tooth profile and trace and part-type normals of different types and sizes. In addition, the traceability of measurement results in internationally recognized standards Physikalisch-Technische Bundesanstalt (PTB).

[2] This section was documented in the technical report associated with the discipline Obligatory Internship (DAS5501) of the Control and Automation Engineering course at the UFSC. The purpose of adding this information to the PFC was requested by the student's supervisor at the WZL institute in order to make the current document a self-contained text.

Figure 13 – Machine P 16.

(a) Workspace P 16.

(b) 3D tracer head.

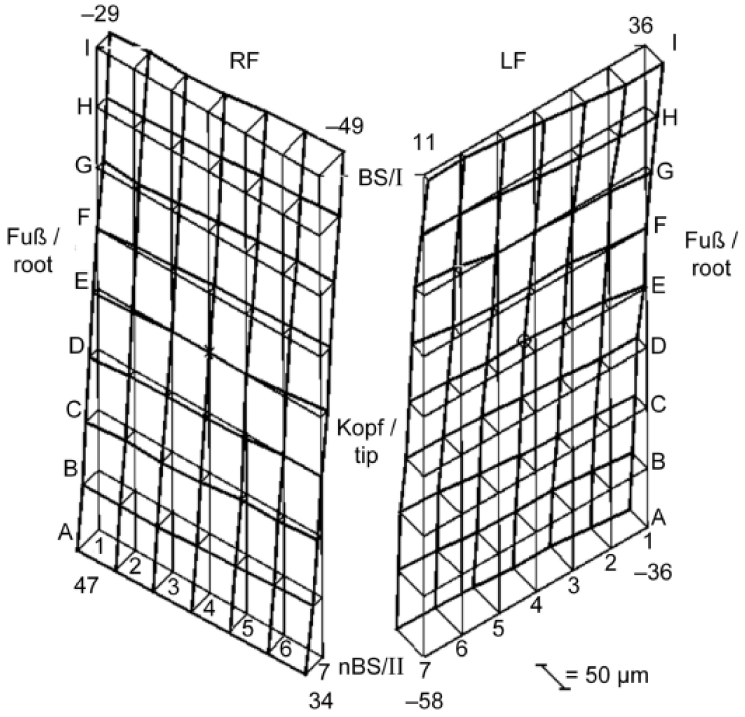
Source: Klingelnberg (2022b).

2.2.3 Flank Topography

In the end, the measurement result can be summarized in the flank topography, which is essential, for example, to determine topographical changes or the calculated simulation of the mashing process. Different representations, such as profile and helix measurements combined in a three-dimensional test pattern, are possible. In addition, a grid of individual probe points is drawn whose number depends on the desired resolution, covering the entire flank, as represented in Figure 14.

In this representation, neighboring points are connected by straight lines in the profile and helix directions. To indicate the grid points, it is recommended to use numbers for the helices and letters for the profile lines. In this way, a point cloud can be structured, and organized into profile lines containing the flank points.

Figure 14 – Topographical representation of the two flanks of a tooth by probing $79 = 63$ grid points each.



Source: VDI/VDE|2612 (2018).

2.2.4 File formats

Each CMM uses its own file format for storing measurement data, the most commonly used being MKA and GDE. In addition, the WZL standard file format for point cloud files is the PKT format. This file format provides a direct interface to the manufacturing simulation and TCA programs. In addition, PKT files are generated by the GearGenerator, SPARTApro and GearGrind3D programs. On the other hand, the MKA format contains information about the topography of the deviation-free tooth flank and the measured topography deviations on the side (BRECHER, P. D.-I. C.; PROF. DR.-ING. THOMAS BERGS, 2021).

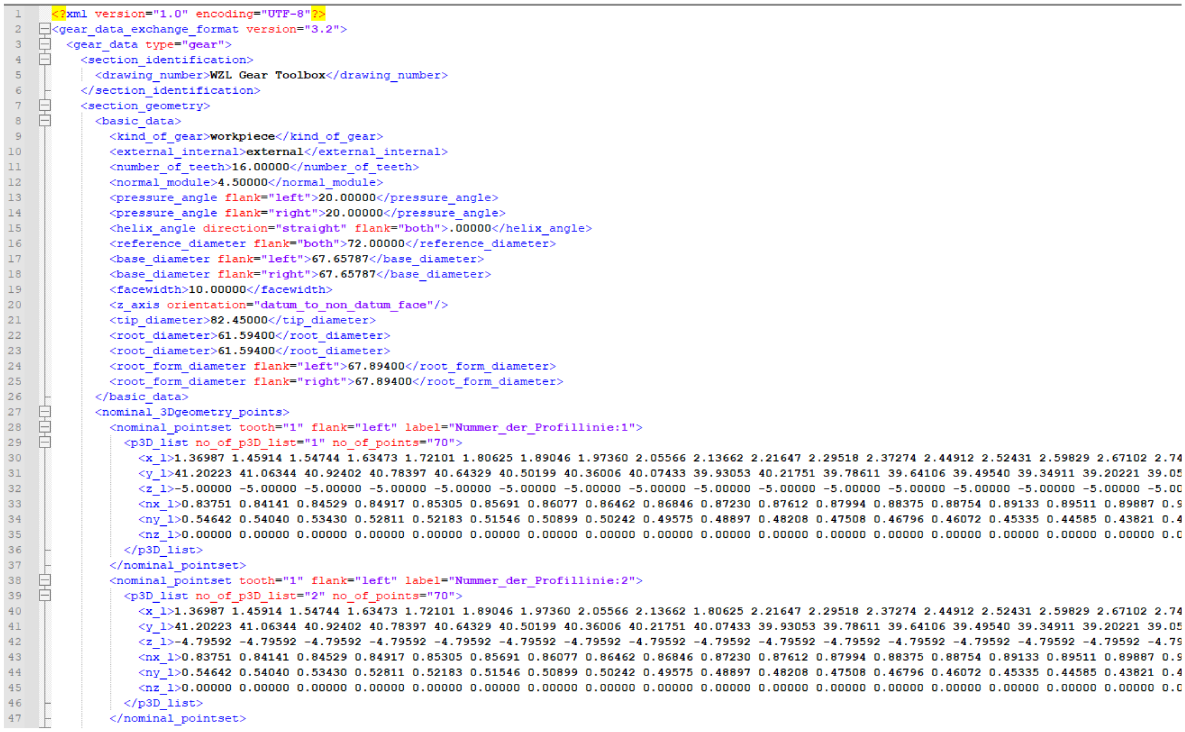
GDE is a file format for exchanging gear data between Computer-Aided Design (CAD), Computer-Aided Engineering (CAE) systems, and other software applications. It allows for transferring gear data such as gear geometry, kinematic properties, load cases, and other data. It is used to share data between design, analysis, and manufacturing systems, and it is a common format for Gear Industry. The GDE format was created by VDI/VDE Technical Committee 3.61 "Measurements of gears and gearboxes" in 2003; currently, it is a 3.3 version (STANDARDS, 2014).

The GDE standard format is defined in the form of a Document Type Definition (DTD) document (VDI, 2021). DTD is a set of grammatical rules for tags designated for

data exchange between programs and can be collected on the official website of VDI (STANDARDS, 2014). Thus, a complete GDE document must include an XML document containing data and a GDE format definition DTD document.

A representation of GDE is seen in Figure 15.

Figure 15 – Representation of a GDE file.



Source: Author (2022).

It is necessary to emphasize that the data in a GDE file is divided into sections consisting of at least one **Datastructure**: *gear_data*. Moreover, the *gear_data* will include the **section_identification** for management data for identification and the **section_geometry**, responsible for the gear geometry data. In addition, it can be used the **section_inspection**, for gear measurement, evaluation, and measurement results, and, in the end, the **section_user** contains user-specific information (STANDARDS, 2014).

2.3 LANGUAGE AND LIBRARIES

This section [3] covers the language and libraries required for gear data structuring and manipulation. It starts by describing the XML format and its general structure. This is

[3] This section was documented in the technical report associated with the discipline Obligatory Internship (DAS5501) of the Control and Automation Engineering course at the UFSC. The purpose of adding this information to the PFC was requested by the student's supervisor at the WZL institute in order to make the current document a self-contained text.

followed by the presentation of the `Libxml2` library which is used to read the data within the XML structure.

2.3.1 XML Format

Juliver (2021) describes that XML itself is a metalanguage to design markup languages, i.e. text language where semantics and structure are added to the content using extra *markup* information enclosed between angle brackets. Thus, XML markup is described as very verbose and has redundancy. That means that every end tag must be supplied; if not, the computer detects errors such as incorrect nesting. HyperText Markup Language (HTML) is the most well-known markup language (W3C, 2021).

The World Wide Web Consortium (W3C, 2021) elucidates that the XML is one of the most widely-used formats for sharing structured information today between:

- programs;
- people;
- computers and people;
- locally and across networks.

Each instance of an XML tag is called an element and the elements are organized in a hierarchy. "XML permits nesting of the data to any depth and the number of single elements per nesting level can be freely defined" (VALENTYN, 2018). The top element is called the `root` element and includes all other elements, which are called `child` elements. In Figure 16, the root is `studentslist`, while `student` is a child and parent, as well.

Furthermore, this work will call *parent* the element that also has *children*, such as `student` and its children `firstName`, `scores`, etc. Besides, using the same example, `id` is the attribute from `student` and `1` is the attribute content.

2.3.1.1 Libxml2

`Libxml2` is the XML C parser and toolkit developed for the Gnome project and it is a free software available under the Massachusetts Institute of Technology (MIT) License. In agreement with the Gnome community, though the library is written in C, a variety of language bindings make it available in other environments (WELLNHOFER, 2022). Some of key points of `Libxml2` are:

- Exports Push (progressive) and Pull (blocking) type parser interfaces for both XML and HTML;
- Does DTD validation at parse time, using a parsed document instance, or with an arbitrary DTD;

Moreover, one of the great highlights of using `Libxml2` is that it allows to navigate in a tree to print element names, i.e. it provides the ability to parse a file to a tree. Using

Figure 16 – Representation of a XML file.

```
<studentsList>
    <student id="1">
        <firstName>Greg</firstName>
        <lastName>Dean</lastName>
        <certificate>True</certificate>
        <scores>
            <module1>70</module1>
            <module12>80</module12>
            <module3>90</module3>
        </scores>
    </student>
    <student ind="2">
        <firstName>Wirt</firstName>
        <lastName>Wood</lastName>
        <certificate>True</certificate>
        <scores>
            <module1>80</module1>
            <module12>80.2</module12>
            <module3>80</module3>
        </scores>
    </student>
</studentsList>
```

Source: Valentyn (2018).

`xmlDocGetRootElement()` to get the root element, then walk through the document and print the entire element name in document order (WELLNHOFER, 2022).

2.4 SIMULATION

Numerical calculation methods allow the use of more complex models to analyse the running and load behaviour. Complex boundary conditions such as tooth corrections and deviations, the influence of different wheel body shapes or the compliance of the shaft-bearing systems can be taken into account (HEMMELMANN, 2007). This makes it possible, for example, to simulate the entire meshing process under load.

2.4.1 Tooth Contact Analysis

According to Litvin and Fuentes (2004), TCA programmes are aimed at solving the equations of the pinion and gear tooth surfaces, the crossing angle and the shortest distance between the axes of rotation. The pinion and gear tooth surfaces are in point contact. In other words, TCA software programs predict meshing and contact of gear tooth areas taking into account the localized bearing contact, which provides the surface contact point. Using the TCA algorithm the following objectives by (RADU et al., 2019) are:

- The transmission errors generated by the misalignment between gear axes;
- The contact route on pinion tooth surfaces;
- Bearing contact as the arrangement of momentary contact ellipses.

The TCA is based on the simulation of the tangency of tooth surfaces in a mesh. The determination of the instantaneous contact ellipse requires knowledge of the principal directions and curvatures of the tooth surfaces that are in tangency. A substantial simplification for the solution to this problem has been achieved due to the expression of principal curvatures and directions for the generated surface by the principal curvatures and directions of the generating tool (LITVIN; FUENTES, 2004).

2.4.1.1 Load-Free Tooth Contact Analysis

The contact geometry that occurs during meshing is defined by manufacturing error, assembly error and load-induced error. The load-induced deviations consist of deformation of the teeth, shafts, housings and bearings, while the manufacturing and assembly-induced deviations result from the load free deviations of these components. and assembly errors are the load-free errors of these components. Load-free tooth contact analysis is therefore often used to investigate the behaviour of mechanical components in situations where there are no external loads, such as when parts are first assembled or disassembled. There is also the tooth contact analysis under load, which is based on the contact distance deviation of the load-free tooth contact analysis, but will not be discussed in detail in this study (HEMMELMANN, 2007).

The results of the load-free tooth contact analysis are the load-free contact pattern, the load-free rotational error, the ease-off, the tooth clearance and the tip clearance. The Ease-Off method is a concept in gear design that refers to the accumulated contact distances of a gear pair, which is essential for ensuring error-free meshing. Essentially, the Ease-Off method provides a means of calculating the deviation in the meshing field, taking into account all necessary corrections. To elaborate, the Ease-Off method is particularly valuable in ensuring that gear pairs operate with the correct clearance and backlash. In addition, the method is useful for predicting the load distribution among gear teeth, which can aid in the selection of appropriate materials for gear construction (BRECHER, P. D.-I. C.; PROF. DR.-ING. THOMAS BERGS, 2021).

2.4.2 ZaKo3D

ZaKo3D is a Finite Element (FE) based tooth contact analysis program developed by WZL. It can investigate the tooth mesh of bevel gears, spur gears and face gears. In addition, simulation with multi-tooth measurement data, tooth contact performance analysis without load and tooth contact analysis under load are possible. The models can also be used to analyze the effect of geometric deviations on a full helical gear, calculate

flank pressure and many other functions (BRECHER, P. D.-I. C.; PROF. DR.-ING. THOMAS BERGS, 2021).

Moreover, the program receives files in PKT that contain coordinates and the normal to the plane at a given point. Because the assignment is done by checking the normal coordinates of a tooth at an average mesh position.

2.5 INTERPOLATION METHODS

Interpolation is a type of estimation, a method of constructing (finding) new data points based on the range of a discrete set of known data points. It is particularly useful when a set of measured data points does not follow a well-defined mathematical function and it is necessary to obtain an estimate of the value of the function at a point where the data is not available. (STEFFENSEN, 2006)

In the gear measurement process, interpolation is used to determine gear parameters with high accuracy. There are several types of interpolation methods used in gear measurement, such as spline interpolation, polynomial interpolation, and harmonic interpolation. The interpolation method used depends on the type of data being measured and the level of accuracy required.

Thus, considering the scope for this work, this section discuss in detail three interpolation methods applied: RBF, BSpline, and Griddata.

2.5.1 Radial Basis Function

The RBF is a method for interpolating multidimensional scattered data (WRIGHT; FORNBERG, 2003). The tool appears for the first time with Hardy as a new analytical method using multiquadratic equations of topography that is based on coordinate data (HARDY, 1971). The concept of RBF is equivalent to fitting a rubber membrane across the measured sample values, minimizing the total surface curvature (ARCGIS PRO 3.0, n.d.). Furthermore, how this rubber membrane will fit the values will be determined by the selected basis function which can be seen in more detail in section 2.5.4.

Accordingly to Skala and Cervenka (2020) meshless interpolations based on RBF offer several significant advantages, namely:

- RBF interpolation is usable generally for N-dimensional problems;
- RBF interpolation is smooth by a definition;
- It is suitable for the interpolation of scattered data, including the interpolation of time-dispersed data as well;
- RBF interpolation can be applied to interpolate scalar and vector fields and visualize both.

However, there are some limitations and weaknesses associated with RBF interpolation methods in general, such as memory usage and computational complexity. According to Skala and Cervenka (2020), in terms of memory it requires $\mathrm{O}(\mathrm{n}^2)$ memory space when n points are provided. In terms of computation complexity it may reach $\mathrm{O}(\mathrm{n}^3)$.

2.5.1.1 RBF Theory Background

The RBF is a Φ function whose value depends on the distance between the input x and a point that can be either the origin:

$$\Phi(\mathrm{x}) = \phi(\|\mathrm{x}\|), \tag{5}$$

or another fixed point x_0 called center,

$$\Phi(\mathrm{x}) = \phi(\|\mathrm{x} - \mathrm{x}_0\|), \tag{6}$$

where $\mathrm{x}_0 \in \mathrm{R}_\mathrm{d}$.

Unless otherwise indicated, we used the euclidean norm

$$\|\mathrm{x}\|_2 = \sqrt{\mathrm{x}_1^2 + ... + \mathrm{x}_\mathrm{d}^2}. \tag{7}$$

2.5.2 Interpolation

The $\Phi(\mathrm{x})$ scalar function is defined for an arbitrary sized variable x and represents a transformation $\mathbb{R}^\mathrm{n} \to \mathbb{R}$,

$$\Phi(\mathrm{x}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \gamma_\mathrm{i} \cdot \phi(\|\mathrm{x} - \mathrm{x}_0\|). \tag{8}$$

At a given point x the value of the RBF is obtained accumulating the interactions with all source points x_0 gained computing the radial distance between x and each $\mathrm{x}_{0\mathrm{i}}$ processed by the radial function Φ, then the transformation is multiplied by the weight γ_i (BIANCOLINI, 2017b).

2.5.3 Improving condition by adding a polynomial

The idea of add a polynomial term is widely used (SOMMER, n.d.), especially due to some stability issues, usually a polynomial $\mathrm{P}_\mathrm{k}(\mathrm{x})$ of a degree k is added (SKALA; CERVENKA, 2020). However, some conditions must be satisfied, for example, the degree of the polynomial must be chosen depending on the type of radial function adopted (BIANCOLINI, 2017a). A good reference in which a table of polynomial augmentations is collected is reported in Boyd and Gildersleeve (2011).

$$\Phi(x) = \sum_{i=1}^{N} \gamma_i \cdot \Phi(\|x - x_0\|) + P_k(x_i). \tag{9}$$

The radial basis fit exists if the weights γ and the coefficients of the polynomial can be found such that the desired function values g_s are obtained at source points, described as:

$$S(x_{0i}) = g_{0i}, 1 \leq i \leq N. \tag{10}$$

Additionally, an unique interpolant exists, or the linear system has a unique solution, if the basis function is a conditionally positive definite function (MONGILLO, 2011). This implicate, for example that it is possible use a linear polynomial when the basis function are conditionally positive definite of order $m \leq 2$.

2.5.4 Radial functions

The behavior of the function between points (interpolation) or outside the dataset (extrapolation) depends on the radial function used. According to Biancolini (2017a), there are two families of radial functions, the compact supported and the global supported. The first one is used when don't is expected interaction between distance points. While global support is used when all cloud points interact with themselves. Despite the problem being difficult, due to a densely populated matrix, it can reach high accuracies, which culminated in the choice of this category for this work. Table 1 can be visualized some common radial basis functions.

Table 1 – Common radial functions

Name	Definition
Thin plane spline	$r^2 \cdot \log(r)$
Cubic	r^3
Quintic	$-r^5$
Multiquadratic	$-\sqrt{(1 + r^2)}$
Inverse multiquadratic	$\frac{1}{\sqrt{(1+r^2)}}$
Gaussian	e^{-r^2}

Another way to improve the condition is using a variable shape parameter ϵ on a basis radial. The shape parameter that scales the input to the RBF, i.e., setting a specified shape parameter will consequently span across many nodes of one basis function and few in another basis. Therefore, each basis function may have a different shape parameter that needs local density measurement to usefully define that parameter (SOMMER, n.d.).

2.5.5 BSpline

B-splines are a mathematical tool for representing curves and surfaces in computer graphics, image processing and computer-aided design. B-splines are defined as piecewise polynomials of degree $\mathtt{k}$ that are continuously differentiable at all interior nodes and have a compact support (PIEGL; TILLER, 1995).

B-splines have many advantages over other curve representation methods. These include their ability to handle arbitrary shapes and their computational efficiency. They are also widely used for interpolation, approximation and smoothing of data (LAI; SCHUMAKER, 2007).

In summary, B-splines are a powerful tool for representing curves and surfaces, and their efficient computational algorithms and widespread use in various applications make them a valuable technique in computer science and related fields.

2.5.6 Griddata

In computational mathematics and computer graphics, the grid-data algorithm is a common interpolation technique. It takes a set of discrete data points on a regular or irregular grid and creates a continuous function.

The grid-data algorithm uses a variety of techniques to interpolate the data. These include linear interpolation, nearest neighbor interpolation and cubic interpolation. These techniques are used to estimate the values of data points at locations where data is not available. The resulting function can be used to generate smooth curves or surfaces through the original data points (SCIPY, 2021).

Several modifications and extensions of the griddata algorithm have been proposed to improve its accuracy and performance, including adaptive interpolation, nonparametric regression, and parallel computing. These techniques can be used to handle larger and more complex data sets and improve the accuracy of the interpolation results.

3 PROBLEM DESCRIPTION AND PROPOSED SOLUTION

This chapter introduces the problem for this work, including its context and significance. Furthermore, the proposed solution to this problem is presented.

3.1 CONTEXT AND PROBLEM DEFINITION

Standardisation enables efficient communication among machines, software, and users, minimizing errors. Nevertheless, the absence of a universal gear data transfer format remains a challenge. However, the GDE format is a notable exception in this domain. Being an open format, GDE is non-proprietary and can be supported by any software program. This feature enhances interoperability among various software applications, reducing the need for data conversion. Moreover, due to its efficiency and robustness, GDE permits the electronic transfer of toothing data from the Design department to Production and Quality Assurance (AG, 2019).

On the other hand, permitting the use of different file formats, which can store varying contents of gears, expands the possibility of utilizing more applications. In this context, the WZL Gear Technology Team is devising a code to execute conversions between different file types containing gear data stored in formats such as General Exchange Format (PKT), GDE, and MKA. Additionally, the institute has developed the WZL Gear Toolbox, a software tool that features several other tools, including gear hobbing simulation, gear geometry generation, and TCA. One of these tools, ZaKo3D, uses TCA to examine the tooth mesh by receiving point clouds and normal vectors from files with the PKT format. It is noteworthy that the PKT format is the institute's standard format for importing data into the program.

However, there are a number of other aspects to consider when moving from measurement to simulation, many of which are not directly related to gears and gearing, such as:

- Measurement instrument errors due to inaccuracies in the measurement instruments used to measure the gears, such as errors in measurement resolution or precision;
- Environmental conditions as ambient temperature, humidity, and dust, can affect the measurements and make them unreliable;
- Incomplete tooth measurements.

Although the accuracy of CMMs has been continuously optimised and improved over the past decades, the positional errors of 3D measurement points are inevitably affected by machine geometric errors (LIN, H. et al., 2020). Thus, the influence of CMM errors on 3D gear measurements is becoming an open and challenging issue. There is also the problem of incomplete tooth measurements. Firstly, it's generally only possible

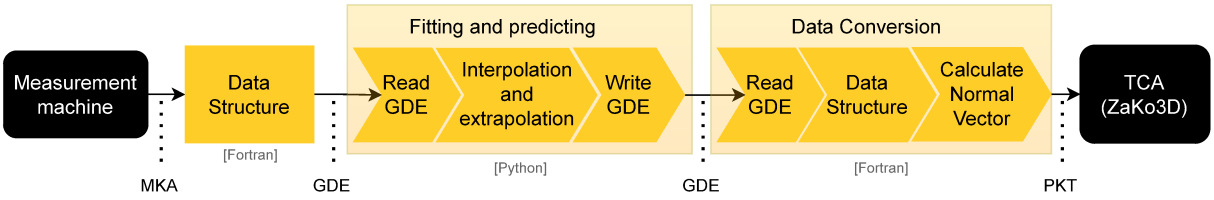
to measure part of a gear's root geometry on a CMM, so for most gear geometries the root can be completed beforehand in other software. Also, in most cases, measuring the entire flank would add unnecessary complexity and cost to the manufacturing and inspection process.

Therefore, with this in mind, another module needs to be added to the process: the processing of data gear topography. This new module aims at correcting possible outliers on the measured surfaces and recreating the entire tooth face and flank surface from a reduced set of points, of which would be a complete measurement.

Taking the aforementioned points, the aim of the work is to develop methods for automated processing of gear topographies. Methods that start from reading the topographic data to preparing the data for input into the simulation software. The proposed solution can be seen in Figure 17, which represents the global pipeline of the project.

First, the MKA files with the measured data from the CMM are converted into GDE files within the institute's code structure. Then the relevant data from the files in GDE format are reread and used for input into the interpolation and extrapolation procedures. After the application of the methods, the point cloud is rewritten into a GDE file with the basic information of the gear. Then, with the aid of the GDE reading routine [1], the data is reloaded into the Fortran code library, and the normal vectors of each point are calculated with the assistance of other preexisting codes in the environment. Finally, through PKT format file writing program, the file is prepared to be input into the ZaKo3D software.

Figure 17 – Overview of the proposed solution



[1] This section was documented in the Technical Report associated with the subject Obligatory Internship (DAS5501) of the Control and Automation Engineering course in the Control and Automation Engineering course at UFSC. However, improvements have been made to the file reading routine in GDE and the sorting routine, which are documented in this PFC. The purpose of adding this information to the PFC was requested by the student's supervisor at the WZL Institute to make the current document a stand-alone text.

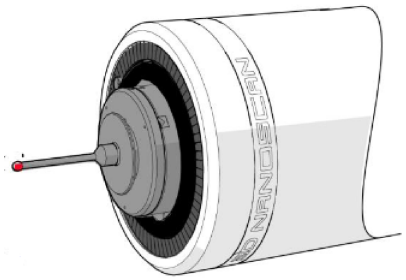
4 IMPLEMENTATION OF THE PROPOSED SOLUTION AND ANALYSIS OF THE OBTAINED RESULTS

In this chapter, we will go through the steps in detail to solve the problem at hand, i.e., beginning at the gear measurement process using the P16 machine, and the GDE reading and sorting routine. Then, we present the interpolation/extrapolation algorithm implemented. The chapter finishes with the results analysis [1].

4.1 MEASUREMENT ON P16

Before gear measurement, cleaning the gears with a lacquer solvent and cleaning thinner is crucial. Even elementary distortion can leave significant errors on the measurement. Thereafter, there is the automatic calibration process of the stylus/probe (Figure 18), for determining the exact dimensions of the stylus element used. In the menu bar of the graphical user interface, there is the calibrate probe option. Clicking on it displays the catalog of available probes. The active probe is highlighted and it is possible to change it by another probe. For this test a Taster - Y 3.0mm was used. Figure 19 shows the interface with some active probes.

Figure 18 – Probe element.



Source: Klingelnberg (2022a).

It takes around two minutes to calibrate. After that period, the probe elements can be used in the operation of measurement. Figure 20 shows the process of calibration in process.

Two different stylus were used in this measurement. With the first one, we aimed to find the space where the subsequent measurement was performed. In contrast, the second stylus was used to perform the process itself. The latter should be positioned

[1] The section 4.1 and section 4.3 were documented in the Technical Report associated with the subject Obligatory Internship (DAS5501) of the Control and Automation Engineering course in the Control and Automation Engineering course at UFSC. However, improvements have been made to the file reading routine in GDE and the sorting routine, which are documented in this PFC. The purpose of adding this information to the PFC was requested by the student's supervisor at the Institute WZL to make the current document a stand-alone text.

Figure 19 – Menu of calibrate probe option.

Source: Klingelnberg (2022a).

Figure 20 – Stylus used in the calibration process.

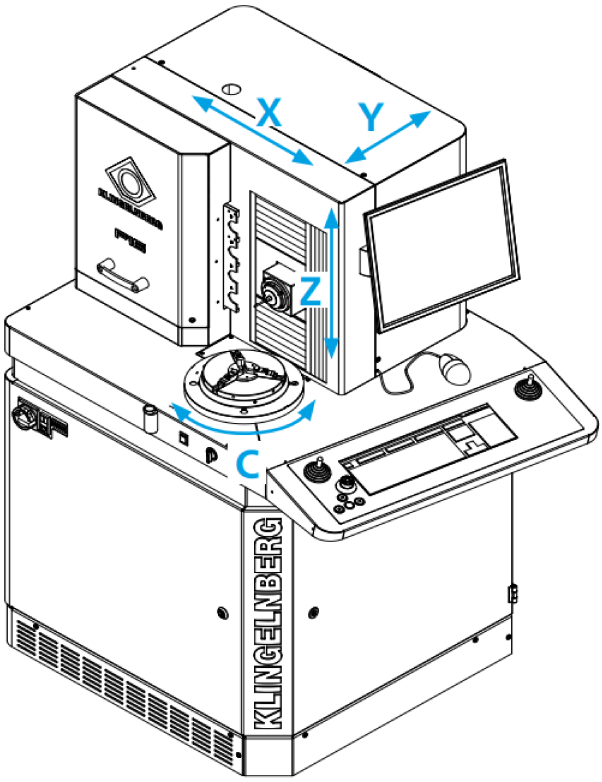
Source: COSTA (2022).

near the bottom of the chosen tooth (see Figure 21 - minimum Z-value) to start the process. In the Figure 21 $< \mathrm{C} >$ is the position of workpiece rotational axis; $< \mathrm{X} >$, $< \mathrm{Y} >$ and $< \mathrm{Z} >$ the positions of linear axis.

For this test, we used the planet gear, of which an example can be seen in Figure 22a and the its specifications in the table 2. Hence, two tests were performed with the same gear. The first considering a gear without significant variations in the surface and the second with a purposeful distortion of the surface. The reason for the second test was to investigate variations that occurred in the pitch measurement.

Once the gear is positioned in the workpiece (Figure 22b), it was selected a type of measurement and a short setup was developed, specifying the number of teeth, number of flank lines, and number of profile lines to be measured per flank. Then the gear machine set a fully automated teeth topography measurement.

Figure 21 – Axis arrangement on a P machine.



Source: Klingelnberg (2022b).

Table 2 – Specifications for Planet Gear tests.

Parameter	Value
No. of teeth	28
Module m	3mm
Pressure angle	20°
Helix angle	0°
Base helix angle angle	0°
Tip Diameter	90mm
Face width	45mm

4.1.1 Output and data import

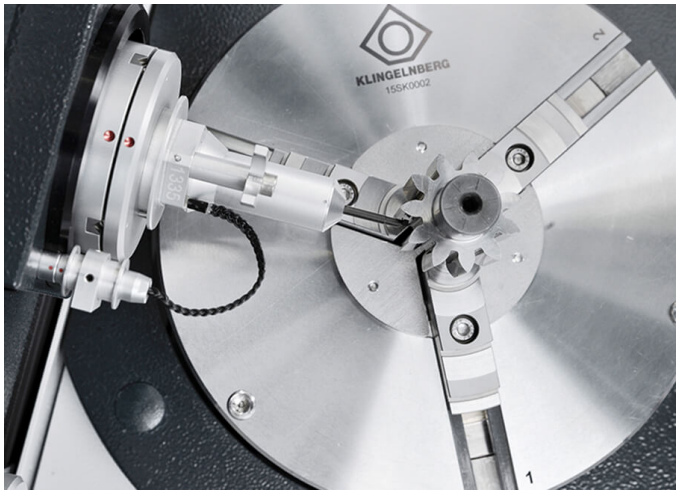
To better comprehend the whole process, during the measurement, there is a display of the probe system and temperature. In this display, we can visualize the change of C, X, Y and Z. If no error, the measurement results can be saved in PDF (Annex A) and MKA format.

Since the output of the P16 machine has been processed, the MKA extension can be imported into a data structure written in Fortran. Thus, the file can be converted to the PKT extension through the overlaying of ideal topography with the deviations of

Figure 22 – Elements.

(a) Planet gear.

(b) Workpiece fixture via three-jaw chuck (motorized).



Source: Adapted from Klingelnberg (2022a).

actual topography. A structured Fortran code is devolved around making conversions between formats, MKA $\rightarrow$ PKT $\rightarrow$ GDE. And the inverse approach, converting GDE to PKT or MKA files, starts with reading the GDE file.

4.2 PROGRAMMING LANGUAGES

The choice of programming languages used for each part of the project was thinking for attempting the system requirements. Hence, Fortran was used to implement the GDE read routine. Also, conversions between file formats and the creation of coordinate normals point happened in Fortran. The reason for still using Fortran is the comprehensive code repository of the WZL Gear Department. In addition, Fortran has High-Performance Computing (HPC), of which large-scale numerical simulation is a subset and easily handled with arrays (ELTON, n.d.). On the other hand, Fortran had no library for a grip with XML structure. Therefore, the best option was the C language, given its interoperability with Fortran and the Libxml2 toolkit (BEHNEL et al., 2005).

To create the interpolation algorithm Python was used, since it has an extensive assortment of libraries and one of them is Scipy, a collection of powerful mathematical algorithms. There are several general facilities available in SciPy for interpolation and smoothing of data in one, two, and higher dimensions for structured and unstructured data as well (SCIPY, 2023).

4.3 GDE ROUTINE

The routine for handling GDE files is divided into two main parts: Fortran and C implementations. In addition, the routine has subroutines for organising the data read

from the file to ensure that the point cloud is arranged in profile lines, see section 2.2.3. The routine was then included in one of the programs available in the internal version of the WZL Gear Tool Box, the Geometric Converter, which can quickly convert gear data file formats.

4.3.1 Implementation of GDE reading routine

The routine for handling GDE files is divided into two main parts: The Fortran and C implementations and the interoperability between the two languages. The core principle of interoperability is that something should work the same way in both languages, in this case Fortran and C (INTEL, 2022). There is a standard practice for generating `procedure` declarations and `derived-type` text and global variables that are interoperable with C. The use of `BIND (C)`, as shown in Figure 23, optionally combined with the specification of a binding name, is responsible for informing the compiler that the symbol is compatible with C. Thus, these variables must be declared in the declaration part of a module *module*.

Figure 23 – Example of code for interoperability C - Fortran.

```
module ftn_2_C
   interface
     subroutine C_Library(sendA, sendB, recv) bind(c, name='C_Library')
        use, intrinsic :: ISO_C_BINDING
        implicit none
        type (C_PTR), value :: sendA
        integer (C_INT), value :: sendB
        type (C_PTR), value :: recv
     end subroutine
   end interface
end module ftn_2_C
```

Source: Author (2022).

Pointers in C are represented by a specific derived type: `type(c_ptr)`, thus one needs to use intrinsic conversion procedures to convert from or to C pointers. Pointers and Targets are innate correlates. As is seen in Figure 24, the second part of the process must be specified, which is the target the pointer is pointing to. Besides, it is necessary to know that Pointers are strongly typed in the sense that a pointer to a REAL scalar target may not point to any other data type (`INTEGER`, `LOGICAL`, e.t.c.), nor may it point to an array. At last, Pointers do not need to be dereferenced because the pointer is just a reference to the memory space that the variable uses.

4.3.2 GDE reading routine

The GDE reading routine allows loading a GDE file and extracting the required information. For the implementation of this process, routine modules were created in Fortran **Read_interface.f90** and **Read_C_caller.f90** (Figure 25) that are used to make the

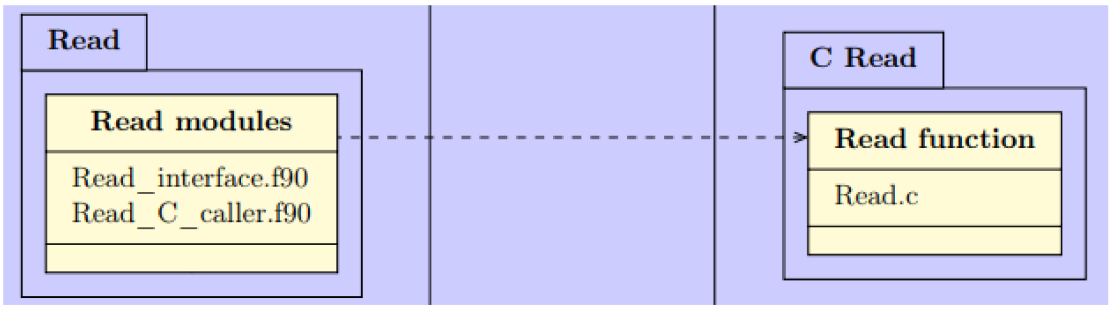
Figure 24 – Example of code for interoperability C - Fortran.

```
use, intrinsic :: ISO_C_BINDING, only: C_INT, C_FLOAT, C_LOC
 use ftn_2_C
 implicit none
 ...
 real (C_FLOAT), target :: send(100)
 integer (C_INT) :: sendB
 integer (C_INT), ALLOCATABLE, target :: recv(:)
 ...
 ALLOCATE( recv(100) )
 ...
 call C_Library(C_LOC(send), sendB, C_LOC(recv))
 ...
```

Source: Author (2022).

Fortran and C features equivalent. The **Read_C** is the representation of where the code written entirely in C is located. In addition, the C routine expects a procedure *pointer* argument and pass it a Fortran procedure, as pointers.

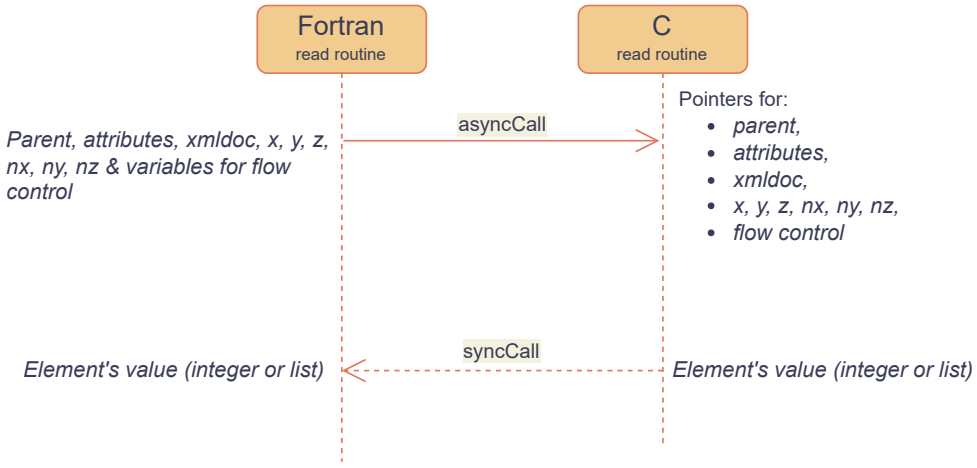
Figure 25 – Schematic of Fortran and C routines.



Source: Adapted from COSTA (2022).

The flow exposed in Figure 26 demonstrates the variable exchange relationship between the two languages. The information of which `parent`, which *attributes* are required and the XML file (*xmldoc*) must always be specified. The C code can then return a pointer to an integer value or to a list of coordinates, in the case of `x, y, z, nx, ny,` nz.

Figure 26 – Simplified flow exchange C - Fortran.

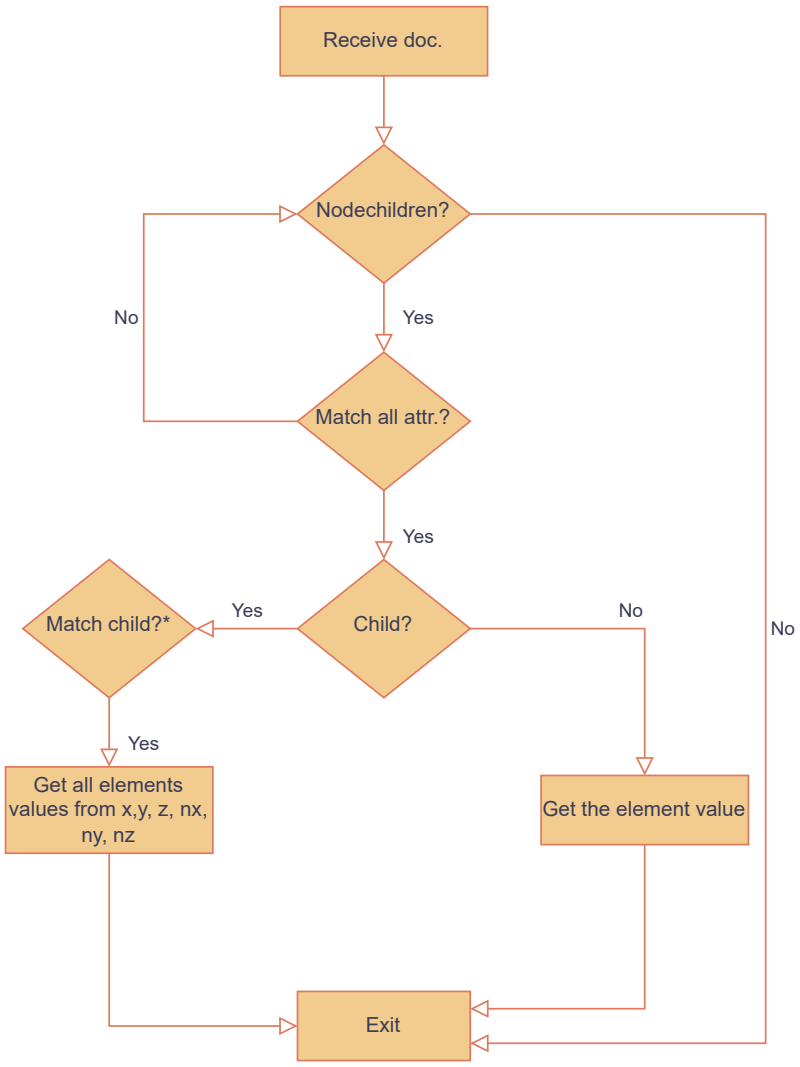


Source: Author (2022).

The flowchart in Figure 27, shows the process of a GDE reading routine in the language C part. The `nodechildren` is node/element with children. In this context, it is understood the function has several input arguments:

- *parent*: the parent XML element to start the search from;
- *child*: the element that the function is looking for;
- *second_property*: the secondary property of the element;
- *attribute1*, *attributecontent1*, *attribute2*, *attributecontent2*, *attribute3* and *attributecontent3*: the attribute name and content that the function is looking for. There are up to three possible attribute name and content pairs that the function can search for;
- *mydocpointer*: a pointer to the XML document;
- *value_elem*: a buffer for the element value;
- *value_nodeX*, *value_nodeY*, *value_nodeZ*, *value_nodeNX*, *value_nodeNY* and *value_nodeNZ*: buffers for the values of the element's child nodes, if any;
- *n*: a pointer to the number of nominal point sets in the current iteration (current profile line);
- *cont*: a pointer to the number of nominal point set elements in the right or left flank (number of profile lines per flank).

Figure 27 – Simplified flowchart reading GDE routine.



Source: Author (2022).

The function starts by initializing some variables and an array that contains the attribute name and content. It then converts the parent string to a C-style string and creates an XPath expression that looks for all the elements with the name parent.

Next, the function uses **libxml2** to evaluate the XPath expression and retrieve the resulting nodeset, which contains all the elements that match the XPath expression. It then loops over the nodeset to find the *child* element, which is the element that the function is looking for.

If the *child* element is found, the function then loops over the *attribute_array* to find if the attribute matches with the current attribute name stored on XML. If the attribute is found, the function then checks if the content of the attribute matches one the attribute content strings.

If both, the attribute and its content match, the function extracts the values of the

element and its child nodes, if any, and stores them in the respective output buffers. The function also updates the *n* and *cont* variables based on the number of nominal point sets and nominal point set elements found. When the element has more then one element, each coordinate *value_nodeX*, *value_nodeY*, *value_nodeZ* and its corresponding normal lines,*value_nodeNX*, *value_nodeNY*, *and value_nodeNZ* are returned. It is worth pointing out that all of them are element siblings.

Since the GDE file is easily editable by users, inconsistencies can occur in the data. Also, the input to the simulation software is by profile lines. In view of an appropriate input into the software, the data sorting algorithm was created.

4.4 SORTING ROUTINE

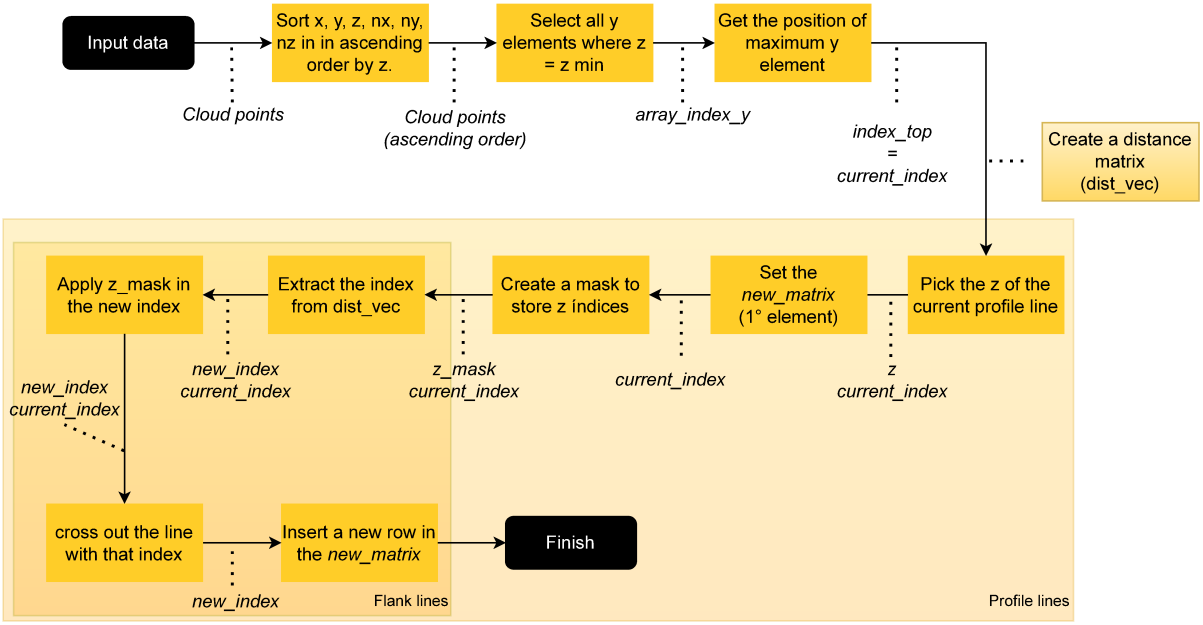
The objective of a sorting routine was to guarantee that the data read from GDE routine are sorted by profile lines to be inserted on ZaKo3D software. This algorithm is a subroutine called `sort` that takes six input arrays `(columnx, columny, columnz, columnNx, columnNy, columnNz)` and outputs a sorted matrix `new_matrix`. The algorithm is intended for use with straight and helical gears, and works by taking a set of points in 3D space that define the geometry of a gear and ordering them along the profile and flank lines of a gear.

For the algorithm to work properly, two requirements must be met:

- Profile lines with the same number of flank lines;
- The same profile line must have the same `z` values.

The algorithm first sorts the input points according to their `x, y,` and `z` coordinates using a subroutine called `ascending_sort`. The sorted points are then stored in a matrix A with six columns, where each row contains the `x, y, z, Nx, Ny, and Nz` values for a single point. The algorithm flowchart is shown in Figure 28.

Figure 28 – Sorting algorithm flowchart



Source: Author (2023).

Next, the algorithm finds the topmost point on the gear by searching for the point with the minimum z coordinate. If multiple points have the same minimum z coordinate, the algorithm selects the one with the maximum y coordinate. A loop is then used to determine the y-coordinates with the lowest z-coordinate, and these are stored in `array_index_y`. The index of the largest y-coordinate in `array_index_y`is then used to determine the starting index for sorting the flanks of the gear teeth.

The algorithm then creates a distance matrix that stores the Euclidean distance 11 between every pair of points in the input matrix. The distance matrix is used to find the next point in each profile line by selecting the point that is closest to the previous point. This process is repeated for each profile line, with the algorithm searching only among the points that have the same z coordinate as the previous point on the profile line.

The Euclidean distance can be defined as (O'NEILL, 2006):

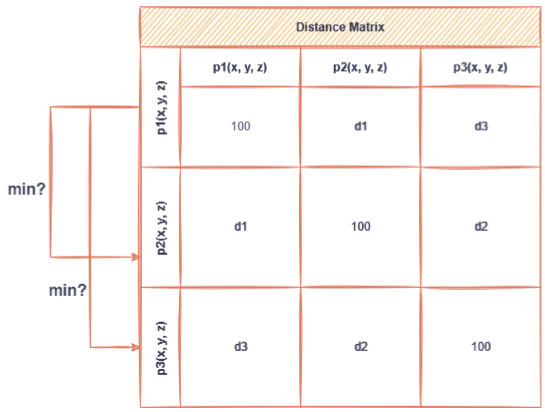
$$||\mathrm{a}-\mathrm{b}||^2 = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{a_i}-\mathrm{b_i})^2 \tag{11}$$

which *n* is the number of measured points, a and b are the corresponding points the three coordinates $(\mathrm{x}, \mathrm{y}, \mathrm{z})$. Figure 29a and 29b depicts what happens after the calculation of the distances between each point. The second point on the distance matrix will be the shortest distance between this point and the top point, the third point the shortest distance between this point and the second point, and so on. All points on the same profile line are found based on the smallest distance from the previous point. After this

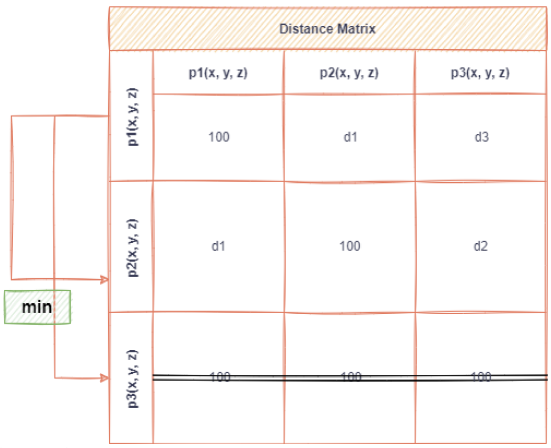
process, the minimum point line is cross out, replacing whole row for a number much bigger than a normal distance.

Figure 29 – Example of the process to find the shortest distance.

(a) step A.



(b) step B.



Source: Author (2023).

In sequence, the algorithm creates an array of row indices (`z_mask`) where the value in the third column of `matrix_of_cloud_points` is equal to the minimum value in the third column of `matrix_of_cloud_points`. The resulting array of row indices (`z_mask`) can then be used to extract the corresponding rows from the `dist_vec`. Compute the index of the row in `matrix_of_cloud_points` that has the minimum value in the column of `dist_vec` corresponding to the set of rows identified by `z_mask` and `current_index`.

Afterwards, convert the index of the minimum value from the local index of `dist_vec` to the global index of `matrix_of_cloud_points` by looking up the corresponding row in `z_mask`. Subsequently, set all elements in the row of `dist_vec` corresponding to the current row (`current_index`) to 100. This is done to avoid selecting the same row twice.

The output of the algorithm is the sorted matrix `new_matrix`, which contains the same points as the input array but ordered along the profile and flank lines of the gear. The `new_matrix` is sorted in such a way that the first point corresponds to the topmost point on the gear, and subsequent points are ordered along the profile and flank lines in a way that ensures that adjacent points are as close to each other as possible.

4.4.1 Results

The process of testing the assertiveness of the algorithm consisted of differentiating between the GDE file entered to read and rewriting the file in GDE format containing the extracted data. A Fortran routine was called to check the validity between the desired GDE file and the calculated one, with a tolerance of 0.0001. Both are almost identical, except for rounding differences after 3 decimal places, which can already be ignored.

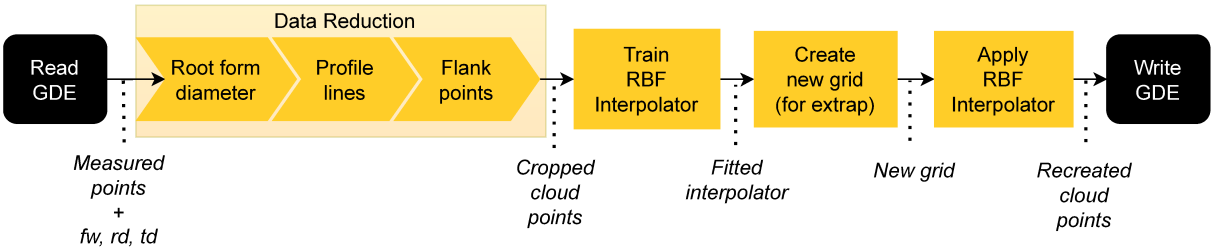
4.5 INTERPOLATION AND EXTRAPOLATION ALGORITHM

In view of the problem presented, a structured algorithm was created to facilitate interpolation and extrapolation. The schematic of the algorithm is shown in Figure 30. First, the point cloud is read from a GDE file. In addition, other essential elements read from the GDE files are Face Width (FW), Root Form Diameter (RD) and Tip Diameter (TP). For the initial tests, two different GDE files were used, one for a spur gear and one for a helical gear. The specification of both can be seen in Table 3 and the 3/28 tooth surface in Annex A. The measurements were complete, but the flank surface had some imperfections. For this reason, a pre-processing step was performed. The concept was to construct a reduced surface for testing before applying a real partly measured surface.

Table 3 – Gear basic data

Gear type	**Planet gear**	**Helical gear**
No. of teeth	28	29
Pressure angle	20°	20°
Helix angle	0	-12°
Module	3mm	3mm
Base helix angle	0	-11.267°
Tip diameter	90 mm	95 mm
Root Form Diameter	79.97 mm	84.96 mm
Face width	45mm	45mm

Figure 30 – Extrapolation Algorithm flowchart.

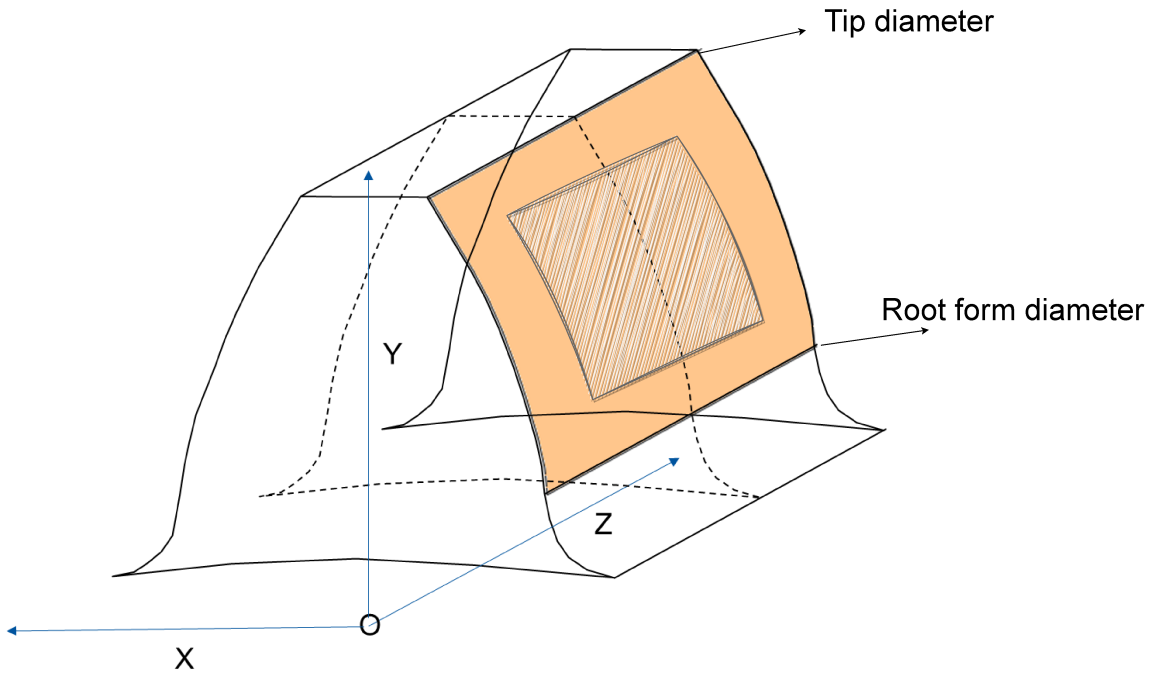


Source: Author (2023).

The first phase was to remove all points below the `root_form_diameter`. Notably, the points below the line of `root_form_diameter` are not measured but they are generated within the WZL code structure. The second and third phase was to reduce the flank surface in the longitudinal direction (reducing the number of profile lines) and in the vertical direction (number of flank points per profile line). Figure 31 shows the area used as input to the interpolator for extrapolating performance, the smaller hatched area. The larger painted area between the tip and root diameters was used for further testing

of the optimisation parameters in section 4.5.1.2. In addition, the limits of flank point reduction and the number of profile lines were determined empirically in an attempt to represent a partially measured surface. Other essential elements read from the GDE files are b, $\mathrm{d_{Ff}}$ and $\mathrm{d_a}$.

Figure 31 – Representation of the tooth in the coordinate system.



Source: Author (2023).

4.5.1 Interpolation

For the selection of the interpolators, the area highlighted in Figure 31, obtained with the topological data of a spur gear, was used.

4.5.1.1 Model selection

The first step of the project was to find the interpolator method that could produce the best interpolations. There is a wide range of interpolators with different mathematical constructions, implementation forms and computational uses. For this study, the `Scipy.interpolate` library was used. This was chosen because the Scipy interpolation subpackage provides several options for solving this problem. These include classes for multivariate interpolation and 2D splines for unstructured data. Thus, within the multivariate interpolation class, alternatives such as `RBFinterpolator` and `Griddata` were tested. For the second class of interpolators, 2D splines for unstructured data, the

Scipy `BSpline` library was used. Other interpolator structures were also considered, but due to their minimal prominence in articles or because they are only branches of the same interpolator family studied in this paper, they were removed. These include the `RegularGridInterpolator` and the `CloughTocher2DInterpolator` (YANG, R.; XING, 2021).

At this initial stage, the aim was to understand how well each library would interpolate the input coordinates. One way of doing this was to analyse the residuals, as shown in Figure 32, with the representation of a heat map. The color-coded values represent the discrepancy in millimeters between the measured x-values and the corresponding predicted x-values obtained from each of the interpolators, indicating the magnitude of the error in the x-coordinate. On the abscissa axis is the z, which indicates the tooth width, and on the ordinate axis is the y, which indicates its height.

In 32a is the interpolation using `scipy.interpolate.bisplrep`, a function from the SciPy library. It computes a B-spline representation of a given set of data points based on the SURFIT routine of FITPACK. In 32b is the representation using the RBF interpolator (`scipy.interpolate.RBFInterpolator`) and in 32c is the heat map obtained using Griddata (`scipy.interpolate.griddata`). It is important to note that the linear transformation was used for this first comparison because it is less complex and is available for all the interpolators tested. Furthermore, the interpolation is in `x` direction (the coordinate system can be seen in Figure 31).

Table 4 gives a summary of the other parameters used.

Table 4 – Model selection

Name	Kernel	Smoothing	Epsilon
Bivariate Spline (bisplrep)	Bilinear	0	Not applicable
RBF	Linear	0	1
Gridata	Linear	Not applicable	Not applicable

In the case of the B-spline, the default parameter, a bicubic spline (`kx=ky=1`), indicates that the spline orders in the x- and y-directions can be specified by means of the optional parameters `kx` and `ky`. Furthermore, no smoothing was desired ($\mathrm{s} = 0$), in contrast to the griddata where there are no smoothing parameters.

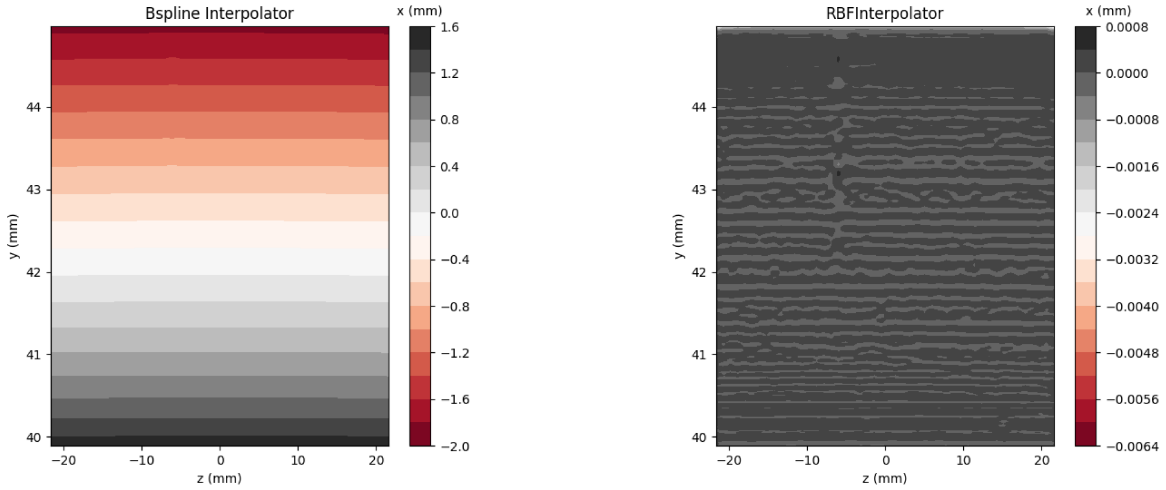
4.5.1.2 Choosing parameters

The `RBFinterpolator` algorithm has some parameters that can be associated with improved performance, depending on the application. Therefore, the smoothing, epsilon, and kernel parameters were changed for this study, as shown in Table 5.

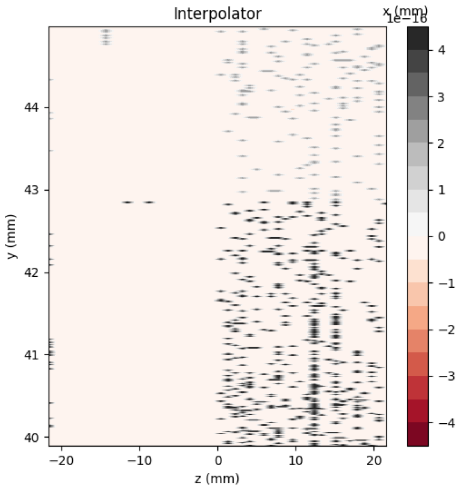
The choice of epsilon values was based on the author's study. In general, larger errors are associated with orders less than 10^{-1} and greater than 10^{1}; however, the smallest error is not found at $\epsilon = 0$ (MONGILLO, 2011). On the other hand, the smoothing

Figure 32 – Interpolation results for different interpolators

(a) BSpline (b) RBFInterpolator



(c) Griddata



Source: Author (2023)

Table 5 – Parameter selection

Paramater	Values
Smoothing	0.1, 0.5, 1.0, 2.0, 5.0
Kernel	Thin plate spline, cubic, quintic, multiquadric, inverse multiquadric, inverse quadratic, gaussian
Epsilon	0.1, 0.5, 1.0, 2.0, 5.0

parameter was selected considering the smoothing effect. The interpolant fits the data perfectly when the smoothing is set to 0, while a more considerable value, will give a smoothing effect.

Therefore, the first step was to evaluate the best interpolator combinations for

the problem using performance metrics. The metrics used were Root Mean Square Error (RMSE) and Maximum Absolute Error (MaxAE). RMSE was chosen because it provides a measure of the average error in terms of the unit of measurement of the variable of interest. While MaxAE gives an idea of the maximum error size to identify cases where the model fails to predict extreme values. The latter is interesting as the vast majority of the data is predicted well, but points close to the surface boundary are mispredicted. Table 6 shows the top 5 models based on the values obtained for both metrics.

Table 6 – Performance Metrics - RBFInterpolator

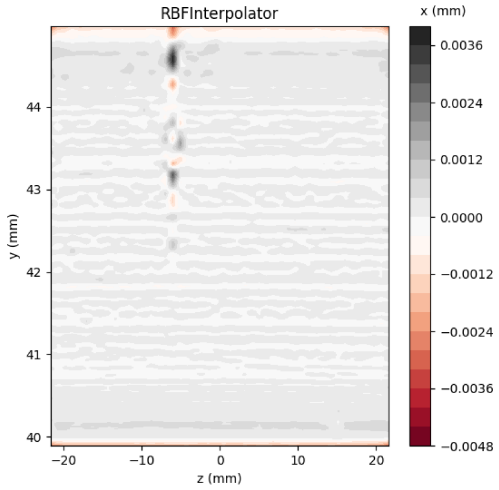
RMSE	**MaxAE**	**Kernel**	**Smoothing**	**Epsilon**
0.000078	0.007357	Thin plate spline	0.1	5
0.000084	0.007947	Linear	0.1	5
0.000105	0.007313	Cubic	0.1	5
0.000117	0.006103	Quintic	0.1	5
0.000153	0.006843	Quintic	1.0	5

However, this initial verification is not sufficient to determine which interpolator produces the best interpolated surfaces, so it was necessary to conduct a visual analysis of the points, studying the behaviour of each parameter separately and where the largest errors are located.

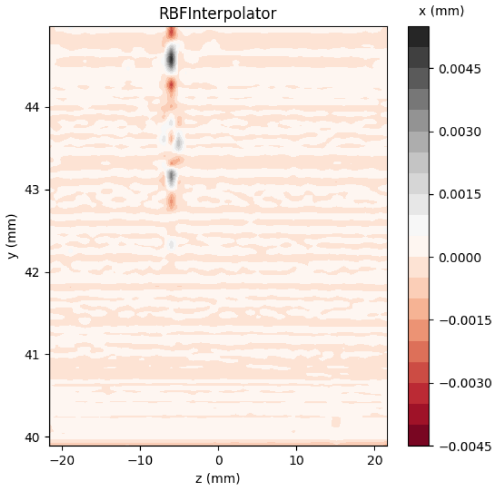
Figure 33 shows the three best kernels found for this problem, `thin_plate_spline`, `cubic` and `quintic`. The values of $\mathrm{s} = 1$ and $\epsilon = 1$ were kept constant for comparison. In general, the kernel affects the performance of the RBF interpolator because it determines how the data is used to make predictions. It can be seen that quintic and cubic are smooth and have a compact support, which means that they only have a significant effect on the interpolated values for data points that are close to the target point. For this reason, the cubic kernel in particular is a good choice to produce an interpolated surface that is both smooth and able to capture non-linear relationships in the data.

Figure 33 – Interpolation results for different kernels

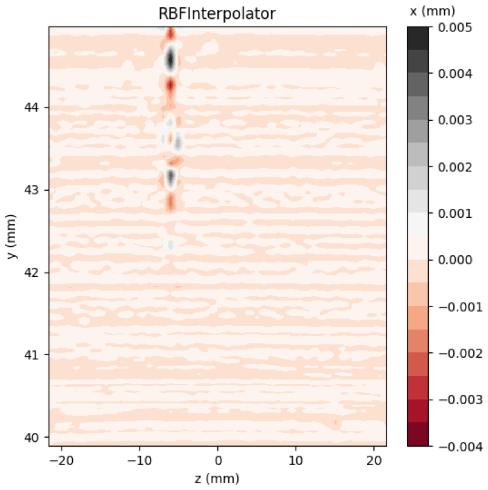
(a) thin_plate_spline



(b) cubic



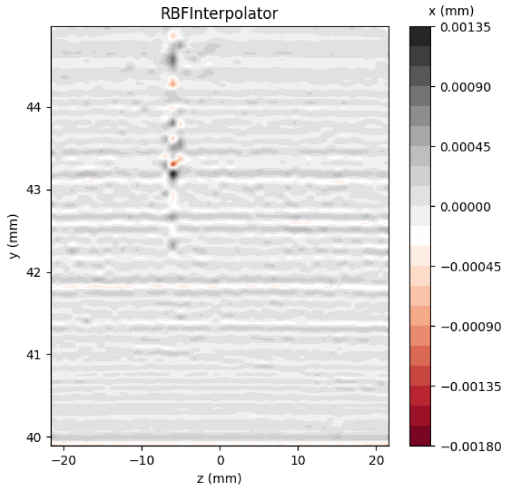
(c) quintic



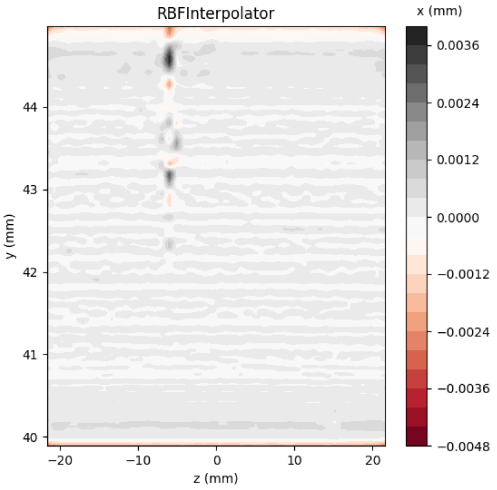
The smoothing parameter was chosen based on the fact that it is able to smooth out distortions on the tooth surface, and this can be verified by the difference in coloration in the region, indicating that the outliers may have been smoothed as desired. Figure 34 shows three interpolation results with $\mathrm{s} = 0.1$, $\mathrm{s} = 1$and $\mathrm{s} = 5$, all using the *thin_plate_spline* function and $\epsilon = 1$.

Figure 34 – Interpolation results for different values of smoothing (*thin_plate_spline*)

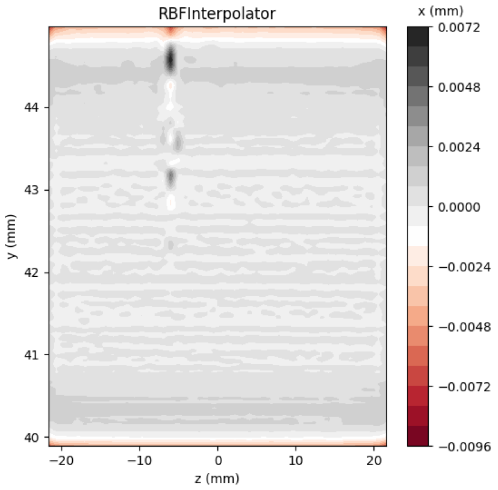
(a) s = 0.1



(b) s = 1



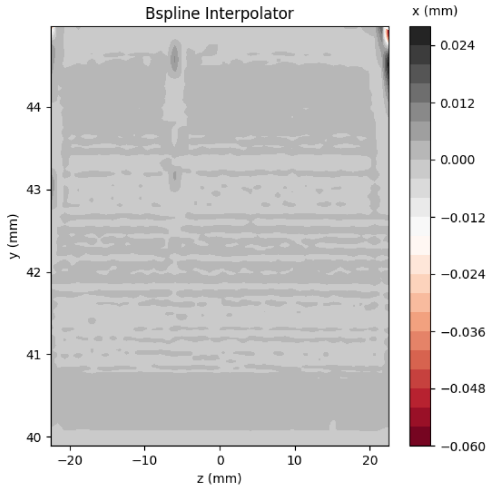
(c) s = 5



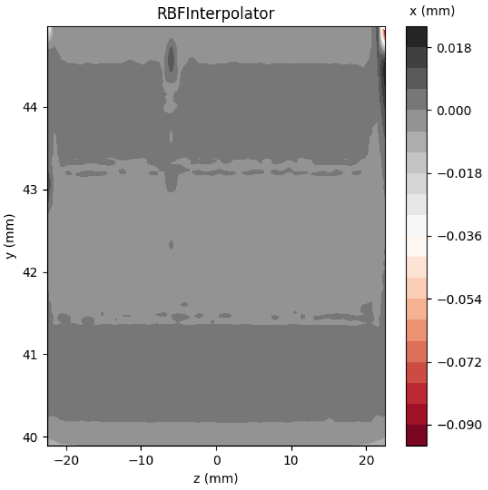
However, after tests with the ZaKo3D simulator, it was found that the level of smoothing would still not be sufficient to make the surface free of distortion, so an attempt was made to increase the level of smoothing, as shown in Figures 36a, 36b, 36c. Nevertheless, in the upper right corner, a variation is observed that increases with smoothing to the order of $90\mu\mathrm{m}$, which is a significant value for this study. The experiments were carried out with $\mathrm{kernel} = \mathrm{cubic}$ and $\epsilon = 1$.

Figure 35 – Interpolation results for different values of smoothing (*cubic*)

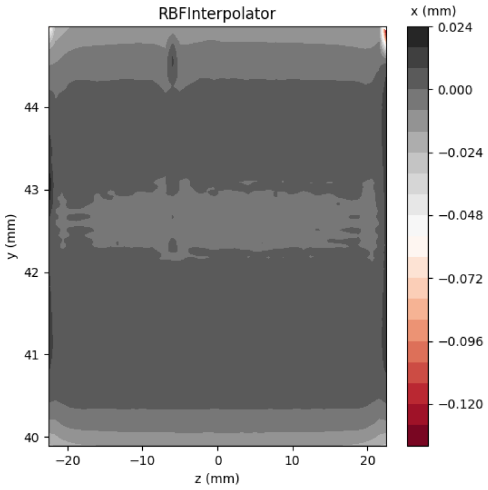
(a) s = 10



(b) s = 100



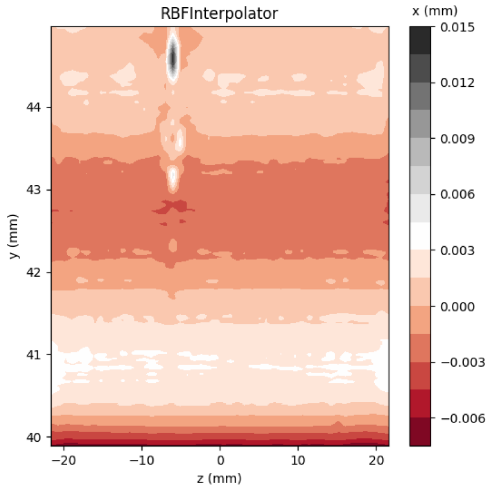
(c) s = 1000



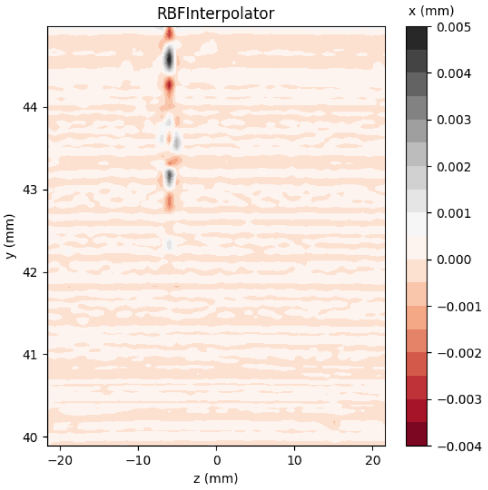
Finally, the behaviour of epsilon was analyzed together with the other parameters in the interpolator. It can be seen that a larger value of epsilon results in a narrower RBF, meaning that the interpolation will be more sensitive to changes in local data points. Conversely, a smaller value of epsilon results in a wider RBF, meaning that the interpolation will be smoother and less sensitive to changes in the local data points. Figure 36 shows a visual comparison of the shape parameter values using `kernel=quintic` and $\mathrm{s} = 1$.

Figure 36 – Interpolation results for different values of epsilon

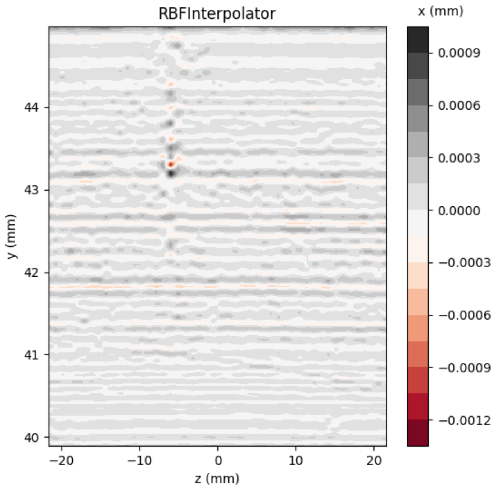
(a) ϵ = 0.1



(b) ϵ = 1



(c) ϵ = 5



Thus, the RBFInterpolator (`kernel=cubic, smoothing=1, epsilon=1`) presented the best compromise when evaluating the images and the performance metrics. Therefore, it is a strong candidate to perform good interpolations and/or extrapolations.

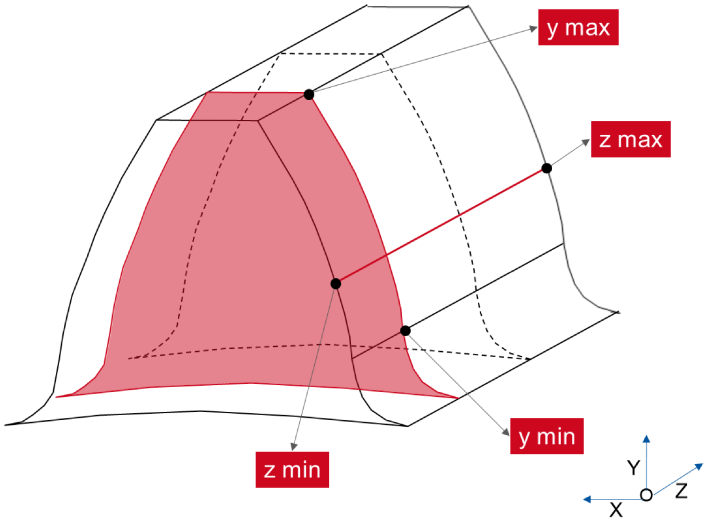
4.5.2 Extrapolation

The extrapolation of tooth surfaces is done by setting new extrapolation boundaries. Then, by user input of the number of profile lines and the number of points in each, the z and y vectors that form the grid for extrapolation are generated. Therefore a new block called `new grid` has been added.

4.5.2.1 New grid

The `new grid` contains the values of the z and y coordinates. Therefore, the first step is to find the grid boundaries, i.e. the minimum and maximum points for y and z. In the Figure 37, the position of the y and z limits necessary for the creation of a new grid is shown.

Figure 37 – Representation of a profile line and a flank line.



Source: Author (2023).

In order to find a valid case for both spur and helical gears, it was necessary to find the minimum and maximum values of y for each profile line. Since helical gears have a helix angle, the values in the y direction will change slightly at the beginning and end of each profile line, which can become significant in a high accuracy extrapolation. Therefore, the criterion used to find y_{max} was the value of the tip diameter, and the criterion used to find y_{min} was the root form diameter (see section 2.1.2). While the minimum and maximum values for z are derived from the face width gear specification. This is because z has a routine in Fortran to take the measurements and organise the point cloud to start in the middle and go up or down.

The initial guess of y and z is entered in the `get_limits` function for each profile line. The initial maximum and minimum values of y are taken from the **partially** measured existing profile lines, while the input z is the z corresponding to the profile line created with user input. Two auxiliary functions have therefore been implemented: `f_tip_d` and `f_root_form_d`. These functions compare the calculated diameter with `tip_diameter` and `root_form_diameter` respectively. The diameter d calculation requires the value of x,

$$d = 2 \cdot \sqrt{x^2 + y^2}. \tag{12}$$

The interpolator is then used to find the value of x, using the input values of y and z. Finally, an optimised function is used to estimate the correct value of $\mathrm{y_{max}}$ and $\mathrm{y_{min}}$. The Secant function from the **scipy.optimize** library was used, see the documentation for more details (SCIPY, n.d.).

4.5.3 File writing

Using the user-defined grid, the interpolator is then applied to this new set of values and a new flank surface is created. The same process is repeated for each flank (right, left) and for all the teeth contained in the gear. Finally, the last stage is to write the new point cloud to a GDE file. This is done by calling a second Python function and entering all the basic geometric information as well as the new topography of the gear flank.

Then, within the WZL data structure, the Geometry Converter programme is used to convert the file written in GDE format to PKT format. In addition, the files in PKT format also store the normals of each point, so another class is called to calculate the normals (see more in section 2.4.2). In this way, the converter creates several files in PKT format, the number of files created depending on the number of teeth stored in the GDE file. Afterwards, all these files are used as input for the ZaKo3D software.

4.6 PREPARING THE TOOTH CONTACT SIMULATION

To perform tooth contact analysis without loads in ZaKo3D requires a few inputs. These include point cloud files for the pinion and the wheel, which can be in the form of `.PKT` files for as many teeth as desired, as well as the base diameter for both gears. The base diameter can be found in the PDF document generated by the measuring machine, and the axial distance must also be provided. Once this information is entered, the user can select the number of pitches to be simulated and the number of rolling positions per pitch.

4.7 RESULTS ANALYSIS

4.7.1 GDE reading routine

Within the context of creating a generic algorithm, in the sense of returning an accurate result for different input gear models and types, 6 models within the selected types were tested, summarised in Table 7. It also read GDE files containing a different number of points in each profile line and the number of profile lines per flank (right and left). The routine correctly imputed all topographic and basic geometry data into the Institute's code library for all gear types tested.

Table 7 – Gear types

Type	Nº of teeth	Nºof profile lines	Nºof flank lines
Planet Gap	16	37	41
Planet Tooth	16	37	41
Pinion Gap	21	29	33
Pinion Tooth	21	29	33
Bevel Pinion	11	21	21
Bevel Wheel	39	21	21

4.7.1.1 Sorting routine

In the sorting routine, two steps were carried out prior to the reading of the file. These are:

- Conversion of the file from GDE to PKT format using Geometry Converter software;
- Data randomization using Matlab software.

The conversion of the files to PKT format was necessary because of the support implemented in Matlab for reading files of this type. Finally, the GDE file was read using the function developed and then the function that allowed the ordering of the profile lines in the dataset was called. This method allowed us to compare the original files with the ordered ones.

After carrying out the tests it was noticed that the algorithm performed correct reading and sorting for spur and helical gears, then, the answer was valid for 4/6 of the selected gears. The reason for this was that there were inconsistencies in finding the next point for bevel gears, resulting in an uneven list. The cause is found in the variations of the Z-values that corroborate the increased complexity of the problem.

However, overall, this algorithm provides an efficient and accurate way to sort points in 3D space for use in gear design and simulation.

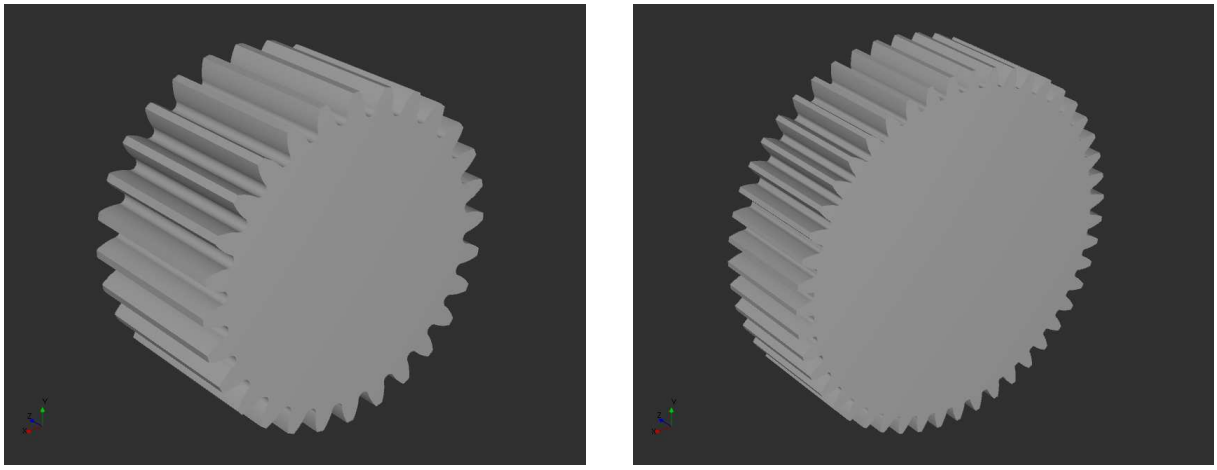
4.7.2 Interpolation

To better understand the results of automation, the example of the Spur Gear (SG) pinion is described in detail. For this case, $1,612,800$ topographic measurements (576 points per flank, along 50 profiles in 28 teeth) must be read from the MKA file, with other essential geometric information. In addition to these $1,612,800$ points, thousands more are generated internally in the Fortran library to create the root surface. The entire topography is then written to the GDE file and re-read in the Python program to recreate the new topography.

Hence the two interpolators chosen to recreate the surface were RBF (`kernel=cubic, smoothing=1, epsilon=1`) and griddata (`kernel=cubic`) as the latter is only an interpo-

Figure 38 – 3D topographies

(a) Pinion (b) Wheel



Source: Author (2023).

lator and has no smoothing options. However, in spite of the good fit of the Griddata to the data as an interpolator in the initial tests carried out in the section 4.5.1.1, some points had non-numerical values (NaN). This could lead to a singular or ill-conditioned matrix generated by the interpolator, but future tests should be carried out. Thus, it was not possible to proceed with the Griddata.

Therefore, the RBF interpolator was successfully applied to a point cloud of 70,000 points distributed in 50 points per flank along 50 profile lines in 28 teeth, removing the `root form diameter`. The generation of new topographies for all teeth took approximately 3 hours and 30 minutes, with an average of approximately 7 minutes per tooth due to the large number of points calculated by the interpolator. The tests were performed on a Windows 10 64-bit computer with an Intel(R) Core(TM) i7-8700 CPU running at 3.20 GHz (up to 4.6 GHz) and 16.0 GB of RAM.

4.7.3 Simulation

For a simulation in the Zako3D software, the SG pinion values were read and a completely new gear topography was created. This has the same basic geometry as the first and is shown in Figure 38. The second component selected for meshing was the wheel, a spur gear with 47 teeth and a diameter of 132.4966 mm. The 3D topographies generated by the software for both gears are illustrated in Figure 38.

Furthermore, as the points below the `root form diameter` were not recreated in this investigation, due to the balance between the complexity required and the time available, only the tooth contact analyses free of load were performed, which does not require the points below the `root form diameter` to perform the calculations.

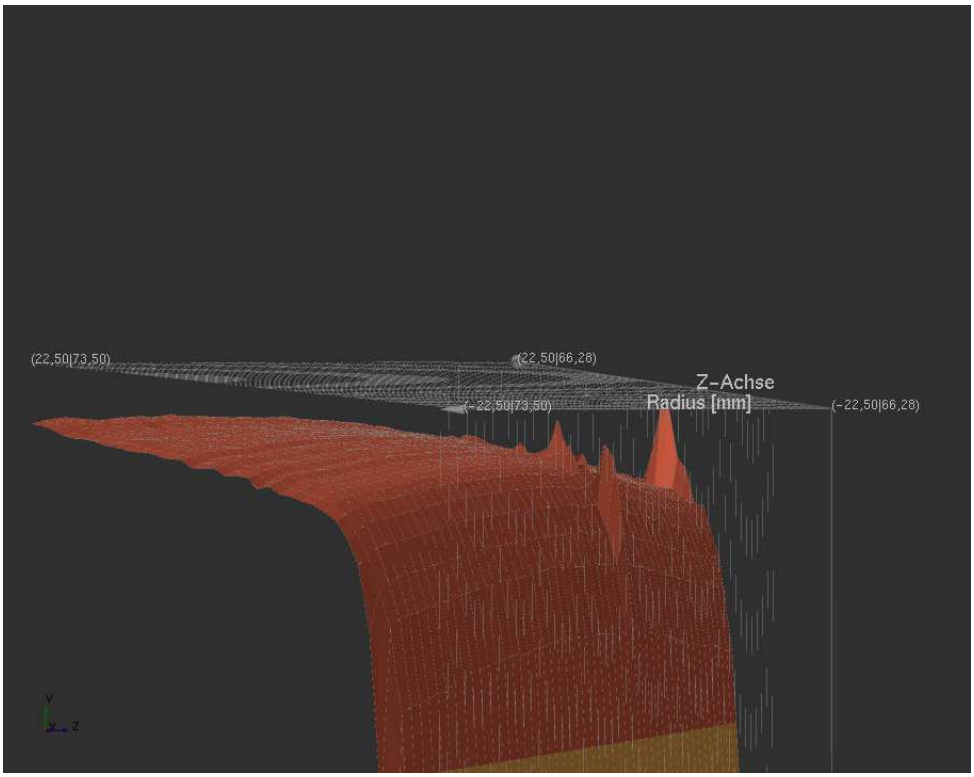
4.7.3.1 Load free Tooth Contact Analysis

The ease-off, contact pattern and transmission error for multi-tooth and single-tooth meshes have been calculated, but the ease-off already gives an indication of the meshing result, and therefore whether the interpolated surface has been able to recreate the gear topography with confidence. Figure 39a shows the ease-off obtained with the original measured topography converted directly from MKA to PKT format, and Figure 39b shows the ease-off obtained with the topography recreated by interpolation. It can be seen that the reconstructed topography has the same curvature as the original and a slight softening in relation with 39a distortions, but still contains them.

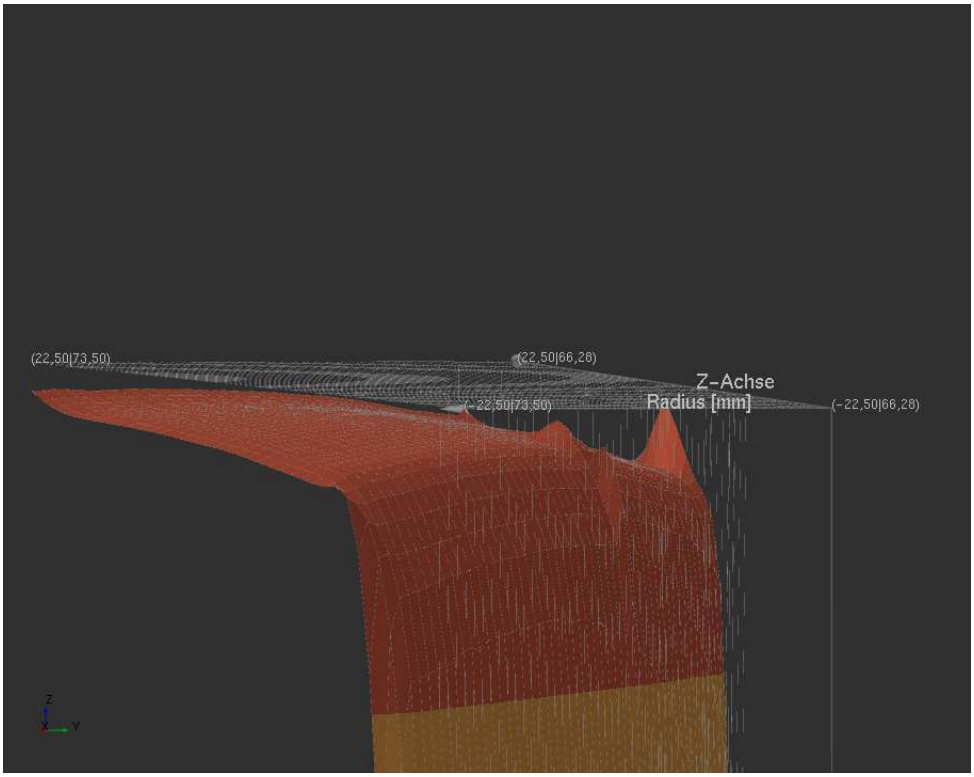
This observation leads one to believe that an increase in smoothing can lead to the total destruction of the deviations, although it has been observed in section 4.5.1.2 that as the smoothing increases, the surface begins to exhibit anomalous upward behaviour.

Figure 39 – Ease-off

(a) Original



(b) Recreated



Source: Author (2023).

5 CONCLUSION

This chapter presents the final considerations and concluding remarks on the presented work. Section 5.1 summarizes the achievements, and Section 5.2 provides a future perspective of the developed project.

5.1 FINAL REMARKS

This work aimed at developing methods for the automated processing of gear topographies as an enabler for virtual and end of line tests. This approach is divided into three macro processes: measurement, data reading and processing, and finally, simulation. Thus, an end-to-end pipeline to simulate tooth contact analyses with partially and ill-measured surfaces was developed.

First, the topography of gear was captured through measurements successfully performed on the Kilgenberg p16. The contribution for the second macro step was the construction of a reading GDE routine for reading GDE files. This routine is inserted into the WZL Kernels, part of the data structure algorithm. The same data structure is capable of doing conversions between different file formats and sorting the input data for other applications. The sorting allows applications like ZaKo3D to receive the data in readable form for the correct calculation of the simulations.

It is important to note that GDE files contain all the topological information and dimensional characteristics of a part. Therefore, reading files in this format is essential to manipulate gear data in a fast, integrated way and with fewer input errors. Hence, once the developed algorithm correctly read the input data of cylindrical gears with different attributes, it was shown to be robust and generic in this process.

During the data processing, the second contribution was the creation of algorithms capable of reconstructing the gear flank surfaces through interpolation and extrapolation methods in order to reduce deviations in them. The solution involved the study of interpolator models and the optimisation of hyperparameters. In this way, it was verified that the RBF presented a better ability to fit the data than the BSpline, applying the Absolute Error (AE) metric between the original values and the predicted ones. In addition, the Griddata interpolator also performed satisfactorily for interpolations, but without the possibility of extrapolation.

The use of interpolators proved to be efficient for the interpolation of gear flank surfaces. Through the ease-off calculation performed in ZaKo3D software, the surfaces showed the same behaviour as the original surface in terms of curvature and tendency. However, the surfaces still exhibited the original deformations to a lesser extent. It is worth noting that these deformations are quite small, in the order of microns, which highlights the complexity of the problem. In view of this, other smoothing values in the order of hundreds were explored in order to reduce the deviations. However, a heat map

was used to check the magnitude of the residuals, which indicated an undesirable point of distortion.

Additionally the interpolation aspects, the algorithm is adaptable and modular, allowing for the creation of extrapolated surfaces with a minimum of user input. However, due to the tight timeline of the project, this step was not taken. Studies using only interpolators can still be improved. In the RBF interpolator, options such as "neighbors" can be executed. This option could reduce the fitting time of the interpolation, as not all of the thousands of points are used. In addition, adapting the sorting routine for bevel gears could increase the ability of the algorithm to be generic.

In conclusion, this work has achieved the proposed tasks of building an end-to-end pipeline to simulate tooth contact analysis, which can reduce the time spent on measurement and data exchange, as well as minimise errors in the process. As a result of this project, other projects within the WZL Institute dealing with data in GDE format are progressing as well.

5.2 OUTLOOK

As a future perspective of the presented work and possible improvements to the developed process, one could suggest:

- Investigate other libraries in Python that implement Spline interpolation;
- Perform extrapolations from partially measured areas;
- Implementing routines to enable interoperability between Python and Fortran, so that the re-created data is entered directly into the Fortran structure, allowing greater efficiency in the final build time;
- Convert this data pre-processing step, with certain improvements, into an internal software available in the Gear ToolBox to carry out the TCA with confidence.

REFERENCES

AG, KISSsoft. **Generating Toothing Data in GDE Format**. 2019. Available from: https://www.kisssoft.com/en/news-and-events/newsroom/generating-toothing-data-in-gde-format. (accessed: 14.01.2022).

ARCGIS PRO 3.0. **How radial basis functions work**. Available from: https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/how-radial-basis-functions-work.htm. (accessed: 03.01.2022).

BANODIYA, Balkishan; KARMA, Vijay Kumar. MEASURMENT OF TRANSMISSION ERROR IN SPUR GEARS. **International Research Journal of Engineering and Technology (IRJET)**, p. 2369–2375, 2017.

BEHNEL, Stefan; FAASSEN, Martijn; BICKING, Ian. **lxml: XML and HTML with Python**. [S.l.]: Lxml, 2005.

BEUTH. **DIN 21772: Gears - Cylindrical involute gears and gear pairs - Definition of deviations**. 2023. Available from: https://www.beuth.de/en/standard/din-21772/125295771. (accessed: 15.02.2023).

BIANCOLINI, Marco Evangelos. **Fast Radial Basis Functions for Engineering Applications**. 1. ed. Roma - Italy: Springer International, 2017a.

BIANCOLINI, Marco Evangelos. Radial Basis Functions. In: FAST Radial Basis Functions for Engineering Applications. Cham: Springer International Publishing, 2017b. P. 9–33. ISBN 978-3-319-75011-8.

BOYD, John P.; GILDERSLEEVE, Kenneth W. Numerical experiments on the condition number of the interpolation matrices for radial basis functions. **Applied Numerical Mathematics**, v. 61, n. 4, p. 443–459, 2011. ISSN 0168-9274.

BRECHER, Prof. Dr.-Ing. Christian; PROF. DR.-ING. THOMAS BERGS, MBA. **ZaKo3D Manual - Tooth Contact Analysis**. 1. ed. Germany, 2021. Last accessed 11 January 2023.

COSTA, MATEUS DE MAGALHÃES BARCELOS. **AUTOMATIC HANDLING OF GEAR DATA FOR INDUSTRY 4.0 APPLICATIONS**. 2022. S. 70. Monography (Graduation thesis) – UNIVERSIDADE FEDERAL DE SANTA CATARINA, Joinville.

ELTON, Dan. **Why physicists still use Fortran**. Available from: http://www.moreisdifferent.com/2015/07/16/why-physicsts-still-use-fortran/. (accessed: 17.01.2022).

HARDY, Rolland L. Multiquadric equations of topography and other irregular surfaces. **JGR Journal of Geophysical Research**, v. 76, n. 10, p. 1905–1915, 1971.

HEMMELMANN, Dipl.-Ing. Jan Erich. **Simulation des lastfreien und belasteten Zahneingriffs zur Analyse der Drehübertragung von Zahnradgetrieben**. 1. ed. Aachen - Germany: [s.n.], 2007.

INTEL. **Standard Fortran and C Interoperability**. [S.l.: s.n.], 2022. Last accessed 17 JUly 2022. Available from: https://www.intel.com/content/www/us/en/develop/documentation/fortran-compiler-oneapi-dev-guide-and-reference/top/compiler-reference/mixed-language-programming/standard-fortran-and-c-interoperability.html.

JELASKA, Damir. **LaTeX: Gears and gear drives**. 2. ed. Noida, India: Thomson Digital, 2012.

JULIVER, Jamie. **XML Files: What They Are How to Open Them**. [S.l.: s.n.], 2021. Last accessed 25 July 2022. Available from: https://blog.hubspot.com/website/what-is-xml-file#:~:text=To%5C%20summarize%5C%3A%5C%20An%5C%20XML%5C%20file,indicate%5C%20the%5C%20type%5C%20of%5C%20element..

KLINGELNBERG. **KLINGELNBERG P 16 Precision Measuring centers**. [S.l.: s.n.], 2022a. Last accessed 28 JUly 2022. Available from: https://klingelnberg.com/fileadmin/Business_Divisions/Precision_Measung_Centers/Downloads/P16_BROCH_EN.pdf.

KLINGELNBERG. **P 16**. [S.l.: s.n.], May 2022b. https://klingelnberg.com/en/industry-solutions/automotive/detail-page?product=183&cHash=a528d1cbb4b3b377996045ec6ac26c49. Last accessed 16 July 2022.

KLOCKE, Fritz; BRECHER, Christian. **Zahnrad- und Getriebetechnik - Auslegung – Herstellung – Untersuchung – Simulation**. 1. ed. München: Carl Hanser, 2017. ISBN 978-3-446-43068-6.

LAI, Ming-Jun; SCHUMAKER, Larry L. **Spline Functions on Triangulations**. [S.l.]: Cambridge University Press, 2007. (Encyclopedia of Mathematics and its Applications).

LEARN ENGINEERING. **Gear Types, Definition, Terms Used, And The Law Of Gearing**. 2021. Available from: `https://medium.com/@learnengineering/gear-types-definition-terms-used-and-the-law-of-gearing-98c7517ef93`. (accessed: 12.02.2023).

LIN, Chen-Hsiang; FONG, Zhang-Hua. Numerical tooth contact analysis of a bevel gear set by using measured tooth geometry data. **Mechanism and Machine Theory**, v. 84, p. 1–24, 2015.

LIN, Hu; KELLER, Frank; STEIN, Martin. Influence and compensation of CMM geometric errors on 3D gear measurements. **Measurement**, Elsevier BV, v. 151, p. 107110, Feb. 2020.

LITVIN, Faydor L.; FUENTES, Alfonso. **GEAR GEOMETRY AND APPLIED THEORY**. 2. ed. UK: Cambridge University, 2004.

MO, Jiahui; SHOU, Huahao; CHEN, Wei. Implicit Surface Reconstruction via RBF Interpolation: A Review. **Recent Patents on Engineering**, v. 16, p. 49–66, 2022.

MONGILLO, Michael. Choosing Basis Functions and Shape Parameters for Radial Basis Function Methods, 2011.

NORMUNG, DIN Deutsches Institut für. **Gears – Cylindrical involute gears and gear pairs – Concepts and geometry (ISO 21771:2007)**. Berlin, Aug. 2014. Last accessed 12 January 2023.

O'NEILL, Barrett. **Elementary Differential Geometry**. [S.l.]: Elsevier, 2006.

PIEGL, Les; TILLER, Wayne. **The NURBS Book**. [S.l.]: Springer Berlin Heidelberg, 1995.

RADU, M C; ANDREI, L; ANDREI, G. A survey on gear meshing quality based on tooth contact analysis. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, v. 514, n. 1, p. 012027, May 2019.

RADZEVICH, Stephen P. **LaTeX: DUDLEY's HANDBOOK OF PRACTICAL GEAR DESIGN and MANUFACTURE**. 3. ed. New York: Taylor & Francis Group, 2016.

SCIPY. **scipy.optimize.newton**. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.newton.html#scipy-optimize-newton.

SCIPY. **griddata**: Interpolate unstructured D-D data. [S.l.], 2021. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.griddata.html.

SCIPY. **Interpolation**. [S.l.: s.n.], 2023. https://docs.scipy.org/doc/scipy/tutorial/interpolate.html. Last accessed 13 February 2022.

SKALA, Vaclav; CERVENKA, Samsul Ariffin Abdul Karim Martin. Finding Points of Importance for Radial Basis Function Approximation of Large Scattered Data. **Computational Science – ICCS 2020**, p. 239–250, 2020.

SOMMER, David. **Stable Radial Basis Function Interpolation for Multi-Physics Simulation Applications**. Available from: https://elib.uni-stuttgart.de/bitstream/11682/10370/1/Bachelorarbeit_Sommer.pdf. (accessed: 07.01.2022).

STADTFELD, Hermann J. **The Basics of Gear Theory**. 2015. Available from: https://www.geartechnology.com/articles/22140-the-basics-of-gear-theory. (accessed: 06.01.2022).

STANDARDS, VDI. **Exchange format for gear data Gear Data Exchange Format (GDE Format) Definition (VDI/VDE 2610)**. Dusseldorf, May 2014. Last accessed 12 January 2023.

STEFFENSEN, Jakob Frederik. **Interpolation**. Second. Mineola, NY: Dover Publications, 2006.

VALENTYN, Sichkar. **XML - files in Python**. [S.l.: s.n.], 2018. Last accessed 20 July 2022. Available from: https://github.com/sichkar-valentyn/XML%5C_files%5C_in%5C_Python/blob/master/example.xml.

VDI. **GDE 3.0**. [S.l.], 2021. Available from: https://www.vdi.de/fileadmin/xml/2610/.

VDI/VDE|2612. **Measurement and testing of gears**. [S.l.], Mar. 2018. Last accessed 02 January 2022. Available from: https://www.vdi.de/fileadmin/pages/vdi_de/redakteure/richtlinien/inhaltsverzeichnisse/2852502.pdf.

W3C. **XML ESSENTIALS**. [S.l.: s.n.], Nov. 2021. Last accessed 29 July 2022. Available from: https://www.w3.org/standards/xml/core#:~:text=What%5C%20is%5C%20XML%5C%3F,more%5C%20suitable%5C%20for%5C%20Web%5C%20use..

WANG, Shenghui; ZHOU, Yuansheng; TANG, Jinyuan; TANG, Kai; LI, Zhengminqing. Digital tooth contact analysis of face gear drives with an accurate measurement model of face gear tooth surface inspected by CMMs. **Mechanism and Machine Theory**, v. 167, 2022.

WELLNHOFER, Nick. **Libxml2**. [S.l.: s.n.], 2022. Last accessed 10 July 2022. Available from: https://gitlab.gnome.org/GNOME/libxml2/-/wikis/home.

WRIGHT, Grady Barrett; FORNBERG, Bengt. **Radial basis function interpolation: numerical and analytical developments**. 2003. MA thesis – United States. Last accessed 08 January 2023.

XU, Bin; SHIMIZU, Yuki; ITO, So; GAO, Wei. Pitch deviation measurement of an involute spur gear by a rotary profiling system. **Precision Engineering**, v. 39, p. 152–160, 2015.

YANG, Lin; REN, Shuai Nan; WANG, Zhen Yu; ZHANG, Peng. Research on spiral bevel gear tooth surface reconstruction based on NURBS. **Applied Mechanics and Materials**, v. 130-134, p. 701–705, 2011.

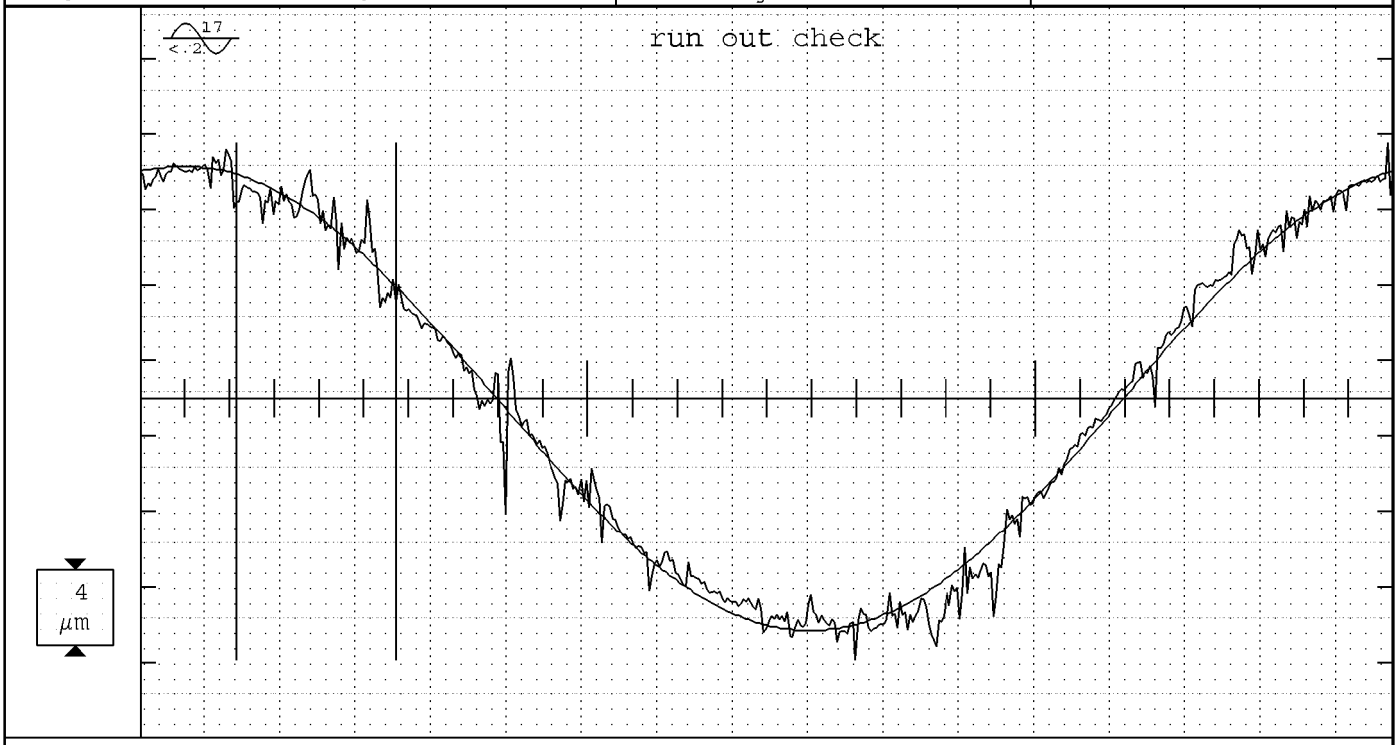
YANG, Ruting; XING, Bing. A Comparison of the Performance of Different Interpolation Methods in Replicating Rainfall Magnitudes under Different Climatic Conditions in Chongqing Province (China). **Atmosphere**, v. 12, n. 10, 2021. ISSN 2073-4433.

ANNEX A – GEAR TOPOGRAPHY AND RUN OUT

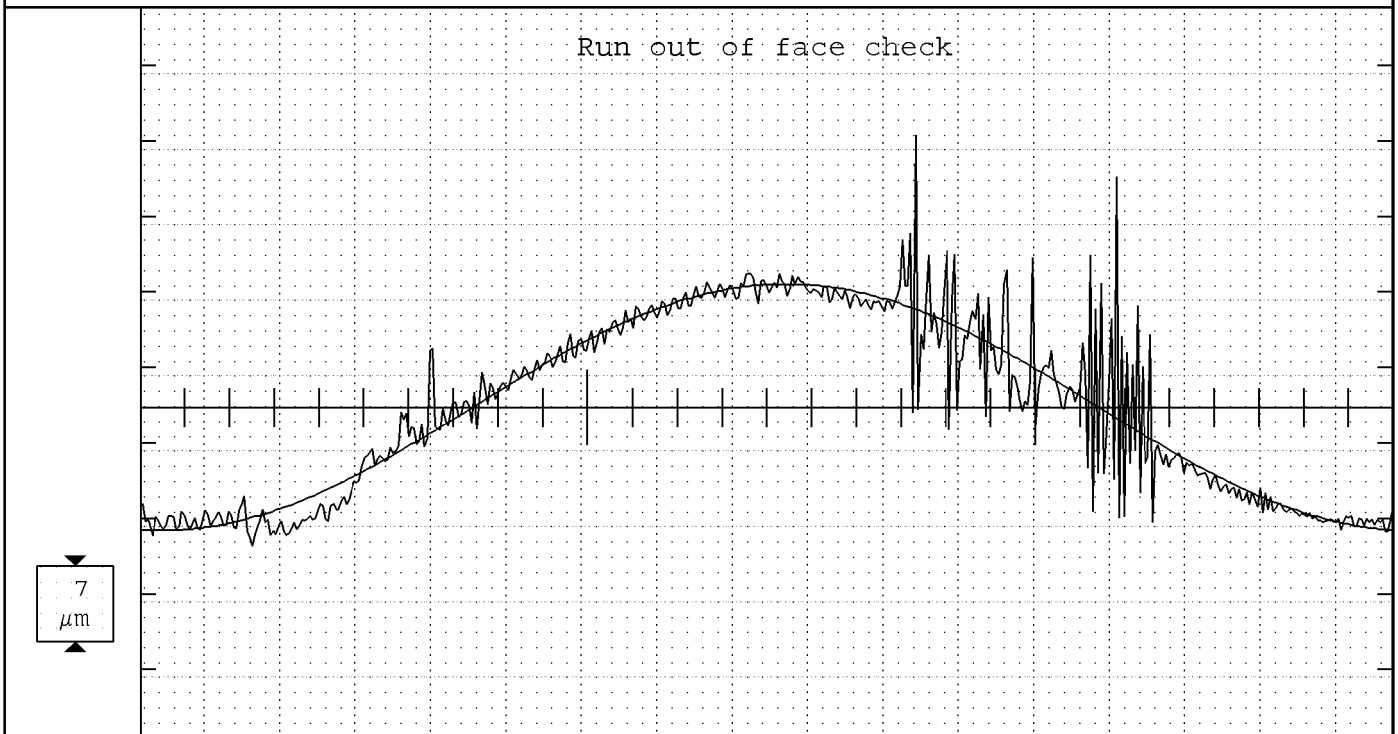
WZL Aachen

Compensation / Journal Ref.

Prog.No.:	GST0434_08 4.23.0 0 P 16	Operator:	nmh_cp	Date:	04.11.2022 13:16
Type:	planet gear	No. of teeth	28	Face Width:	45mm
Drawing No.:	PLN-SG-28-3-04	Module m	3mm		
Order No.:	pln-sg-28-3-04_v10	Pressure angle	20°		
Cust./Mach. No.:		Helix angle	0°		
Loc. of check:	p16	Base Cir.-Ø db	78.9342mm	Stylus-Ø	'17(#6)5mm
Condition:	finished	Base Helix angle	0°	Add.Mod.Coeff x	-.039



Amplitude(s)	:	0.0123 mm [.2 mm]
Max. - tooth	:	1
Min. - tooth	:	15
X: -.002 Y: 22.522 Z: -6.816 Z s: -44.375 mm		



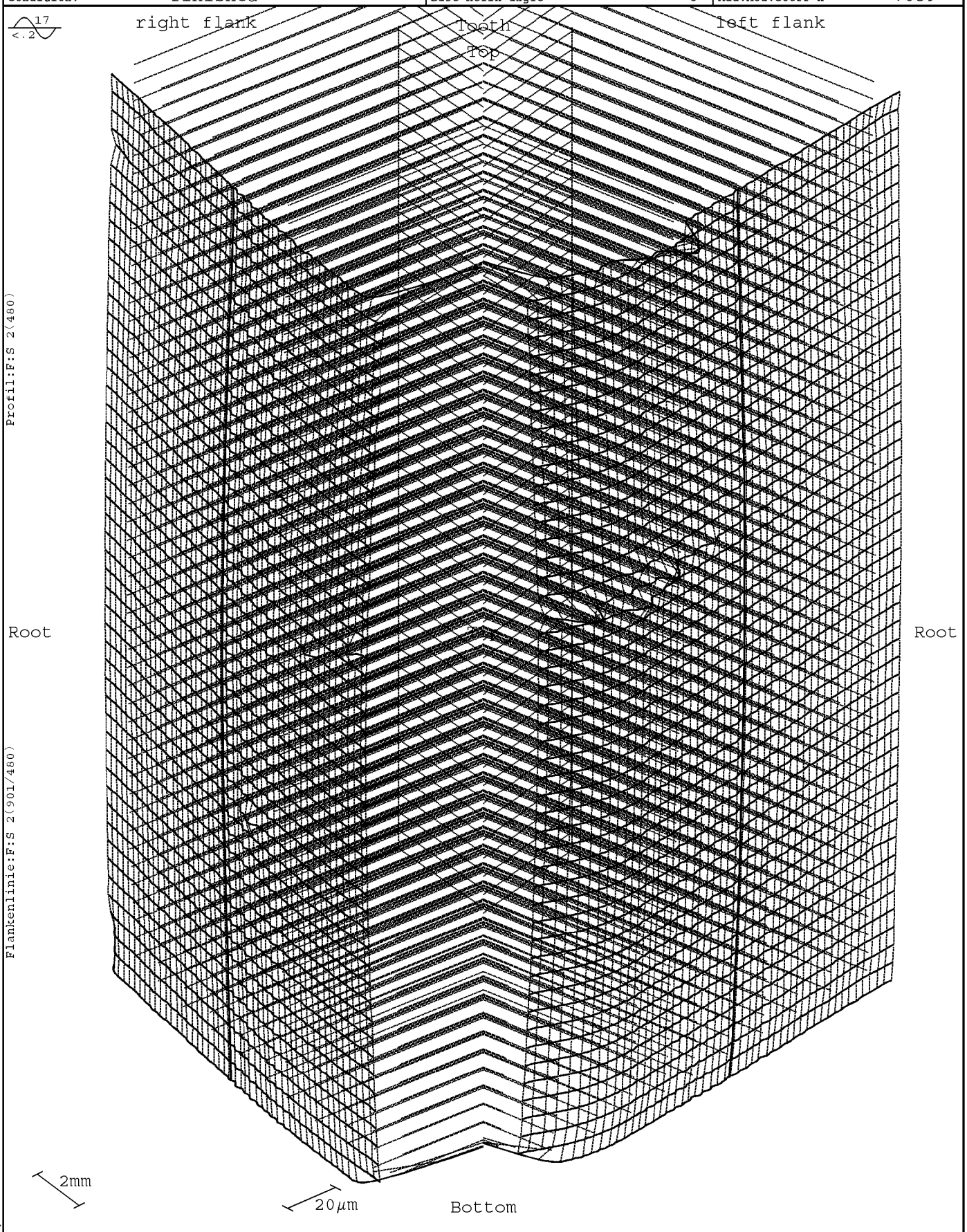
Amplitude(s)	:	0.0114 mm [.2 mm]
Max. - tooth	:	14.4
Min. - tooth	:	.4
X: 1.469 Y: 32.742 Z: 8.02 Z s: -44.375 mm		

KLINGELNBERG

WZL Aachen

Gear Topography

Prog.No.:	GST0434_08 4.23.0 0 P 16	Operator:	nmh_cp	Date:	04.11.2022 13:16
Type:	planet gear	No. of teeth	28	Face Width:	45mm
Drawing No.:	PLN-SG-28-3-04	Module m	3mm	Length Ev. La	10.91mm
Order No.:	pln-sg-28-3-04_v10	Pressure angle	20°	Length Ev. Lß	44.1mm
Cust./Mach. No.:		Helix angle	0°	Appr. Length M1	7.93mm
Loc. of check:	p16	Base Cir.-Ø db	78.9342mm	Stylus-Ø	(#8)1mm
Condition:	finished	Base Helix angle	0°	Add.Mod.Coeff x	-.039



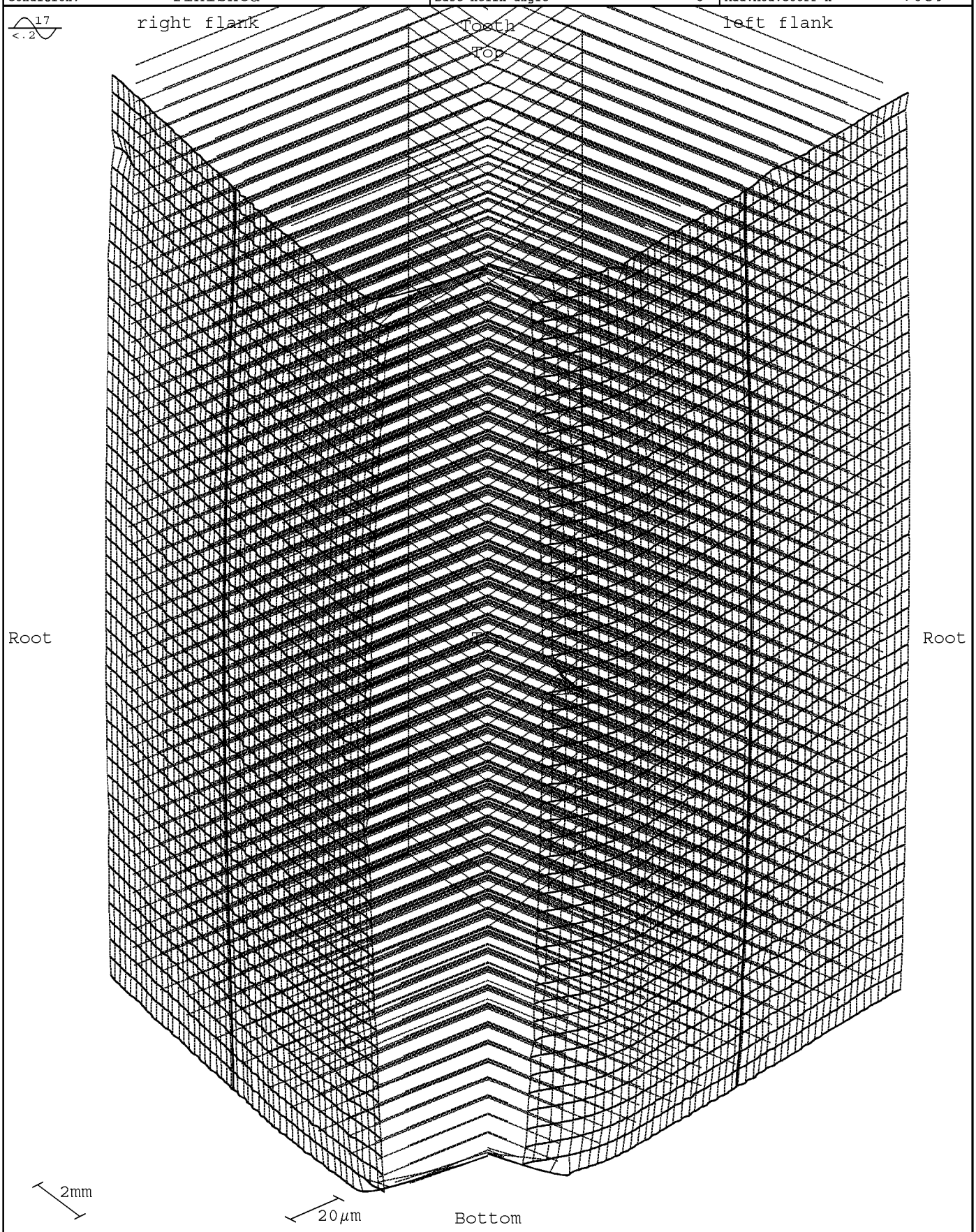
Vb=5:1 Va=500:1 Zahn-Nr.: 1

KLINGELNBERG

WZL Aachen

Gear Topography

Prog.No.:	GST0434_08 4.23.0 0 P 16	Operator:	nmh_cp	Date:	04.11.2022 13:16
Type:	planet gear	No. of teeth	28	Face Width:	45mm
Drawing No.:	PLN-SG-28-3-04	Module m	3mm	Length Ev. La	10.91mm
Order No.:	pln-sg-28-3-04_v10	Pressure angle	20°	Length Ev. Lß	44.1mm
Cust./Mach. No.:		Helix angle	0°	Appr. Length M1	7.93mm
Loc. of check:	p16	Base Cir.-Ø db	78.9342mm	Stylus-Ø	(#8)1mm
Condition:	finished	Base Helix angle	0°	Add.Mod.Coeff x	-.039



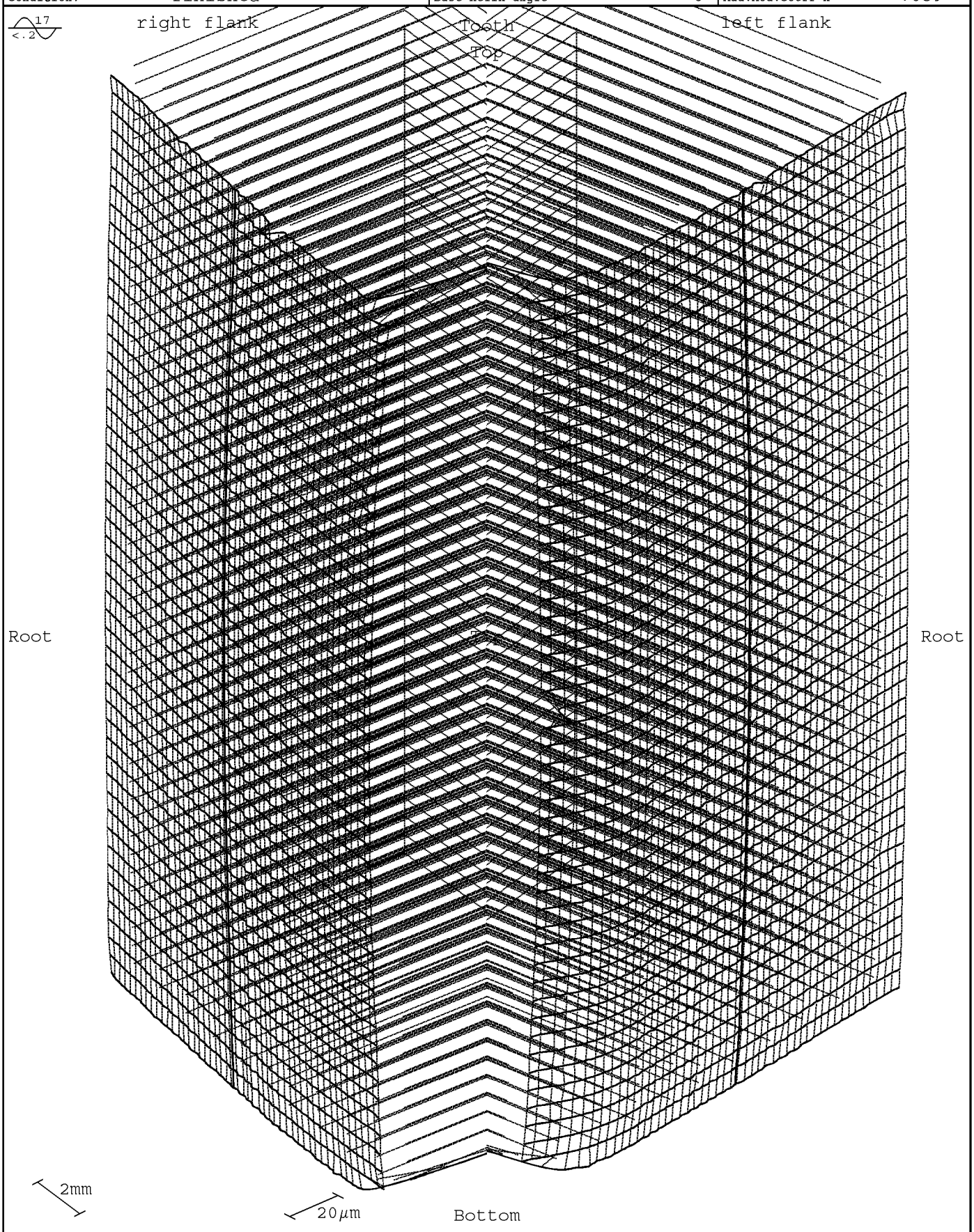
Vb=5:1 Va=500:1 Zahn-Nr.: 2

KLINGELNBERG

WZL Aachen

Gear Topography

Prog.No.:	GST0434_08 4.23.0 0 P 16	Operator:	nmh_cp	Date:	04.11.2022 13:16
Type:	planet gear	No. of teeth	28	Face Width:	45mm
Drawing No.:	PLN-SG-28-3-04	Module m	3mm	Length Ev. La	10.91mm
Order No.:	pln-sg-28-3-04_v10	Pressure angle	20°	Length Ev. Lß	44.1mm
Cust./Mach. No.:		Helix angle	0°	Appr. Length M1	7.93mm
Loc. of check:	p16	Base Cir.-Ø db	78.9342mm	Stylus-Ø	(#8)1mm
Condition:	finished	Base Helix angle	0°	Add.Mod.Coeff x	-.039



Vb=5:1 Va=500:1 Zahn-Nr.: 3

KLINGELNBERG