Spectral Extrapolation from Limited Rough Surface Data of Additively Manufactured Parts

by

Loren E. Baugh

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science in Mechanical Engineering

Auburn, Alabama
December 14, 2024

Keywords: Surface Roughness, Additive Manufacturing, Spectral Analysis

Copyright 2024 by Loren E. Baugh

Approved by

Robert L. Jackson, Chair, Professor of Mechanical Engineering
Kyle D. Schulze, Professor of Mechanical Engineering
Dan Marghitu, Professor of Mechanical Engineering

Abstract

The proliferation of metal additive manufacturing has brought into focus the crucial role of surface characterization in determining the properties and performance of printed parts across diverse applications. This study aims to explore methods for analyzing surface roughness parameters of laser powder bed fusion additively manufactured as-built specimens through a variety of spectral techniques. The discussion focuses on critical parameters, data analysis methods, and practicality of usage. This study explores utilizing two-dimensional datasets (line scans) to define three-dimensional based parameters and generate relevant surfaces to provide further insight into the measurements. The surface generation methods utilize spectral information and fractal parameters from the two-dimensional datasets to produce artificial surfaces. These artificial surfaces are evaluated against three-dimensional measurements (area scans) using established areal surface roughness parameters. The proposed approach enables a comprehensive assessment of surface parameters acquired from both optical and contact profilometry methods, facilitating comparative analysis. It also provides an opportunity to gain a deeper insight into the factors that influence roughness beyond traditional linear measurements, allowing for a more comprehensive understanding.

Table of Contents

Abstract 2

Chapter 1: Background and Introduction 12

1.1 General material and printing 13

1.1.1 Ti-6Al-4V 13

1.1.2 Additive printing process and general fatigue background 14

1.1.3 XCT coupon design 16

1.2 Roughness, Waviness and Form 18

1.3 Objectives 21

Chapter 2: Data Collection and post-processing 23

2.1 Measurement methods overview (working principles) 24

2.1.1 Line measurements 25

2.1.2 Area measurements 27

2.2 Data collection (parameters) 30

2.2.1 Surface roughness line data parameters (CSP) 30

2.2.2 Surface roughness area data parameters (SWLI) 37

2.3 Data Post-processing 39

2.3.1 Line scan initial post-processing 40

2.3.2 Area scan initial post-processing 42

2.3.3 Line selection from SWLI area scans 47

Chapter 3: Advanced surface characterization parameters 49

3.1 Measured dataset exploration 49

3.1.1 Bearing Area Curve calculation methodology 50

3.1.2 Spectrum introduction 54

3.1.3 Fractal dimension overview 55

3.1.4 Fractal dimension calculation using Fourier spectral method 56

3.1.5 Fractal dimension calculation using Roughness length method 60

3.2 Relating Area datasets to Line datasets through parameters 62

Chapter 4: Surface area Generation/Extrapolation 64

4.1 Extrapolation from line profile data to a generated area 64

4.1.1 Extrapolation using Fourier orthogonal spectrum method from a line dataset to an area dataset 64

4.1.2 Surface generation utilizing Hurst Component from Fourier method using fractal artificial surface program 69

4.1.3 Surface generation utilizing Hurst Component from roughness length method using fractal artificial surface program 72

4.2 Comparison of measured areas to extrapolated and generated areas 73

4.2.1 Extrapolation using orthogonal spectrum method 73

4.2.2 Surface generation utilizing Hurst Component fractal artificial surface 75

Chapter 5: Comparison of surfaces and evaluation of results across specimens 80

5.1 Surfaces extrapolated using the Fourier orthogonal spectrum method 80

5.1.1 Fourier orthogonal spectrum method from CSP datasets 80

5.1.2 Fourier orthogonal spectrum method from SWLI datasets 83

5.2 Fractal Dimension/Hurst Component and RMS roughness across specimens 86

5.3 Surfaces generated using fractal artificial surface code 88

5.3.1 Fractal artificial surfaces from CSP datasets 89

5.3.2 Fractal artificial surfaces from SWLI datasets 92

5.4 Skewness and Kurtosis from generated surfaces 95

5.4.1 Skewness and Kurtosis of surfaces created using CSP profile 95

5.4.2 Skewness and Kurtosis of surfaces created using SWLI profile 97

Chapter 6: Conclusions and future work 99

References 101

Appendix 106

Appendix A: 1.2 Roughness, Waviness and Form Figure generation 106

Appendix B: Distribution and Histogram Figure generation 106

Appendix C: Example surface generation and area BAC figure generation 107

Appendix D: Main Code 108

Appendix E: Data interpolation function 133

Appendix F: planeflat function 136

Appendix H: Sample 001 supplementary figures 137

Appendix I: Sample 002 supplementary figures 146

Appendix J: Sample 003 supplementary figures 154

Appendix K: Sample 004 supplementary figures 162

Appendix L: Sample 009 supplementary figures 170

Appendix M: Sample 010 supplementary figures 178

Appendix N: Sample 016 supplementary figures 186

Appendix O: Sample 017 supplementary figures 194

Appendix P: Sample 018 supplementary figures 202

Appendix Q: Sample 024 supplementary figures 210

Appendix R: Sample 025 supplementary figures 218

Appendix S: Sample 031 supplementary figures 226

Appendix T: Sample 032 supplementary figures 234

Appendix U: Sample 033 supplementary figures 242

Appendix V: Parameter variation between measured and generated surfaces 250

List of Tables

Table 1: Chemical composition of the Ti-6Al-4V alloy provided by AP&C..... 14

Table 2: EOS default process parameters for Ti-6Al-4V..... 14

Table 3: The numbering of specimens and their corresponding DOE designation numbers..... 17

List of Figures

Figure 1: a) Ti-6Al-4V specimen geometry with the area of interest and b) schematic of specimen in the build platform 17

Figure 2: XCT coupon specimen photo 18

Figure 3: a) raw combined surface profile b) form c) waviness d) surface roughness 20

Figure 4: Metrology measurement spectrum 24

Figure 5: Veeco Dektak 150 contact profilometer 26

Figure 6: a) contact profilometry working principle 26

Figure 7: Bruker scanning white light interferometer (SWLI) 28

Figure 8: b) optical profilometry scanning white light interferometry working principle 29

Figure 9: Maximum profile peak height (Rp) & maximum profile valley depth (Rv) 31

Figure 10: Arithmetic mean deviation (average line roughness - Ra) 32

Figure 11: Root mean square deviation (RMS line roughness – Rq) 32

Figure 12: Gaussian distribution with standard deviations shown 34

Figure 13: histogram of sample 001 with 25 bins 35

Figure 14: Skewness (Rsk) 35

Figure 15: Kurtosis (Rku) 36

Figure 16: Maximum areal valley depth (Sv), maximum areal peak height (Sp), maximum areal height (Sz), areal distribution parameters, skewness (Ssk) and kurtosis (Sku) 38

Figure 17: a) Raw profile measurement and fit curve sample 004 b) flattened line profile from sample 004 42

Figure 18: a) Unflattened SWLI raw surface scan sample 004 b) plane flattened SWLI surface 004 43

Figure 19: a) Unflattened SWLI raw surface scan sample 004 b) line by line flattened SWLI surface 004 45

Figure 20: Difference between plane flattening and line by line flattening for sample 004 46

Figure 21: Flattened selected line profile from SWLI dataset sample 004 47

Figure 22: a) Example line profile with method determining percentage of the profile interacting with a line passing through it b) the resulting bearing area curve, or material ratio curve showing the percentage of the surface profile interacting with the line as 51

Figure 23: Example surface with method determining percentage of the surface interacting with a plane passing through it 54

Figure 24: Fourier analysis and linear fit from Dektak line profile data sample 004 59

Figure 25: Roughness window calculation method for line profile 60

Figure 26: Roughness length method for calculating H from Dektak profile sample 004 61

Figure 27: Original Dektak line scan profile and Fourier recreated line profile for sample 004 . 66

Figure 28: RMS roughness with respect to scale 68

Figure 29: a) orthogonally extrapolated surface from Dektak line scan for sample 004 b) top-down view of orthogonally extrapolated surface from Dektak line scan for sample 004 68

Figure 30: a) artificial_surf generated surface using Fourier method from Dektak line scan for sample 001 b) top-down view of artificial_surf generated surface using Fourier method from Dektak line scan for sample 001 71

Figure 31: a) artificial_surf generated surface using roughness length method from Dektak line scan for sample 001 b) top-down view of artificial_surf generated surface using roughness length method from Dektak line scan for sample 001 72

Figure 32: a) SWLI area scan sample 001 b) surface extrapolated from Dektak line scan of sample 001 using Fourier orthogonal spectrum method..... 74

Figure 33: a) SWLI area scan sample 001 b) surface extrapolated from SWLI line profile of sample 001 using Fourier orthogonal spectrum method..... 75

Figure 34: SWLI scan of sample 001 b) fractal artificial surface generated surface using Fourier method from Dektak line scan for sample 001 c) fractal artificial surface generated surface using roughness length method from Dektak line scan for sample 001 77

Figure 35: SWLI scan of sample 001 b) fractal artificial surface generated surface using Fourier method from SWLI line profile for sample 001 c) fractal artificial surface generated surface using roughness length method from SWLI line profile for sample 001 78

Figure 36: Average Roughness, CSP measured (Ra) & orthogonal (Sa) vs SWLI measured (Sa)81

Figure 37: Valley Depth, Dektak measured (Rv) & orthogonal (Sv) vs SWLI measured (Sv).... 82

Figure 38: Total Valley Depth, Dektak measured (Rvk*) & orthogonal (Svk*) vs SWLI measured (Svk*)..... 83

Figure 39: Average Roughness, SWLI measured (Ra) & orthogonal (Sa) vs SWLI measured (Sa) 84

Figure 40: Valley Depth, SWLI measured (Rv) & orthogonal (Sv) vs SWLI measured (Sv)..... 85

Figure 41: Total Valley Depth, SWLI measured (Rvk*) & orthogonal (Svk*) vs SWLI measured (Svk*)..... 86

Figure 42: a) Fractal Dimensions b) Hurst Components 87

Figure 43: RMS roughness for Dektak line scan and SWLI line profile 88

Figure 44: Average Roughness, Dektak measured (Ra) & Fourier generated (Sa) & roughness length generated (Sa) vs SWLI measured (Sa) 89

Figure 45: Maximum Valley Depth, Dektak measured (Rv) & Fourier generated (Sv) & roughness length generated (Sv) vs SWLI measured (Sv) 90

Figure 46: Total Valley Depth, Dektak measured (Rvk*) & Fourier generated (Svk*) & roughness length generated (Svk*) vs SWLI measured (Svk*) 91

Figure 47: Average Roughness, SWLI measured (Ra) & Fourier generated (Sa) & roughness length generated (Sa) vs SWLI measured (Sa) 92

Figure 48: Maximum Valley Depth, SWLI measured (Rv) & Fourier generated (Sv) & roughness length generated (Sv) vs SWLI measured (Sv)..... 93

Figure 49: Total Valley Depth, SWLI measured (Rvk*) & Fourier generated (Svk*) & roughness length generated (Svk*) vs SWLI measured (Svk*)..... 94

Figure 50: a) skewness parameter surfaces from CSP profile b) kurtosis parameter surfaces from CSP profile 96

Figure 51: a) skewness parameter surface from SWLI profile b) kurtosis parameter surface from SWLI profile 98

Chapter 1: Background and Introduction

Measuring surfaces of metal components can often be used as a method of quality control in industry to ensure components are produced according to their design specifications. These measurements are often performed with cost as a factor heavily considered. The general goal is to acquire as much valid surface information as possible for as little incurred cost as permissible. This results in many incredibly effective measurement techniques being rendered unused in industry due to practicality despite providing insightful data and detailed information. It is imperative to utilize lower-cost measurement techniques to their fullest potential to ensure that surface information and parameters accurately represent the specimens they are tasked with characterizing.

Efforts to minimize costs have propelled additive manufacturing into many industrial sectors including but not limited to automotive, aerospace, and medical applications (1). The utility for rapid prototyping was immediately evident with polymer additive manufacturing. Once a larger variety of feedstocks were introduced and metal additive manufacturing technologies developed, more interest grew in fabricating working components not just prototypes. The ability to manufacture components directly rather than prototypes that would then be manufactured by another means using another stock material opened new avenues of design. Assembled components that were made previously with subtractive manufacturing individually could be combined into one complex additive component. Conventionally complex structures could be directly printed, such as programmable valves and fluidic devices (2). With the opportunity for simpler, more direct designs, the desire to ensure that as-built specimens directly off the build plate had comparable material quality and properties with respect to their conventionally manufactured counterparts grew. The benefits of additive manufacturing are not as profound if additional

subtractive steps are needed to allow the components to be utilized. Therefore, understanding metal as-built components and their surface quality is important in order to determine if these components can be utilized in demanding environments and use cases.

1.1 General material and printing

There are many metals that have been used as feedstock for metal 3D printing processes (3). When using laser-based printing methods that rely on a powder bed, the feedstock can even be recycled through multiple uses (4). This flexibility allows for materials commonly used in subtractive manufacturing processes to be adopted in additive processes as well. This creates more consistency when comparing traditionally manufactured components with new additive components.

Many additively printed components are rough in nature due to their surfaces being affected by the printing process. These rough surfaces result from many factors such as stair-stepping, splattering, unstable melt pools, and partially melted powder particles (5). These rough surfaces have an adverse effect on component fatigue life (6–8). Evaluating fatigue life directed the design of the components measured and evaluated in this study.

1.1.1 Ti-6Al-4V

One of the most commonly utilized Titanium alloys, Ti-6Al-4V was utilized in this study for industrial relevancy. This is an alloy that is being used in a variety of applications due to its resilience in a variety of environments and its cost of desirable traits (9). Ti-6Al-4V (known as

Ti64) is an α+β titanium alloy with higher mechanical properties, low density, high fracture toughness, biocompatibility and superior corrosion resistance and being very widely used in the aeronautical, automotive, chemical, and medical industries (10–12). The chemical composition of Ti-6Al-4V powder produced by AP&C used in this study is listed in Table 1. The specimens were fabricated by an EOS M290, a laser-based powder bed fusion (L-PBF) system, using default process parameters provided by the manufacturer. The details of process parameters are listed in Table 2.

Table 1: Chemical composition of the Ti-6Al-4V alloy provided by AP&C.

Elements	Al	V	Fe	O	C	N	H	Y	Ti
Wt. %	6.31	3.96	0.20	0.14	0.01	0.01	0.002	< 0.001	Bal

Table 2: EOS default process parameters for Ti-6Al-4V

Laser power	Scanning speed	Hatching distance	Layer thickness	Layer rotation angle
280 W	1200 mm/s	0.14 mm	30 µm	67°

1.1.2 Additive printing process and general fatigue background

Metal additive technologies are rapidly being integrated into a variety of industries and provide unique advantages to traditional methods (1,13–16). Traditional subtractive manufacturing techniques impose restrictions on the design of components as the surfaces need to be accessed by tools. Additive manufacturing can bypass such restrictions by creating contours and features without the need for additional material removal. This is accomplished through a 'ground-up'

method of fabrication whereby material is added layer by layer until a completed component exists(17). The previous advantage is largely applicable if the as-built surfaces are within desirable tolerances. This restriction demonstrates the importance of surface roughness and surface roughness measurement techniques. It is imperative that surfaces can be understood through available measurement techniques in order to ensure components will operate as intended. The presence of surface and near-surface defects, such as porosity, inclusions, and microcracks, is a significant contributing factor to the fatigue life of metal additively manufactured parts (8,18,19).

To better understand these metal additive surfaces, it is important to understand the methods by which they are created. The surfaces measured and explored in this work were formed by laser beam powder bed fusion (LB_PBF) (20). This method of manufacturing uses a focused laser beam to melt metallic powder. The powder is deposited over the surface of a build plate (i.e. base plate, substrate) (17) and the laser melts the powder according to the geometry of the desired part, to create a single layer of metallic particles fused together. The powder is then deposited over the surface and the process repeats iteratively until the component has been created. As with any manufacturing method there are some drawbacks including but not limited to, the need for support structures, the complexity of the process, and the associated costs. The roughness of resulting parts greatly depends on the thermal history of the material during the building process. Many independent variables impact the thermal history of these parts, and it is hard to accurately predict the microstructure of additively manufactured components given only knowledge of feedstock properties and print process parameters (21). While many machine conditions are known and deliberately tuned for the print process, material properties and environmental conditions can introduce unknown variables to the process which can affect things like local solidification rates which greatly impact the final product of the printing process. The parts pertinent to this work are

as-built components, so the vast arrays of post-processing techniques will not be explored. Instead, byproducts of the printing process such as partially-melted powder particles affixed to the part's surface will be accounted for in the measurements performed, which attempt to characterize the complex surfaces of the as-built specimens (22). These complex surfaces have been shown to greatly impact the fatigue life of the specimens. Fatigue strength is a primary interest, as cyclical loading often induces failure in additive parts. This failure has been shown to occur due to sharp deep valleys in the surface, which can be introduced by many factors in the printing process such as (but not limited to) melt pool cooling rates and the presence or lack of presence of metal particles (6,7).

1.1.3 XCT coupon design

A batch of samples resembling geometry of ASTM standardized fatigue testing specimens were fabricated using laser beam powder bed fusion (LBPF) method. The printed sample has both a rectangular and cylindrical portion. The rectangular portion is 15mm x 6 mm (L x W). The cylindrical portion has the same diameter (5mm) as the gauge section of the standard fatigue test specimen. The reason behind this is to maintain all the surface features present in a fatigue specimen so that a direct comparison can be drawn in terms of surface parameters. The top and front views of the specimens with their dimensions have been illustrated in figure 1 a). A batch of 81 samples with varying print parameters was printed and among them, 14 samples will be analyzed in this work comprising a wide combination of process parameters (laser power, scan speed, infill, layer height) to involve variability in surface features. Figure 1 b) shows the schematic of the print bed with the orientation of the specimens along with argon gas flow and recoater

directions. The design of experiment (DOE) attempts to ensure that a variety of print conditions can be characterized.

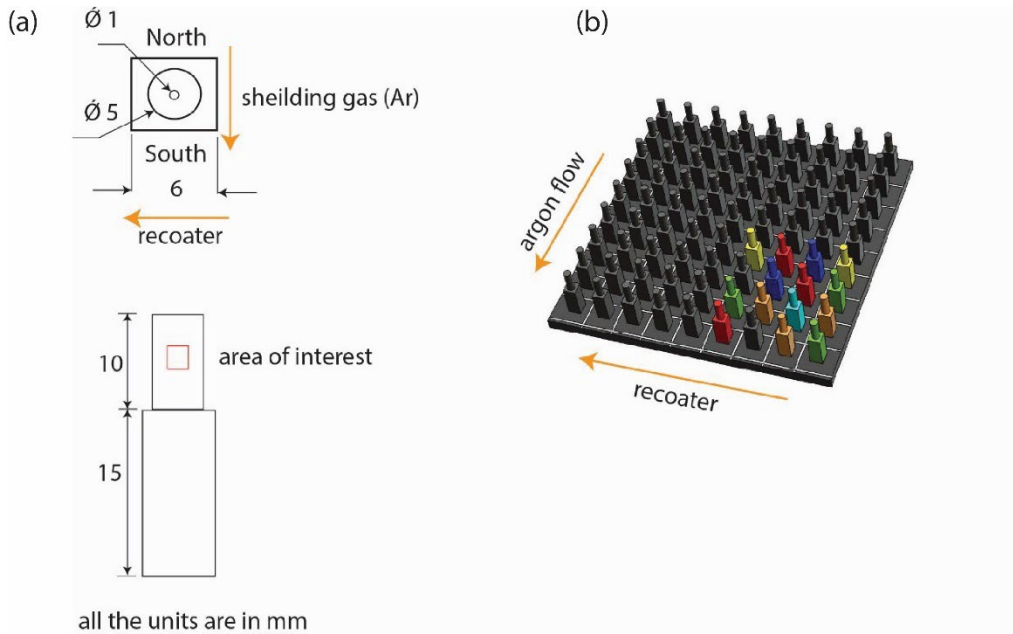

Figure 1: a) Ti-6Al-4V specimen geometry with the area of interest and b) schematic of specimen in the build platform

Different print parameters correspond with different colored specimens. Table 2 shows the print parameters for the specimens. The preliminary region of interest (ROI) was shown in figure 1(a) with an effort to capture the surface feature in a cylindrical surface allowing comparison between different methodologies when mapping surfaces from a region. In order to ensure consistent comparisons, the north sides (direction of gas flow) of the specimens were chosen to be measured.

The specimens are either described using their DOE numbers or their numbering 1-14. This is outlined in Table 3.

Table 3: The numbering of specimens and their corresponding DOE designation numbers

Specimen Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DOE Specimen	001	002	003	004	009	010	016	017	018	024	025	031	032	033

The samples were kept as-built to preserve the native characteristics of the topographic features generated from the printing process. The figure below shows an image of the coupon where the topography is visible to the eye.

Figure 2: XCT coupon specimen photo

1.2 Roughness, Waviness and Form

Now that the specimens have been outlined, the roughness information must be extracted and analyzed. It is important to isolate relevant roughness data from other height data brought in from any other feature than asperities under consideration. When removing anything from surface data, overfitting becomes a concern, so it is imperative that only known form and waviness features be removed to retain the integrity of the surface data. If an insufficient amount of data is removed,

the asperities could be exaggerated or muted by factors existing outside the realm of consideration. If too much data is removed, the asperities could be removed or muted by the data removal. Engineering surfaces can be separated into three main groups of data: roughness, waviness, and form. Form is relegated to the macro-scale shape of the object being analyzed and exists on a much larger scale than the roughness profile being evaluated. For example, the cylindrical surface that was outlined in the previous section could have the curvature of the cylinder removed from the measurements in order to accurately report the roughness of the curved surface. The differences between surface roughness and form are generally well understood, however the distinction between waviness and roughness is far more ambiguous (23,24). The image below conveys a more simplistic approach to understanding a manner by which roughness can be separated from waviness.

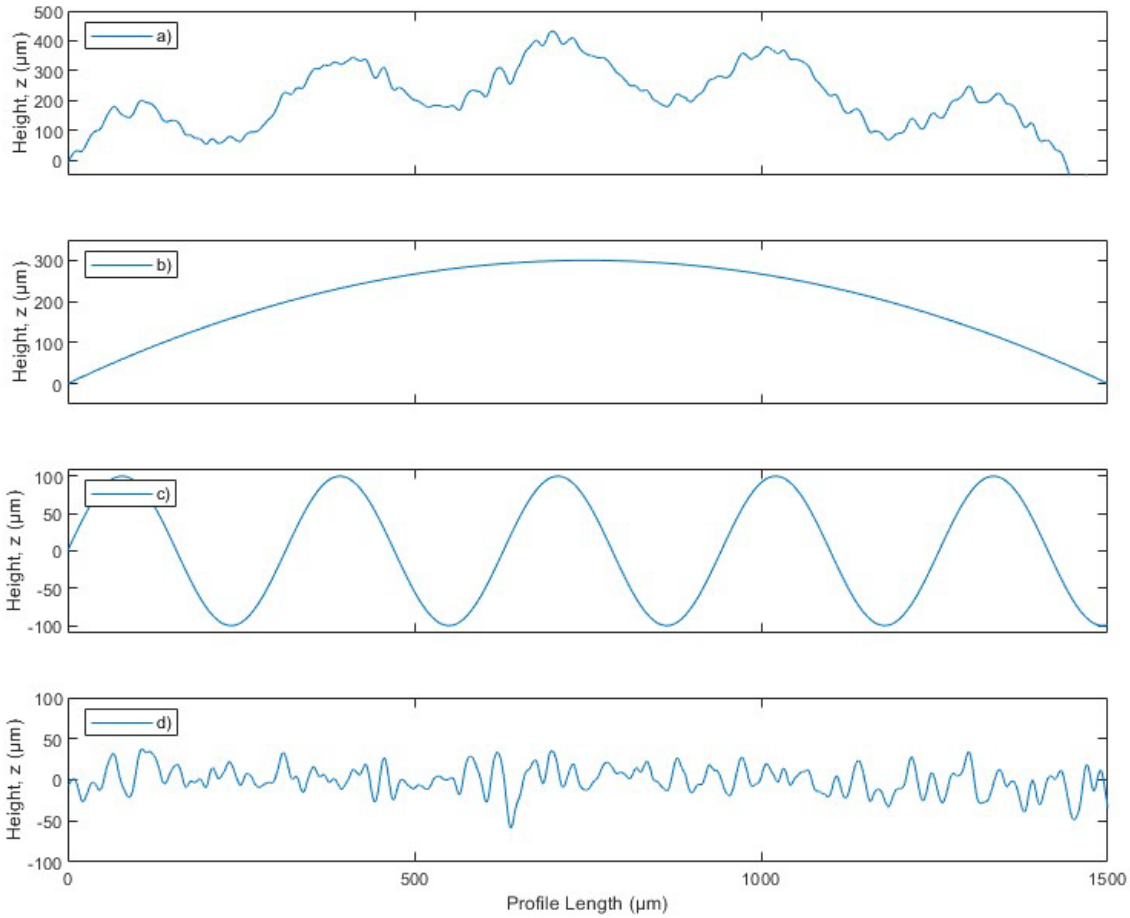

Figure 3: a) raw combined surface profile b) form c) waviness d) surface roughness

More often than not, measured surfaces do not often exhibit a single sin wave that can be easily distinguished as waviness. This results in user discretion determining the number of curves considered for removal when correcting a measurement to eliminate waviness. The line drawn between roughness and waviness often is delineated based on the size of the features of interest. In this case, the ambiguity introduced through the printing process results in difficulties in selecting cutoff wavelengths for filtering the data, as simple particle size cannot be employed as a cutoff.

1.3 Objectives

Using a line measurement technique such as stylus profilometry provides useful global roughness parameters and information regarding the surface texture that is important to understand when evaluating metal additive specimens. The ability to extrapolate additional significant information from a limited dataset allows for a deeper understanding of rough surfaces without additional investment into the physical measurement process. Critical features, or rather features that introduce an avenue for failure or otherwise impact the behavior of components are ideally encountered by the selected measurement technique. When considering a line scan method such as a stylus profilometer, the limitations of a line scan must be acknowledged. A line drawn across a surface will not reliably interact with critical features and cannot realistically be considered as an absolute evaluator of extreme values, or maximum valley depth or peak height of a surface. Stylus profilometers can reliably report height values and generally possess finely tuned vertical resolutions on modern equipment. These accurate height measurements are only useful if they encounter features and asperities that require consideration. Even if the line scan encounters an extreme feature such as a valley, it is not certain that the stylus will encounter the deepest point of that valley, and this is especially true with rough as-built additive samples that have random and irregular surface features.

A practical approach to surface topography investigations is a case dependent endeavor, as surfaces vary widely with respect to the critical features they possess (24). The stylus profilometer is readily employed in industry, and is a well-established method of surface evaluation (25–27). Many methods of area measurement and even some methods of volumetric measurement offer distinct advantages with respect to the amount of area covered, but largely incur higher costs to accomplish this. The ability to derive information using numerical, parameter driven extrapolation

techniques can assist in estimating extreme values that could exist on a surface outside the measured features detected by a contact profilometer. For a particular length scale, spectral analysis techniques can also provide valuable insight into the potential for various height features to appear in a surface. These data exploration techniques can allow a single line scan to compete with area scan measurement techniques in terms of efficiently characterizing a surface. The subsequent sections aim to establish the procedures and methods used to obtain line and area measurements of surfaces, and then explore techniques to develop new parameters and metrics from the line scans that approximate the known area scan values and parameters. Comparisons will be drawn with lines taken directly from the area scans and with line scans obtained from different equipment with different working principles to ensure that comparisons between the line and area measurements are not solely valid with specific equipment and scan parameters.

Chapter 2: Data Collection and post-processing

Selecting a measurement technique and equipment to analyze a surface requires some knowledge about the surface being measured. Understanding the amount of the surface that needs to be characterized is important, and understanding the vertical range of your measurement is also important. These can limit which techniques and equipment will be practical to use. Methods such as X-ray computed tomography (XCT) can provide not only data surrounding the internal structure of components but also their surface topography providing that the voxel size is small enough to capture surface asperities and features. In-depth techniques such as this provide a lot of information about surfaces, but are not practical for most industry use due to their cost and scale limitations (28–30). Given the emphasis on measuring surfaces rather than sub-surface features, optical measurement and contact profilometry were prioritized in this study. Many popular methods for measuring surfaces in great detail at the nanometer scale lack the vertical range to accommodate for rough metal additive surfaces. Many additive surfaces measured in this study have average roughness values on the order of tens of micrometers. These high roughness parameter values require equipment with a relatively large height value measurement range to accommodate the cylindrical form.

Once surface areal scans and line profile scans are obtained from measurement equipment, the raw datasets must be contextualized and analyzed. For the additive Ti6Al4V specimens utilized in this study, fatigue life is an important characteristic to understand. Surface roughness has been found to greatly influence fatigue life (6–8,18,19). To parametrize surface roughness, the roughness must be separated from intended defined features of a component. The use case for a

measurement often determines what aspects of a surface are considered to be roughness. Typically, this is how much a surface deviates from its macro-scale features.

2.1 Measurement methods overview (working principles)

The size of asperities and features under consideration can in some cases dictate the measurement techniques capable of being employed, as different working principles bring their own inherent limitations to studies. Surface roughness has many definitions that are largely case-dependent (31,32). It can generally be said that surface roughness is the degree to which a surface deviates from its macro-scale shape. The figure below provides some insight into which methods can assist with measuring which aspects of surface metrology.

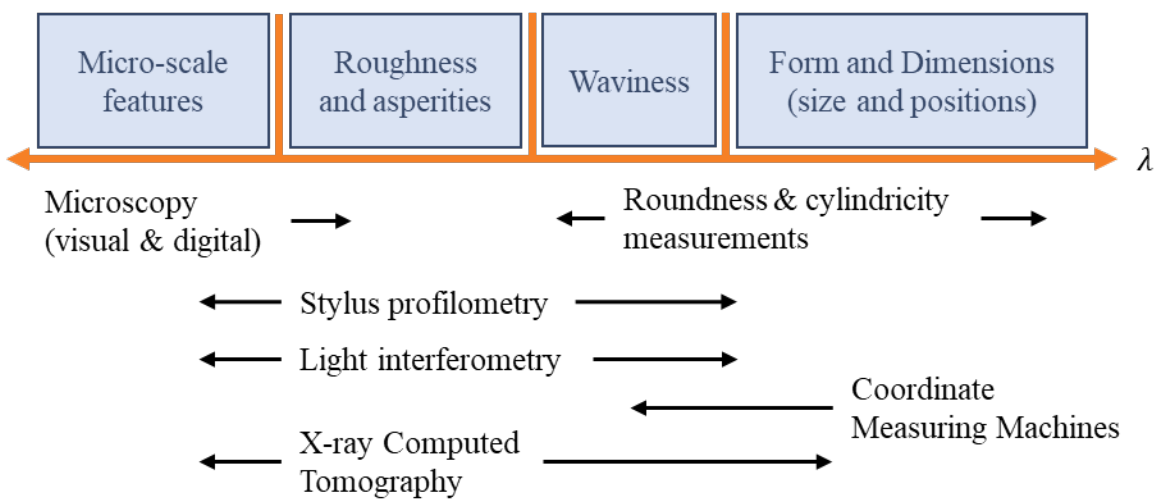

Figure 4: Metrology measurement spectrum

When selecting measurement techniques to be employed in the study, advantages and disadvantages of different equipment can be acknowledged so that surfaces can be accurately

represented by the data obtained. The specimens being measured are as-built rough metal pieces with a cylindrical shape. Given that the measurements occur on the cylindrical portion of the specimen, measurement techniques must be able to span the range of roughness, up to form (33). This ensures that form and waviness can be removed and accounted for in the measurement if it is relevant, leaving only the roughness data. The presence of a cylindrical surface introduces challenges in measurement, as techniques must be able to not only have a large enough vertical range to accommodate for curvature, but also have a fine enough resolution to detect relevant surface asperities and features.

2.1.1 Line measurements

The line profile measurements were obtained using a Veeco Dektak 150 profilometer. This contact stylus profilometer (CSP) is pictured in figure 5. In this study Dektak and CSP are used interchangeably to represent this measurement technique.

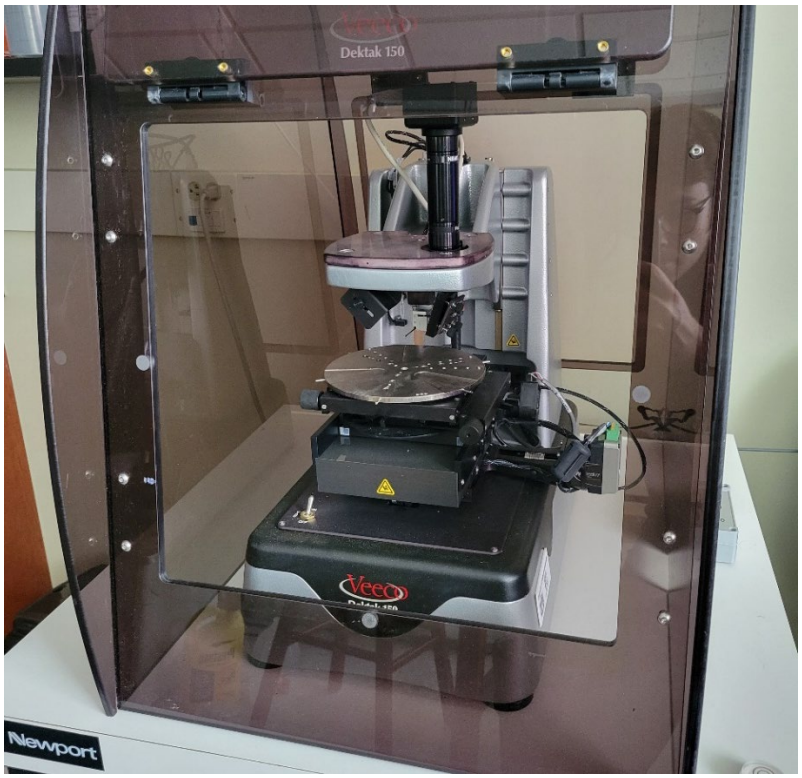

Figure 5: Veeco Dektak 150 contact profilometer

The stylus contact profilometer provides line profiles of a surface, and the working principle is shown in Figure 6.

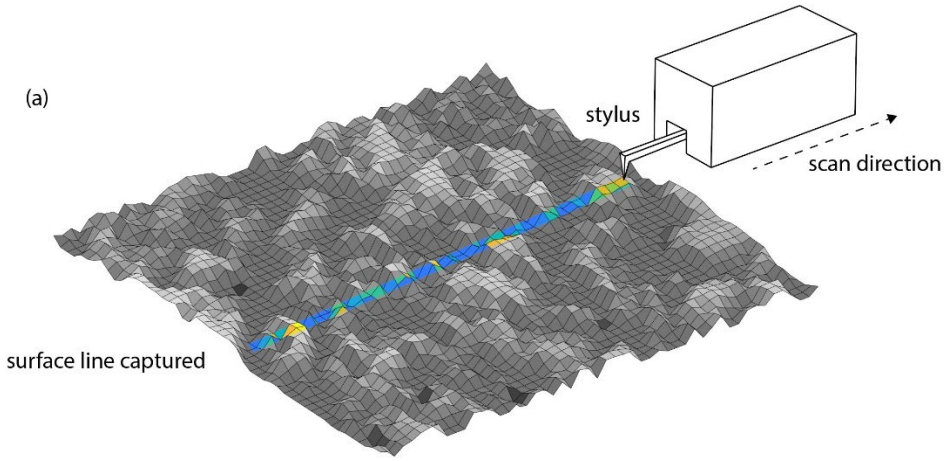

Figure 6: a) contact profilometry working principle

The data utilized in subsequent sections from the Dektak 150 profilometer was obtained with a single line scan per specimen. This device has a 524 µm vertical range and has been set to conduct line scans of a length of 3 mm. The stylus tip is a 2 µm tip radius. Other stylus tip sizes can be selected for measurements of an even finer resolution. The tip size acts as the lateral resolution of the measurement, as the tip radius determines how much of a surface the stylus interacts with, and the minimum space that can be allowed between asperities for detection. The stylus tip size also determines how deep it can travel into narrow valleys, and if it can travel into valleys at all. The vertical resolution of the Dektak is approximately 1 nanometer. The line scans conducted are on the cylindrical portion of the coupon specimens along the principal axis of the specimen. This line consists of a vector of 4800 datapoints of raw height values representing the detected surface profile.

2.1.2 Area measurements

The scanning white light interferometer (SWLI) can be used to extract area scans of a surface, and is shown in Figure 7. This machine is a benchtop system non-contacting surface metrology measurements.

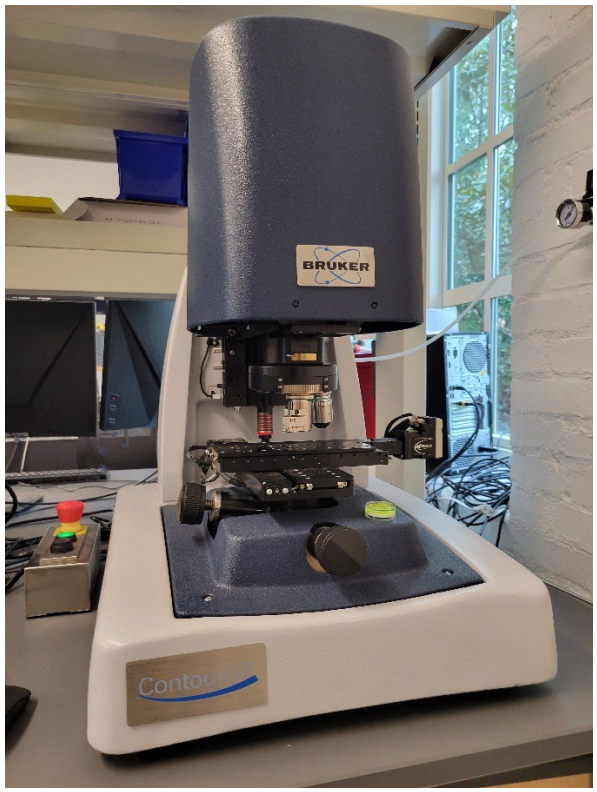

Figure 7: Bruker scanning white light interferometer (SWLI)

This method of surface capture falls under the umbrella of optical measurement techniques and utilizes light wave interference to assess surfaces. The working principle is shown below in figure 8.

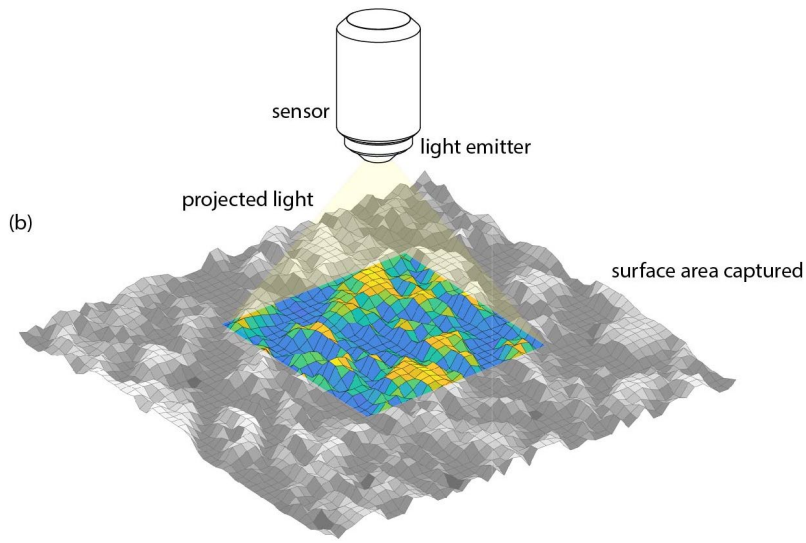

Figure 8: b) optical profilometry scanning white light interferometry working principle

White and green light waves can be bounced off a surface, and the interference between the waves provides insight into how far away the surface is from the sensor in the interferometer. This height distance can be mapped over a surface and stitched together to provide a full surface height scan. This method of surface evaluation can incur difficulties when measuring surfaces with sharp asperities, and steep features that would reflect the light waves away from the sensor in the unit (34). Coatings or surfaces that do not reflect light well also introduce challenges for this technique, as the light must be emitted, reflected off the surface, and detected by the sensor for valid height readings and measurements.

The data utilized in subsequent sections was obtained using SWLI. The scans covered a 1.5 mm by 1.5 mm area from the north side of the XCT coupons. The pixel size is 0.489687 μm/pixel. The data is provided in a .ASC format that is reshaped from 3 columns into a data matrix of size 3063 x 3068 data points.

2.2 Data collection (parameters)

2.2.1 Surface roughness line data parameters (CSP)

The data utilized in subsequent sections from the Dektak 150 profilometer can be used to calculate line roughness parameters. This is a contact stylus profilometer (CSP). These parameters are calculated from line profiles containing height values at defined *x* coordinate positions (35–37). These parameters are commonly used to characterize surfaces and provide a general understanding of the surface topography. Understanding the extreme values of a height value dataset can dictate which measurement ranges are required to accurately characterize a surface. The extreme values are calculated as a distance between the extrema of a feature and the mean line of the surface profile. The calculation of this mean value directly contributes to the extrema value calculations as a result. It is imperative that the mean calculations remain consistent. If the scale of the measurement is sufficiently small, even a single asperity can shift the mean line, and thus the extrema calculations. The illustration for obtaining these extreme peak heights and valley depths is shown in figure 9 below.

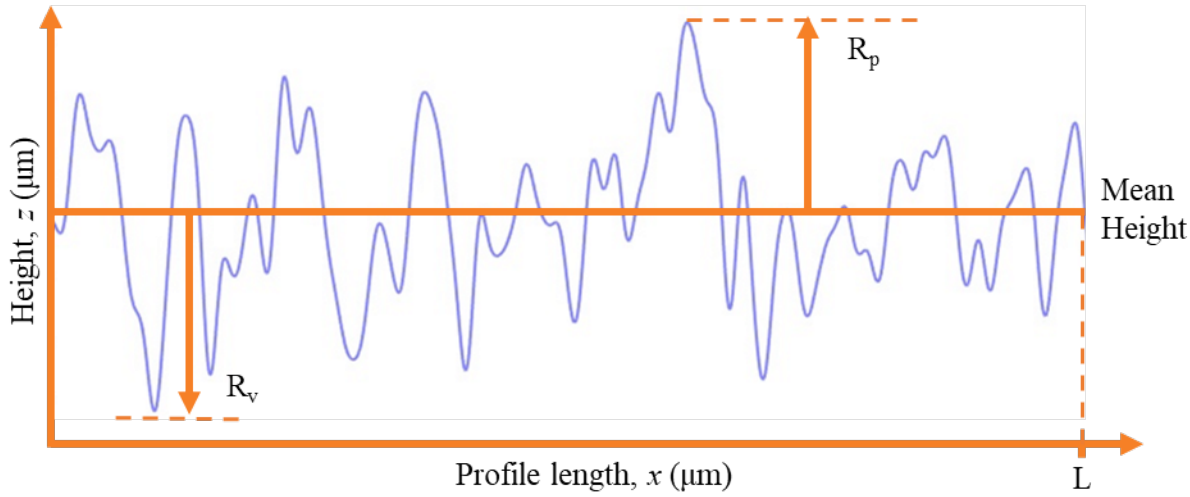

Figure 9: Maximum profile peak height (R_p) & maximum profile valley depth (R_v)

Once the maximum peak height and maximum valley depth values have been obtained, a total height of the surface can be extracted using the equation below.

$$R_z = R_p + |R_v| \tag{1}$$

The extreme values are not the only measure of a surface. In many cases, a more global parameter can provide insight into a surface compared to local extremes such as the maxima. An example absolute value roughness of a surface is shown below in figure 10 and demonstrates how far on average the features deviate from the mean line of the dataset. This distance is calculated from the absolute value of the profile, and as such the absolute value of the profile from figure 9 is shown below in figure 10.

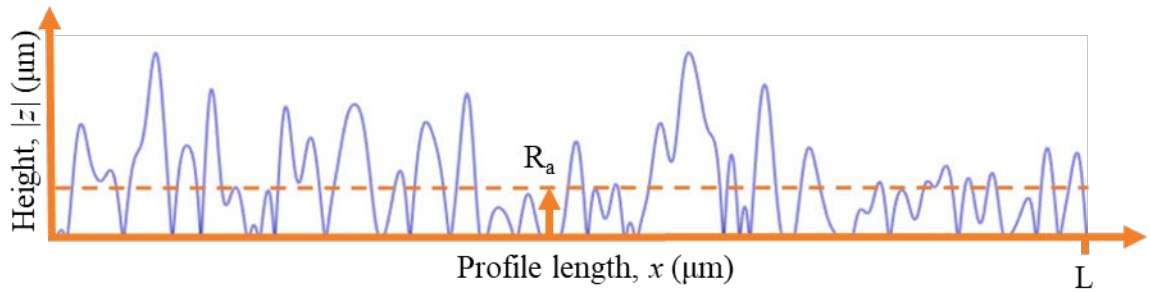

Figure 10: Arithmetic mean deviation (average line roughness - R_a)

The equation to calculate the average line roughness of a surface profile is shown below.

$$R_a = \frac{1}{L}\int_0^L |z(x)|dx \tag{2}$$

Another global surface profile parameter is the root mean square roughness (RMS roughness) which is a more sensitive, but comparable calculation to the average line roughness. This parameter is shown below in figure 11.

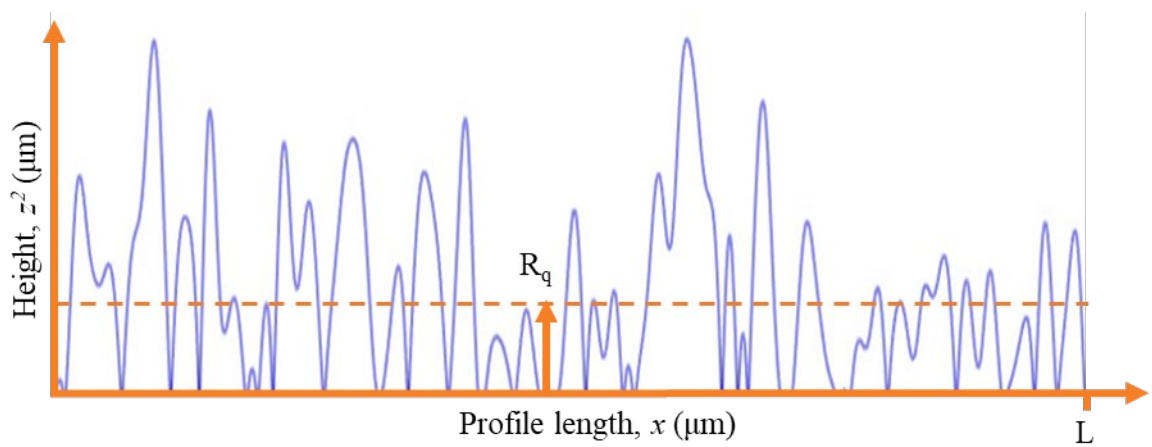

Figure 11: Root mean square deviation (RMS line roughness – R_q)

The RMS roughness measurement is also a prolific choice for characterizing rough surfaces. This parameter is widely used with a variety of rough surfaces (31). This parameter

represents the standard deviation of the surface heights, representing variability of height datapoints from the mean of the dataset. The equation for RMS roughness is shown below.

$$R_q = \sigma = \sqrt{\frac{1}{L}\int_0^L |z^2(x)|dx} \tag{3}$$

It can be beneficial to analyze the distribution of a *z* value height dataset to understand the general trends of the surface. The parameters defined above are extracted from height values in the *z* coordinate plotted with respect to their position on an *x* coordinate line. The resulting profile often has the same height value present at multiple places along the profile. The same dataset can be fit to a distribution where the height values ranging from the minimum to the maximum are the *x* axis of the plot, and the probability density is the *y* axis of the plot. The probability density conveys the likelihood of a point occurring for a distribution. The area under the distribution shown is equal to 1, indicating the total of the possible outcomes. An example Gaussian distribution is shown below in figure 12.

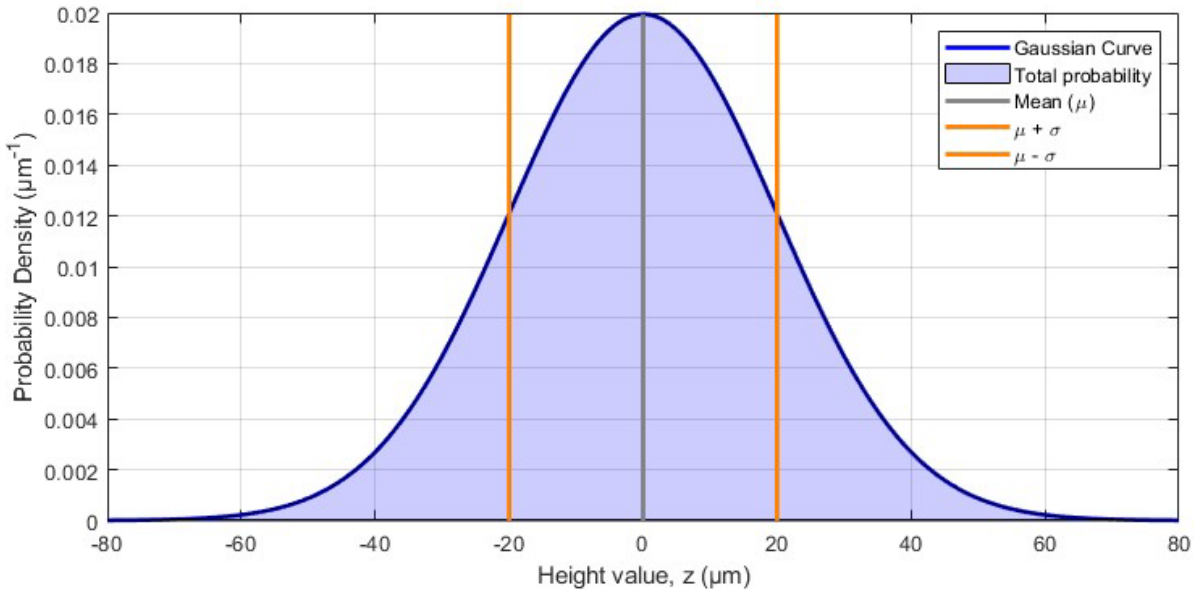

Figure 12: Gaussian distribution with standard deviations shown

The above curve is shown to represent data with an RMS roughness of 20 µm, which means the data has a standard deviation of 20 µm signified by σ. The mean of the dataset for this distribution is 0. The distribution shown above is normal and symmetrical. This is not the case for all distributions of measured datasets. Another way to represent a distribution is through a histogram plot, which has an x axis similar to the gaussian distribution above, however the data is divided into bins. The y axis is the number of datapoints that fall into the respective bins. An example histogram from a single Dektak line scan of sample 001 is shown in figure 13 below.

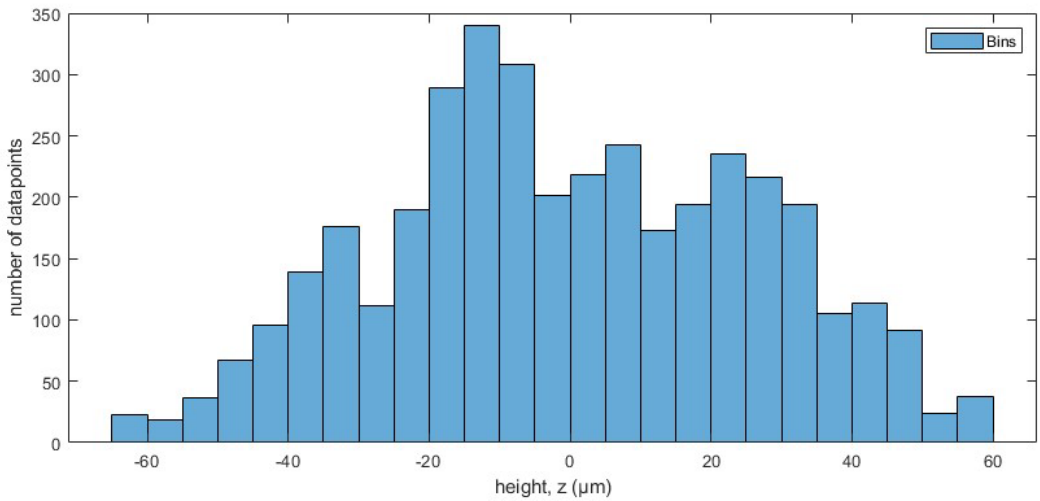

Figure 13: histogram of sample 001 with 25 bins

The presence of a large number of peaks or valleys, or the presence of large peaks or valleys can be detected through the skewness parameter. This parameter provides insight into the general presence of large or small *z* values in the dataset. Skewness is illustrated in figure 14 below.

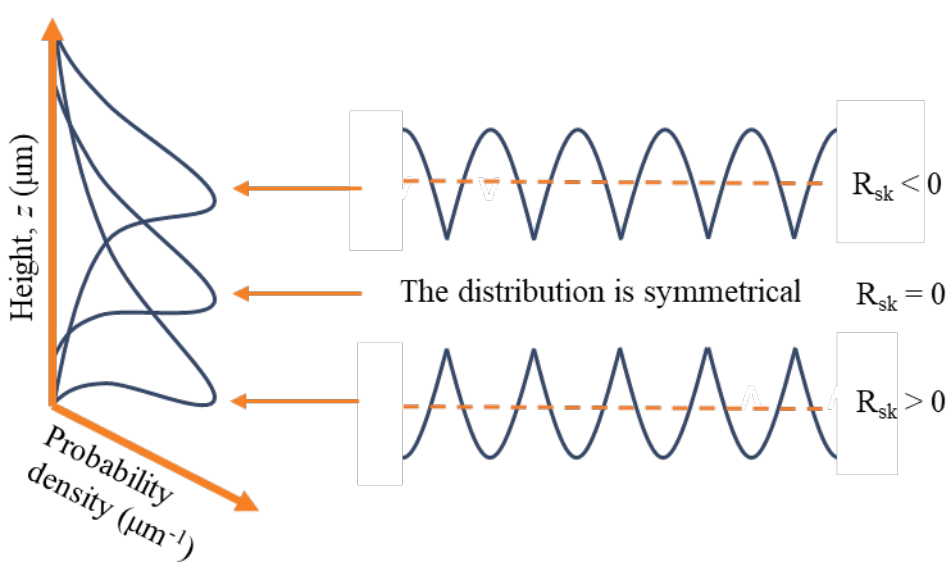

Figure 14: Skewness (R_{sk})

The skewness of a height dataset is based on the height values to the third power and the inverse of the RMS roughness to the third power. The necessary equation is shown below.

$$R_{sk} = \frac{1}{R_q^3}\left[\frac{1}{L}\int_0^L z^3(x)dx\right] \tag{4}$$

The thickness in shape of a height distribution (also referred to as sharpness or bluntness) can often be as insightful as the tendency to lean to the positive or negative direction. This measure is referred to as kurtosis and can illustrate how much of a dataset is near the mean line. The "skinnier" a distribution, the more datapoints are near the mean line. The "fatter" a distribution, the more datapoints are near the extreme values of the dataset, or the more extreme the values of the dataset are. Kurtosis is shown below in figure 15.

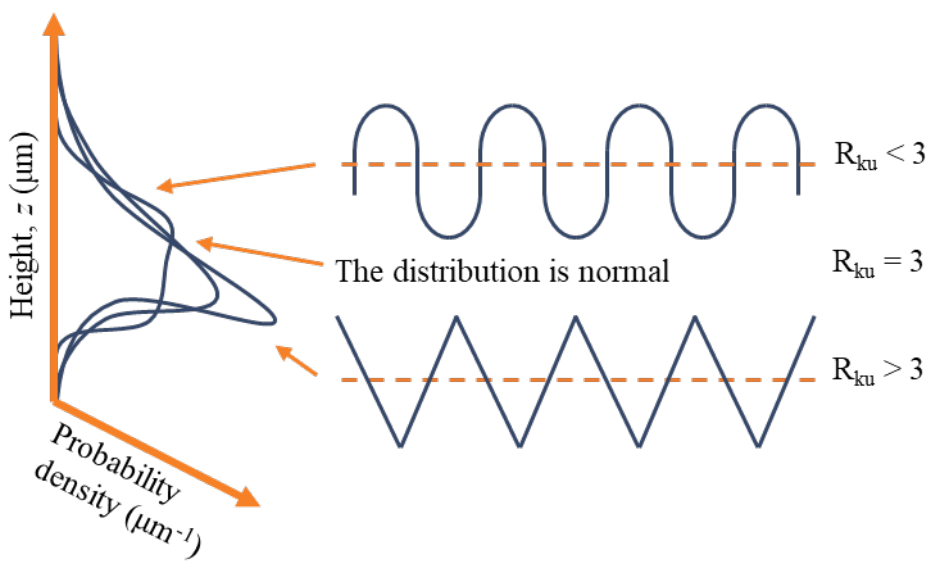

Figure 15: Kurtosis (R_{ku})

For kurtosis values less than 3, the asperities and valley features are typically smoother features, with a larger amount of comparable datapoints surrounding the extreme values. For a Kurtosis

value greater than 3, a larger portion of the datapoints are tails. These thinner distributions have a greater tendency for extrema to be present. For low kurtosis surfaces, the distributions have lighter tails. These low kurtosis surfaces have fewer extrema. The equation for kurtosis depends on the inverse of the RMS roughness to the fourth power, and the height dataset to the fourth power. The equation for kurtosis is shown below.

$$R_{ku} = \frac{1}{R_q^4}\left[\frac{1}{L}\int_0^L z^4(x)dx\right] \tag{5}$$

2.2.2 Surface roughness area data parameters (SWLI)

The data utilized in subsequent sections from the Bruker SWLI can be used to calculate areal roughness parameters. These parameters are calculated from area surfaces containing height values at defined x and y coordinate positions (35). These areal parameters have calculations very similar to the surface profile parameters,

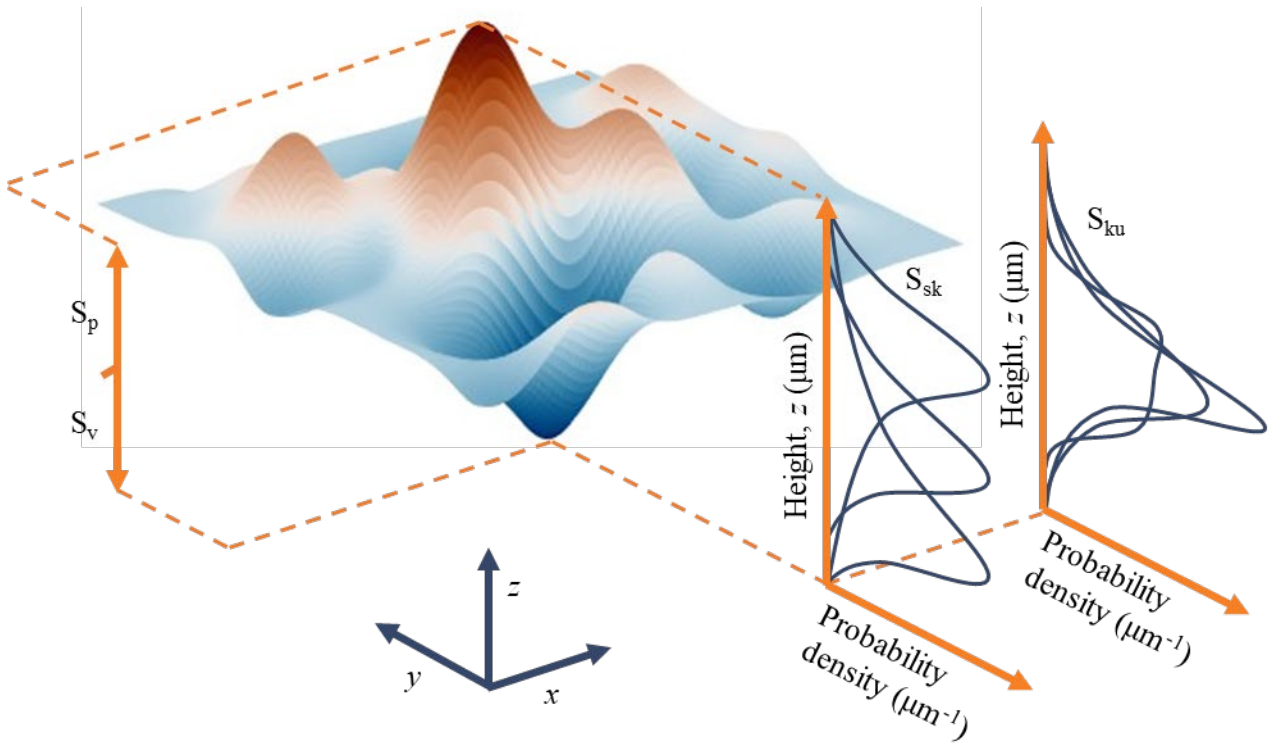

Figure 16: Maximum areal valley depth (S_v), maximum areal peak height (S_p), maximum areal height (S_z), areal distribution parameters, skewness (S_{sk}) and kurtosis (S_{ku})

In comparable fashion the total height of the areal surface can be determined by the extreme values in the positive z and negative z directions. Instead of considering a single line profile, an entire surface dataset is considered, and the single maxima points are extracted from a two-dimensional area dataset rather than a one-dimensional line dataset. By considering multiple line profiles at once through the analysis of an area measurement, the likelihood of extreme values with high degrees of variability from the mean line/plane increases compared to a single line profile (38). The equation for the total height of a surface is shown in the equation below.

$$S_z = S_p + |S_v| \tag{6}$$

Like the line profile measurements, there is also an areal average roughness parameter, which is calculated the same way, but considers a second orthogonal dimension to be integrated across. This is shown in the equation below.

$$S_a = \frac{1}{A}\int_0^y \int_0^x |z(x,y)|\, dxdy \tag{7}$$

The ability to calculate a standard deviation parameter is useful for analyzing distributions and understanding surface trends, and the equation below shows the RMS areal roughness calculation.

$$S_q = \sqrt{\frac{1}{A}\int_0^y \int_0^x |z^2(x,y)|\, dxdy} \tag{8}$$

The same methods for creating distributions from line profiles can be applied to area scans as well. These are still taken with respect to a comparable range of height values, however now the number of datapoints represented in the histogram changes. Skewness and kurtosis are also present for areal measurements, and can be evaluated with the equations below.

$$S_{sk} = \frac{1}{S_q^3}\left[\frac{1}{A}\int_0^y \int_0^x z^3(x,y)\, dxdy\right] \tag{9}$$

$$S_{ku} = \frac{1}{S_q^4}\left[\frac{1}{A}\int_0^y \int_0^x z^4(x,y)\, dxdy\right] \tag{10}$$

2.3 Data Post-processing

The element of form can greatly influence roughness height values when the height information is not filtered. This is present in area scans, which even if aligned along the principal axis of the specimen, will still experience falloff as the cylindrical shape slopes downward. Form

often influences the height values on a scale much larger than the roughness, but this is not always the case. Knowing the specifications of the specimen geometry is beneficial when filtering this out of the raw height data. When specific geometries are unknown, approximations can be made if the scale and general shape of the form is known.

Eliminating waviness from measurements is a sensitive process. The distinction between waviness and roughness is largely subjective (23,39,40). Eliminating unwanted noise without over filtering data is a challenging process when the exact geometry and waviness is not known.

2.3.1 Line scan initial post-processing

The Dektak profilometer measures a surface profile by contacting the surface, and sliding along it in order to construct a vector of height values sampled at a consistent distance from each other. This requires the stylus to be moving at a constant speed across the surface. The profilometer builds in some distance to accelerate to a constant scan velocity, however since the built in distance is unknown, the datasets were curtailed in order to ensure that the datapoints were sampled at a consistent pace. As per the ISO 4287 standard, the measurements were conducted according to the delineated methods. After the raw data was captured, 10% of the datapoints were removed from each end of the line scan. This brought the 3 mm line consisting of 4800 datapoints down to a 2.4 mm line scan consisting of 3840 datapoints. This dataset has a resolution of 0.625×10^{-6} μm/datapoint. This is far below the lateral resolution of the stylus tip which has a 2 μm radius. However, the high resolution scan was kept due to the vertical resolution of the Dektak equipment which is 1 nm.

Assuming ideal alignment between the specimens and the path of the stylus, curvature should not affect the line scan measurements provided they travel along the center axis of the cylindrical portion of the coupon. Any slight misalignment with the sample could potentially introduce the effect of form into the height measurements. The specimens were manually aligned in the Dektak machine, meaning that good alignment is not guaranteed for contact measurements. Given the likelihood of the presence of form in the measurements, a 2nd order polynomial was fitted to each Dektak line scan. This fit curve is intended to account for the influence of crossing over the peak of the cylindrical portion of the specimen. The fit curve is subtracted from the raw dataset yielding the flattened dataset consisting solely of roughness features that are to be quantified and parameterized. This is shown in the following equations.

$$z_{poly2}(x) = p_0 + p_1 x + p_2 x^2 \tag{11}$$

$$z_{lineflat}(x) = z(x) - z_{poly2}(x) \tag{12}$$

This process is shown below in figure 17 using the Dektak line scan from specimen 004 as an example calculation.

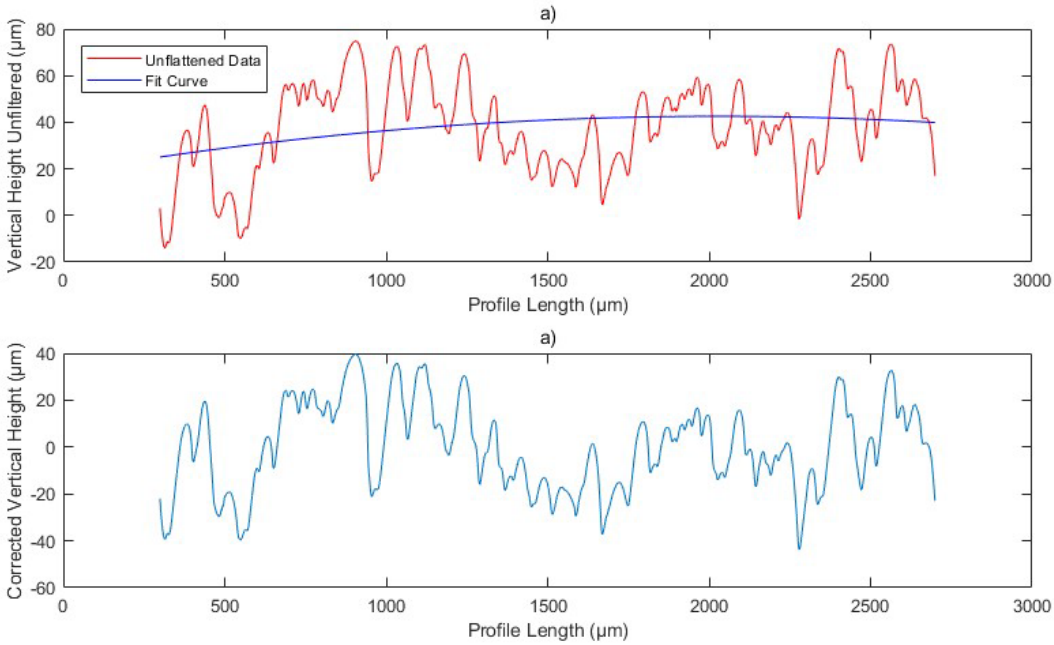

Figure 17: a) Raw profile measurement and fit curve sample 004 b) flattened line profile from sample 004

2.3.2 Area scan initial post-processing

The raw data from the SWLI can have data dropout where the light was not able to return to the sensor after bouncing off the surface. This can occur for a variety of reasons, including but not limited to steep asperity angles, nonreflective surface features, or shadows from asperities. The data was interpolated using a technique (Shown in Appendix X) that fills in empty datapoints with the average of the surrounding points and iterates through the space. This allows the data to be filled from the edges of missing sections inward towards the center of the section. The average values are stored, and the loop iterates until the data section is filled. This method ensures that for smaller areas or one-off data dropout instances, the filled data does not introduce features or points that are not consistent with the existing surface.

The curvature is removed from the raw SWLI measurement to produce a flattened surface. This method differs from the polynomial flattening method, which considers only a line profile of data. This flattening process for specimen 004 is depicted in figure 18.

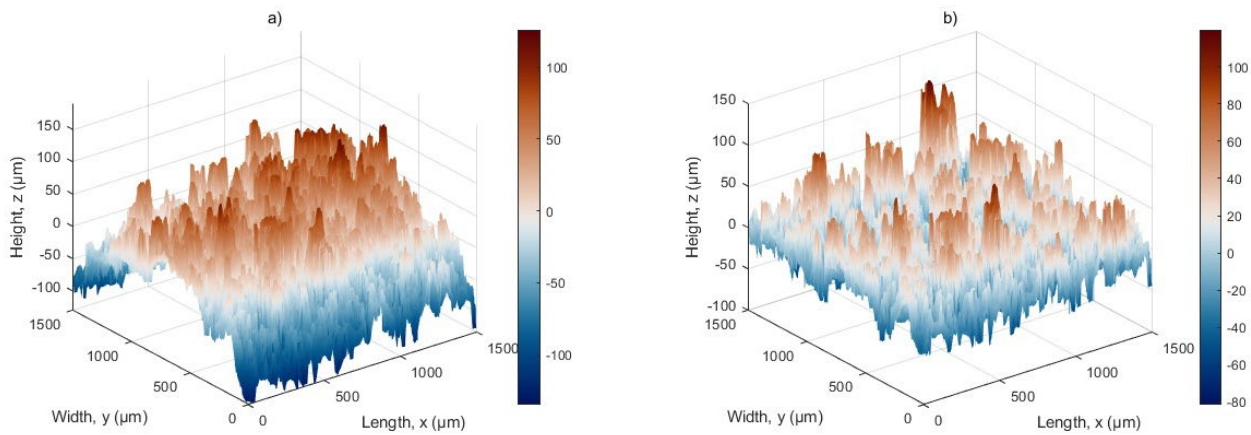

Figure 18: a) Unflattened SWLI raw surface scan sample 004 b) plane flattened SWLI surface 004

The natural curvature of the surface can be seen clearly in the scan as the cylindrical portion of the specimen was analyzed. In order to extract meaningful surface roughness parameters from the surface, it must be flattened so that desired features and asperities can be considered without large-scale form and geometry affecting height measurements. There are multiple methods for removing curvature from a surface, and below two such methods are explored and compared in the subsequent writing. Understanding how methods of curvature correction impact the respective datasets is imperative when comparing different raw datasets in differing formats. Single line scan

flattening is more similar to the line-by-line method than the holistic plane flattening. The first, shown in figure 18, generates a smoothed contour plane beneath the surface and then removes it to leave only the asperities and small-scale features behind. This smoothed contour plane is a second degree polynomial surface. The process is outlined below.

$$z_{poly22}(x, y) = p_{00} + p_{10}x + p_{01}y + p_{20}x^2 + p_{11}xy + p_{02}y^2 \tag{13}$$

$$z_{planeflat}(x, y) = z - z_{poly22}(x, y) \tag{14}$$

The above equations utilize p as the constant and polynomial coefficients for the plane removal flattening calculation.

In order to ensure consistency in the flattening methods, a line-by-line second order polynomial flattening technique was employed on the surface measurement data as well. The line-by-line method was evaluated in both the *x* direction, and the *y* direction producing two different surfaces each flattened iteratively in the same fashion as a single Dektak profile. The curvature of the surface should only exist with respect to the *y* direction of the scans if the scans are properly aligned. Since perfect alignment is not possible, the plane flattening method above utilized polynomials in the *x* and *y* directions. The Dektak line profiles exist isolated from the surrounding topography. The flattening technique applied to the SWLI considers nearby lines, thus it varies from the flattening process for a single line. To explore the differences between the line and area flattening techniques, a new area flattening method was developed where each line in an area was considered independently of the lines adjacent to it.

This method has individual lines' curvature removed from them iteratively. The result is a surface where each line is individually corrected and brought down to a zero average height. This means that if there is tilt in the orthogonal direction to the fit, it will be accounted for in the

curvature removal process. The line flattening method is identical to the method used for a single Dektak line scan. The difference is this line-by-line flattening method treated each individual surface profile in the length (x) direction independently and iterated through the surface in the width (y) direction. These directions were selected so the lines being flattened by the second order polynomial are aligned with the specimen in the same direction as the Dektak line scans. The results from the line-by-line flattening technique are shown below in Figure 19.

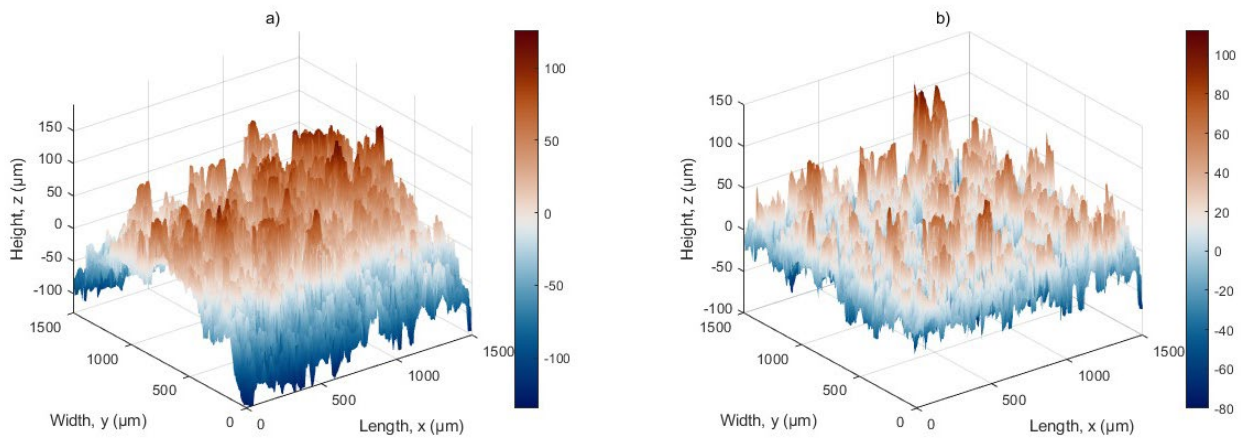

Figure 19: a) Unflattened SWLI raw surface scan sample 004 b) line-by-line flattened SWLI surface 004

Now that two methods of flattening the surfaces have been explored, a comparison can be drawn to understand the difference between the methods. The line method is especially useful for stylus area measurements, because it does not let nearby profiles affect the flattening of a single profile. Stylus area measurements can have misalignment between profile scans and this method accounts for that. An example of this is shown in Appendix X. In order to better understand the effectiveness of the two respective surface flattening functions it is important to directly compare the outputs from the two functions on the same raw surface dataset. The surface obtained from

line-by-line flattening was subtracted from the plane flattened surface to observe the difference between the methods, denoted as δ. It is important to note that δ is on the order of the surface heights and has a significant impact on the results. The results of this are shown in figure 20 below.

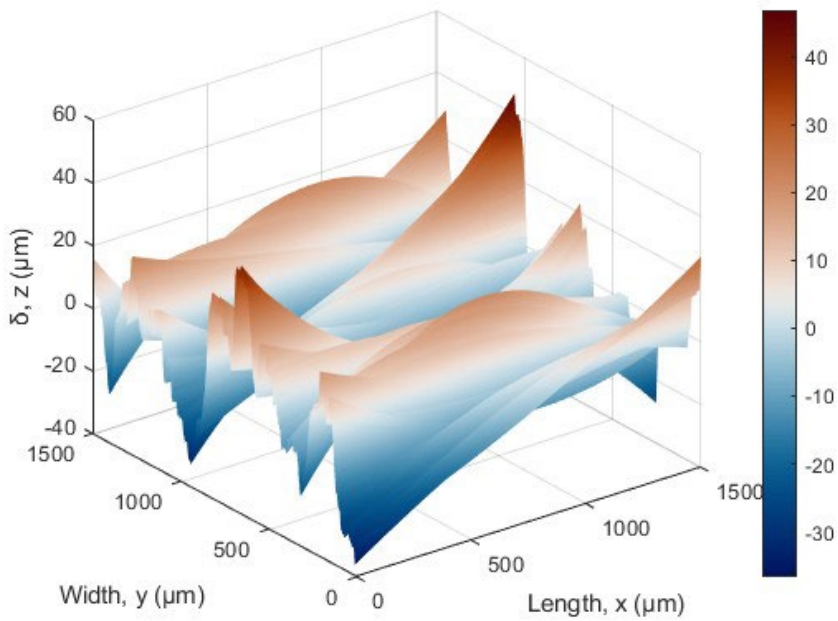

Figure 20: Difference between plane flattening and line-by-line flattening for sample 004

The results from the line-by-line flattening technique appear to remove more information from the raw scan than the plane flattening technique. This is critical when considering line scans rather than full areal roughness scans. The results of the difference indicate that the resulting height values vary significantly with respect to each flattening technique as the surface is iterated through. The difference between the methods is on the order of the surface roughness features measured for the specimen. The variation could indicate that if a surface that has been flattened by a plane fit method, some line profiles in the surface undergo a considerably larger amount of flattening than if they were considered independently. This work involves isolating a single line and considering

its parameters and qualities and investigates the validity of using that line to approximate area measurements and parameters. It is important to ensure that the surfaces and surface profiles being evaluated are flattened and normalized using consistent and equivalent methods.

2.3.3 Line selection from SWLI area scans

A critical idea mentioned earlier is the extrapolation of a dataset with a limited scope in hopes of generating further useful data. A single line was chosen at random for evaluation from the SWLI scan. This line serves as a representation of a profilometer line scan across the measured surface, however the data has been obtained with the same working principle as the area scans. This line profile is shown below in Figure 21.

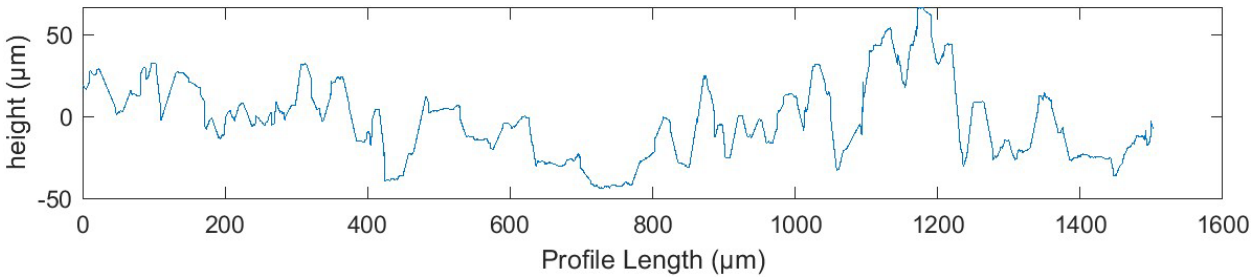

Figure 21: Flattened selected line profile from SWLI dataset sample 004

This surface profile will be characterized in multiple ways and methods of extrapolating useful information will be employed to approximate the parameters and information able to be derived using an area measurement technique. This line can serve as a part of a larger dataset for comparative analysis and can be related to Dektak line scans. This comparison, while not exactly the same surface profile, was measured at the same general location on the specimen and can

illustrate the differences in working principles between the equipment and how those impact resulting surface profile measurements.

Chapter 3: Advanced surface characterization parameters

The previously defined roughness parameters provide valuable insight into the surface topography of specimens. The parameters can accurately describe surfaces, however recreating a surface when only provided with those parameters would likely be an insurmountable task. The following sections search for additional parameters to further define surfaces in hopes that additional parameters could assist with generating surfaces that approximate real and measured surfaces with limited-scale datasets. Understanding extrema of a surface only serves to define a small percentage of a surface dataset. Understanding global or averaged parameters can describe more datapoints in a measurement, however these can neglect extrema and aspects such as shape and frequency of features. If a relevant surface can be generated using certain parameters, these parameters are arguably significant when defining surface topography. The definition of relevant generated surface data is one that is not clearly defined. For the purposes of this investigation, surfaces generated using statistical methods and further defined parameters will be evaluated against the measured surface datasets. The parameters and statistical extrapolation methods will be employed with a limited portion of these datasets and will be evaluated against the whole.

3.1 Measured dataset exploration

The advanced surface parameters explored in section 3.1 are based on single line height profiles. The profiles explored in subsequent sections are taken from both Dektak line scans and single lines taken from SWLI area scans. These are of different lengths with different resolutions,

however the calculations performed are consistent between contact and optical measurement techniques.

3.1.1 Bearing Area Curve calculation methodology

Understanding specific extrema features can assist with identifying points of failure and abnormalities that can affect a component's behavior. Extrema values such as maximum profile valley depth have been consistently correlated with component fatigue performance (5). Extreme features by nature can exist without affecting large portions of an area, so global parameters that take into consideration and represent an entire measured dataset can describe surfaces in a more holistic manner. The Bearing Area Curve (BAC), sometimes referred to as the Material Ratio, are commonly used in contact mechanics as parameters that can potentially correlate with friction and wear measurements (41). The use of the BAC extends beyond plane contact analysis, as it can also illustrate the proportion of deep valleys and asperities to the rest of the dataset. The BAC also can indicate how the surface is structured with respect to different heights, and is directly related to the height distribution. Figure 22 below illustrates how the BAC is extracted from a surface profile according to ISO standards (42).

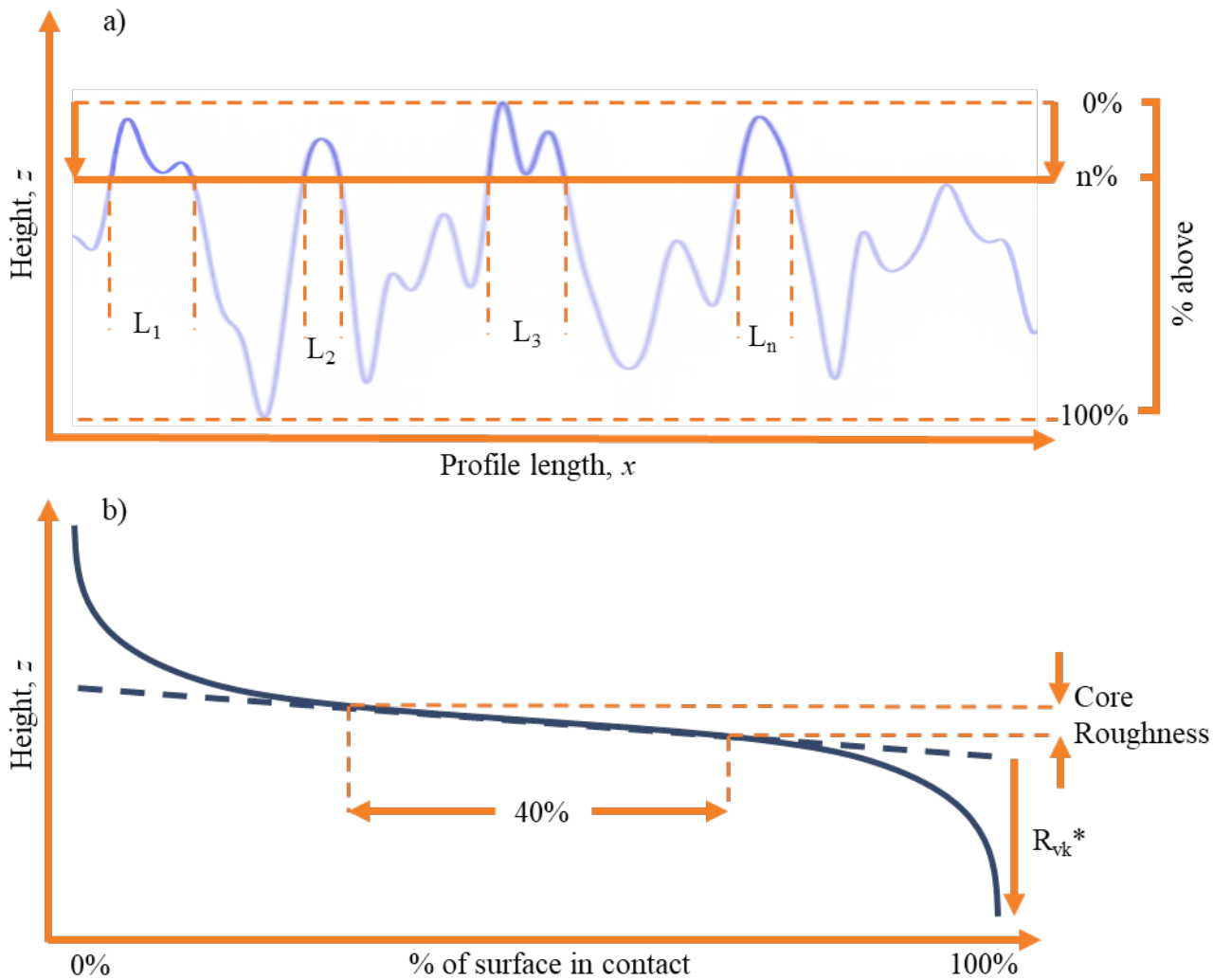

Figure 22: a) Example line profile with method determining percentage of the profile interacting with a line passing through it b) the resulting bearing area curve, or material ratio curve showing the percentage of the surface profile interacting with the line as

The BAC is best suited for digital calculation, where the height values of the surface are discrete z direction numbers in a vector or array. The ISO standard then utilizes the equation below to calculate a BAC from the part a) of Figure 22. It must be recognized that L_i is a length portion

of the surface profile above the set iteration point, and is effectively calculated through a direct summation shown below.

$$BAC(h) = 100 * \frac{(\sum_{i=1}^{L} I(z_i \geq h))}{L}, \quad for\ h\ \in [R_v, R_p] \tag{15}$$

$$I(z_i \geq h) = \begin{cases} 1 & if\ z_i \geq h \\ 0 & if\ z_i < h \end{cases}$$

The calculations for the BAC were performed using both a CSP line scan data profile, and a single line height dataset taken from a SWLI area scan. The full area dataset obtained from the SWLI will be discussed in a subsequent section. The line datasets underwent the same analysis processes. The span of the datasets was defined as the total height of the measurements from the maximum valley depth to the maximum peak height. A loop is run, that iterates the same number of times as the span of the dataset. This loop begins at the *Rp* maximum peak height and iterates downward towards the Rv maximum valley depth of the profile. The loop records the number of height values greater than or equal to the height value at a specific instance. As this passes downward through the profile it represents a straight line descending through the surface taking into account the number of datapoints that are at or above that line. It must be acknowledged that this BAC calculation does not consider contact mechanics principles or material properties that would describe material deformation as contact occurs. The BAC is a statistical analysis method intended to describe an undeformed surface. The BAC is then calculated by converting the vector representing the changing amount of datapoints in contact with the flat line into a percentage. In order to calculate a fit line to the BAC curve, the beginning 30% and ending 30% of the BAC curve data is omitted and a straight line is fit to the center 40% of the dataset (43). The center 40% of the dataset in the BAC is utilized to determine the core roughness depth (R_k), which conventionally is used as a means to separate the extrema from the central parts of the dataset. A

line is fit to this center 40% of the core roughness to determine behavior of the surface. A small negative slope indicates that much of the core roughness falls at the mean line of the surface, often indicating a flatter surface. A steeper negative slope indicates a rougher surface with more deviation in the height values from the mean. For applications such as fatigue analysis, analyzing extrema can provide valuable insights (7,8,19). The maxima of a dataset are important, but understanding maxima relative to the rest of the dataset is arguably more useful. The reduced valley depth (R_{vk}) is defined in the ISO standards as the average depth of the profile valleys projecting through the core roughness profile (44). This parameter is commonly used to contextualize valley extrema. The total valley depth (R_{vk}*) is defined as the difference between the end of the BAC fit line and the R_v valley depth (45,46).

The BAC curves calculated for these rough additive specimens do not always replicate the typical curve shown in figure 22 b). This is due to multiple factors relating to both the surface structure, and the characterization techniques. The rough specimens have partially attached metal powder particles that are represented as prominent and large asperities. These metal additive components have a very high surface roughness, and this renders the typical R_{vk} parameter difficult to calculate. This is due to the "core roughness" not adhering to a typical height distribution, and a bearing area curve that crosses the fit line at multiple points. The difficulties for rough specimens meant the R_{vk}* parameter was far easier to use and provided valuable insight into the surfaces by considering the core roughness to the same degree as R_{vk} while lending more weight to the extrema valley measurement.

The BAC can also be calculated for an area as well as for a line profile. For an area the surface will effectively have a flat plane pass through it and record the percentage of area height

datapoints from the area dataset that are at or above the plane. This process is shown in the figure 23 below as z_i is iterated from the maximum peak height to the maximum valley depth.

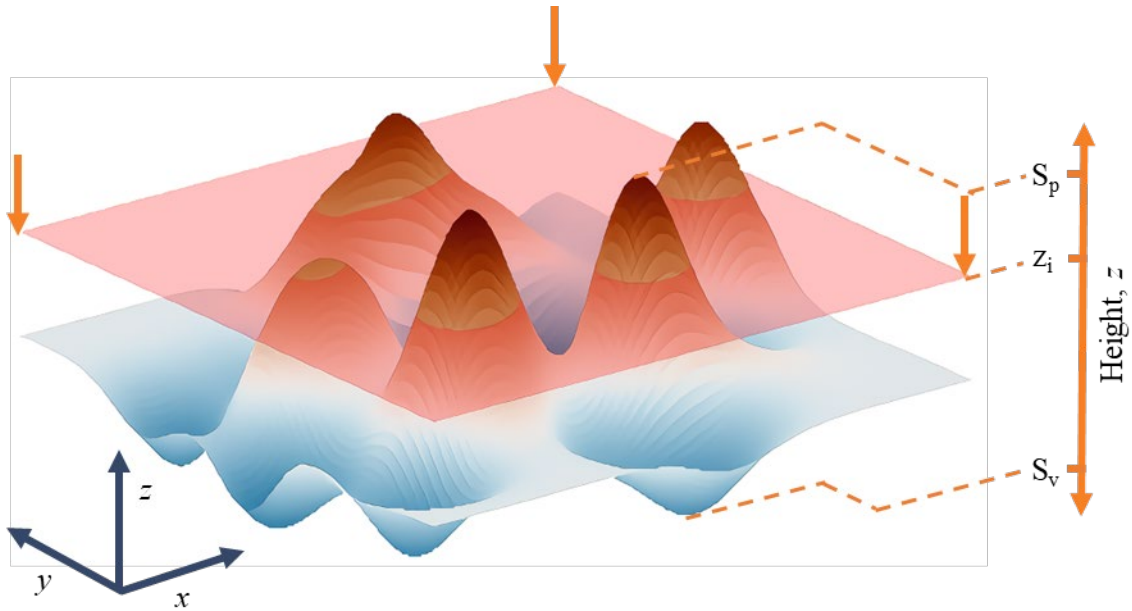

Figure 23: Example surface with method determining percentage of the surface interacting with a plane passing through it

The resulting calculation provides the areal total valley depth (S_{vk}*) that is derived in the same process shown in figure 22 b). This method is unchanged between line and area dataset calculations.

3.1.2 Spectrum introduction

When surface geometries are analyzed across scales, relating and defining features can be difficult. When an analysis is intended to be conducted on features at a particular scale, features outside the range of analysis can have a measurable impact on results (47). Measurements recorded

by CSP and SWLI report height values without consideration of factors such as larger scale influence of things such as form and waviness. To isolate roughness from waviness and form, post-processing methods were discussed, however there are parameters that can assist with understanding how features relate to the scale they exist on. The spectrum of the dataset can also provide insights into the surface structure across scales. The additive surfaces being considered are multiscale surfaces that consist of various features across different scales. The spectrum of a dataset is a property that can be analyzed to better understand a dataset's behavior with respect to scale.

The spectrum of the dataset records amplitudes of the surface features with respect to frequency. The lower frequency components are often on the macro-scale. These are generally larger and broader features that can span a significant portion of a measurement. This can impact the overall shape of the surface and exists largely from the scale of the form down to the waviness. The higher frequency components largely exist on the nanoscale and smaller. These features are sometimes smaller than measurement equipment resolutions and can sometimes be noise and not real surface features. The spectrum frequencies thus correspond to different wavelengths (i.e. scales) of surface features. The amplitudes correspond to the magnitude or prominence of the surface features at the set scale.

3.1.3 Fractal dimension overview

In many cases, the ability to define a surface with random features across a large range of scales is very difficult, as the presence of certain features is unpredictable. One way of approximating rough surfaces is to define them as fractal surfaces. To define a fractal surface, first

a fractal must be defined. A fractal is a middle ground between strict Euclidean geometric order and chaotic mathematics (48). Fractals are defined as self-similar or self-affine features existing across a set defined range of scales (49,50). These features retain their shape identity but vary in size. This restriction allows for more complex surfaces to be approximated and analyzed across scales.

Since fractal surfaces follow a structure across a range of scales, a scale-independent parameter is necessary to characterize rough fractal surfaces. A widely used parameter is the fractal dimension *D* (50–52). This parameter relates the ratio of the change in detail to the change in scale. The *D* for a line is 1, the *D* for a flat plane is 2, a rough fractal profile should fall in between. The *D* for a volumetric space is 3, and a rough surface would be expected to fall between 2 and 3 (53). There are multiple ways to obtain *D*, two of which will be explored in the following work. Metal additive surfaces from as-built specimens are incredibly complex due to their means of creation. L-PBF processes result in large amounts of partially melted particles on the surface of specimens, so the range of features spans from the form of the specimen down to, and even beyond the metal particle size. For fatigue applications, and due to the limitations of the equipment employed in this study, the analysis was largely done from the scale of the form of the specimen down to just below the metal particle size. A computationally efficient way to approximate these rough surfaces is through the utilization of fractal geometries (51).

3.1.4 Fractal dimension calculation using Fourier spectral method

The Fourier transform effectively deconstructs a space (or time) based function into a frequency based function. This is performed on discrete height values for the subsequent analysis. The exploration of the line data was conducted using both the Discrete Fourier Transform (DFT)

and the Fast Fourier Transform algorithm (FFT) (54,55). The DFT and FFT methods were both utilized and compared to ensure consistency in the data processing techniques employed in this study. The equation for the DFT utilized can be found below. Z[k] is the DFT of the sequence (surface profile) z[n].

$$Z[k] = \sum_{n=0}^{N-1} z[n] * e^{-j\frac{2\pi}{N}nk} \tag{16}$$

This calculation provides an input signal of line height data. This calculation for a discrete height dataset is performed so that the *N* number of samples taken corresponds to the length of the sample in terms of datapoints. The line dataset length is also the same as the number of frequency bins calculated in the DFT process. To render the DFT direct results from the previous equation useful, they must be normalized. The normalization process is shown in the equation below.

$$Z_{norm}[k] = |2 * Z[k]| = \left| 2 * \sum_{n=0}^{N-1} z[n] * e^{-j\frac{2\pi}{N}nk} \right| \tag{17}$$

The DFT result is doubled in the normalization process because the result contains both positive and negative frequency components, and only the full amplitudes are desired to be considered. This is done to analyze a single-sided amplitude spectrum of a real-valued signal, as the full spectrum is symmetrical. The absolute value magnitude is taken to ensure that positive full amplitude results are obtained.

The FFT algorithm is a method developed to calculate a discrete Fourier transform more efficiently (56). For larger datasets the DFT function can require time to compute on standard equipment, as the function must iterate through the length of the line measurement for both the scale sample and the frequency bin loops. This calculation has a complexity of $O(N^2)$. The FFT

algorithm brings this complexity down to O(N*log(N)) (55). This is accomplished by recursively breaking down a large DFT problem into smaller DFT problems. The symmetries of the DFT matrix allows for this reduction in computational requirements. When the FFT result is calculated using the MATLAB embedded function, the normalization process must be identical to the DFT. The result is doubled, and the absolute value is taken. The phase and amplitude of the spectrum can provide valuable insight when analyzing these rough surfaces. The phase of the spectrum represents the frequency component and can be calculated using the "angle" function in MATLAB. The following expression represents the mathematical process the angle function calculates (57). a_n is the imaginary component of the spectrum in the n-th position. b_n represents the real component of the spectrum in the n-th position.

$$\varphi_n = \tan^{-1}\left(\frac{a_n}{b_n}\right) \tag{18}$$

The equation solves for the phase angle of the n-th frequency component. for calculating the amplitude Δ, the magnitude of the spectrum must be taken, and the following equation delineates the process (57).

$$|\Delta_n| = \sqrt{a_n^2 + b_n^2} \tag{19}$$

To extract a meaningful fractal parameter from a Fourier dataset, the results of the spectrum must be plotted, and a line fit must be obtained (58). The plot of the spectrum is a scatter plot with the log of the frequency on the x axis and the log of the magnitude squared on the y axis. This is shown below in figure 24.

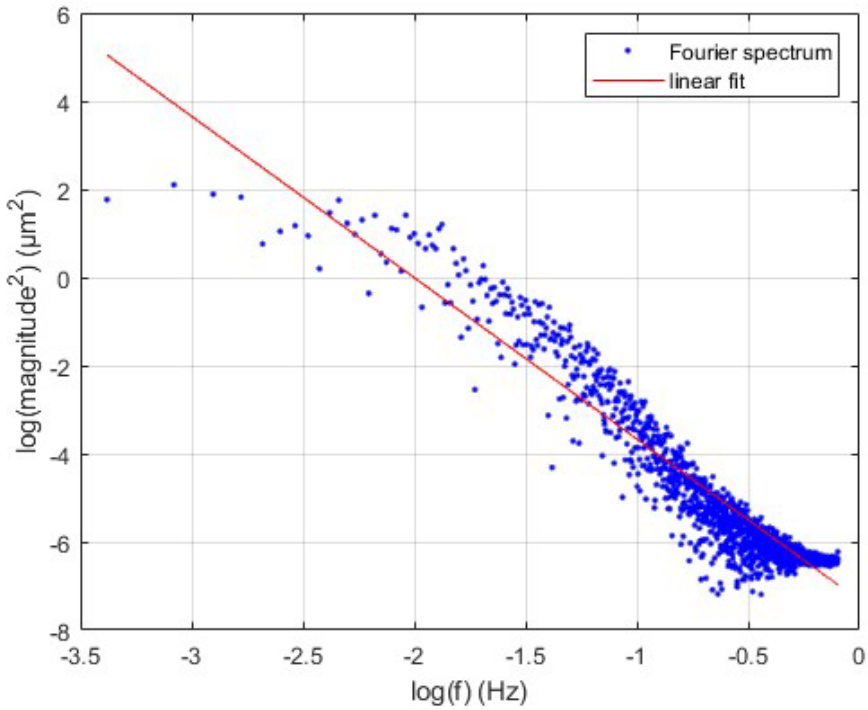

Figure 24: Fourier analysis and linear fit from Dektak line profile data sample 004

The Fourier series shown in figure 20 on a log scale allows for a linear fit slop to represent the value beta (β). This beta is related to the fractal dimension (D) through the following equation (49).

$$D = \frac{4 + \beta}{2} \tag{20}$$

Once the fractal dimension is obtained through this Fourier method, more analysis can be conducted and other datasets with the same fractal dimension can be explored to determine if conclusions can be drawn based on this parameter. It must be noted that while the fractal dimension can be an insightful parameter, and the Fourier log plot may have a linear tendency, the Fourier log plot is not likely to have a perfectly linear relationship between variables. This is because the real measured surfaces are not perfectly fractal in nature. D is a simplification of the relationship,

and as such omits some details of the surface. The degree to which the relationship between beta and the Fourier log plot is accurate will determine the relevance of the fractal dimension parameter calculated with this method.

3.1.5 Fractal dimension calculation using Roughness Length method

The fractal dimension calculation is not exclusively based on the beta parameter from the Fourier method. There are multiple ways to calculate the fractal dimension when provided with a dataset. One way to extract a fractal dimension from a dataset is to use the roughness length method (49,59,60). The parameter R_q is utilized to determine roughness, and a changing window length is used to determine scale. The change in scale process is depicted below in figure 25.

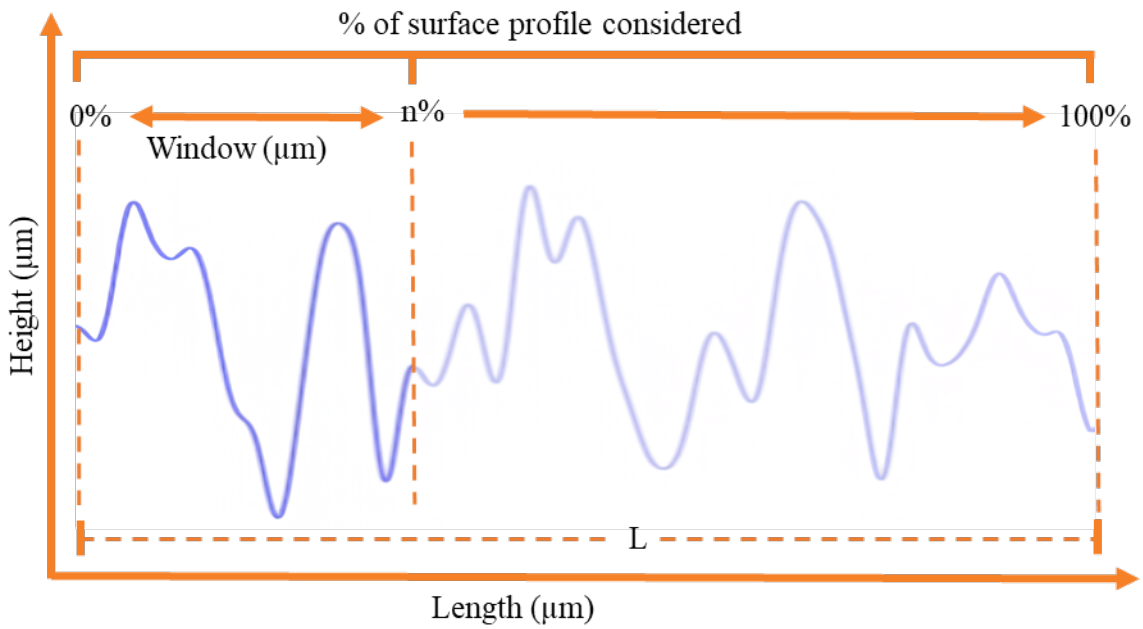

Figure 25: Roughness window calculation method for line profile

This method takes into consideration a window that is simply a portion of the line profile data available. The window begins with the smallest number of datapoints that are still able to calculate a RMS roughness value using Equation 3. This minimum is 2 datapoints, and the RMS roughness is calculated for this smallest window size and then the window is increased a single datapoint at a time and the RMS roughness is calculated for each window size iteration. This calculation is performed until the window size is scaled up to the length of the line profile measurement. The RMS roughness with respect to window length is plotted on a log scale and is expected to have a relatively linear relationship if the surface is perfectly Weierstrass-Mandelbrot fractal. The resulting plot then has a straight line fit to the discrete dataset, and the slope of the fit line is obtained. This slope of the fit line is the Hurst Component (*H*) (49,51). This is shown below in figure 26.

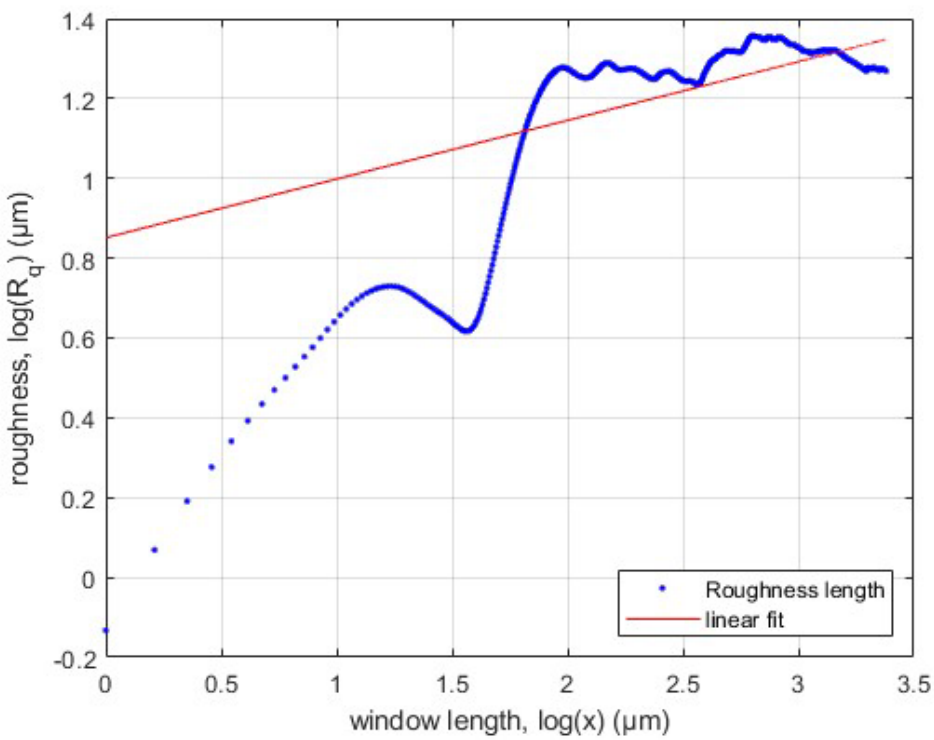

Figure 26: Roughness length method for calculating *H* from Dektak profile sample 004

This parameter can be related to the fractal dimension through the following equation.

$$D = 2 - H \tag{21}$$

This calculation provides insight into how the RMS roughness changes with respect to incremental changes in scale. This method of generating a fractal dimension creates a generalization by assuming the RMS roughness on the log scale has a linear tendency. Fractal surfaces with similar RMS roughness over scale behavior to measured surfaces can be analyzed to potentially generate meaningful datasets. Work done by Zhang and Jackson et al. suggests that the roughness length method correlates well with the actual *D* and is more effective than other methods for determining *D* (53).

3.2 Relating Area datasets to Line datasets through parameters

The purpose of relating line measurements to area measurements is largely derived from ensuring minimal resources can be being utilized to generate meaningful results. In addition, stylus measurements may be more accurate than other methods, albeit slower. Generally, the more data generated from a measurement the better. Costs can affect this generalization, as resource allocation is an important aspect of both research and industry. Cost-benefit analysis can assist in determining how much data is required for a specific project. Quantity of data aside, the availability of measurement techniques can also factor into which datasets will be available to represent a surface.

Inherently line-based parameters are different from area-based parameters as they represent aspects of datasets with different dimensions. On a core level the aspects they represent are similar despite the difference in dimensions they account for. The maximum valley depth from a single

profile in a SWLI area scan will not provide the same information as a maximum valley depth from the entire area dataset. While it is not valid to directly compare line profile parameters to area parameters, understanding the differences between them can provide insight into how surfaces are characterized using different measurement techniques.

Chapter 4: Surface area Generation/Extrapolation

4.1 Extrapolation from line profile data to a generated area

Increasing the dimension of a dataset from a line to an area is not as simple as extending the line profile in an additional dimension. For rough additive specimens, the asperities present are not grooves and peaks occurring at regular intervals. The standard parameters discussed in section 2.2.1 for line measurements may not directly correlate with their areal counterparts outlined in section 2.2.2. The parameters will generally increase when larger measurements are taken. A longer line scan will generally have higher roughness than a shorter line scan. For this reason, the advanced parameters outlined in section 3 can provide a means to extrapolate from a line to an area. The utility of scale independent properties of surfaces shines when datasets are being considered over multiple scales. These parameters can assist in providing a baseline from which the extrapolated surfaces can be derived from. The goal is to utilize a dataset on a limited scale to extrapolate a potentially relevant dataset on a larger scale. The advanced parameters discussed can assist by bringing in scale independent aspects of the datasets and insightful roughness analysis.

4.1.1 Extrapolation using Fourier orthogonal spectrum method from a line dataset to an area dataset

The ability to break a complex line profile into component sin waves can provide valuable information about the sizes of features at various scales. This is accomplished by analyzing features with respect to their wavelengths, or lateral scale. Composing the component waves back

into the original signal can assist in ensuring that the data accurately represents the original profile dataset. The method employed utilizes the orthogonality of the x and y axes and reconstructs a surface along those two directions. This relationship is why this reconstruction method is called the "Fourier orthogonal spectrum method" in this work. Once line profile features can be deconstructed and recreated, the same method with respect to an additional orthogonal dimension can be used to extrapolate a surface.

To recreate a line profile from Fourier spectrum data, a summation of waves must be conducted across both the length dimension (x-axis) and the scale dimension. This summation will result in a complex line profile with features matching the original measured profile. The Fourier transform deconstructs the complex signal (profile) into amplitude and frequency data, and the inverse calculation returns the complex signal. The equation for this calculation is the inverse of the higher order full discrete Fourier definition (61).

$$z(n) = \sum_{i=2}^{scale} \sum_{n=2}^{x-dir} |\Delta(i)| \left(\cos\left(\frac{2\pi(n-1)\delta_x}{\lambda(i)} + \varphi(i) \right) \right) \tag{22}$$

This equation normalizes the amplitudes from the FFT by the length of the spectrum of the dataset (L_s) internally in the software utilized for calculation. The equation also depends on the length interval between the datapoints (δ_x) which is calculated by dividing the scan length in the x direction by one less than the length of the spectrum. This accounts for the space between each datapoint and is stored in the attached code in units of meters. The attached code in appendix X calculates the above equation iteratively. The first loop iterates through half the length of the spectrum, due to the symmetry of the spectrum. The first loop ensures that the phase and wavelength are calculated at each spectrum interval. The phase is calculated utilizing an "angle"

internal function, and the wavelength is denoted as a portion of the scan length for each step. The second (internal) loop calculates the equation above, and iterates through the full length of the spectrum. This is due to the original scan length and the spectrum having the same number of datapoints; thus, the recreated line must also be of the same length. An image of the original and inverse Fourier recreated profiles overlayed is shown below in Figure 27. This image simply serves as verification of the code and algorithm implemented.

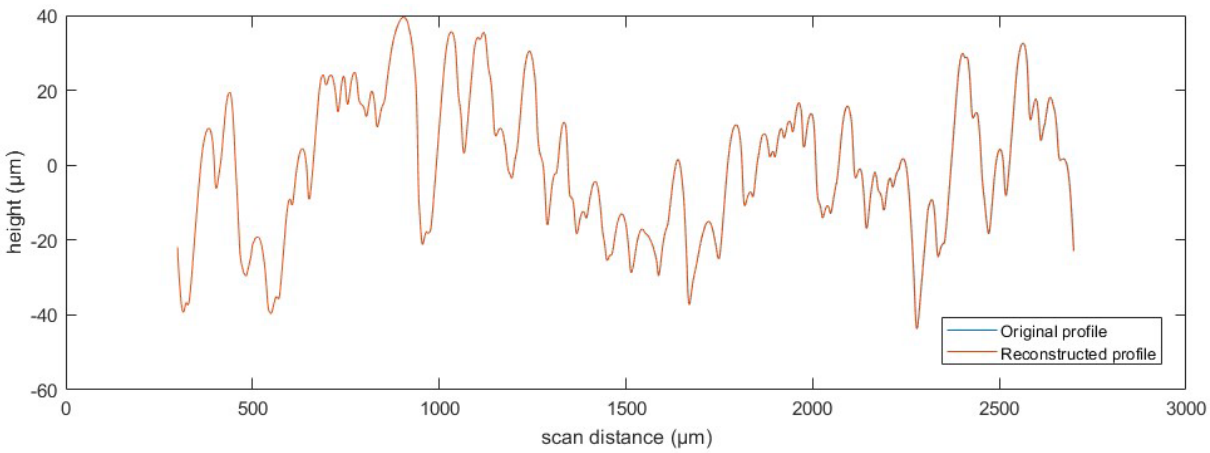

Figure 27: Original Dektak line scan profile and Fourier recreated line profile for sample 004

Once the features of a single line could be replicated, the orthogonal method could be used to produce a surface dataset based directly on the spectrum of a single Dektak contact line profile. The following equation can be used to extrapolate potential surface values from a single line to a surface with a set length and width.

$$z(n) = \sum_{i=2}^{scale} \sum_{n=2}^{x-dir} \sum_{m=2}^{y-dir} |\Delta(i)| \left(\sin\left(\frac{2\pi(n-1)\delta_x}{\lambda(i)} + \varphi(i) \right) + \cos\left(\frac{2\pi(m-1)\delta_y}{\lambda(i)} + \varphi(i) \right) + \right) \quad (23)$$

For consistency's sake the dimensions of the SWLI area scan were selected to determine the bounds of the extrapolated area so a direct comparison could be made between the extrapolated area and the original SWLI line scan. The resolution in the y direction was set to be 3063 datapoints, and the resolution in the x direction was set to be 3068 datapoints matching SWLI scan dimensions. The size of the measurement is defined where δ_x is the x direction resolution and δ_y is the resolution in the y direction. The datasets from a specific sample can then be compared to gain insight into how well an extrapolated surface from a single CSP line profile could compare to a full optical area measurement. The x and y dimensions of the resulting height value matrix (generated surface) are defined by the user, but the wavelengths considered in the calculation directly determine the amount of scales calculated in the outer summation. The higher frequencies account for smaller and smaller wavelengths which represent finer details and features. The general trends of the surface can be seen from the longer wavelength features. 11 surfaces were extrapolated using this Fourier orthogonal method, and the resulting RMS roughness of each surface was plotted. Figure 28 shows the variation of the RMS areal roughness with respect to increases in the scale indices.

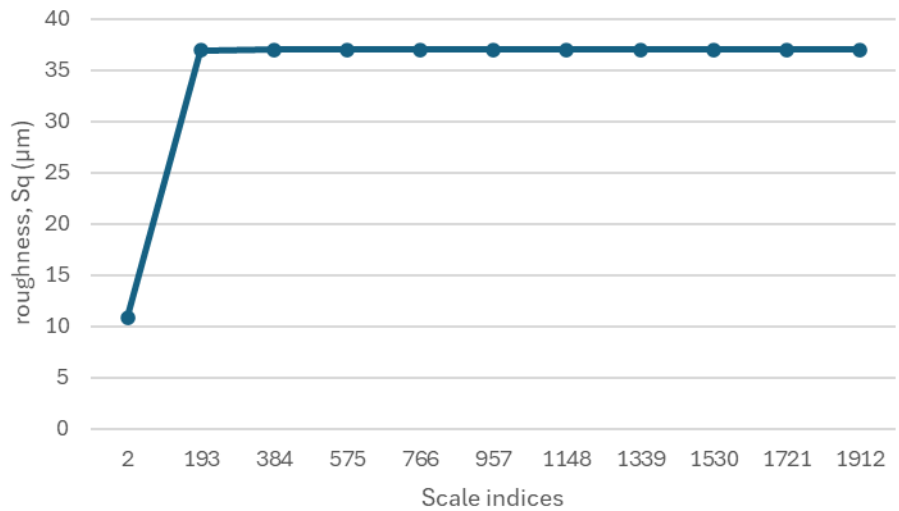

Figure 28: RMS roughness with respect to scale

The scale dimension greatly affects calculation time, for this reason instead of running the loop from 2 to half the length of the spectrum (1912), the loop was run from 2 to 500. This effectively filters out some of the smaller scales.

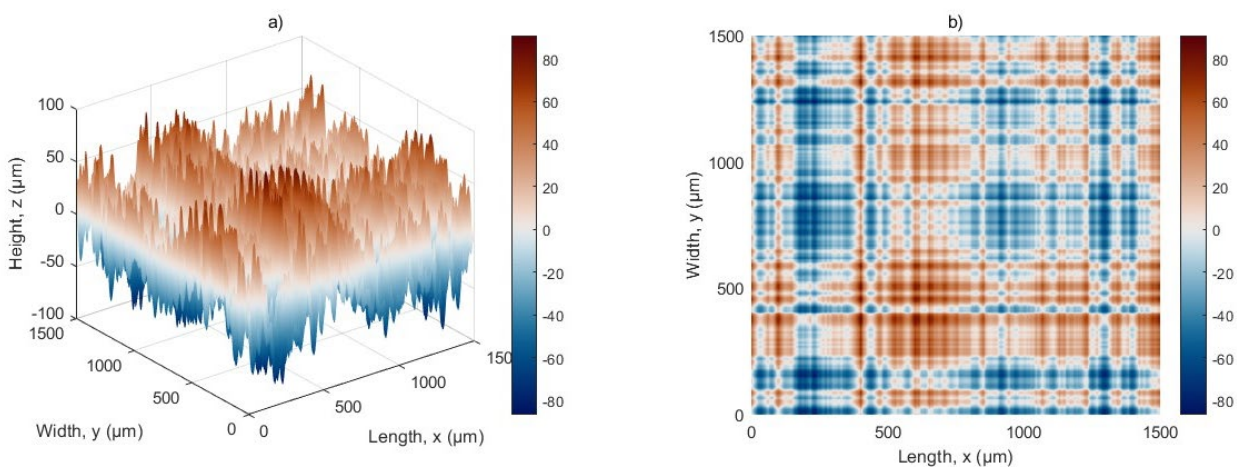

Figure 29: a) orthogonally extrapolated surface from Dektak line scan for sample 004 b) top-down view of orthogonally extrapolated surface from Dektak line scan for sample 004

The same method of data extrapolation was performed on a single line taken from the SWLI scan that the extrapolated surface will be evaluated against. This method takes a portion of the dataset, extrapolates that portion back to its original dimensions and then evaluates it directly against its original entirety. The SWLI single line was selected indiscriminately, and to ensure some consistency between samples the 1500th line of the area scan was selected for all specimens. This single line underwent the same process outlined for the Dektak line profile.

The resulting surfaces were compared to their SWLI scan measured counterparts for each of the 14 specimens measured and evaluated. These results will be presented in subsequent sections. The effects of the differences in working principles can be seen when a single profile from the SWLI is evaluated against a single profile from the Dektak.

4.1.2 Surface generation utilizing Hurst Component from Fourier method using fractal artificial surface program

The utility of the Fourier method for characterizing a surface has been explored in the literature, and the generation of surfaces based on the fractal dimension calculated through this method is also established (51–53). *D* was calculated for each specimen using Eq. 20 from section 3.1.4 with a beta value being extracted from each dataset. Eq. 20 and Eq. 21 from section 3.1.5 must be combined relating the two methods of calculating fractal dimensions in order to convert the beta parameter into a Hurst component input for the fractal artificial surface generator. This process of calculating a Hurst component when provided with a beta value is shown in the equation below.

$$H = 2 - D = 2 - \frac{(4+\beta)}{2} \tag{24}$$

This parameter was used in an fractal artificial surface generator code (62). This code executes a surface generation method that utilizes wavevectors and symmetry in conjunction with the inverse fast Fourier transform (IFFT) to generate a surface from a provided RMS roughness, Hurst component, and set of length matrix dimensions. The function begins by validating the inputs from the main file, ensuring the parameters provided match the number of input arguments given in the call to execute the function. The bounds of the *x* and *y* dimensions are then validated as even numbers, or are modified to be even numbers. The length of the measurement is provided in the function input line along with the dimensions in pixels for the desired area. These values are then utilized to calculate the distance between pixels and corresponding dimensions in the length unit provided in the function input. The next step of the fractal artificial surface generation function is to create wavevectors of the surface. These are found utilizing the following relationship between wavevector, q wavelength λ (63,64).

$$q = \frac{2\pi}{\lambda} \tag{25}$$

These wavevectors are then scaled and centered using the unwrap and fftshift MATLAB internal functions.

The Power Spectral Density (PSD) is calculated producing a two dimensional matrix of height values that are not yet scaled to match the desired surface output (65). This calculation prevents the PSD from becoming too large at small frequencies and iterates through each *x* and *y* position of the desired height value matrix output. The calculation describes how the surface topography magnitude is distributed across different spatial frequencies. This dataset is then scaled

to match the RMS roughness input. A random phase is utilized, and the inverse fast Fourier transform is applied to generate the surface from the spectral dataset. This creates a surface that has set dimensions, and has features set according to not only the scale-independent Hurst Component, but the RMS roughness as well. For specimen 001 the described Fourier methodology used data from a line profile to generate a 3-D surface shown in Figure 30 below.

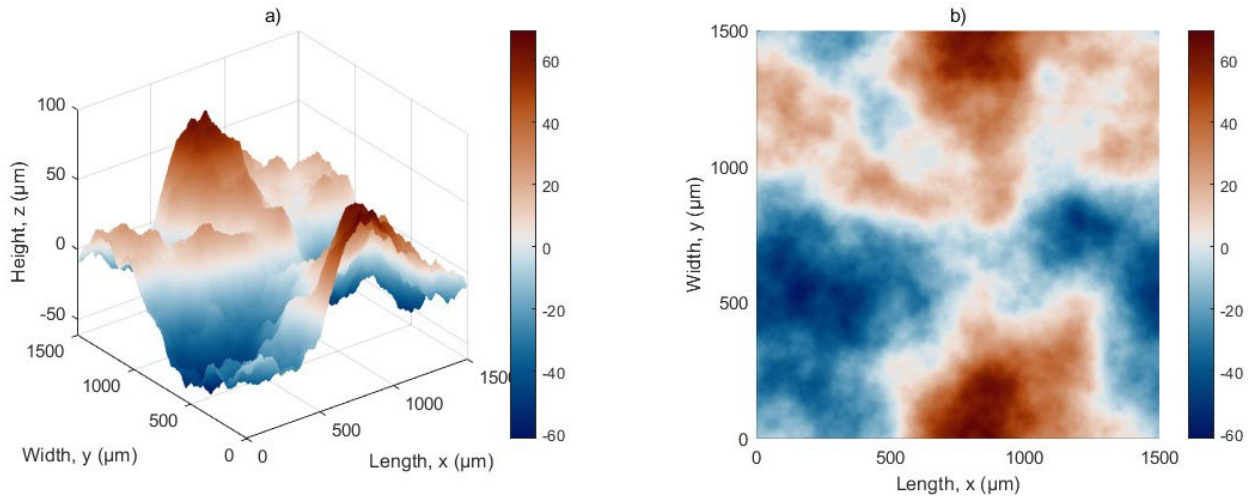

Figure 30: a) artificial_surf generated surface using Fourier method from Dektak line scan for sample 001 b) top-down view of artificial_surf generated surface using Fourier method from Dektak line scan for sample 001

For sample 001 the RMS roughness value input into the program was 26.15 µm as measured. The Hurst Component calculated from the beta parameter using the Fourier spectral method outlined in Equation 20 and input into the program was 0.9954. The resulting surface is approximately the same size as the orthogonally extrapolated surface; however the density and structure of features is markedly different, and the orthogonal method yielded a surface with consistent repeated features that are not present in the Fourier spectral method generated surface.

The Dektak lines were not the only lines evaluated using this method; the single line from the SWLI area scan for each specimen was also used to generate a comparable surface. The results of the comparisons of these surfaces will be shown in subsequent sections.

4.1.3 Surface generation utilizing Hurst Component from roughness length method using fractal artificial surface program

The roughness length method extracts a Hurst Component directly from a profile, and the same process for generating a surface with the fractal artificial surface generation function is employed. The only difference between the two generation methods is the means by which the Hurst component is obtained. The surface generated using the roughness length method is shown in Figure 31 below.

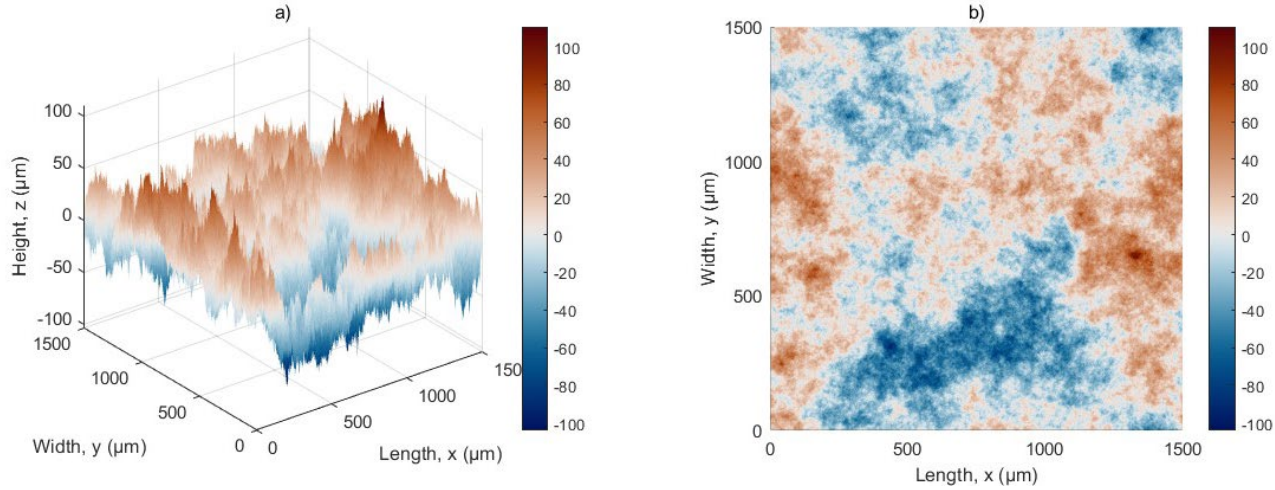

Figure 31: a) artificial_surf generated surface using roughness length method from Dektak line scan for sample 001 b) top-down view of artificial_surf generated surface using roughness length method from Dektak line scan for sample 001

Since the input line was the same, the RMS roughness value provided was also the same 26.15 μm. However, the Hurst component extracted from the roughness length method for sample 001's Dektak line scan was 0.3287 which is very different from the previous component calculated using the Fourier method. Similar discrepancies were found in the previous literature of Zhang and Jackson et al. where the extracted fractal dimensions varied based on analysis method (49,53). This calculation produced a surface with smaller and more distinct features, more closely mimicking the rough additive measured surfaces (see Fig. 18). The comparison between this surface and others generated using the roughness length method will be shown in subsequent sections.

4.2 Comparison of measured areas to extrapolated and generated areas

4.2.1 Extrapolation using orthogonal spectrum method

The entirety of a SWLI area scan can be analyzed using the areal roughness parameters outlined in section 2.2.2. This can describe the measured surface in greater detail than a single line profile. Now that the methods for extrapolating a surface from a line profile using the Fourier spectral method are defined, they can be employed using a CSP line (similar to figure 27 in section 4.1.1). The surface scanned by the SWLI, and the surface extrapolated from a single Dektak line using the Fourier orthogonal spectrum method are shown below in figure 32.

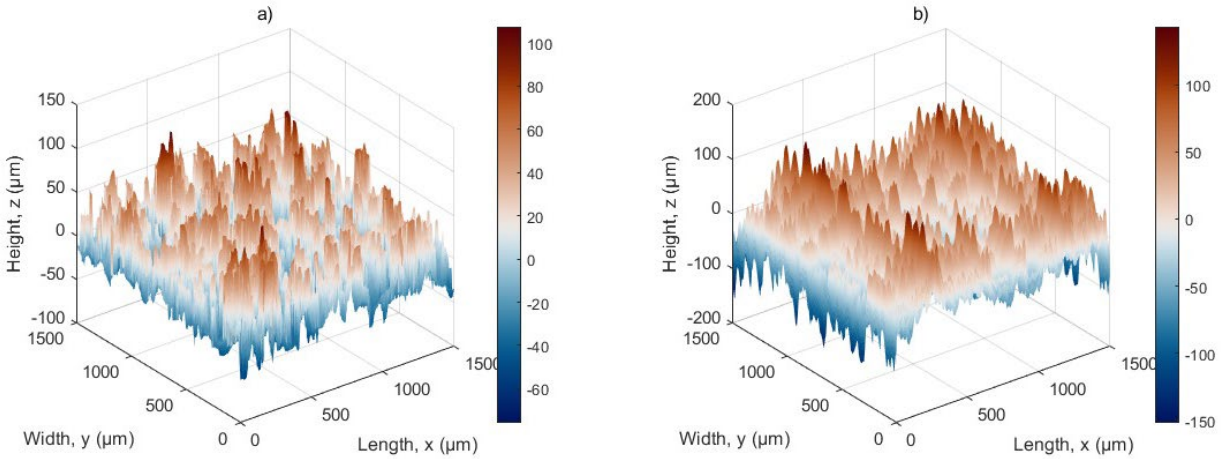

Figure 32: a) SWLI area scan sample 001 b) surface extrapolated from Dektak line scan of sample 001 using Fourier orthogonal spectrum method

A single line scan is not guaranteed to encounter, and capture the same surface features as the SWLI area scan, as the scans were not taken on the same exact part of the cylindrical portion of the samples. The Dektak also has a different working principle, being a contact-based measurement technique. The effects of these differences can be mitigated by applying the same surface extrapolation process to a line taken from the SWLI area scan dataset. The SWLI line profile was obtained through the same working principle as the area scan, and exists in the same location on the specimen. This is shown below in figure 33.

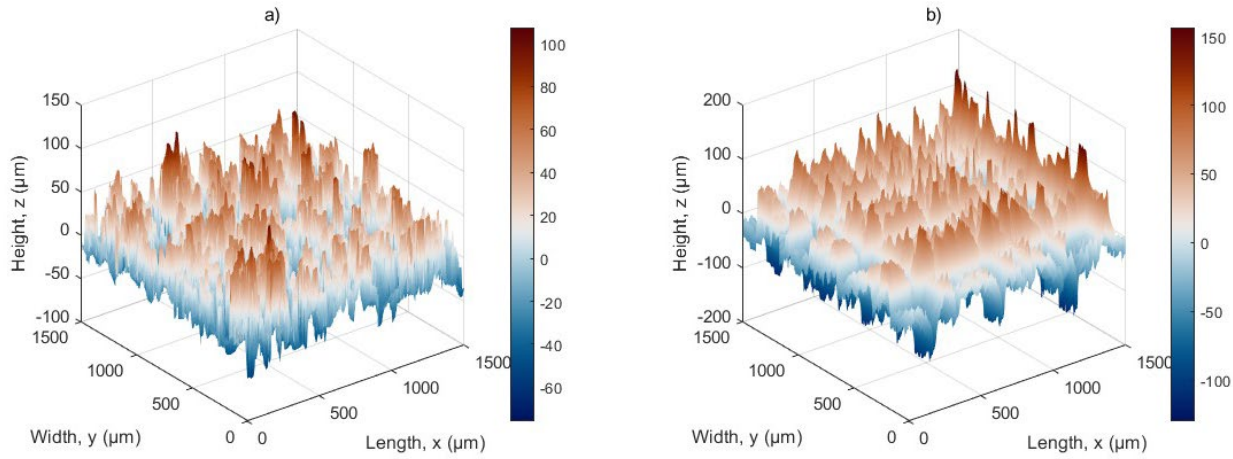

Figure 33: a) SWLI area scan sample 001 b) surface extrapolated from SWLI line profile of sample 001 using Fourier orthogonal spectrum method

Upon initial inspection, the surface extrapolated from the SWLI line profile appears to contain consistently higher peaks and deeper valleys than the surface extrapolated from the Dektak line profile. From visual inspection the extrapolated surfaces do not appear to vary much from each other in morphology. These surfaces and their characteristics will be described in subsequent sections utilizing the previously defined roughness parameters.

4.2.2 Surface generation utilizing Hurst Component fractal artificial surface

The Hurst component was obtained from a single line through two methods: the Fourier spectral analysis method, and the roughness length method. This results in two potential scale-independent parameters per line profile. Again, two line profiles were considered, one from the Dektak profilometer and one from the SWLI scan. The Dektak scan profile was used to generate two surfaces, each with a different morphology and Hurst component. They have been outlined in sections 4.1.2 and 4.1.3. These generated surfaces are depicted with the SWLI scan as a reference

in figure 34. The surfaces are displayed adjacent to each other oriented vertically. This allows for the surfaces of equal lateral size to be visually compared.

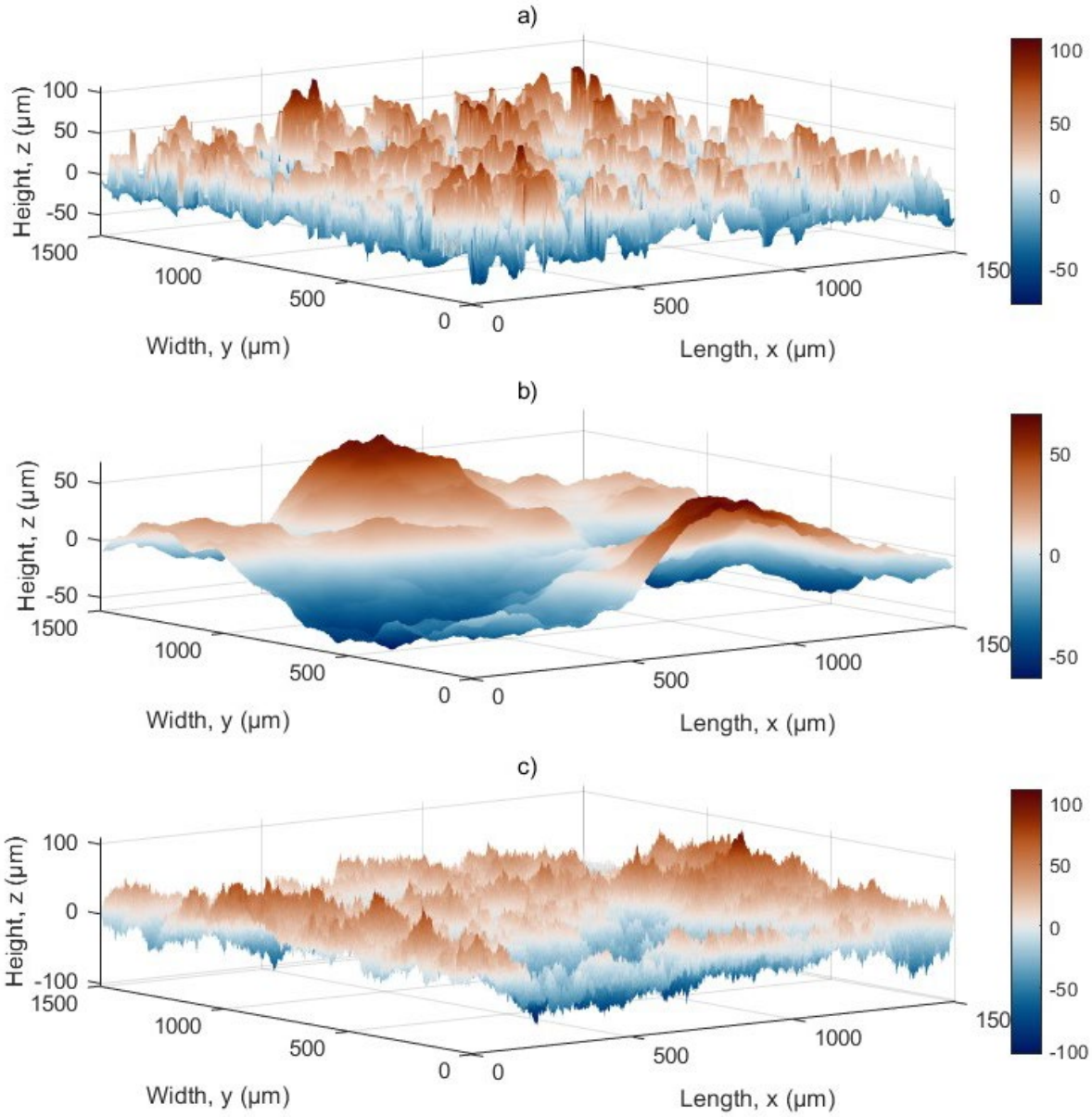

Figure 34: SWLI scan of sample 001 b) fractal artificial surface generated surface using Fourier method from Dektak line scan for sample 001 c) fractal artificial surface generated surface using roughness length method from Dektak line scan for sample 001

Figure 35 shows the surfaces generated using the SWLI line profile in similar fashion.

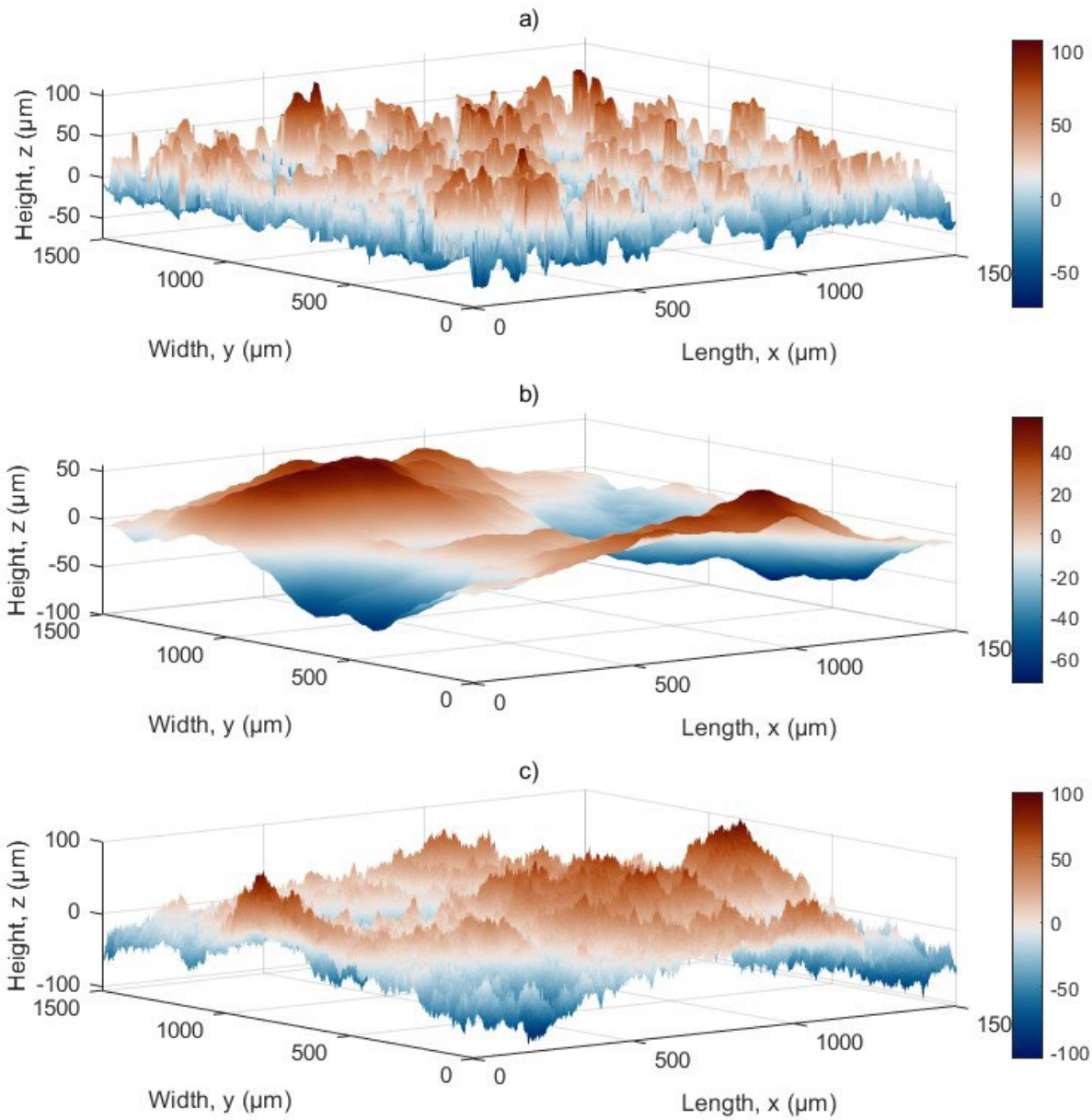

Figure 35: SWLI scan of sample 001 b) fractal artificial surface generated surface using Fourier method from SWLI line profile for sample 001 c) fractal artificial surface generated surface using roughness length method from SWLI line profile for sample 001

It is clear upon visual inspection that the Fourier spectral method of obtaining the Hurst component from β yields surfaces with larger scale features than the measured SWLI scan surface. These features are less pronounced and almost appear as if the surfaces are simply lacking small scale asperities that are characteristic of the as-built metal additive specimens. The roughness-length method yielded surfaces more similar to that measured by the SWLI. The fractals can be considering self-affine features over scales that are not present in the real additive surfaces. There are small features and asperities present, however there are also larger scale features that are not present in the flattened SWLI dataset. These surfaces are characterized in terms of their roughness parameters in the subsequent sections. All 14 specimens have undergone the same processes and have had surfaces extrapolated and generated from their datasets. Each surface measured, extrapolated, and generated has been characterized and recorded.

Chapter 5: Comparison of surfaces and evaluation of results across specimens

5.1 Surfaces extrapolated using the Fourier orthogonal spectrum method

The orthogonal spectrum surface extrapolation method was employed with both Dektak line scans and single profiles taken from SWLI scans on all 14 specimens considered in this study. Reviewing results of the extrapolation process for many specimens can ensure outlier results to not skew results. Once the orthogonal spectrum method extrapolates area datasets, they can be characterized using the surface roughness parameters defined in section 2.2.2. R_a/S_a and R_v/S_v/ are two parameters analyzed in the sections below. The BAC can also be explored and a Svk* parameter can also be extracted from the extrapolated surfaces. Generally, 1D profile parameters cannot be compared directly to 2D areal parameters, however in the context of this study the profile is used to produce both 1D and 2D parameters. This comparison can illustrate how much certain parameters change in magnitude with an increase in dimension. The areal parameters generated from the Dektak line and SWLI single profile are compared to their SWLI 3D surface scans for each specimen respectively. The graphs display the results for all specimens utilizing their numbering 1-14 to designate the results for each specimen independently.

5.1.1 Fourier orthogonal spectrum method from CSP datasets

In the figure below, the Dektak line profile average roughness and the average roughness of the orthogonally generated surface from the Dektak line are compared to the average roughness of the SWLI area scan for all 14 specimens.

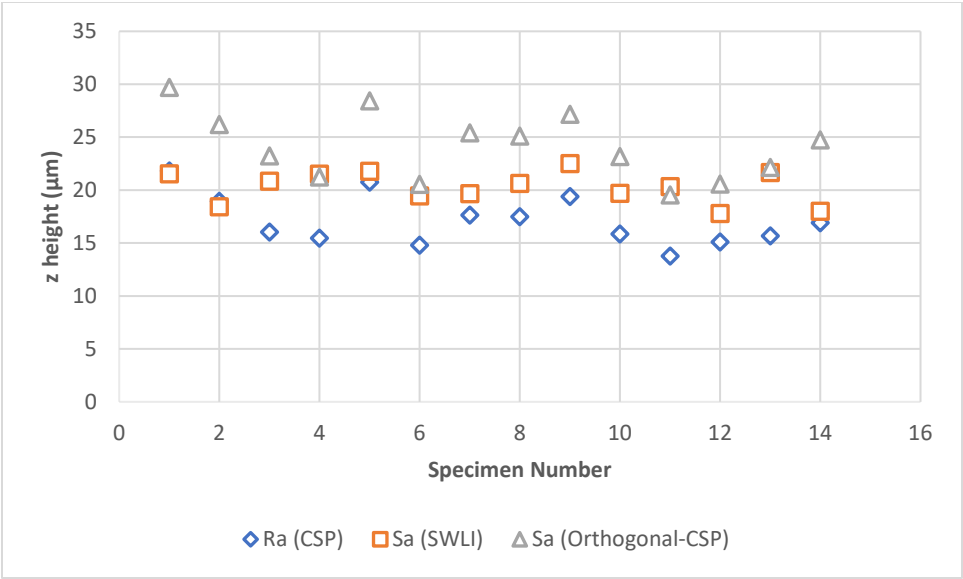

Figure 36: Average Roughness, CSP measured (R_a) & orthogonal (S_a) vs SWLI measured (S_a)

The surface generated from the Dektak line scan using the orthogonal spectrum extrapolation method has a slightly higher average roughness than the SWLI area scan across nearly all specimens. Unsurprisingly, the average roughness of the CSP line scan is lower than the areal parameters.

Now that the average roughness of the specimens has been better explored, the extrema of the extrapolated datasets can be analyzed. For relevancy to adjacent fatigue behavior exploration of metal additive specimens, maximum valley depth was chosen as an extrema value to be explored in this work. The maximum valley depth from the line datasets from both Dektak and SWLI scans were compared to the SWLI area scan, and the surfaces generated using the orthogonal spectrum method. Figure 37 below shows the comparison for Dektak-based data.

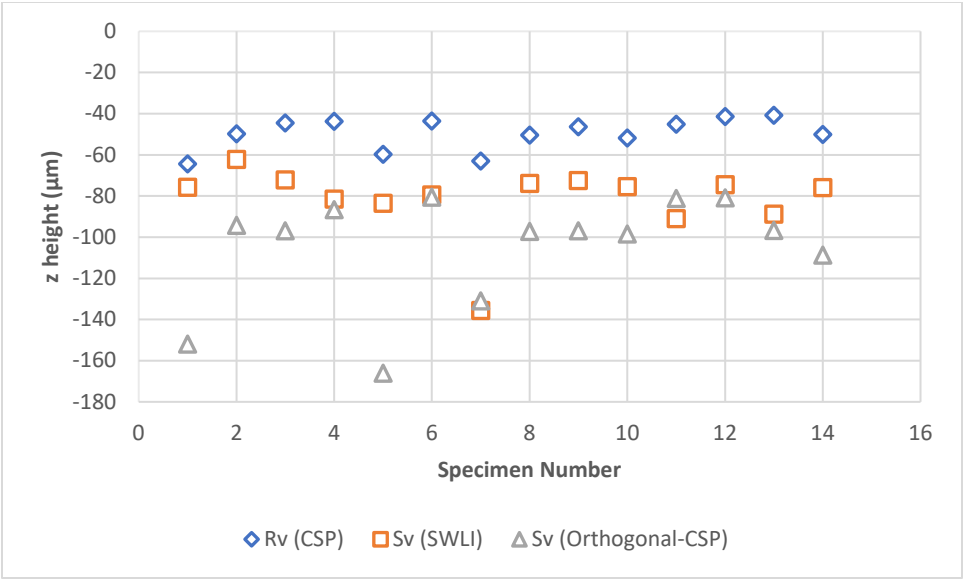

Figure 37: Valley Depth, Dektak measured (R_v) & orthogonal (S_v) vs SWLI measured (S_v)

The valley depths from the orthogonally generated surfaces were typically lower than the measured surfaces, both line profile and area scan. The line scans from the Dektak unsurprisingly had far less deep valley measurements for many reasons. The stylus may have skipped over the deepest parts of valleys, or may not have been aligned with the valleys in a manner that would have provided deep measurements. The surface extrapolated from the Dektak line profile had maximum valley depths closer to the area measured and the extrapolated values were usually larger in magnitude, providing a conservative estimate.

The extrema of the surfaces are not the only items of interest in the realm of fatigue. Cracks may often form in multiple locations on a surface not just the deepest point. Interactions between valleys and surrounding features often dictates the site of failure (7,8,22). A parameter being explored is the R_{vk}*(S_{vk}*) parameter which relates the extrema of a maximum valley depth to the core roughness of a specimen. The calculation method for this parameter was outlined in section 3.1 1, and the method was performed for all 14 specimens, for both line and area datasets.

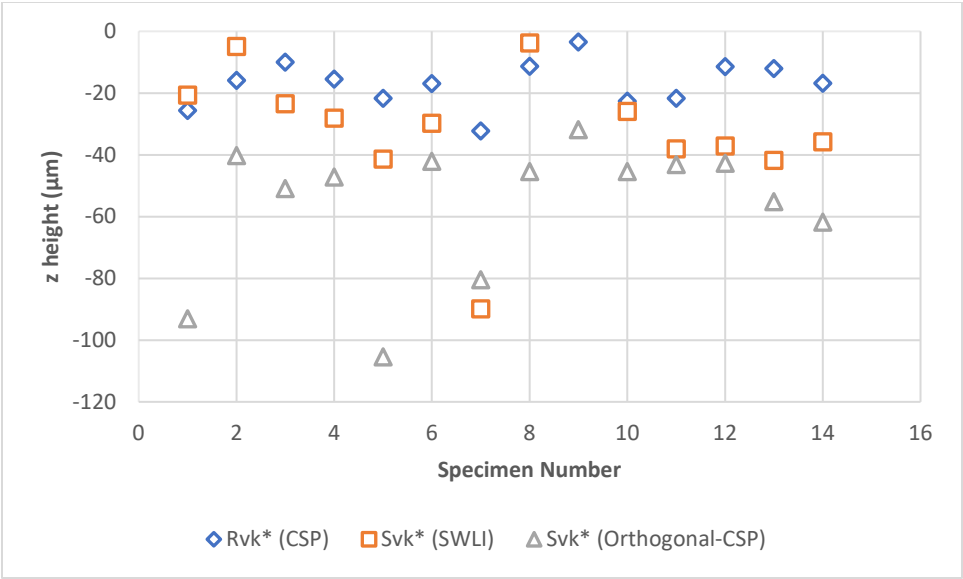

Figure 38: Total Valley Depth, Dektak measured (R_{vk}*) & orthogonal (S_{vk}*) vs SWLI measured (S_{vk}*)

The S_{vk}* for the surfaces extrapolated using the orthogonal spectrum method from a Dektak profile is deeper than the measured surface scan except for one case. The S_{vk}* parameters vary across a large range for the specimens, as they are dependent not only on the extrema, but on the core roughness as well. This results in large variations between specimens.

5.1.2 Fourier orthogonal spectrum method from SWLI datasets

In the figure below, the SWLI line profile average roughness and the average roughness of the orthogonally generated surface from the SWLI line profile are compared to the average roughness of the SWLI area scan for all 14 specimens.

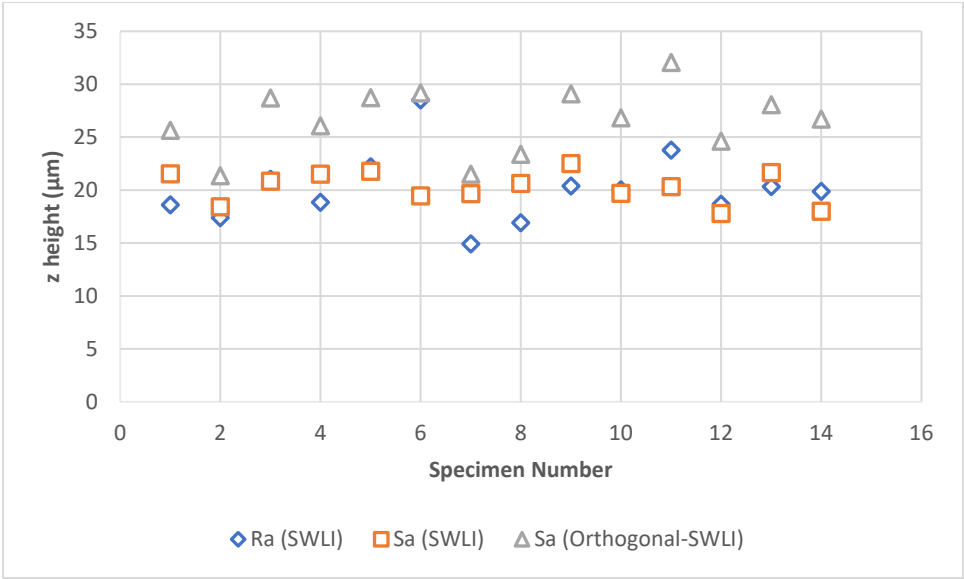

Figure 39: Average Roughness, SWLI measured (R_a) & orthogonal (S_a) vs SWLI measured (S_a)

The surface generated from the SWLI line profile dataset using the orthogonal spectrum method has significantly higher average roughness parameter values than the measured surface. There is a larger variation between the measured surface and the surface extrapolated from the SWLI profile than the surface extrapolated from the Dektak profile.

Figure 40 below shows the analysis for the extrapolated orthogonal method surface using a single profile from the SWLI scan.

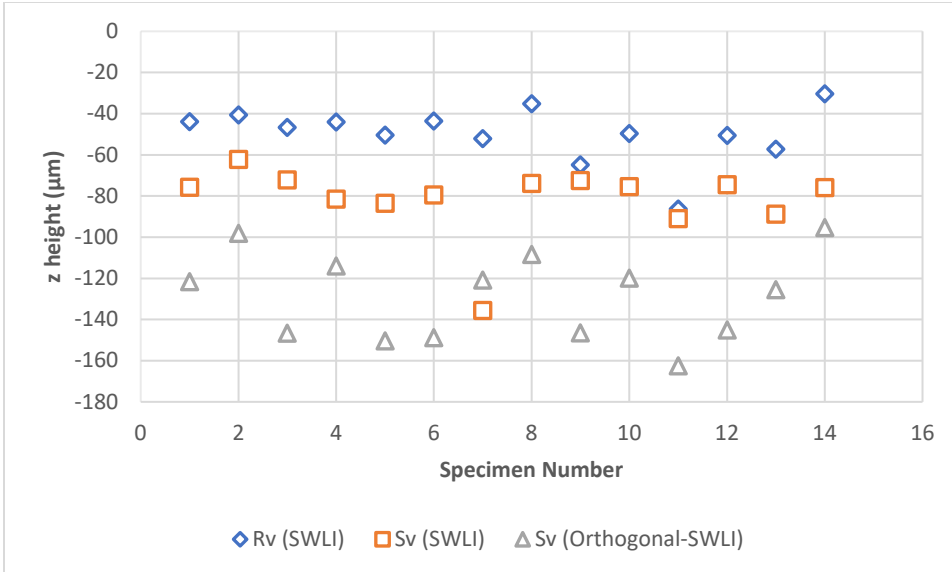

Figure 40: Valley Depth, SWLI measured (R_v) & orthogonal (S_v) vs SWLI measured (S_v)

The maximum valley depths from the SWLI line profiles are closer to the areal measured maximum valley depths suggesting that the working principle of the measurement technique can have an impact on the degree to which valleys are reported. The extrapolated surface from the SWLI line profile greatly overshot the valley depths recorded from the SWLI area scan. Since a deeper valley is more likely to cause surface fatigue, this method provides a conservative approximation. This suggests that the method employed can assist in predicting extrema from a surface when a limited contact-based measurement is taken. It must be reiterated that the comparison of these extrapolated surfaces and profile datasets is not direct with the surface height values but is with an optical area scan of the surface.

Similar analysis was performed for the SWLI line profile and extrapolated surface regarding the R_{vk}*/S_{vk}* parameter. This is shown below in figure 41.

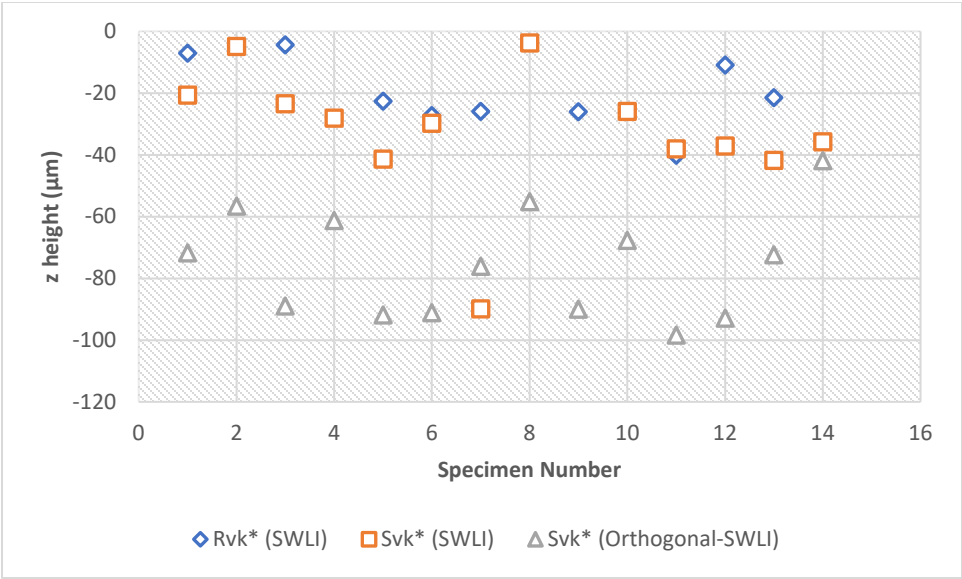

Figure 41: Total Valley Depth, SWLI measured (R_{vk}*) & orthogonal (S_{vk}*) vs SWLI measured (S_{vk}*)

Similar trends can be seen between the Dektak line scan and extrapolated surface, and for the SWLI line profile and extrapolated surface. The consistency in increased S_{vk}* valley depth suggests that the extrapolation method is producing the deeper parameters rather than the differences in working principles of the measurement techniques. There remails large variations in S_{vk}* parameters across different surfaces.

5.2 Fractal Dimension/Hurst Component and RMS roughness across specimens

The Hurst component was also used to generate surfaces with similar characteristics to the measured SWLI surface scan. The following figure 42 shows the Fractal Dimension and the directly related Hurst Component for all 14 specimens. These parameters were calculated from the

Dektak line scan data and the SWLI line profile using both the Fourier spectrum method (section 3.1.4) and the roughness length method (section 3.1.5).

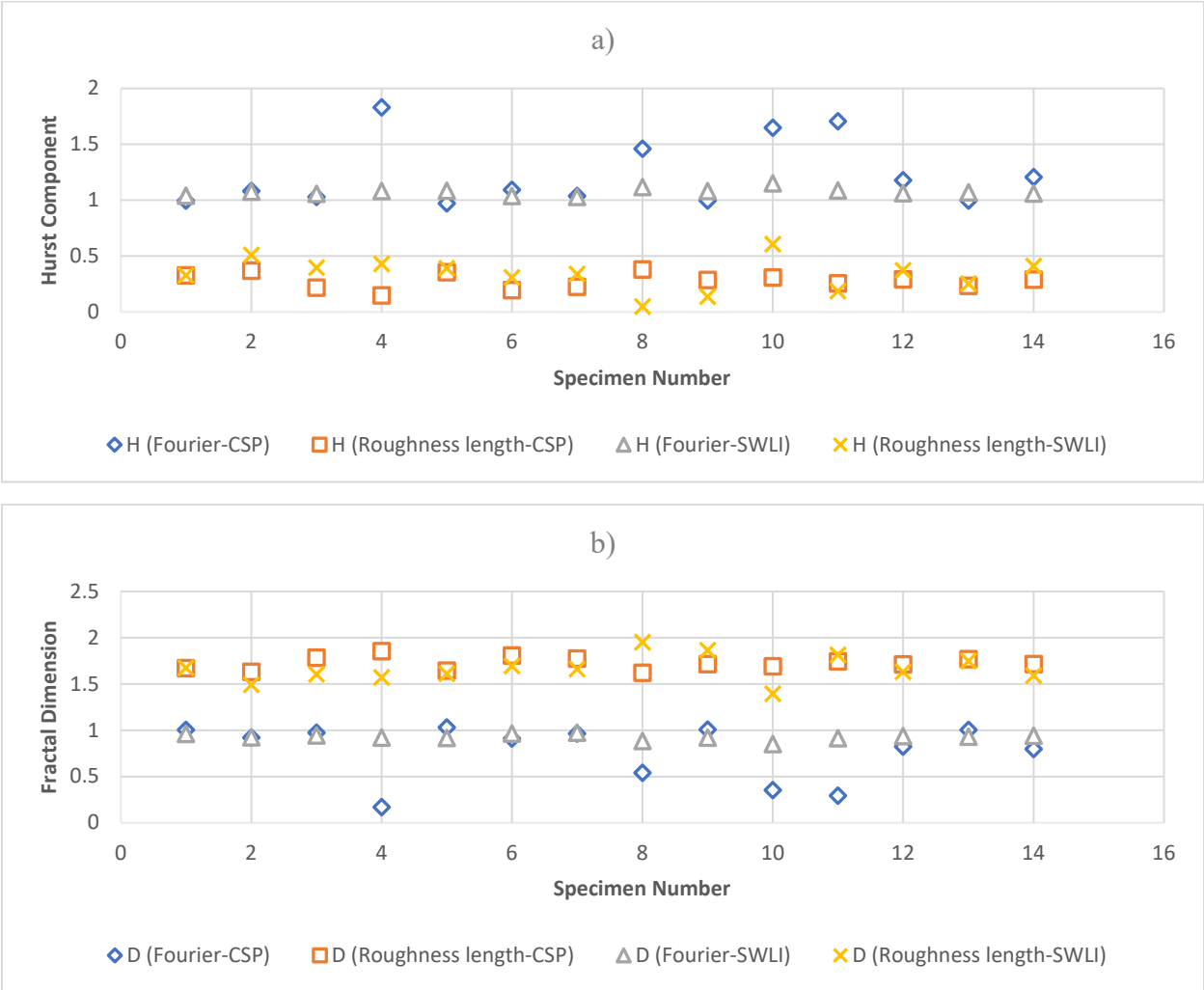

Figure 42: a) Fractal Dimensions b) Hurst Components

The Fractal Dimension, *D*, was consistently higher for the roughness length method of calculation compared to the Fourier analysis. The higher Fractal Dimension (lower Hurst Component) is characteristic of a rougher line, as *D* of 2 represents a plane and *D* of 1 represents a straight line. The Hurst Components appear relatively consistent for each method across all the specimens. The method of surface generation outlined in sections 3.1.4 and 3.1.5 utilizes both the

Hurst Component and the RMS roughness for each dataset from each specimen. Figure 43 below shows the RMS roughness for the Dektak line scan and SWLI line profile for each of the 14 specimens.

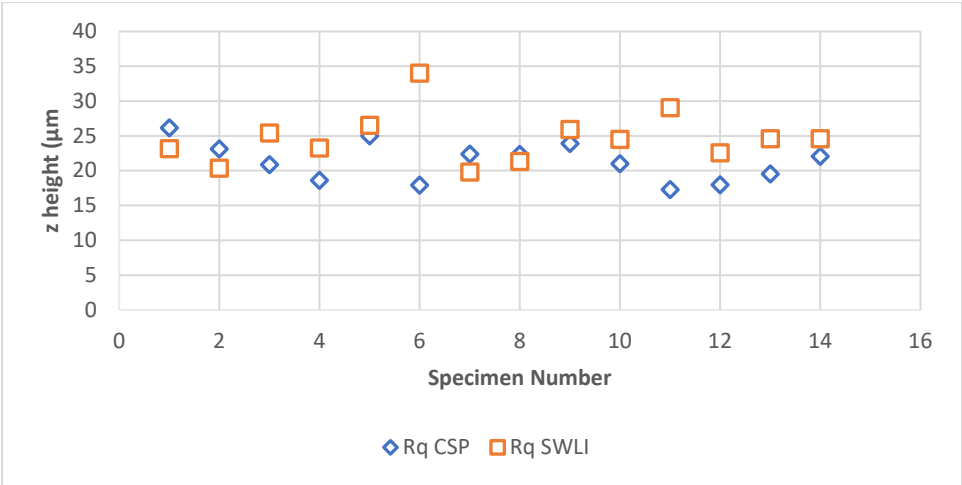

Figure 43: RMS roughness for Dektak line scan and SWLI line profile

The RMS roughness for the SWLI profile does not appear to directly correlate with the Dektak line scan data for the same samples. This can be attributed to many reasons. The profiles were obtained with different working principles, the profiles represent different locations on the specimens, and more. The SWLI RMS roughness is generally slightly higher than the RMS roughness of the Dektak line profile across most specimens. The degree to which the RMS roughness is higher varies significantly.

5.3 Surfaces generated using fractal artificial surface code

The "artificial_surf" (referenced in section 4.1.2 also called "fractal artificial surface) surface generation method was employed with both Dektak line scans and single profiles taken

from SWLI scans on all 14 specimens considered in this study. Once the function generates area datasets, they can be characterized using the surface roughness parameters defined in section 2.2.2. Average roughness and maximum valley depth are two surface roughness parameters emphasized in the sections below. The BAC can also be explored and a S_{vk}* parameter can also be extracted from the generated surfaces.

5.3.1 Fractal artificial surfaces from CSP datasets

Two different methods were used to extract Hurst Components from the Dektak line scan data. This results in two different surfaces generated from the same original line profile dataset. The Average roughness of the original Dektak line scan and the average roughness of the two generated surfaces is compared to the average roughness of the SWLI scan. This comparison is shown in the figure below.

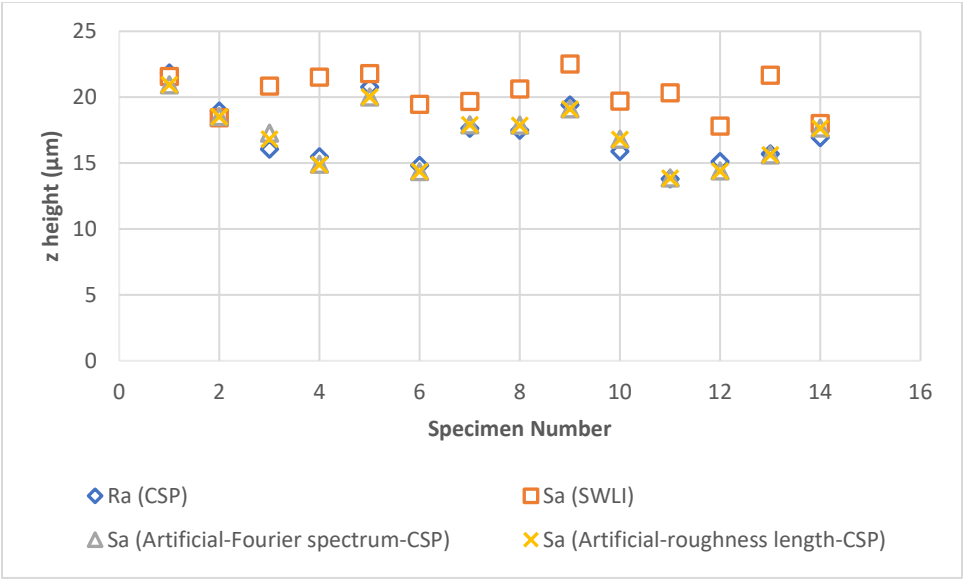

Figure 44: Average Roughness, Dektak measured (R_a) & Fourier generated (S_a) & roughness length generated (S_a) vs SWLI measured (S_a)

The average roughness values are relatively consistent between the surfaces. The average roughness from the Dektak line profile is the only line roughness parameter displayed in the plot above. The SWLI area scan average roughness is the highest, but in some cases, the roughness length generated surface approaches the same average roughness. This is because the generated surfaces were scaled according to RMS roughness.

The extrema of maximum valley depth were also analyzed in similar fashion to the orthogonally extrapolated surfaces. This is shown in figure 45 below.

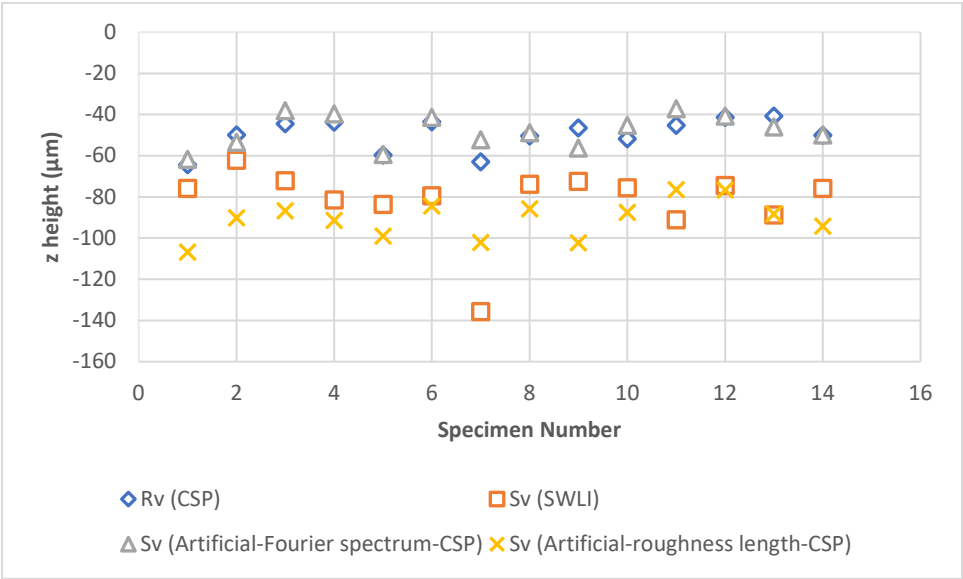

Figure 45: Maximum Valley Depth, Dektak measured (R_v) & Fourier generated (S_v) & roughness length generated (S_v) vs SWLI measured (S_v)

The line profile maximum valley depth from the Dektak scan used to generate the two surfaces is consistently shallower than the areal maximum valley depths from the generated surfaces. The Fourier method of calculating the Hurst Component provided larger H values than the roughness length method. That larger H produced surfaces that had far shallower valleys than the surface generated using the smaller H value from the roughness length method. The roughness

length method generated surface approximates the maximum valley depth of the SWLI area scan fairly accurately across the 14 specimens.

The BAC was also considered for these generated surfaces as it was for the orthogonally generated specimens. The R_{vk}*/S_{vk}* parameter has been calculated for all 14 specimens, and the results are shown in figure 46.

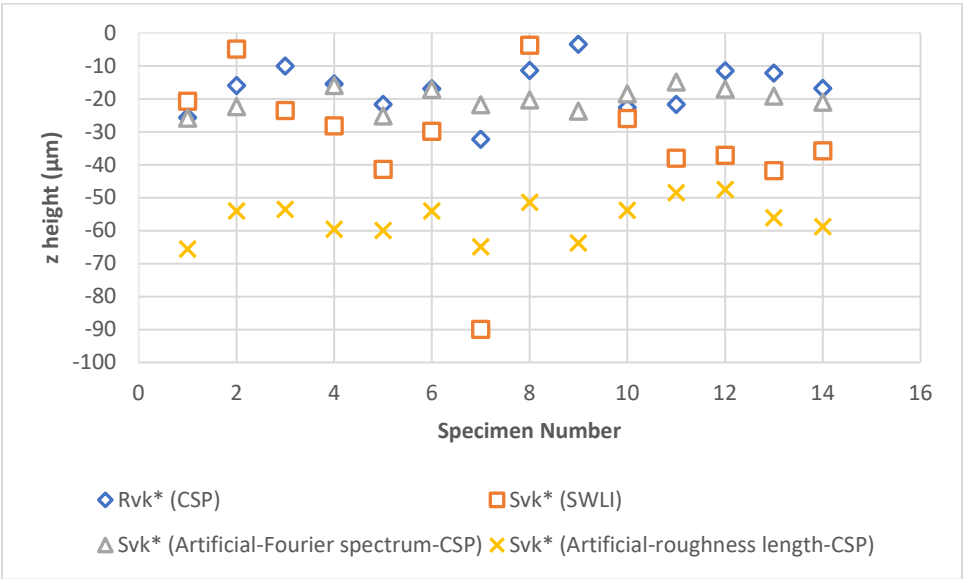

Figure 46: Total Valley Depth, Dektak measured (R_{vk}*) & Fourier generated (S_{vk}*) & roughness length generated (S_{vk}*) vs SWLI measured (S_{vk}*)

The S_{vk}* values for the SWLI area scan cover a wide height range. This is largely due to differences in the slope of the core roughness fit to the BAC curve. The S_v values were shown to vary in the previous figure, however they did not vary nearly to the degree that the S_{vk}* parameter does across the 14 specimens. As for the previous calculations, the roughness length generated surfaces produce deeper S_{vk}* parameter values than the Fourier method. These deep approximations are conservative with respect to fatigue life analysis. It should be noted that the fit of the core roughness varies so much for the measured SWLI surface that for specimen number 9

the S_{vk}* parameter was not able to be calculated as the fit overshot the measured S_v value for that surface. The surface generated using the Fourier analysis method was not able to be characterized with the S_{vk}* parameter for specimen 3 either. The parameter varies across a wide range for the other surfaces, but was ultimately able to be extracted for all other surfaces and profiles.

5.3.2 Fractal artificial surfaces from SWLI datasets

The Hurst component calculation method was also varied for the SWLI line profile. The average roughness was compared for these SWLI datasets as it was for the Dektak datasets in section 5.3.1. The results of the comparison are shown in figure 47 below.

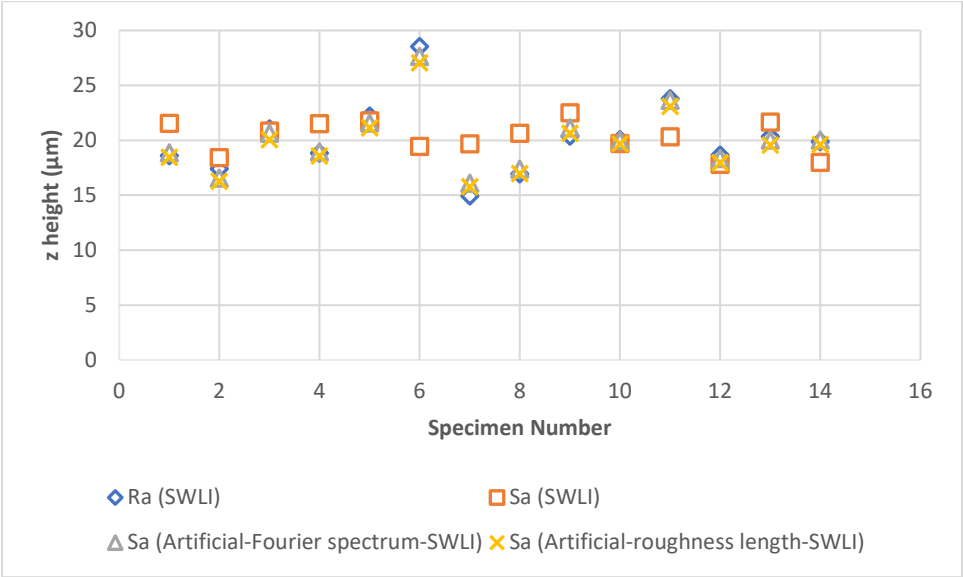

Figure 47: Average Roughness, SWLI measured (R_a) & Fourier generated (S_a) & roughness length generated (S_a) vs SWLI measured (S_a)

The SWLI area scan average roughness is not significantly higher or lower than the average roughness of the two generated surfaces. The variation of the average roughness of the measured

SWLI surfaces between samples is minimal. This is again because the generated surfaces are scaled to the RMS roughness.

The maximum valley depth was explored in similar fashion to the Dektak dataset in section 5.3.1. Figure 48 below conveys the maximum valley depths of the SWLI profile, SWLI original line, and two generated surfaces.

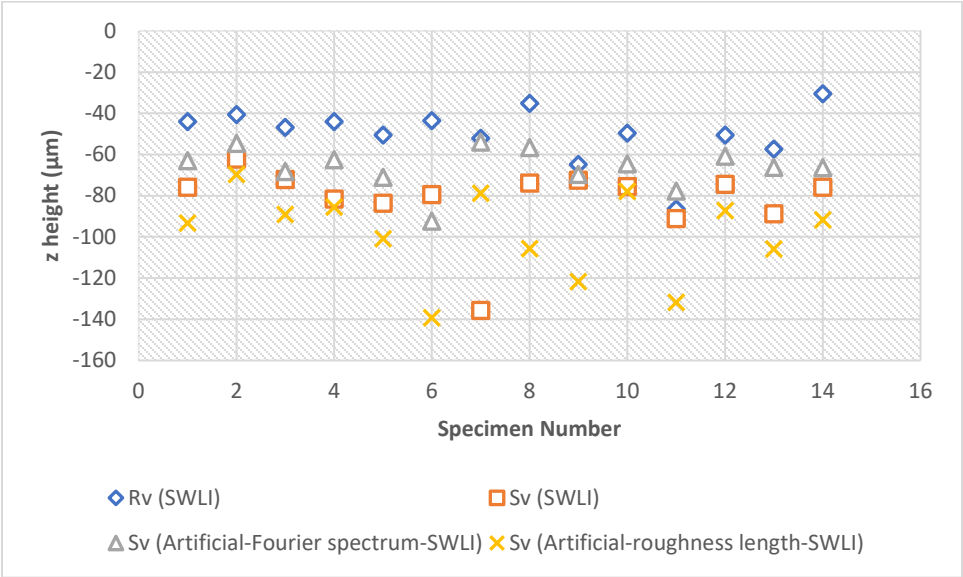

Figure 48: Maximum Valley Depth, SWLI measured (R_v) & Fourier generated (S_v) & roughness length generated (S_v) vs SWLI measured (S_v)

The maximum valley depth comparison performed for the Dektak line scan was done for the SWLI line data. Unsurprisingly the SWLI line profile maximum valley depth was reduced compared to the area scan, as it considers only a single profile of the area represented by the SWLI area scan. The Fourier method of calculating *H* produced larger values than the roughness length method in the same fashion as for the Dektak data. This time the maximum valley depths generated using the Fourier method were more accurate to the original SWLI scan, and the roughness length

method seems to slightly overshoot the valley depths recorded by the area scan, providing a conservative approximation.

The BAC was considered for the SWLI datasets. The dependance of the *Rvk**/*Svk** parameter on the slope of the core roughness fit line is more evident with the SWLI datasets than with the Dektak datasets. Some specimens were not able to be characterized with these parameters. The results of the BAC analysis on the SWLI-based surfaces and profiles are shown in figure 49 below.

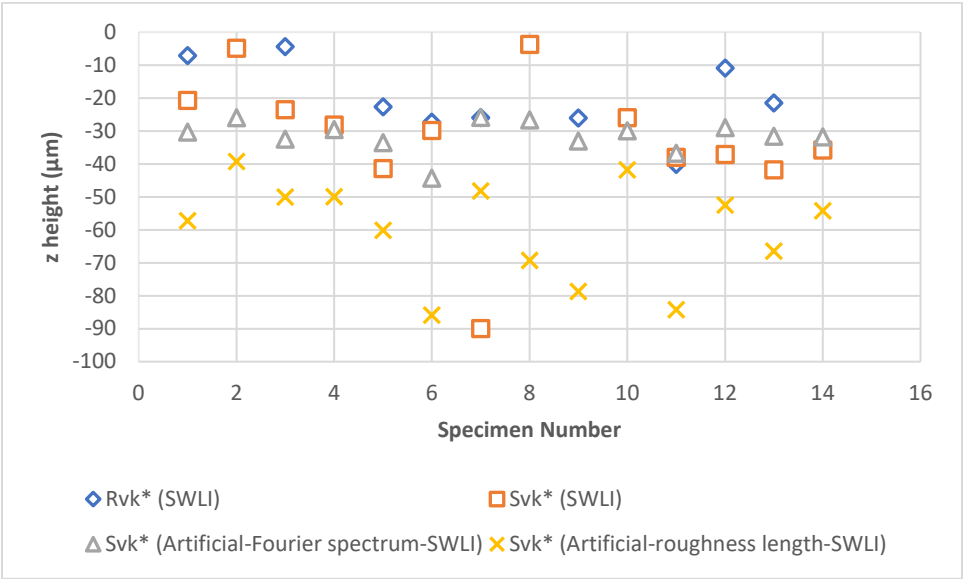

Figure 49: Total Valley Depth, SWLI measured (R_{vk}*) & Fourier generated (S_{vk}*) & roughness length generated (S_{vk}*) vs SWLI measured (S_{vk}*)

The single profile from the SWLI dataset proved difficult to characterize with this parameter. The SWLI line profiles from specimens 2, 4, 8, 10 & 14 were not able to be characterized with the R_{vk}* parameter, however all surfaces generated from the SWLI line profiles were able to be characterized with the S_{vk}* parameter. The S_{vk}* parameter for the SWLI area measurement varies so greatly it is difficult to determine if another surface provides a good

correlation, however the Fourier method appears to generate a surface with similar S_{vk}* parameters. The roughness length method from the SWLI area profile appears to overshoot the measured area for the SWLI measurement in the same manner as for the Dektak line scan data.

5.4 Skewness and Kurtosis from generated surfaces

The skewness and kurtosis parameters (defined in section 2.2.1 and 2.2.2) have been calculated for all generated and extrapolated surfaces. These parameters can be compared to those calculated from the measured surface taken from SWLI scans. The line profile scans used for generation also can produce parameters that can be compared to the surface parameters.

5.4.1 Skewness and Kurtosis of surfaces created using CSP profile

The CSP profile was used to generate an orthogonal surface and two fractal artificial surfaces. The global skewness and kurtosis parameters are plotted in the subsequent figure.

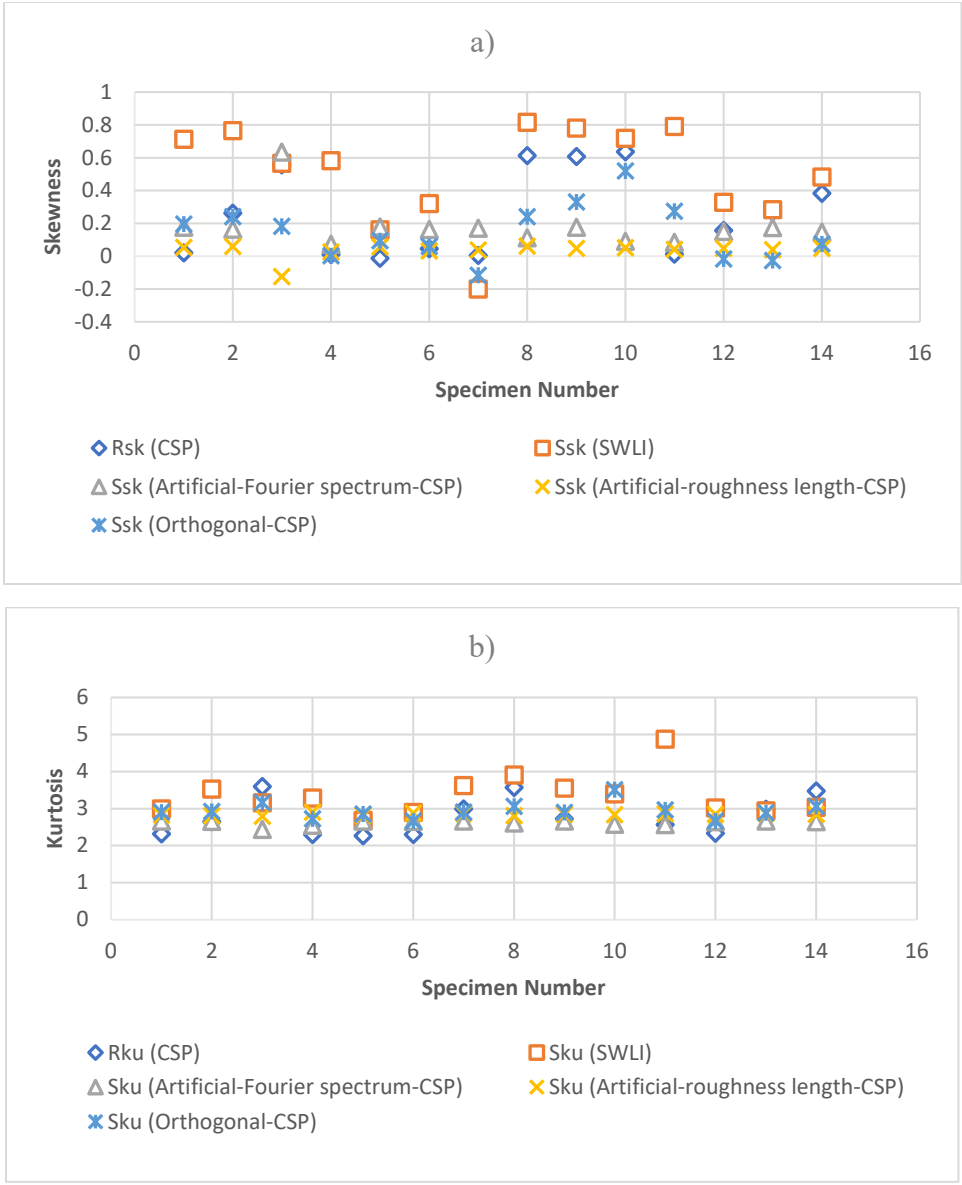

Figure 50: a) skewness parameter surfaces from CSP profile b) kurtosis parameter surfaces from CSP profile

Apart from specimen 7 (sample 016) the measured surface appears to have a consistently higher skewness parameter than the generated surfaces. The orthogonal surface has a slightly more positive trend in skewness across all specimens than the fractal artificial surfaces. The Fourier spectrum method surfaces appear to have a very slight positive trend compared to the

roughness length method surfaces with respect to their skewness. The kurtosis across the board is very near to 3 (Gaussian value). The measured SWLI surface has a slightly higher kurtosis than the others. There do not appear to be notable trends between the other surfaces. They all have a similar distribution around the mean.

5.4.2 Skewness and Kurtosis of surfaces created using SWLI profile

The selected SWLI profile was used to generate an orthogonal and two fractal artificial surfaces. The global skewness and kurtosis parameters are plotted in the subsequent figure.

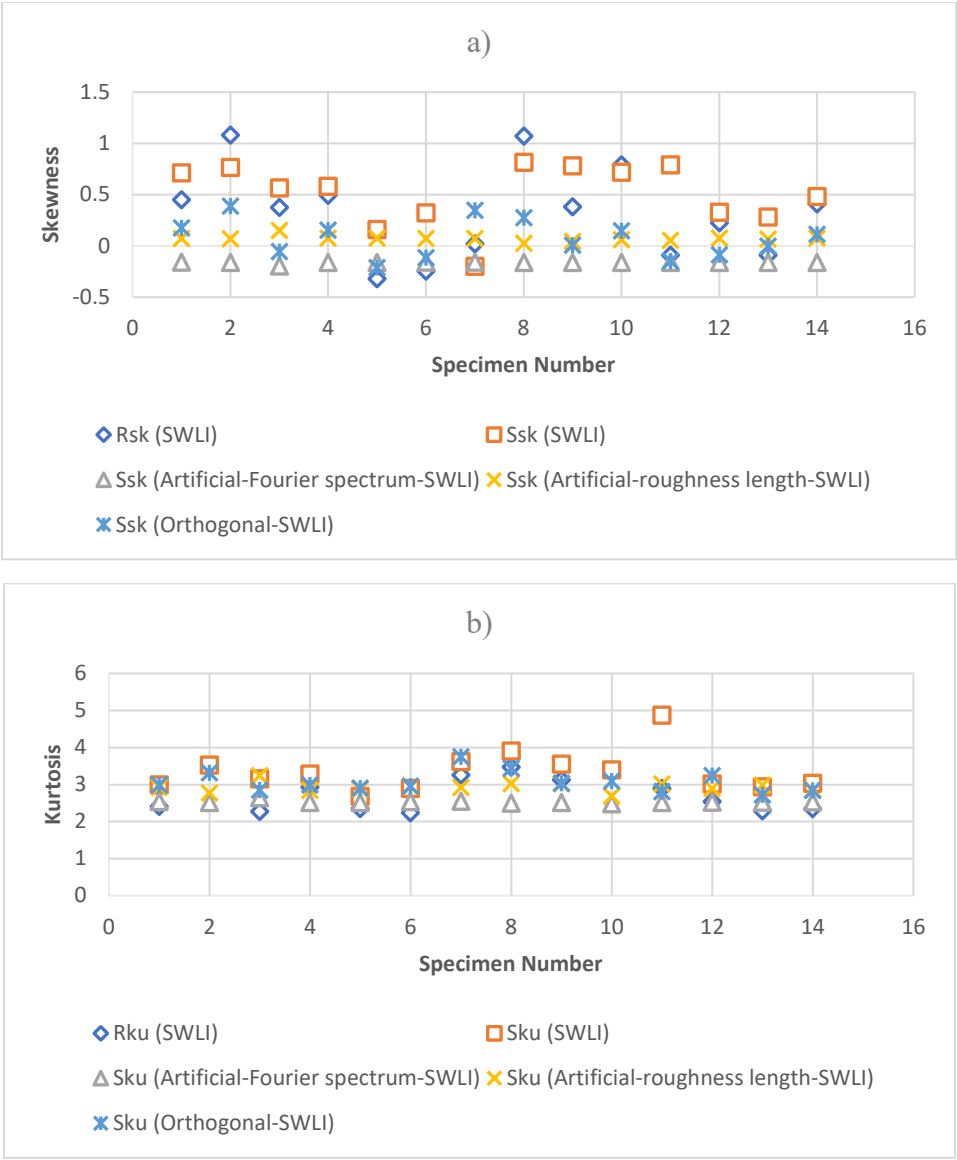

Figure 51: a) skewness parameter surface from SWLI profile b) kurtosis parameter surface from SWLI profile

The skewness parameters calculated for surfaces generated/extrapolated from the SWLI profile have more variation than those extracted from surfaces created using the CSP profile. The Fourier spectrum method and the roughness length method are very close to 0 with the Fourier method producing negative skewness values. The kurtosis variation is still small, and very comparable to the values in figure 50 centered around a value of 3.

Chapter 6: Conclusions and future work

It was shown that the original surface flattening post-processing methods have a non-negligible effect on the surface roughness parameters calculated. For the purpose of characterizing rough additive surfaces with relation to fatigue life, the orthogonal method consistently produces conservative estimates for both average roughness and extrema parameters. The parameters from the orthogonal method are typically of a greater magnitude than the measured surface's parameters. This means the maximum valleys in the orthogonal surface are significantly deeper than those from the original line profile and may more accurately represent an area compared to the single profile. The fractal artificially generated surfaces appear to have parameters closer to the measured surface than the orthogonal surfaces. The fractal artificially generated surfaces typically have roughness parameters of a lesser magnitude than the measured SWLI surfaces. These also appear to produce surfaces that are more gaussian in nature than the measured surfaces. The application of the surface measurement and characterization can determine which surface generation/extrapolation method is most applicable. For a close estimation to the measured surface, a fractal artificial surface may provide comparable results, this is largely due to the RMS roughness scaling the fractal artificial surfaces. For fatigue applications, understanding the extreme deep valleys can provide insight into crack initiation. To best estimate extrema for surfaces when provided with a single profile, the orthogonal method may best provide relevant and conservative surface predictions.

The total valley depth parameter (R_{vk}^{*}) is difficult to calculate for rough additive surfaces where many points on the profiles and area datasets are not close to the mean of the dataset. When the extrema do not differ significantly from the calculated core roughness the parameter can be

invalid. Since this parameter accounts for both a core roughness as well as the extrema (R_v/S_v) it may be a desirable parameter for fatigue life correlations. The difficulty in calculating this parameter means not every profile of rough additive surfaces can provide this parameter.

Standardization of a line-by-line flattening method could ideally make line profiles and area measurements more comparable. The area measurement post-processing techniques may vary depending on the technique and scan parameters used, but more work may be done to allow for better comparisons between line scans and area scans. Since the fractal artificial surfaces are scaled to RMS roughness, the results are muted. Statistical methods in future work could potentially be employed to modify the RMS value that is scaling the generated surfaces.

References

1. Kalender M, Kilic SE, Ersoy S, Bozkurt Y, Salman S. Additive Manufacturing and 3D Printer Technology in Aerospace Industry. In: 2019 9th International Conference on Recent Advances in Space Technologies (RAST) [Internet]. Istanbul, Turkey: IEEE; 2019 [cited 2024 Feb 21]. p. 689–94. Available from: https://ieeexplore.ieee.org/document/8767881/

2. Keating SJ, Gariboldi MI, Patrick WG, Sharma S, Kong DS, Oxman N. 3D Printed Multimaterial Microfluidic Valve. Pisignano D, editor. PLoS ONE. 2016 Aug 15;11(8):e0160624.

3. Das S, Bourell DL, Babu SS. Metallic materials for 3D printing. MRS Bull. 2016 Oct;41(10):729–41.

4. Soltani-Tehrani A, Isaac JP, Tippur HV, Silva DF, Shao S, Shamsaei N. Ti-6Al-4V powder reuse in laser powder bed fusion (L-PBF): The effect on porosity, microstructure, and mechanical behavior. International Journal of Fatigue. 2023 Feb;167:107343.

5. Maleki E, Bagherifard S, Bandini M, Guagliano M. Surface post-treatments for metal additive manufacturing: Progress, challenges, and opportunities. Additive Manufacturing. 2021 Jan;37:101619.

6. Pegues J, Roach M, Scott Williamson R, Shamsaei N. Surface roughness effects on the fatigue strength of additively manufactured Ti-6Al-4V. International Journal of Fatigue. 2018 Nov;116:543–52.

7. Pegues JW, Shamsaei N, Roach MD, Williamson RS. Fatigue life estimation of additive manufactured parts in the as-built surface condition. Mat Design & Process Comms [Internet]. 2019 Jun [cited 2024 Feb 22];1(3). Available from: https://onlinelibrary.wiley.com/doi/10.1002/mdp2.36

8. Lee S, Ahmadi Z, Pegues JW, Mahjouri-Samani M, Shamsaei N. Laser polishing for improving fatigue performance of additive manufactured Ti-6Al-4V parts. Optics & Laser Technology. 2021 Feb;134:106639.

9. Gurrappa I. Characterization of titanium alloy Ti-6Al-4V for chemical, marine and industrial applications. Materials Characterization. 2003 Oct;51(2–3):131–9.

10. Liu S, Shin YC. Additive manufacturing of Ti6Al4V alloy: A review. Materials & Design. 2019 Feb;164:107552.

11. Harun WSW, Manam NS, Kamariah MSIN, Sharif S, Zulkifly AH, Ahmad I, et al. A review of powdered additive manufacturing techniques for Ti-6al-4v biomedical applications. Powder Technology. 2018 May;331:74–97.

12. Nouira H, Salgado JA, El-Hayek N, Ducourtieux S, Delvallée A, Anwer N. Setup of a high-precision profilometer and comparison of tactile and optical measurements of standards. Meas Sci Technol. 2014 Apr 1;25(4):044016.

13. Vafadar A, Guzzomi F, Rassau A, Hayward K. Advances in Metal Additive Manufacturing: A Review of Common Processes, Industrial Applications, and Current Challenges. Applied Sciences. 2021 Jan 28;11(3):1213.

14. Mohd Yusuf S, Cutler S, Gao N. Review: The Impact of Metal Additive Manufacturing on the Aerospace Industry. Metals. 2019 Nov 29;9(12):1286.

15. Prashar G, Vasudev H, Bhuddhi D. Additive manufacturing: expanding 3D printing horizon in industry 4.0. Int J Interact Des Manuf. 2023 Oct;17(5):2221–35.

16. Mohanavel V, Ashraff Ali KS, Ranganathan K, Allen Jeffrey J, Ravikumar MM, Rajkumar S. The roles and applications of additive manufacturing in the aerospace and automobile sector. Materials Today: Proceedings. 2021;47:405–9.

17. Thompson SM, Bian L, Shamsaei N, Yadollahi A. An overview of Direct Laser Deposition for additive manufacturing; Part I: Transport phenomena, modeling and diagnostics. Additive Manufacturing. 2015 Oct;8:36–62.

18. Itoga H. Effect of surface roughness on step-wise S–N characteristics in high strength steel. International Journal of Fatigue. 2003 May;25(5):379–85.

19. Hu YN, Wu SC, Withers PJ, Zhang J, Bao HYX, Fu YN, et al. The effect of manufacturing defects on the fatigue life of selective laser melted Ti-6Al-4V structures. Materials & Design. 2020 Jul;192:108708.

20. Dev Singh D, Mahender T, Raji Reddy A. Powder bed fusion process: A brief review. Materials Today: Proceedings. 2021;46:350–5.

21. Shamsaei N, Yadollahi A, Bian L, Thompson SM. An overview of Direct Laser Deposition for additive manufacturing; Part II: Mechanical behavior, process parameter optimization and control. Additive Manufacturing. 2015 Oct;8:12–35.

22. Gockel J, Sheridan L, Koerper B, Whip B. The influence of additive manufacturing processing parameters on surface roughness and fatigue life. International Journal of Fatigue. 2019 Jul;124:380–8.

23. Raja J, Muralikrishnan B, Fu S. Recent advances in separation of roughness, waviness and form. Precision Engineering. 2002 Apr;26(2):222–35.

24. Jacobs TDB, Pastewka L, Guest Editors. Surface topography as a material parameter. MRS Bull. 2022;47(12):1205–10.

25. Lee DH, Cho NG. Assessment of surface profile data acquired by a stylus profilometer. Meas Sci Technol. 2012 Oct 1;23(10):105601.

26. Lu E, Zhang R, Liu J, Yin S, Ohmori H. Observation of ground surface roughness values obtained by stylus profilometer and white light interferometer for common metal materials. Surface & Interface Analysis. 2022 Jun;54(6):587–99.

27. Buchenau T, Mertens T, Lohner H, Bruening H, Amkreutz M. Comparison of Optical and Stylus Methods for Surface Texture Characterisation in Industrial Quality Assurance of Post-Processed Laser Metal Additive Ti-6Al-4V. Materials. 2023 Jul 4;16(13):4815.

28. Tammas-Williams S, Zhao H, Léonard F, Derguti F, Todd I, Prangnell PB. XCT analysis of the influence of melt strategies on defect population in Ti–6Al–4V components manufactured by Selective Electron Beam Melting. Materials Characterization. 2015 Apr;102:47–61.

29. Sun W, Brown S. An overview of industrial X-ray computed tomography. [Internet]. National Physical Laboratory; 2012 Jan p. 55. (NPL Report). Report No.: 32. Available from: http://eprintspublications.npl.co.uk/id/eprint/5385

30. Naresh K, Khan KA, Umer R, Cantwell WJ. The use of X-ray computed tomography for design and process modeling of aerospace composites: A review. Materials & Design. 2020 May;190:108553.

31. Myers NO. Characterization of surface roughness. Wear. 1962 May;5(3):182–9.

32. Thomas TR. Characterization of surface roughness. Precision Engineering. 1981 Apr;3(2):97–104.

33. Murthy TSR, Rao SY, Peters J. A Simple Approach for Evaluation of Cylindrical Surfaces. CIRP Annals. 1981;30(1):441–4.

34. Gao F, Leach RK, Petzing J, Coupland JM. Surface measurement errors using commercial scanning white light interferometers. Meas Sci Technol. 2008 Jan 1;19(1):015303.

35. Surface Roughness Measurement—Parameters [Internet]. Available from: https://www.olympus-ims.com/en/metrology/surface-roughness-measurement-portal/parameters/#!cms[focus]=cmsContent14708

36. Kastner J, Heinzl C. X-Ray Tomography. In: Ida N, Meyendorf N, editors. Handbook of Advanced Nondestructive Evaluation [Internet]. Cham: Springer International Publishing; 2020 [cited 2024 Feb 21]. p. 1–73. Available from: https://link.springer.com/10.1007/978-3-319-30050-4_5-2

37. IS0-25178-2:2012 Geometric product specification (GPS)- surface texture: Area-part 2: Terms, definition and surface texture parameters. 2012.

38. Sayles RS, Thomas TR. Surface topography as a nonstationary random process. Nature. 1978 Feb;271(5644):431–4.

39. Dobrzanski P, Pawlus P. Digital filtering of surface topography: Part I. Separation of one-process surface roughness and waviness by Gaussian convolution, Gaussian regression and spline filters. Precision Engineering. 2010 Jul;34(3):647–50.

40. Le Goic G, Favreliere H, Samper S, Formosa F. Multi scale modal decomposition of primary form, waviness and roughness of surfaces. Wiley Online Library. 2011 Sep;33(5):271–395.

41. Zhu S, Huang P. Influence mechanism of morphological parameters on tribological behaviors based on bearing ratio curve. Tribology International. 2017 May;109:10–8.

42. Geometrical Product Specificaition (GPS) - Surface texture: Profile method; Surfaces having stratified functional properties- Part 2: Height characterization using the linear material ratio curve [Internet]. 1996. Available from: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://cdn.standards.iteh.ai/samples/22280/22426a423a044669ac1fe13f70d7e9f2/ISO-13565-2-1996.pdf

43. Mike Stewart. A New Approach to the Use of Bearing Area Curve [Internet]. Society of Manufacturing Engineers; 1990. Available from: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://numericalengineering.com/wp-content/uploads/2018/03/Training-A-New-Approach-to-the-Use-of-Bearing-Area-Curve-FC900229.pdf

44. ISO 21920-2:2021 Geometrical product specifications (GPS) — Surface texture: Profile Part 2: Terms, definitions and surface texture parameters [Internet]. 2021. Available from: https://www.iso.org/standard/72226.html

45. Precision Devices. SURFACE ROUGHNESS TERMINOLOGY AND PARAMETERS [Internet]. Available from: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.predev.com/pdffiles/surface_roughness_terminology_and_parameters.pdf

46. DIN 4776 Rauheitsmessung; Kenngrößen RK, RPK, RVK, Mr1, Mr2 zur Beschreibung des Materialanteils im Rauheitsprofil; Meßbedingungen und Auswerteverfahren [Internet]. 1990. Available from: https://www.dinmedia.de/en/standard/din-4776/1541840

47. Jackson RL. The Effect of Scale-Dependent Hardness on Elasto-Plastic Asperity Contact between Rough Surfaces. Tribology Transactions. 2006 Jul;49(2):135–50.

48. Glen D. Johnson, Arkady Tempelman, Ganapati P. Patil. FRACTAL BASED METHODS IN ECOLOGY: A REVIEW FOR ANALYSIS AT MULTIPLE SPATIAL SCALES. Special Statistical Ecology Issue. 1995 Autumn;10(2/3):123–321.

49. Zhang X, Jackson RL. An Analysis of the Multiscale Structure of Surfaces with Various Finishes. Tribology Transactions. 2017 Jan 2;60(1):121–34.

50. Fractal geometry: what is it, and what does it do? Proc R Soc Lond A. 1989 May 8;423(1864):3–16.

51. Wang Y, Azam A, Wilson MC, Neville A, Morina A. Generating fractal rough surfaces with the spectral representation method. Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology. 2021 Dec;235(12):2640–53.

52. Kogut L, Jackson RL. A Comparison of Contact Modeling Utilizing Statistical and Fractal Approaches. Journal of Tribology. 2006 Jan 1;128(1):213–7.

53. Zhang X, Xu Y, Jackson RL. An analysis of generated fractal and measured rough surfaces in regards to their multi-scale structure and fractal dimension. Tribology International. 2017 Jan;105:94–101.

54. Julius O. Smith. Introduction to the DFT. In: Mathematics of the Discrete Fourier Transform (DFT): With Audio Applications [Internet]. 2nd ed. W3K Publishing; 2007. Available from:

https://books.google.com/books?hl=en&lr=&id=fTOxS9huzHoC&oi=fnd&pg=PA1&dq=Discrete+Fourier+Transform+dft&ots=DYg7AKMibt&sig=XKWvTXttxN8IKAodJeSXNh1WGMg#v=onepage&q=Discrete%20Fourier%20Transform%20dft&f=false

55. Paul Heckbert. Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm. 1998 Jan 27;Computer Graphics(2). Available from: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/99/pub/www/notes/fourier/fourier.pdf

56. Cooley JW, Tukey JW. An algorithm for the machine calculation of complex Fourier series. Math Comp. 1965;19(90):297–301.

57. Huw Fox, Bill Bolton. Mathematics for Engineers and Technologists [Internet]. Elsevier; 2002 [cited 2024 Jun 13]. Available from: https://linkinghub.elsevier.com/retrieve/pii/B9780750655446X50000

58. Majumdar A, Tien CL. Fractal characterization and simulation of rough surfaces. Wear. 1990 Mar;136(2):313–27.

59. Mandelbrot BB. The fractal geometry of nature. Updated and augm. ed. New York: W. H. Freeman; 1986.

60. Malinverno A. A simple method to estimate the fractal dimension of a self-affine series. Geophysical Research Letters. 1990 Oct;17(11):1953–6.

61. Eric W. Weisstein. Fourier Series [Internet]. Wolfram MathWorld; Available from: https://mathworld.wolfram.com/FourierSeries.html

62. Mona Mahboob Kanafi. Surface generator: artificial randomly rough surfaces [Internet]. Available from: https://www.mathworks.com/matlabcentral/fileexchange/60817-surface-generator-artificial-randomly-rough-surfaces

63. Walter Beneson, John W.Harris. Handbook of Physics [Internet]. illustrated ed. Springer Science & Business Media, 2006; 2006. 1186 p. Available from: https://books.google.com/books?id=c60mCxGRMR8C&source=gbs_navlinks_s

64. Gujrati A, Sanner A, Khanal SR, Moldovan N, Zeng H, Pastewka L, et al. Comprehensive topography characterization of polycrystalline diamond coatings. Surf Topogr: Metrol Prop. 2021 Mar 1;9(1):014003.

65. Persson BNJ. On the Fractal Dimension of Rough Surfaces. Tribol Lett. 2014 Apr;54(1):99–106.

Appendix

Appendix A: 1.2 Roughness, Waviness and Form Figure generation

```
%% 1.2 Roughness, Waviness and Form

% Parameters
n = 3000; % ~ resoluiton of SWLI
x = linspace(0, 1500, n); % 1.5 mm SWLI line profile
amplitude = 300; % arbitrary
waviness_amplitude = 100; % arbitrary
waviness_frequency = 0.02; % arbitrary
roughness_amplitude = 100; %
nominal_shape = amplitude * (1 - (2*x/1500 - 1).^2);
waviness = waviness_amplitude * sin(waviness_frequency * x);
roughness = roughness_amplitude * smoothdata(randn(1, n), 'gaussian', 50);
combined_surface = nominal_shape + waviness + roughness;
% figure generation
figure
subplot(4, 1, 1);
plot(x, combined_surface);
% title('Combined Surface: Nominal Shape + Waviness + Roughness');
ylabel('Height, z (µm)')
legend('a)', 'Location', 'northwest');
set(gca, 'XTickLabel', []);
ylim([-50, 500]);
subplot(4, 1, 2);
plot(x, nominal_shape);
% title('Nominal Shape (Form)');
ylabel('Height, z (µm)')
legend('b)', 'Location', 'northwest');
set(gca, 'XTickLabel', []);
ylim([-50, 350]);
subplot(4, 1, 3);
plot(x, waviness);
% title('Sinusoidal Waviness');
ylabel('Height, z (µm)')
legend('c)', 'Location', 'northwest');
set(gca, 'XTickLabel', []);
ylim([-110, 110]);
subplot(4, 1, 4);
plot(x, roughness);
% title('Small Scale Roughness');
xlabel('Profile Length (µm)')
ylabel('Height, z (µm)')
legend('d)', 'Location', 'northwest');
ylim([-100, 100]);
```

Appendix B: Distribution and Histogram Figure generation

```
%% 2.2.1 Distribution and Histogram
```

```
%Gaussian distribution
mean = 0; % Mean
sigma = 20; % Standard deviation
x = linspace(mean - 4*sigma, mean + 4*sigma, 1000);
PDF = (1/(sigma * sqrt(2*pi))) * exp(-0.5 * ((x - mean)/sigma).^2);
% Plot the Gaussian Dsitribution (probability density function)
figure;
plot(x, PDF, 'Color', [0 0 1], 'LineWidth', 2, 'DisplayName', 'Gaussian Curve');
hold on;
fill([x fliplr(x)], [PDF zeros(1, length(PDF))], [ 0 0 1] , 'FaceAlpha', 0.2,
'DisplayName', 'Total probability');
plot([mean mean], [0 max(PDF)], 'Color', [0.5 0.5 0.5], 'LineWidth', 2,
'DisplayName', 'Mean (\mu)');
plot([mean + sigma mean + sigma], [0 max(PDF)], 'Color', [1 0.5 0], 'LineWidth', 2,
'DisplayName', '\mu + \sigma');
plot([mean - sigma mean - sigma], [0 max(PDF)], 'Color', [1 0.5 0], 'LineWidth', 2,
'DisplayName', '\mu - \sigma');
xlabel('Height value, z (µm)');
ylabel('Probability Density (µm^-^1)');
legend;
% Add grid
grid on;
hold off;

% histogram from sample 001 line scan
% load the scan
N1 = xlsread('N1.xlsx'); % 3 mm line scan
Lmatrix = length(N1);
start = .1 * Lmatrix;
finish = .9 * Lmatrix;
pz_D(:,1) = N1(start:finish,1);
pz_D(:,2) = N1(start:finish,2);
Length_D = 0.8*3000; % um
% flatten the line profile
pz_flat_D = polyfit(pz_D(:,1),pz_D(:,2),2);
zval_D = polyval(pz_flat_D,pz_D(:,1));
pz_D(:,2) = pz_D(:,2) - zval_D;
% histogram plot
figure
H_Dektak = histogram(pz_D(:,2));
xlabel('height, z (µm)')
ylabel('number of datapoints')
legend('Bins')
```

Appendix C: Example surface generation and area BAC figure generation

```
%% 2.2.2 surface generation

load CrameriColourMaps7.0.mat; % colormap

% Define parameters
```

```
[x, y] = meshgrid(-5:0.1:5, -5:0.1:5);
num_peaks = 50;
% Generate random peak heights and valley depths
peak_heights = rand(1, num_peaks) * 2;
valley_depths = -rand(1, num_peaks) * 2;
z = zeros(size(x));
for i = 1:num_peaks
    peak_x = randn * 2;
    peak_y = randn * 2;
    z = z + peak_heights(i) * exp(-((x - peak_x).^2 + (y - peak_y).^2));
    valley_x = randn * 2;
    valley_y = randn * 2;
    z = z + valley_depths(i) * exp(-((x - valley_x).^2 + (y - valley_y).^2));
end

figure
surf(x, y, z, 'EdgeColor', 'none');
axis off
set(gca,'XColor', 'none','YColor','none')
set(gca, 'color', 'none');
colormap(vik)


%% 3.1.1 BAC

% use existing surface
figure
surf(x, y, z, 'EdgeColor', 'none');
axis off
set(gca,'XColor', 'none','YColor','none')
set(gca, 'color', 'none');
colormap(vik)
% Add a transparent flat plane
h_plane = 2; % Set the height of the plane
hold on
z_plane = ones(size(x)) * h_plane; % Create a matrix of the plane height
h = surf(x, y, z_plane, 'FaceColor', 'red', 'FaceAlpha', 0.25, 'EdgeColor', 'none');
% Add the plane
hold off
```

Appendix D: Main Code

The code in this section was run for all 14 specimens. There are a few marked fields that are varied to change which datasets are loaded into the code, however all calculations remain consistent between specimens. The example shown below is for sample 001.

```
% ROUGH SURFACE ANALYSIS
% MASTERS THESIS 2024
% Loren Baugh
```

```
clc;
close all;
clear all;

%% 2.1.1 Loading Line Data Dektak scan
% Scan Information:
% 3 mm line scan
N1 = xlsread('N1.xlsx'); % -- change depending on specimen being considered
Lmatrix = length(N1);
start = .1 * Lmatrix; % -- crop 10% off each side
finish = .9 * Lmatrix;
pz_D(:,1) = N1(start:finish,1);
pz_D(:,2) = N1(start:finish,2);
Length_D = 0.8*3000; % µm

%% 2.1.2 Loading Area Data SWLI scan
% Scan Information:
% 1.5 x 1.5 mm area scan
% Specify range and delimiter
opts = delimitedTextImportOptions("NumVariables", 3);
opts.DataLines = [14, Inf];
opts.Delimiter = ",";
% Specify column names and types
opts.VariableNames = ["WykoASCIIDataFileFormat1", "VarName2", "VarName3"];
opts.VariableTypes = ["double", "double", "double"];
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";
% Import the data
X4 = readtable("N.ASC", opts);
load CrameriColourMaps7.0.mat;
% clear temporary variables
clear opts
close all
data=table2array(X4);
x=[0:1:3067];
y=[0:1:3062];
px= 0.489687; % -- pixel size
x1=x.*px;
y1=y.*px;
height=3063;
width=3068;
z=reshape(data(:,3),[height,width])./1000; % control units

%% 2.3.1 Line scan flattening Dektak data

pz_flat_D = polyfit(pz_D(:,1),pz_D(:,2),2);
zval_D = polyval(pz_flat_D,pz_D(:,1));

figure('Position', [100, 100, 900, 300]);
subplot(2,1,1);
plot(pz_D(:,1),pz_D(:,2),'r')
hold on
plot(pz_D(:,1),zval_D,'b')
title("a)",'FontWeight','Normal')
```

```
% legend('Unflattened Data','Fit Curve','location','northwest')
xlabel('Profile Length (µm)')
ylabel(['Height raw (µm)'])
set(gca,"FontSize",11)
pbaspect([6 1 1]) % or [10,1,1]
pz_D(:,2) = pz_D(:,2) - zval_D;
subplot(2,1,2);
plot(pz_D(:,1),pz_D(:,2))
title("a)",'FontWeight','Normal')
xlabel('Profile Length (µm)')
ylabel(['Height flat (µm)'])
set(gca,"FontSize",11)
pbaspect([6 1 1]) % or [10,1,1]
Line_gca_D = gcf;
exportgraphics(Line_gca_D,'Line_Flattening_Dektak.jpg')

% roughness parameter calculation for Dektak line scan
Ra_Dektak = sum(abs(pz_D(:,2)))/(length(pz_D));
Rv_Dektak = min(pz_D(:,2));
Rp_Dektak = max(pz_D(:,2));
Rz_Dektak = abs(Rv_Dektak) + abs(Rp_Dektak);
Rq_Dektak = sqrt((1/(length(pz_D)))*(sum(pz_D(:,2).^2)));
Sk_Dektak = skewness(pz_D(:,2));
Ku_Dektak = kurtosis(pz_D(:,2));

%% 2.3.2 Data Fill with Interpolation (Dr. Jackson's technique) SWLI - external
% Appendix E
[zfill] = interpdatafill(z);

%% 2.3.2 SWLI plane flattening / line by line flattening comparison SWLI

[X, Y] = meshgrid(x1, y1);
points = [X(:), Y(:), zfill(:)];
points(isnan(points))=0;

% plane flattening - Appendix F
[flatsurf] = planeflat(points, zfill, height, width);

% area plot plane flattened surface
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1);
s = surf(x1,y1,zfill,'edgecolor','none');
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
zlim([-130 190])
subplot(1,2,2);
s1 = surf(x1,y1,flatsurf,'edgecolor','none');
hold on
```

```
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
Surf_gca = gcf;
exportgraphics(Surf_gca,'Plane_Flattening_SWLI.jpg')

% areal roughness parameters calculations for plane flattened surface
f=reshape(flatsurf,[],1);
sa_flatsurf=(sum(abs(f)))/(length(f));
sq_flatsurf=sqrt(sum(abs(((f.^2))))/length(f));
sp_flatsurf=max(f);
sv_flatsurf=min(f);
sz_flatsurf=abs(sp_flatsurf)+abs(sv_flatsurf);
ssk_flatsurf=skewness(f);
sku_flatsurf=kurtosis(f);

% plane flattening line by line function - Appendix G
[flatlinesurf_y,flatlinesurf_x] = planelineflat(zfill, height, width, x1, y1);
% Flatlinesurf_y iterates in the y direction and considers profiles along
% the x axis
% Flatlinesurf_x iterates in the x direction and considers profiles along
% the y axis
croplengthy = height - 1;
y1flatlinesurf = y1(1:croplengthy);

% area plot line by line flattened surface
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1);
s = surf(x1,y1,zfill,'edgecolor','none');
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
direction = [1 1 1];
zlim([-130 190])
subplot(1,2,2);
s2 = surf(x1,y1flatlinesurf,flatlinesurf_y,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
direction = [1 1 1];
Surf_linebyline_gca = gcf;
exportgraphics(Surf_linebyline_gca,'LinebyLine_Flattening_SWLI.jpg')
```

```
% plane flattening line by line y direction method
% area plot flattened surface
croplengthy = height - 1;
y1flatlinesurf = y1(1:croplengthy);

% areal roughness parameters calculations from line flattened surface
fline_y=reshape(flatlinesurf_y,[],1);
sa_flatlinesurf_y=(sum(abs(fline_y)))/(length(fline_y));
sq_flatlinesurf_y=sqrt(sum(abs(((fline_y.^2))))/length(fline_y));
sp_flatlinesurf_y=max(fline_y);
sv_flatlinesurf_y=min(fline_y);
sz_flatlinesurf_y=abs(sp_flatlinesurf_y)+abs(sv_flatlinesurf_y);
ssk_flatlinesurf_y=skewness(fline_y);
sku_flatlinesurf_y=kurtosis(fline_y);
% comparison of plane flattening and line by line y flattening
flatsurfcrop_y = flatsurf(1:croplengthy,1:width);
differenceflatsurfs_y = flatsurfcrop_y - flatlinesurf_y;
figure('Position', [100, 100, 450, 300]);
s3 = surf(x1,y1flatlinesurf,differenceflatsurfs_y,'edgecolor','none');
hold on
colorbar
colormap(vik)
% title('difference between plane and line by line y flattened surfaces')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('δ, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
Surf_deltaflattening_gca = gcf;
exportgraphics(Surf_deltaflattening_gca,'Delta_Flattening_SWLI.jpg')

%% 2.3.3 choose line to evaluate from SWLI area scan

% line below selects the single Line considered from the area
LineSelect = 1500; % randomized
Line = flatsurf(LineSelect,:);
Line = Line';
Line_x = x1';
Line_ = [Line_x(:),Line];
zfill_Line = zfill(LineSelect,:);
zfill_Line = zfill_Line';
zfill_Line_ = [Line_x(:),zfill_Line];

% flattened line plot
figure('Position', [100, 100, 900, 250]);
plot(Line_(:,1),Line_(:,2))
% title('flattened original line')
xlabel('Profile Length (µm)')
ylabel('height (µm)')
pbaspect([6 1 1])
set(gca,"FontSize",11)
Line_gca = gcf;
exportgraphics(Line_gca,'Line_SWLI.jpg')
```

```
% line roughness parameters calculations flattened line
Ra_flatline=(sum(abs(Line_(:,2))))/(length(Line_(:,2)));
Rq_flatline=sqrt(sum(abs(((Line_(:,2).^2))))/length(Line_(:,2)));
Rp_flatline=max(Line_(:,2));
Rv_flatline=min(Line_(:,2));
Rz_flatline=abs(Rp_flatline)+abs(Rv_flatline);
Rsk_flatline=skewness(Line_(:,2));
Rku_flatline=kurtosis(Line_(:,2));

% convert the entire area scan into a single line
flatsurfline = reshape(flatsurf,1,[])';

%% 3.1.1 bearing area curve Dektak line scan

Line_HvalsN1_D = pz_D(:,2);
span_D = round(Rp_Dektak + abs(Rv_Dektak));
mr_D = zeros(span_D,1);
Line_above_D = zeros(span_D,1);
Hi_D = 1;
for H_D = round(Rp_Dektak):-1:round(Rv_Dektak)
    Line_above_D(Hi_D) = sum(Line_HvalsN1_D >= H_D);
    Hi_D = Hi_D + 1;
end
mr_D = 100*Line_above_D/length(Line_HvalsN1_D);
Hthresh_D = [round(Rp_Dektak):-1:round(Rv_Dektak)]';
startmr_D = round(0.3*span_D);
endmr_D = round(0.7*span_D); % -- use center 40% for determining the fit
mr_cut_D = mr_D(startmr_D:endmr_D);
Hthresh_cut_D = Hthresh_D(startmr_D:endmr_D);
[mrxData_D,mryData_D] = prepareCurveData(mr_cut_D,Hthresh_cut_D);
ft_D = fittype('poly1');
[mrfitresult_D,gof] = fit(mrxData_D,mryData_D,ft_D);
mrslopeN1_D = mrfitresult_D.p1;
last_point_D = length(Hthresh_D);
last_height_D = mrfitresult_D.p1 * mr_D(last_point_D) + mrfitresult_D.p2;
% calculate the difference between the fit and deepest valley
Rvk_D = Rv_Dektak - last_height_D;
Rvk_Dektak = Rvk_D;

%% 3.1.1 bearing area curve line SWLI scan

Line_Hvals = Line_(:,2);
span = round(Rp_flatline + abs(Rv_flatline));
mr = zeros(span,1);
Line_above = zeros(span,1);
Hi = 1;
for H = round(Rp_flatline):-1:round(Rv_flatline)
    Line_above(Hi) = sum(Line_Hvals >= H);
    Hi = Hi + 1;
end
mr = 100*Line_above/length(Line_Hvals);
Hthresh = [round(Rp_flatline):-1:round(Rv_flatline)]';
startmr = round(0.3*span);
endmr = round(0.7*span);
mr_cut = mr(startmr:endmr);
```

```
Hthresh_cut = Hthresh(startmr:endmr);
[mrxData,mryData] = prepareCurveData(mr_cut,Hthresh_cut);
ft = fittype('poly1');
[mrfitresult,gof] = fit(mrxData,mryData,ft);
mrslope = mrfitresult.p1;
last_point = length(Hthresh);
last_height = mrfitresult.p1 * mr(last_point) + mrfitresult.p2;
% calculate the difference between the fit and deepest valley
Rvk_S = Rv_flatline - last_height;
Rvk_SWLI = Rvk_S;

%% 3.1.1 bearing area curve surface SWLI scan

surf_Hvals =  flatsurf;
surfspan = round(sp_flatsurf + abs(sv_flatsurf));
mrsurf = zeros(surfspan,1);
surf_above = zeros(surfspan,1);
Hnsurf = 1;
for Hsurf = round(sp_flatsurf):-1:round(sv_flatsurf)
    surf_above(Hnsurf) = sum(surf_Hvals(:) >= Hsurf);
    Hnsurf = Hnsurf + 1;
end
[Lmrsurf,Wmrsurf] = size(surf_Hvals);
mrsurf = 100*surf_above/(Lmrsurf*Wmrsurf);
Hthreshsurf = [round(sp_flatsurf):-1:round(sv_flatsurf)]';
startmrsurf = round(0.3*surfspan);
endmrsurf = round(0.7*surfspan);
mrsurf_cut = mrsurf(startmrsurf:endmrsurf);
Hthreshsurf_cut = Hthreshsurf(startmrsurf:endmrsurf);
[mrsurfxData,mrsurfyData] = prepareCurveData(mrsurf_cut,Hthreshsurf_cut);
surfft = fittype('poly1');
[mrsurffitresult,gof] = fit(mrsurfxData,mrsurfyData,surfft);
mrsurfslope = mrsurffitresult.p1;
last_point_S = length(Hthreshsurf);
last_height_S = mrsurffitresult.p1 * mrsurf(last_point_S) + mrsurffitresult.p2;
% calculate the difference between the fit and deepest valley
Svk_S = sv_flatsurf - last_height_S;
Svk_SWLI = Svk_S;

%% 3.1.4 FFT Work Fourier and conjugate methods from Dektak line

samplingrate_D = 1/(mean(diff(pz_D(:,1))));

% sample calculation of DFT
[dft_spectrum_D, dft_spectrum_norm_D] = dft(pz_D(:,2)); % -- external function

% spectrum from fft funciton
spectrum_D = 2*fft(pz_D(:,2)); % -- the factor of two accounts for positive and
negative frequency components
spectrum_norm_D = abs(spectrum_D/(length(spectrum_D)));
spectrum_half_D = spectrum_norm_D(1:floor((length(spectrum_norm_D))/2+1)); % --
positive frequency components of normalized spectrum
% evaluation of sample DFT and fft functions
max_diff_spectrum_D = max(abs(dft_spectrum_D - spectrum_D));
% amplitude & frequency calculations
```

```
lambda_D = 1./((0:length(spectrum_D)-1)*samplingrate_D/length(spectrum_D));
amplitude_D = abs(spectrum_norm_D(2:end));
amplitude_half_D = abs(spectrum_half_D(2:end)); % use spectrum_half to plot
% phase calculation
phase_D = angle(spectrum_D);
phase_4Quad_D = imag(spectrum_D);
lambda_plot_D = lambda_D(1:1920); % -- manually selected 0.5 Dektak line
% alternative to spectrum obtained using the complex conjugate
zconj_D = fft(pz_D(:,2))/length(pz_D(:,2)); % zf = fft(z)/N;
zconj_D(:) = sqrt(zconj_D(:).*conj(zconj_D(:))); %zf(:)=sqrt(zf(:).*conj(zf(:)));
F_amplitude_D = zconj_D(:); %-- use in single sided amplitude spectrum instead of
spectr1
F_amplitude_half_D = F_amplitude_D(1:floor((length(F_amplitude_D))/2+1));

fplotvalues_D = samplingrate_D*(0:(length(spectrum_D)/2))/length(spectrum_D);
% Data is prepared in order to set up scatterplot for D
slope_y_values_D = log10(spectrum_half_D.^2);
slope_x_values_D = log10(fplotvalues_D);
[xData_D,yData_D] = prepareCurveData(slope_x_values_D,slope_y_values_D);
ft_D = fittype('poly1');
[fitresult_D,gof] = fit(xData_D,yData_D,ft_D);
Beta_D = fitresult_D.p1; % slope resulting from trendline
% calculate fractal dimension assuming the KPZ equation which describes statistical
growth
FractalDimensionKPZ_Line_Dektak = (4 + Beta_D)/2;
H_Fourier_Dektak = 2 - FractalDimensionKPZ_Line_Dektak;
% generate graph for paper
figure
plot(fitresult_D,xData_D,yData_D)
legend('Fourier spectrum','linear fit','Location','NorthEast');
xlabel('log(f) (Hz)');
ylabel('log(magnitude^2) (µm^2)');
grid on
set(gca,"FontSize",11)
Fourier_D = gcf;
exportgraphics(Fourier_D,'Fourier_Dektak.jpg')


%% 3.1.4 FFT Work Fourier and conjugate methods from SWLI line


samplingrate = 1/(mean(diff(Line_(:,1))));

% sample calculation of DFT
[dft_spectrum, dft_spectrum_norm] = dft(Line); % -- external function

% spectrum from fft funciton
spectrum = 2*fft(Line_(:,2)); % -- the factor of two accounts for positive and
negative frequency components
spectrum_norm = abs(spectrum/(length(spectrum)));
spectrum_half = spectrum_norm(1:floor((length(spectrum_norm))/2+1)); % -- positive
frequency components of normalized spectrum
% evaluation of sample DFT and fft functions
max_diff_spectrum = max(abs(dft_spectrum - spectrum));
% amplitude & frequency calculations
```

```
lambda = 1./((0:length(spectrum)-1)*samplingrate/length(spectrum));
amplitude = abs(spectrum_norm(2:end));
amplitude_half = abs(spectrum_half(2:end)); % use spectrum_half to plot
% phase calculation
phase = angle(spectrum);
phase_4Quad = imag(spectrum);
lambda_plot = lambda(1:1534);
% alternative to spectrum obtained using the complex conjugate
zconj = fft(Line_(:,2))/length(Line_(:,2)); % zf = fft(z)/N;
zconj(:) = sqrt(zconj(:).*conj(zconj(:))); %zf(:)=sqrt(zf(:).*conj(zf(:)));
F_amplitude = zconj(:); %-- use in single sided amplitude spectrum instead of spectr1
F_amplitude_half = F_amplitude(1:floor((length(F_amplitude))/2+1));
F_amplitude_sum = sum(F_amplitude(:)); % sum the amplitudes (if all peaks
% are lined up this is the highest amplitude expected)-not sure about this tbh

fplotvalues = samplingrate*(0:(length(spectrum)/2))/length(spectrum);
% Data is prepared in order to set up scatterplot for D
slope_y_values = log10(spectrum_half.^2);
slope_x_values = log10(fplotvalues);
[xData,yData] = prepareCurveData(slope_x_values,slope_y_values);
ft = fittype('poly1');
[fitresult,gof] = fit(xData,yData,ft);
Beta = fitresult.p1; % slope resulting from trendline
FractalDimensionKPZ_Line = (4 + Beta)/2;
H_Fourier_SWLI = 2 - FractalDimensionKPZ_Line;

%% 3.1.5 plot RMS roughness wrt scale single line Dektak data

% RMS Roughness Section from line selected
R_qn_D = 2; % -- start of the sliding window
R_zvals_D = pz_D(:,2); % -- height values considered for the R_q calculations
R_qave_D = mean(R_zvals_D(1:2)); % -- average of 1st two height values
s_qzvals_D = (R_zvals_D(1:2)-R_qave_D).^2; % -- squared differences between first two
height values and their average
LR_D = length(spectrum_D) - 1;
R_q_D = zeros(1,LR_D);
R_q_D(1) = sqrt(sum(s_qzvals_D)/R_qn_D); % -- calculates and assignes first R_q value
R_qindex_D = 2;
for R_qi_D = 3:length(R_zvals_D);
    R_qzi_D = R_zvals_D(1:R_qi_D); % -- z vals considered for each window
    R_qn_D = length(R_qzi_D);
    R_qave_D = mean(R_qzi_D);
    s_qzvals_D = (R_qzi_D - R_qave_D).^2;
    R_q_D(R_qindex_D) = R_q_D(R_qindex_D) + sqrt(sum(s_qzvals_D)/R_qn_D);
    R_qindex_D = R_qindex_D+1;
end
R_q_log_D = log10(R_q_D);
cutoff_R_q_log_D = 2;
R_q_log_inrange_D = R_q_log_D < cutoff_R_q_log_D;
R_q_log_cut_D = R_q_log_D(R_q_log_inrange_D);
R_q_x_distance_D = linspace(1, Length_D, length(R_q_log_cut_D));
slope_x_values_R_q_log_D = log10(R_q_x_distance_D);
[xData_R_q_D,yData_R_q_D] = prepareCurveData( slope_x_values_R_q_log_D, R_q_log_cut_D
);
ft_R_q_D = fittype('poly1');
```

```
[fitresult_R_q_D, gof] = fit(xData_R_q_D,yData_R_q_D,ft_R_q_D);
Hurst_R_q_D = fitresult_R_q_D.p1; % -- Hurst Component
FractalDimension_R_q_D = 2 - Hurst_R_q_D; % -- Fractal Dimension
H_Rq_Dektak = Hurst_R_q_D; % -- rename variable

% plot for paper
figure
plot(fitresult_R_q_D,xData_R_q_D,yData_R_q_D)
legend( 'Roughness length', 'linear fit', 'Location', 'SouthEast' );
xlabel('window length, log(x) (µm)');
ylabel('roughness, log(R_q) (µm)');
grid on
set(gca,"FontSize",11)
Rq_D = gcf;
exportgraphics(Rq_D,'Roughness_length_Dektak.jpg')

%% 3.1.5 plot RMS roughness wrt scale single line SWLI data

% RMS Roughness Section from line selected
R_qn = 2; % -- start of the sliding window
R_zvals = Line_(:,2); % -- height values considered for the R_q calculations
R_qave = mean(R_zvals(1:2)); % -- average of 1st two height values
s_qzvals = (R_zvals(1:2)-R_qave).^2; % -- squared differences between first two
height values and their average
LR = length(spectrum) - 1;
R_q = zeros(1,LR);
R_q(1) = sqrt(sum(s_qzvals)/R_qn); % -- calculates and assignes first R_q value
R_qindex = 2;
for R_qi = 3:length(R_zvals);
    R_qzi = R_zvals(1:R_qi); % -- z vals considered for each window
    R_qn = length(R_qzi);
    R_qave = mean(R_qzi);
    s_qzvals = (R_qzi - R_qave).^2;
    R_q(R_qindex) = R_q(R_qindex) + sqrt(sum(s_qzvals)/R_qn);
    R_qindex = R_qindex+1;
end
R_q_log = log10(R_q);
cutoff_R_q_log = 2;
R_q_log_inrange = R_q_log < cutoff_R_q_log;
R_q_log_cut = R_q_log(R_q_log_inrange);
slope_x_values_R_q_log = log10([1:length(R_q_log_cut)]);
[xData_R_q,yData_R_q] = prepareCurveData( slope_x_values_R_q_log, R_q_log_cut );
ft_R_q = fittype('poly1');
[fitresult_R_q, gof] = fit(xData_R_q,yData_R_q,ft_R_q);
Hurst_R_q = fitresult_R_q.p1; % -- Hurst Component
FractalDimension_R_q = 2 - Hurst_R_q; % -- Fractal Dimension

%% 4.1.1 replicate original line from Dektak line scan (Orthogonal Spectrum Method)

% is starting with the second term in the spectrum, wavelength = L/(i-1)
rng(1); % sets reproducibility
phase_random_D = ((randi(90,1,length(dft_spectrum_D)))*2*pi)/180;
phase_anglesolution_D = angle(dft_spectrum_D);
height_manual_D = zeros(1,length(dft_spectrum_D));
n_D = 2;
```

```
% i defines spectrum
% n defines x
scanlength_D=0.0024; % 2.4 mm for Dektak scan (3 mm - 20%)
deltax_D=scanlength_D/(length(dft_spectrum_D)-1);
for i_D=2:(length(dft_spectrum_D)/2)
    phase_line_D(i_D) = atan(imag(dft_spectrum_D(i_D))/real(dft_spectrum_D(i_D))); %
evaluate against phase_anglesolution
    lambda_line_D(i_D)=scanlength_D/(i_D-1);
    for n_D = 1:length(dft_spectrum_D);
        height_manual_D(n_D) =
height_manual_D(n_D)+dft_spectrum_norm_D(i_D)*(cos(2*pi*(n_D-
1)*deltax_D/lambda_line_D(i_D)+phase_anglesolution_D(i_D)));
    end
end

% figure for paper
figure('Position', [100, 100, 1200, 300]);
plot(pz_D(:,1),pz_D(:,2))
xlabel('scan distance (µm)')
ylabel('height (µm)')
pbaspect([6 1 1]) % or [10,1,1]
set(gca,"FontSize",11)
hold on
% plot generated surface
plot(pz_D(:,1),height_manual_D(:))
xlabel('scan distance (µm)')
ylabel('height (µm)')
pbaspect([6 1 1]) % or [10,1,1]
set(gca,"FontSize",11)
Line_gca = gcf;
legend('Original profile','Reconstructed profile')
exportgraphics(Line_gca,'Line_replicate_ortho_D.jpg')


%% 4.1.1 replicate original line from SWLI line from area scan (Orthogonal Spectrum
Method)
% ------- dft_spectrum used to experiment
% is starting with the second term in the spectrum, wavelength = L/(i-1)
rng(1); % sets reproducibility
phase_random = ((randi(90,1,length(dft_spectrum)))*2*pi)/180;
phase_anglesolution = angle(dft_spectrum);
height_manual = zeros(1,length(dft_spectrum));
n = 2;
% i defines spectrum
% n defines x
scanlength=0.0015; % 1.5 mm for SWLI scan
deltax=scanlength/(length(dft_spectrum)-1);
for i=2:(length(dft_spectrum)/2)
phase_line(i) = atan(imag(dft_spectrum(i))/real(dft_spectrum(i))); % evaluate against
phase_anglesolution
lambda_line(i)=scanlength/(i-1);
for n = 1:length(dft_spectrum);
    height_manual(n) = height_manual(n)+dft_spectrum_norm(i)*(cos(2*pi*(n-
1)*deltax/lambda_line(i)+phase_anglesolution(i)));
end
```

```
end

% plot original surface
figure('Position', [100, 100, 1200, 300]);
plot(Line_(:,1),Line_(:,2))
xlabel('scan distance (um)')
ylabel('height (um)')
pbaspect([6 1 1]) % or [10,1,1]
set(gca,"FontSize",11)
Line_gca = gca;
hold on
% plot generated surface
plot(Line_(:,1),height_manual(:))
xlabel('scan distance (um)')
ylabel('height (um)')
pbaspect([6 1 1]) % or [10,1,1]
set(gca,"FontSize",11)
Line_gca = gcf;
legend('Original profile','Reconstructed profile')
exportgraphics(Line_gca,'Line_replicate_ortho_S.jpg')

%% 4.1.1 Surface Extrapolation Area from single Dektak line Sin/Cos orthogonal method

% -- with the phase_random in the y direction are the roughness values
% similar to area measurements?
% -- try running this with 1000 x 1000 or 500 x 500 to get it in a testable
% speed
% i defines spectrum
% n defines x
% m defines y
% scanlength_D=0.0024;
deltay_area_D= scanlength/height;
deltax_area_D= scanlength/width;
y_resolution = height; % round(y1(end))
x_resolution = width; % round(x1(end))
rng(1); % sets reproducibility
height_manual_area_D = zeros(y_resolution,x_resolution); % length(spectr) -instead of
500
n_D = 2;
m_D = 2;
for i_D=2:500 %length(spectr)/2 -instead of 50
phase_area_D(i_D) = atan(imag(spectrum_D(i_D))/real(spectrum_D(i_D)));
lambda_area_D(i_D)=scanlength/(i_D-1);
for m_D = 1:y_resolution;
for n_D = 1:x_resolution;  %length(spectr);
    height_manual_area_D(m_D,n_D) =
height_manual_area_D(m_D,n_D)+spectrum_norm_D(i_D)*(sin(2*pi*(n_D-
1)*deltax_area_D/lambda_area_D(i_D)+phase_anglesolution_D(i_D)))+(spectrum_norm_D(i_D
)*(cos(2*pi*(m_D-1)*deltay_area_D/lambda_area_D(i_D)+phase_anglesolution_D(i_D))));
end
end
end

% roughness parameters from extrapolated area
heightmanualarea__D=reshape(height_manual_area_D,[],1);
```

```
sa_heightmanualarea_D=(sum(abs(heightmanualarea__D)))/(length(heightmanualarea__D));
area_min_D = min(height_manual_area_D(:));
area_max_D = max(height_manual_area_D(:));
y_ortho = linspace(0, 1500, size(height_manual_area_D, 2)); % X-axis from 0 to 1500
micrometers
x_ortho = linspace(0, 1500, size(height_manual_area_D, 1)); % Y-axis from 0 to 1500
micrometers
[H_D, W_D] = meshgrid(y_ortho, x_ortho);

% plot for paper
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1)
surf(H_D,W_D,height_manual_area_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2)
surf(H_D,W_D,height_manual_area_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
view(2)
Surf_ortho_D_gca = gcf;
exportgraphics(Surf_ortho_D_gca,'Surf_ortho_D.jpg')

%% 4.1.1 Surface Extrapolation Area from single SWLI line Sin/Cos orthogonal method
% i defines spectrum
% n defines x
% m defines y
scanlength=0.0015;
deltay_area= scanlength/height;
deltax_area= scanlength/width;
y_resolution = height; % round(y1(end))
x_resolution = width; % round(x1(end))
rng(1); % sets reproducibility
height_manual_area = zeros(y_resolution,x_resolution);
n = 2;
m = 2;
for i=2:500 %length(spectr)/2
phase_area(i) = atan(imag(spectrum(i))/real(spectrum(i)));
lambda_area(i)=scanlength/(i-1);
for m = 1:y_resolution;
for n = 1:x_resolution;  %length(spectr);
```

```
    height_manual_area(m,n) = height_manual_area(m,n)+spectrum_norm(i)*(sin(2*pi*(n-
1)*deltax_area/lambda_area(i)+phase_anglesolution(i)))+(spectrum_norm(i)*(cos(2*pi*(m
-1)*deltay_area/lambda_area(i)+phase_anglesolution(i)))); % <- functional +
end
end
end

% roughness parameters from extrapolated area
heightmanualarea_=reshape(height_manual_area,[],1);
sa_heightmanualarea=(sum(abs(heightmanualarea_)))/(length(heightmanualarea_));
area_min = min(height_manual_area(:));
area_max = max(height_manual_area(:));
y_ortho = linspace(0, 1500, size(height_manual_area, 2)); % X-axis from 0 to 1500
micrometers
x_ortho = linspace(0, 1500, size(height_manual_area, 1)); % Y-axis from 0 to 1500
micrometers
[H, W] = meshgrid(y_ortho, x_ortho);

% plot for paper
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1)
surf(H,W,height_manual_area,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2)
surf(H,W,height_manual_area,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
view(2)
Surf_ortho_S_gca = gcf;
exportgraphics(Surf_ortho_S_gca,'Surf_ortho_S.jpg')

%% 3.1.1 bearing area curve - orthogonal Dektak surface

manualarea_Hvals_D =  height_manual_area_D;
areaspan_D = round(area_max_D + abs(area_min_D));
mrarea_D = zeros(areaspan_D,1);
area_above_D = zeros(areaspan_D,1);
Hn_D = 1;
for Harea_D = round(area_max_D):-1:round(area_min_D)
    area_above_D(Hn_D) = sum(manualarea_Hvals_D(:) >= Harea_D);
    Hn_D = Hn_D + 1;
```

```
end
[Lmrarea_D,Wmrarea_D] = size(manualarea_Hvals_D);
mrarea_D = 100*area_above_D/(Lmrarea_D*Wmrarea_D);
Hthresharea_D = [round(area_max_D):-1:round(area_min_D)]';
startmrarea_D = round(0.3*areaspan_D);
endmrarea_D = round(0.7*areaspan_D);
mrarea_cut_D = mrarea_D(startmrarea_D:endmrarea_D);
Hthresharea_cut_D = Hthresharea_D(startmrarea_D:endmrarea_D);
[mrareaxData_D,mrareayData_D] = prepareCurveData(mrarea_cut_D,Hthresharea_cut_D);
areaft_D = fittype('poly1');
[mrareafitresult_D,gof] = fit(mrareaxData_D,mrareayData_D,areaft_D);
mrareaslope_D = mrareafitresult_D.p1;
last_point_D_ortho = length(Hthresharea_D);
last_height_D_ortho = mrareafitresult_D.p1 * mrarea_D(last_point_D_ortho) +
mrareafitresult_D.p2;
Svk_ortho_D = area_min_D - last_height_D_ortho;

%% 3.1.1 bearing area curve - orthogonal SWLI surface

manualarea_Hvals =  height_manual_area;
areaspan = round(area_max + abs(area_min));
mrarea = zeros(areaspan,1);
area_above = zeros(areaspan,1);
Hn = 1;
for Harea = round(area_max):-1:round(area_min)
    area_above(Hn) = sum(manualarea_Hvals(:) >= Harea);
    Hn = Hn + 1;
end
[Lmrarea,Wmrarea] = size(manualarea_Hvals);
mrarea = 100*area_above/(Lmrarea*Wmrarea);
Hthresharea = [round(area_max):-1:round(area_min)]';
startmrarea = round(0.3*areaspan);
endmrarea = round(0.7*areaspan);
mrarea_cut = mrarea(startmrarea:endmrarea);
Hthresharea_cut = Hthresharea(startmrarea:endmrarea);
[mrareaxData,mrareayData] = prepareCurveData(mrarea_cut,Hthresharea_cut);
areaft = fittype('poly1');
[mrareafitresult,gof] = fit(mrareaxData,mrareayData,areaft);
mrareaslope = mrareafitresult.p1;
last_point_S_ortho = length(Hthresharea);
last_height_S_ortho = mrareafitresult.p1 * mrarea(last_point_S_ortho) +
mrareafitresult.p2;
Svk_ortho_S = area_min - last_height_S_ortho;


%% 4.1.2, 4.1.3 using artificial_surf external function & 3.1.1 bearing area curve
calculations

% Dektak line surface generaiton
% artificial_surf Dektak line Fourier H
[z_artificial_F_D , PixelWidth , PSD_artificial] = artificial_surf(Rq_Dektak,
H_Fourier_Dektak, scanlength, round(x1(end)) , round(y1(end)));
[H_artificial_F_D,W_artificial_F_D] =
meshgrid(1:size(z_artificial_F_D,2),1:size(z_artificial_F_D,1));
surf_max = max(z_artificial_F_D(:));
```

```
surf_min = min(z_artificial_F_D(:));
% bearing area curve (material ratio) - artificial_surf Dektak line F H
surf_Hvals =  z_artificial_F_D;
surfspan = round(surf_max + abs(surf_min));
mrsurf = zeros(surfspan,1);
surf_above = zeros(surfspan,1);
Ha = 1;
for Hsurf = round(surf_max):-1:round(surf_min)
    surf_above(Ha) = sum(surf_Hvals(:) >= Hsurf);
    Ha = Ha + 1;
end
[Lmrsurf,Wmrsurf] = size(surf_Hvals);
mrsurf = 100*surf_above/(Lmrsurf*Wmrsurf);
Hthreshsurf = [round(surf_max):-1:round(surf_min)]';
startmrsurf = round(0.3*surfspan);
endmrsurf = round(0.7*surfspan);
mrsurf_cut = mrsurf(startmrsurf:endmrsurf);
Hthreshsurf_cut = Hthreshsurf(startmrsurf:endmrsurf);
[mrsurfxData,mrsurfyData] = prepareCurveData(mrsurf_cut,Hthreshsurf_cut);
surfft = fittype('poly1');
[mrsurffitresult,gof] = fit(mrsurfxData,mrsurfyData,surfft);
mrsurfslope = mrsurffitresult.p1;
last_point_D_artsur = length(Hthreshsurf);
last_height_D_artsur = mrsurffitresult.p1 * mrsurf(last_point_D_artsur) +
mrsurffitresult.p2;
% calculate the difference between the fit and deepest valley
Svk_artsur_F_D = surf_min - last_height_D_artsur;
% plot for paper
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1)
surf(H_artificial_F_D,W_artificial_F_D,z_artificial_F_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2)
surf(H_artificial_F_D,W_artificial_F_D,z_artificial_F_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
view(2)
Surf_artsurf_D_gca = gcf;
exportgraphics(Surf_artsurf_D_gca,'Surf_artsurf_F_D.jpg')
```

```
% artificial_surf Dektak line Rq H
[z_artificial_Rq_D , PixelWidth , PSD_artificial] = artificial_surf(Rq_Dektak,
H_Rq_Dektak, scanlength, round(x1(end)) , round(y1(end)));
[H_artificial_Rq_D,W_artificial_Rq_D] =
meshgrid(1:size(z_artificial_Rq_D,2),1:size(z_artificial_Rq_D,1));
surf_max = max(z_artificial_Rq_D(:));
surf_min = min(z_artificial_Rq_D(:));
% bearing area curve (material ratio) - artificial_surf Dektak line Rq H
surf_Hvals =  z_artificial_Rq_D;
surfspan = round(surf_max + abs(surf_min));
mrsurf = zeros(surfspan,1);
surf_above = zeros(surfspan,1);
Ha = 1;
for Hsurf = round(surf_max):-1:round(surf_min)
    surf_above(Ha) = sum(surf_Hvals(:) >= Hsurf);
    Ha = Ha + 1;
end
[Lmrsurf,Wmrsurf] = size(surf_Hvals);
mrsurf = 100*surf_above/(Lmrsurf*Wmrsurf);
Hthreshsurf = [round(surf_max):-1:round(surf_min)]';
startmrsurf = round(0.3*surfspan);
endmrsurf = round(0.7*surfspan);
mrsurf_cut = mrsurf(startmrsurf:endmrsurf);
Hthreshsurf_cut = Hthreshsurf(startmrsurf:endmrsurf);
[mrsurfxData,mrsurfyData] = prepareCurveData(mrsurf_cut,Hthreshsurf_cut);
surfft = fittype('poly1');
[mrsurffitresult,gof] = fit(mrsurfxData,mrsurfyData,surfft);
mrsurfslope = mrsurffitresult.p1;
last_point_D_artsur = length(Hthreshsurf);
last_height_D_artsur = mrsurffitresult.p1 * mrsurf(last_point_D_artsur) +
mrsurffitresult.p2;
% calculate the difference between the fit and deepest valley
Svk_artsur_Rq_D = surf_min - last_height_D_artsur;
% plot for paper
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1)
surf(H_artificial_Rq_D,W_artificial_Rq_D,z_artificial_Rq_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2)
surf(H_artificial_Rq_D,W_artificial_Rq_D,z_artificial_Rq_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
```

```
set(gca,"FontSize",11)
view(2)
Surf_artsurf_D_gca = gcf;
exportgraphics(Surf_artsurf_D_gca,'Surf_artsurf_Rq_D.jpg')



% SWLI line surface generation
% artificial_surf SWLI line Fourier H
[z_artificial_F_S , PixelWidth , PSD_artificial] = artificial_surf(Rq_flatline,
H_Fourier_SWLI, scanlength, round(x1(end)) , round(y1(end)));
[H_artificial_F_S,W_artificial_F_S] =
meshgrid(1:size(z_artificial_F_S,2),1:size(z_artificial_F_S,1));
surf_max = max(z_artificial_F_S(:));
surf_min = min(z_artificial_F_S(:));
% bearing area curve (material ratio) - artificial_surf SWLI line F H
surf_Hvals =  z_artificial_F_S;
surfspan = round(surf_max + abs(surf_min));
mrsurf = zeros(surfspan,1);
surf_above = zeros(surfspan,1);
Ha = 1;
for Hsurf = round(surf_max):-1:round(surf_min)
    surf_above(Ha) = sum(surf_Hvals(:) >= Hsurf);
    Ha = Ha + 1;
end
[Lmrsurf,Wmrsurf] = size(surf_Hvals);
mrsurf = 100*surf_above/(Lmrsurf*Wmrsurf);
Hthreshsurf = [round(surf_max):-1:round(surf_min)]';
startmrsurf = round(0.3*surfspan);
endmrsurf = round(0.7*surfspan);
mrsurf_cut = mrsurf(startmrsurf:endmrsurf);
Hthreshsurf_cut = Hthreshsurf(startmrsurf:endmrsurf);
[mrsurfxData,mrsurfyData] = prepareCurveData(mrsurf_cut,Hthreshsurf_cut);
surfft = fittype('poly1');
[mrsurffitresult,gof] = fit(mrsurfxData,mrsurfyData,surfft);
mrsurfslope = mrsurffitresult.p1;
last_point_S_artsur = length(Hthreshsurf);
last_height_S_artsur = mrsurffitresult.p1 * mrsurf(last_point_S_artsur) +
mrsurffitresult.p2;
% calculate the difference between the fit and deepest valley
Svk_artsur_F_S = surf_min - last_height_S_artsur;
% plot for paper
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1)
surf(H_artificial_F_S,W_artificial_F_S,z_artificial_F_S,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2)
```

```
surf(H_artificial_F_S,W_artificial_F_S,z_artificial_F_S,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
view(2)
Surf_artsurf_S_gca = gcf;
exportgraphics(Surf_artsurf_S_gca,'Surf_artsurf_F_S.jpg')


% artificial_surf SWLI line Rq H
[z_artificial , PixelWidth , PSD_artificial] = artificial_surf(Rq_flatline,
Hurst_R_q, scanlength, round(x1(end)) , round(y1(end)));
[H_artificial,W_artificial] =
meshgrid(1:size(z_artificial,2),1:size(z_artificial,1));
surf_max = max(z_artificial(:));
surf_min = min(z_artificial(:));
% bearing area curve (material ratio) - artificial_surf SWLI line Rq H
surf_Hvals =  z_artificial;
surfspan = round(surf_max + abs(surf_min));
mrsurf = zeros(surfspan,1);
surf_above = zeros(surfspan,1);
Ha = 1;
for Hsurf = round(surf_max):-1:round(surf_min)
    surf_above(Ha) = sum(surf_Hvals(:) >= Hsurf);
    Ha = Ha + 1;
end
[Lmrsurf,Wmrsurf] = size(surf_Hvals);
mrsurf = 100*surf_above/(Lmrsurf*Wmrsurf);
Hthreshsurf = [round(surf_max):-1:round(surf_min)]';
startmrsurf = round(0.3*surfspan);
endmrsurf = round(0.7*surfspan);
mrsurf_cut = mrsurf(startmrsurf:endmrsurf);
Hthreshsurf_cut = Hthreshsurf(startmrsurf:endmrsurf);
[mrsurfxData,mrsurfyData] = prepareCurveData(mrsurf_cut,Hthreshsurf_cut);
surfft = fittype('poly1');
[mrsurffitresult,gof] = fit(mrsurfxData,mrsurfyData,surfft);
mrsurfslope = mrsurffitresult.p1;
last_point_S_artsur = length(Hthreshsurf);
last_height_S_artsur = mrsurffitresult.p1 * mrsurf(last_point_S_artsur) +
mrsurffitresult.p2;
% calculate the difference between the fit and deepest valley
Svk_artsur_Rq_S = surf_min - last_height_S_artsur;
% plot for paper
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1)
surf(H_artificial,W_artificial,z_artificial,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
```

```
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2)
surf(H_artificial,W_artificial,z_artificial,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
view(2)
Surf_artsurf_S_gca = gcf;
exportgraphics(Surf_artsurf_S_gca,'Surf_artsurf_Rq_S.jpg')

%% plot the extrapolated & generated areas next to the original SWLI scan

% Dektak line Fourier orthogonal spectrum method
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1);
s1 = surf(x1,y1,flatsurf,'edgecolor','none');
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2);
surf(H_D,W_D,height_manual_area_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
Surf_ortho_D_gca = gcf;
exportgraphics(Surf_ortho_D_gca,'Surf_og_ortho_D.jpg')

% SWLI line Fourier orthogonal spectrum method
figure('Position', [100, 100, 1800, 600]);
subplot(1,2,1);
s1 = surf(x1,y1,flatsurf,'edgecolor','none');
hold on
colorbar
colormap(vik)
```

```
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(1,2,2);
surf(H_D,W_D,height_manual_area,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
Surf_ortho_S_gca = gcf;
exportgraphics(Surf_ortho_S_gca,'Surf_og_ortho_S.jpg')

% Dektak and SWLI line fourier orthogonal spectrum method
figure('Position', [100, 100, 800, 800]);
subplot(3,1,1)
s1 = surf(x1,y1,flatsurf,'edgecolor','none');
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(3,1,2);
surf(H_D,W_D,height_manual_area_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(3,1,3);
surf(H_D,W_D,height_manual_area,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("c)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
```

```
Surf_ortho_D_S_gca = gcf;
exportgraphics(Surf_ortho_D_S_gca,'Surf_og_ortho_D_S.jpg')

% original vs artificial_surf D
figure('Position', [100, 100, 800, 800]);
subplot(3,1,1)
s1 = surf(x1,y1,flatsurf,'edgecolor','none');
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(3,1,2)
surf(H_artificial_F_D,W_artificial_F_D,z_artificial_F_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(3,1,3)
surf(H_artificial_Rq_D,W_artificial_Rq_D,z_artificial_Rq_D,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("c)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
Surf_artsurf_D_gca = gcf;
exportgraphics(Surf_artsurf_D_gca,'Surf_artsurf_D.jpg')

% original vs artificial_surf S
figure('Position', [100, 100, 800, 800]);
subplot(3,1,1)
s1 = surf(x1,y1,flatsurf,'edgecolor','none');
hold on
colorbar
colormap(vik)
title("a)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(3,1,2)
```

```
surf(H_artificial_F_S,W_artificial_F_S,z_artificial_F_S,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("b)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
subplot(3,1,3)
surf(H_artificial,W_artificial,z_artificial,'edgecolor','none')
hold on
colorbar
colormap(vik)
title("c)",'FontWeight','Normal')
xlabel('Length, x (µm)')
ylabel('Width, y (µm)')
zlabel('Height, z (µm)')
set(gca,"FontSize",11)
direction = [1 1 1];
Surf_artsurf_S_gca = gcf;
exportgraphics(Surf_artsurf_S_gca,'Surf_artsurf_S.jpg')


%% Parameter Consolidation Section


% Parameter final calculation section
% Orthogonal Surface from Dektak line area parameter calculations
Ortho_D = heightmanualarea__D;
Sa_Ortho_D = (sum(abs(Ortho_D)))/(length(Ortho_D));
Sq_Ortho_D = sqrt(sum(abs(((Ortho_D.^2))))/length(Ortho_D));
Sp_Ortho_D = max(Ortho_D);
Sv_Ortho_D = min(Ortho_D);
Sz_Ortho_D = abs(Sp_Ortho_D)+abs(Sv_Ortho_D);
Ssk_Ortho_D = skewness(Ortho_D);
Sku_Ortho_D = kurtosis(Ortho_D);
% Orthogonal Surface from SWLI line area parameter calculations
Ortho_S = heightmanualarea_;
Sa_Ortho_S = (sum(abs(Ortho_S)))/(length(Ortho_S));
Sq_Ortho_S = sqrt(sum(abs(((Ortho_S.^2))))/length(Ortho_S));
Sp_Ortho_S = max(Ortho_S);
Sv_Ortho_S = min(Ortho_S);
Sz_Ortho_S = abs(Sp_Ortho_S)+abs(Sv_Ortho_S);
Ssk_Ortho_S = skewness(Ortho_S);
Sku_Ortho_S = kurtosis(Ortho_S);
% artificial_surf Dektak line Fourier H
artsur_F_D = reshape(z_artificial_F_D,[],1);
Sa_artsur_F_D = (sum(abs(artsur_F_D)))/(length(artsur_F_D));
Sq_artsur_F_D = sqrt(sum(abs(((artsur_F_D.^2))))/length(artsur_F_D));
Sp_artsur_F_D = max(artsur_F_D);
Sv_artsur_F_D = min(artsur_F_D);
Sz_artsur_F_D = abs(Sp_artsur_F_D)+abs(Sv_artsur_F_D);
Ssk_artsur_F_D = skewness(artsur_F_D);
```

```
Sku_artsur_F_D = kurtosis(artsur_F_D);
% artificial_surf Dektak line Rq H
artsur_Rq_D = reshape(z_artificial_Rq_D,[],1);
Sa_artsur_Rq_D = (sum(abs(artsur_Rq_D)))/(length(artsur_Rq_D));
Sq_artsur_Rq_D = sqrt(sum(abs(((artsur_Rq_D.^2))))/length(artsur_Rq_D));
Sp_artsur_Rq_D = max(artsur_Rq_D);
Sv_artsur_Rq_D = min(artsur_Rq_D);
Sz_artsur_Rq_D = abs(Sp_artsur_Rq_D)+abs(Sv_artsur_Rq_D);
Ssk_artsur_Rq_D = skewness(artsur_Rq_D);
Sku_artsur_Rq_D = kurtosis(artsur_Rq_D);
% artificial_surf SWLI line Fourier H
artsur_F_S = reshape(z_artificial_F_S,[],1);
Sa_artsur_F_S = (sum(abs(artsur_F_S)))/(length(artsur_F_S));
Sq_artsur_F_S = sqrt(sum(abs(((artsur_F_S.^2))))/length(artsur_F_S));
Sp_artsur_F_S = max(artsur_F_S);
Sv_artsur_F_S = min(artsur_F_S);
Sz_artsur_F_S = abs(Sp_artsur_F_S)+abs(Sv_artsur_F_S);
Ssk_artsur_F_S = skewness(artsur_F_S);
Sku_artsur_F_S = kurtosis(artsur_F_S);
% artificial_surf SWLI line Rq H
artsur_Rq_S = reshape(z_artificial,[],1);
Sa_artsur_Rq_S = (sum(abs(artsur_Rq_S)))/(length(artsur_Rq_S));
Sq_artsur_Rq_S = sqrt(sum(abs(((artsur_Rq_S.^2))))/length(artsur_Rq_S));
Sp_artsur_Rq_S = max(artsur_Rq_S);
Sv_artsur_Rq_S = min(artsur_Rq_S);
Sz_artsur_Rq_S = abs(Sp_artsur_Rq_S)+abs(Sv_artsur_Rq_S);
Ssk_artsur_Rq_S = skewness(artsur_Rq_S);
Sku_artsur_Rq_S = kurtosis(artsur_Rq_S);


% Parameter organization section - line profiles
% Original Dektak line parameters
Rsk_Dektak = Sk_Dektak;
Rku_Dektak = Ku_Dektak;
R_Dektak =
[Ra_Dektak,Rq_Dektak,Rv_Dektak,Rp_Dektak,Rz_Dektak,Rsk_Dektak,Rku_Dektak,Rvk_Dektak];
% Original SWLI line parameters
Ra_SWLI = Ra_flatline;
Rq_SWLI = Rq_flatline;
Rp_SWLI = Rp_flatline;
Rv_SWLI = Rv_flatline;
Rz_SWLI = Rz_flatline;
Rsk_SWLI = Rsk_flatline;
Rku_SWLI = Rku_flatline;
R_SWLI = [Ra_SWLI,Rq_SWLI,Rv_SWLI,Rp_SWLI,Rz_SWLI,Rsk_SWLI,Rku_SWLI,Rvk_SWLI];
% Parameter organization section - area scans
% Original SWLI area parameters
Sa_SWLI = sa_flatsurf;
Sq_SWLI = sq_flatsurf;
Sp_SWLI = sp_flatsurf;
Sv_SWLI = sv_flatsurf;
Sz_SWLI = sz_flatsurf;
Ssk_SWLI = ssk_flatsurf;
Sku_SWLI = sku_flatsurf;
S_SWLI = [Sa_SWLI,Sq_SWLI,Sv_SWLI,Sp_SWLI,Sz_SWLI,Ssk_SWLI,Sku_SWLI, Svk_SWLI];
```

```
% Orthogonal Surface from Dektak line area parameters
S_Ortho_Dektak =
[Sa_Ortho_D,Sq_Ortho_D,Sv_Ortho_D,Sp_Ortho_D,Sz_Ortho_D,Ssk_Ortho_D,Sku_Ortho_D,Svk_o
rtho_D];
% Orthogonal Surface from SWLI line area parameters
S_Ortho_SWLI =
[Sa_Ortho_S,Sq_Ortho_S,Sv_Ortho_S,Sp_Ortho_S,Sz_Ortho_S,Ssk_Ortho_S,Sku_Ortho_S,Svk_o
rtho_S];
% artificial_surf Dektak line Fourier H parameters
S_artsur_F_Dektak =
[Sa_artsur_F_D,Sq_artsur_F_D,Sv_artsur_F_D,Sp_artsur_F_D,Sz_artsur_F_D,Ssk_artsur_F_D
,Sku_artsur_F_D,Svk_artsur_F_D];
% artificial_surf Dektak line Rq H parameterS
S_artsur_Rq_Dektak =
[Sa_artsur_Rq_D,Sq_artsur_Rq_D,Sv_artsur_Rq_D,Sp_artsur_Rq_D,Sz_artsur_Rq_D,Ssk_artsu
r_Rq_D,Sku_artsur_Rq_D,Svk_artsur_Rq_D];
% artificial_surf SWLI line Fourier H parameters
S_artsur_F_SWLI =
[Sa_artsur_F_S,Sq_artsur_F_S,Sv_artsur_F_S,Sp_artsur_F_S,Sz_artsur_F_S,Ssk_artsur_F_S
,Sku_artsur_F_S,Svk_artsur_F_S];
% artificial_surf SWLI line Rq H parameters
S_artsur_Rq_SWLI =
[Sa_artsur_Rq_S,Sq_artsur_Rq_S,Sv_artsur_Rq_S,Sp_artsur_Rq_S,Sz_artsur_Rq_S,Ssk_artsu
r_Rq_S,Sku_artsur_Rq_S,Svk_artsur_Rq_S];


% Parameter upload section - no fractal_surface
parameters = [R_Dektak,R_SWLI,S_SWLI , S_Ortho_Dektak,S_Ortho_SWLI ,
S_artsur_F_Dektak,S_artsur_Rq_Dektak,S_artsur_F_SWLI,S_artsur_Rq_SWLI];
variable_names =
{'Ra_Dektak','Rq_Dektak','Rv_Dektak','Rp_Dektak','Rz_Dektak','Rsk_Dektak','Rku_Dektak
','Rvk_Dektak' ,
'Ra_SWLI','Rq_SWLI','Rv_SWLI','Rp_SWLI','Rz_SWLI','Rsk_SWLI','Rku_SWLI','Rvk_SWLI' ,
'Sa_SWLI','Sq_SWLI','Sv_SWLI','Sp_SWLI','Sz_SWLI','Ssk_SWLI','Sku_SWLI','Svk_SWLI' ,
'Sa_Ortho_D','Sq_Ortho_D','Sv_Ortho_D','Sp_Ortho_D','Sz_Ortho_D','Ssk_Ortho_D','Sku_O
rtho_D','Svk_ortho_D' ,
'Sa_Ortho_S','Sq_Ortho_S','Sv_Ortho_S','Sp_Ortho_S','Sz_Ortho_S','Ssk_Ortho_S','Sku_O
rtho_S','Svk_ortho_S' ,
'Sa_artsur_F_D','Sq_artsur_F_D','Sv_artsur_F_D','Sp_artsur_F_D','Sz_artsur_F_D','Ssk_
artsur_F_D','Sku_artsur_F_D','Svk_artsur_F_D' ,
'Sa_artsur_Rq_D','Sq_artsur_Rq_D','Sv_artsur_Rq_D','Sp_artsur_Rq_D','Sz_artsur_Rq_D',
'Ssk_artsur_Rq_D','Sku_artsur_Rq_D','Svk_artsur_Rq_D' ,
'Sa_artsur_F_S','Sq_artsur_F_S','Sv_artsur_F_S','Sp_artsur_F_S','Sz_artsur_F_S','Ssk_
artsur_F_S','Sku_artsur_F_S','Svk_artsur_F_S' ,
'Sa_artsur_Rq_S','Sq_artsur_Rq_S','Sv_artsur_Rq_S','Sp_artsur_Rq_S','Sz_artsur_Rq_S',
'Ssk_artsur_Rq_S','Sku_artsur_Rq_S','Svk_artsur_Rq_S'};
excel_file = 'F:\ResearchTribology\Surface Regeneration\Line2Area_SWLI\Old SD
N\ConsolidatedParameters_v2.xlsx';
excel_sheet = 'AllData';
writecell(variable_names, excel_file, 'Sheet', excel_sheet, 'Range', 'B2:BU2');
writematrix(parameters, excel_file, 'Sheet', excel_sheet, 'Range', 'B3:BU3'); % edit
for correct row/specimen

variable_names_H_RMS = {'Rq_D','H_F_D','H_Rq_D','Rq_S','H_F_S','H_Rq_S'}
```

```
parameters_H_RMS = [Rq_Dektak, H_Fourier_Dektak, H_Rq_Dektak, Rq_flatline,
H_Fourier_SWLI, Hurst_R_q,]
writecell(variable_names_H_RMS, excel_file, 'Sheet', excel_sheet, 'Range',
'B20:G20');
writematrix(parameters_H_RMS, excel_file, 'Sheet', excel_sheet, 'Range', 'B21:G21');
% edit for correct row/specimen
```

Appendix E: Data interpolation function

A function from Swarna Saha provided by Samsul Mahmood and Dr. Jackson. I have modified into a function titled "interpdatafill.m" which takes an input "z" and provides an output "zfill". The input data typically consisted of a 3063 x 3068 matrix of height values representing an approximately 1.5 mm x 1.5 mm optical area scan. The SWLI equipment has limitations due to its method of surface height capturing, namely reflective surfaces and steep angles of surface asperities. Steep angles and certain valley geometries can cause light to deviate and not be intercepted by the detector. This creates gaps in the dataset which are represented by empty elements. The function is shown below.

```
function [zfill] = interpdatafill(z)

zfill = z;
for k=1:3063
    for j=1:3068
        average=0;
        count=0;
        check=isequaln(zfill(k,j),NaN);
        if check==1
            if j==1 && k==1
                check1=isequaln(zfill(k+1,j),NaN);
                check2=isequaln(zfill(k,j+1),NaN);
                if check1~=1
                    average=average+zfill(k+1,j);
                    count=count+1;
                end
                if check2~=1
```

```
        average=average+zfill(k,j+1);
        count=count+1;
    end
elseif j==1 && k==3063
    check1=isequaln(zfill(k-1,j),NaN);
    check2=isequaln(zfill(k,j+1),NaN);
    if check1~=1
        average=average+zfill(k-1,j);
        count=count+1;
    end
    if check2~=1
        average=average+zfill(k,j+1);
        count=count+1;
    end
elseif j==1 && k~=1 && k~=3063
    check1=isequaln(zfill(k-1,j),NaN);
    check2=isequaln(zfill(k+1,j),NaN);
    check3=isequaln(zfill(k,j+1),NaN);
    if check1~=1
        average=average+zfill(k-1,j);
        count=count+1;
    end
    if check2~=1
        average=average+zfill(k+1,j);
        count=count+1;
    end
    if check3~=1
        average=average+zfill(k,j+1);
        count=count+1;
    end
elseif j==3068 && k==1
    check1=isequaln(zfill(k,j-1),NaN);
    check2=isequaln(zfill(k+1,j),NaN);
    if check1~=1
        average=average+zfill(k,j-1);
        count=count+1;
    end
    if check2~=1
        average=average+zfill(k+1,j);
        count=count+1;
    end
elseif j==3068 && k==3063
    check1=isequaln(zfill(k,j-1),NaN);
    check2=isequaln(zfill(k-1,j),NaN);
    if check1~=1
        average=average+zfill(k,j-1);
        count=count+1;
    end
    if check2~=1
        average=average+zfill(k-1,j);
        count=count+1;
    end
elseif j==3068 && k~=1 && k~=3063
    check1=isequaln(zfill(k-1,j),NaN);
    check2=isequaln(zfill(k,j-1),NaN);
```

```
        check3=isequaln(zfill(k+1,j),NaN);
        if check1~=1
            average=average+zfill(k-1,j);
            count=count+1;
        end
        if check2~=1
            average=average+zfill(k,j-1);
            count=count+1;
        end
        if check3~=1
            average=average+zfill(k+1,j);
            count=count+1;
        end
    elseif k==1 && j>1 && j<3068
        check1=isequaln(zfill(k,j-1),NaN);
        check2=isequaln(zfill(k,j+1),NaN);
        check3=isequaln(zfill(k+1,j),NaN);
        if check1~=1
            average=average+zfill(k,j-1);
            count=count+1;
        end
        if check2~=1
            average=average+zfill(k,j+1);
            count=count+1;
        end
        if check3~=1
            average=average+zfill(k+1,j);
            count=count+1;
        end
    elseif k==3063 && j>1 && j<3068
        check1=isequaln(zfill(k-1,j),NaN);
        check2=isequaln(zfill(k,j-1),NaN);
        check3=isequaln(zfill(k,j+1),NaN);
        if check1~=1
            average=average+zfill(k-1,j);
            count=count+1;
        end
        if check2~=1
            average=average+zfill(k,j-1);
            count=count+1;
        end
        if check3~=1
            average=average+zfill(k,j+1);
            count=count+1;
        end

    else
        check1=isequaln(zfill(k-1,j),NaN);
        check2=isequaln(zfill(k,j-1),NaN);
        check3=isequaln(zfill(k+1,j),NaN);
        check4=isequaln(zfill(k,j+1),NaN);
        if check1~=1
            average=average+zfill(k-1,j);
            count=count+1;
        end
```

```
                if check2~=1
                    average=average+zfill(k,j-1);
                    count=count+1;
                end
                if check3~=1
                    average=average+zfill(k+1,j);
                    count=count+1;
                end
                if check4~=1
                    average=average+zfill(k,j+1);
                    count=count+1;
                end
            end
            zfill(k,j)=average/count;
        end
    end
end
```

Appendix F: planeflat function

```
function [flatsurf] = planeflat(points, zfill, height, width)

planefit=fit([points(:,1),points(:,2)],points(:,3),'poly22');
planef=points;
planef(:,3)=planefit(points(:,1,1),points(:,2,1));
planeff=zeros(height,width);
planeff=reshape(planef(:,3),height,width);
flatsurf(:,:,1)=zfill-planeff;
```

Appendix G: planelineflat function

```
function [flatlinesurf_y,flatlinesurf_x] = planelineflat(zfill, height, width, x1,
y1)
% y
h = 1;
while h < height
    Ra_swli_y(h,1) = mean(abs(zfill(h,:)));
    zflat_y = polyfit(x1,zfill(h,:),2);
    zval_y(h,:) = polyval(zflat_y,x1);
    flatlinesurf_y(h,:) = zfill(h,:) - zval_y(h,:);
    Raflat_y(h,1) = mean(abs(flatlinesurf_y(h,:)));
    zlevel_y(h,:) = flatlinesurf_y(h,:) - mean(zfill(h,:));
    h = h + 1;
end
Raflat_y(any(isnan(Raflat_y),2),:) = [];
Raflat_mean_y = mean(Raflat_y);
Raflat_std_y = std(Raflat_y);
% x
w = 1;
while w < width
    Ra_swli_x(1,w) = mean(abs(zfill(:,w)));
```

```
    zflat_x = polyfit(y1,zfill(:,w),2);
    zval_x(:,w) = polyval(zflat_x,y1);
    flatlinesurf_x(:,w) = zfill(:,w) - zval_x(:,w);
    Raflat_x(1,w) = mean(abs(flatlinesurf_x(:,w)));
    zlevel_x(:,w) = flatlinesurf_x(:,w) - mean(zfill(:,w));
    w = w + 1;
end
Raflat_x(any(isnan(Raflat_x),2),:) = [];
Raflat_mean_x = mean(Raflat_x);
Raflat_std_x = std(Raflat_x);
```

Appendix H: DFT function

```
function [dft_spectrum, dft_spectrum_norm] = dft(Line)

dft_spectrum = zeros(length(Line),1);
for dft_k = 1:length(Line)
    dft_spectrum(dft_k) = 0;
    for dft_n = 1:length(Line)
        dft_spectrum(dft_k) = dft_spectrum(dft_k) + (Line(dft_n)*exp((-
1j)*2*pi*(dft_n - 1)*(dft_k - 1)/length(Line)));
    end
end
dft_spectrum = 2.* dft_spectrum;
dft_spectrum_norm = abs(dft_spectrum/(length(dft_spectrum)));
```

Appendix H: Sample 001 supplementary figures

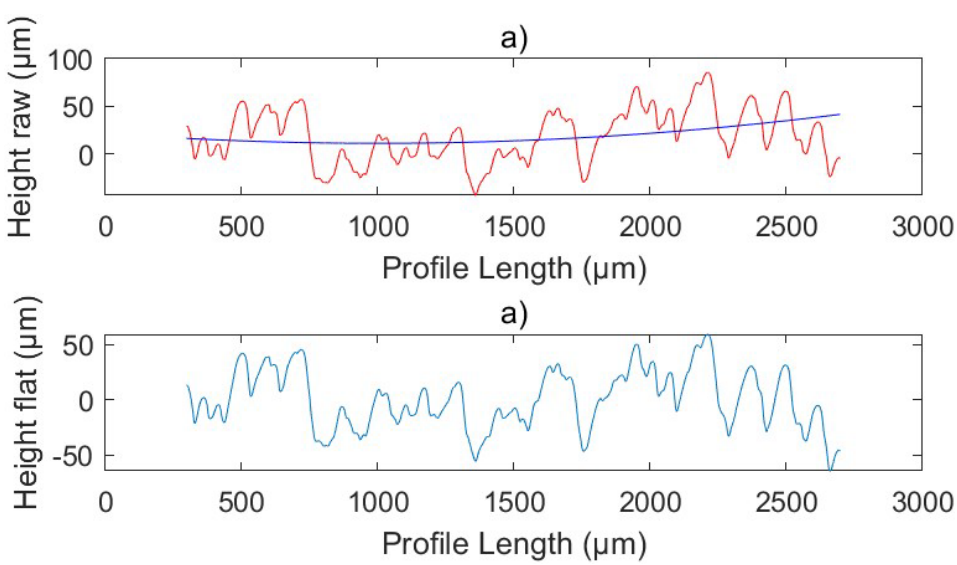

Figure H1: a) Raw profile measurement and fit curve sample 001 b) flattened line profile from sample 001

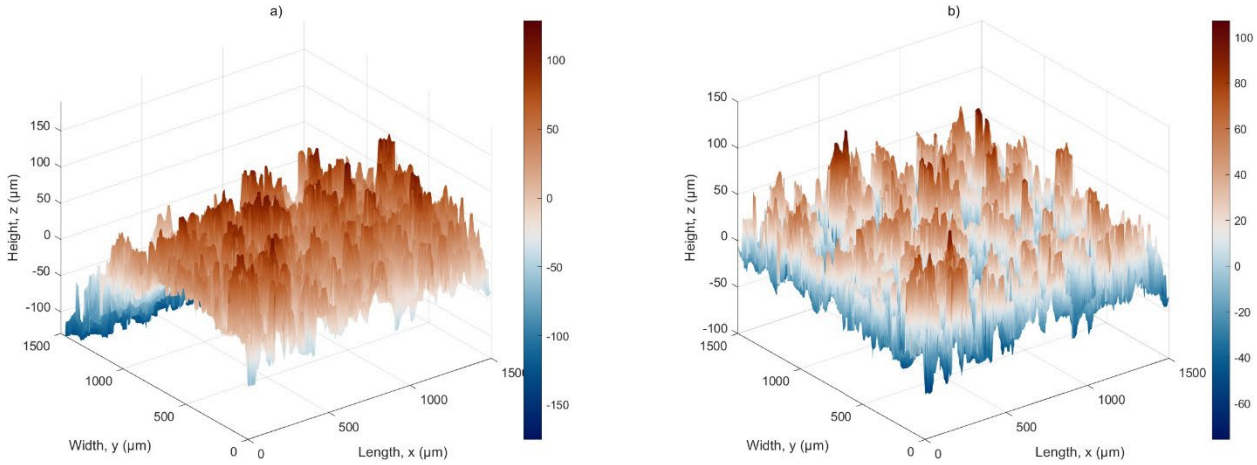

Figure H2: a) Unflattened SWLI raw surface scan sample 001 b) plane flattened SWLI surface 001

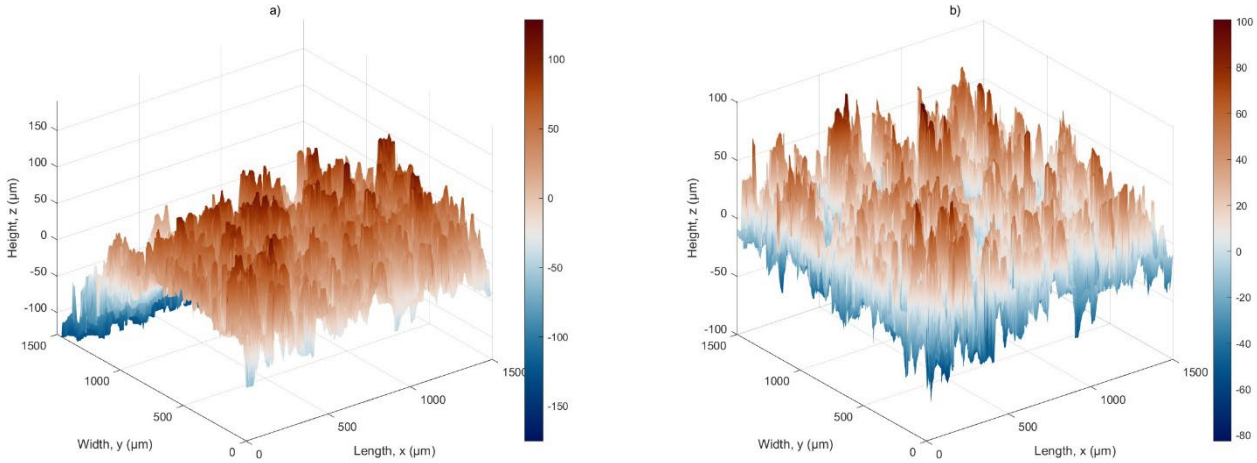

Figure H3: a) Unflattened SWLI raw surface scan sample 001 b) line by line flattened SWLI surface 001

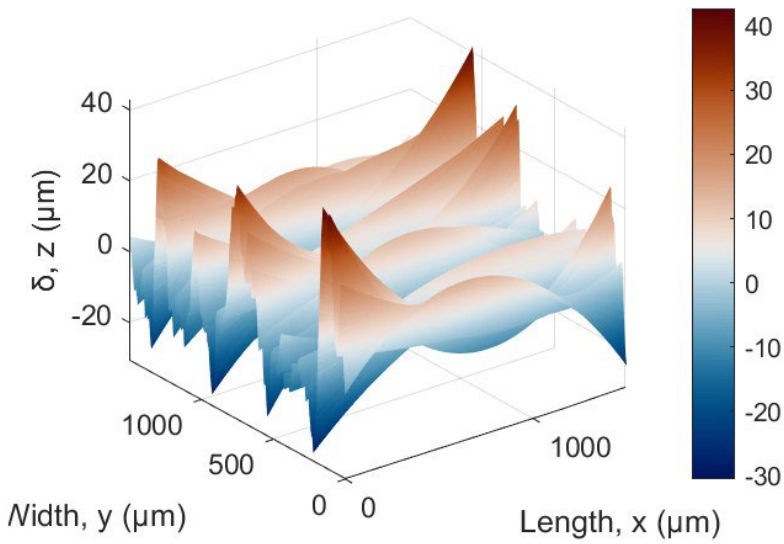

Figure H4: Difference between plane flattening and line by line flattening for sample 001

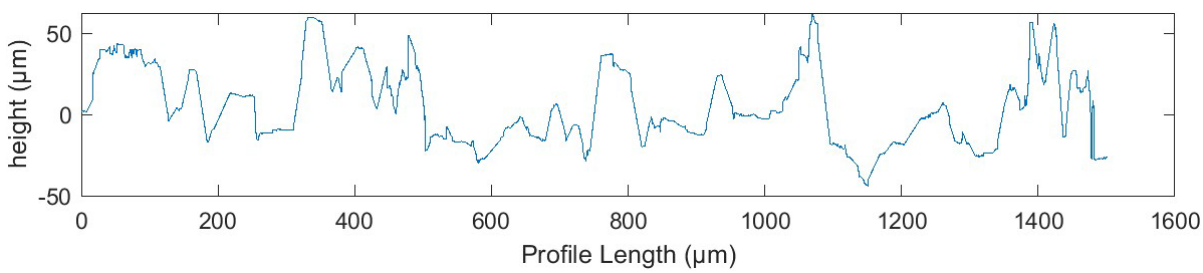

Figure H5: Flattened selected line profile from SWLI dataset sample 001

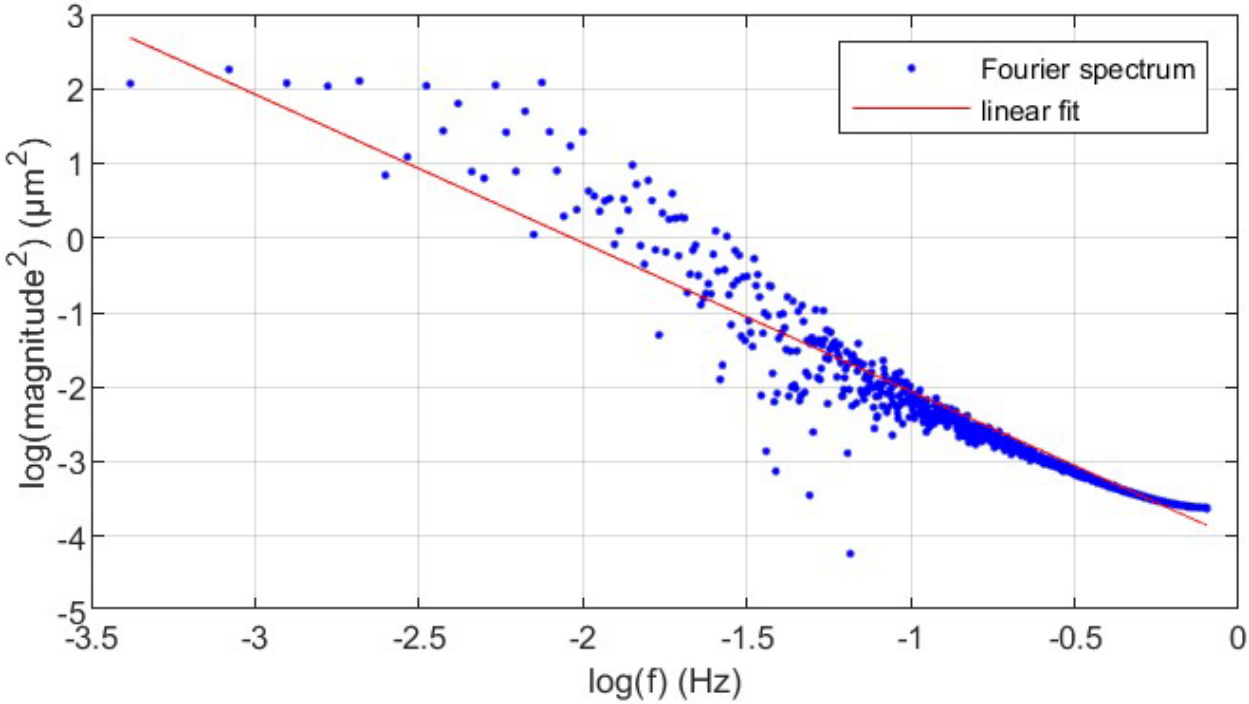

Figure H6: Fourier analysis and linear fit from Dektak line profile data sample 001

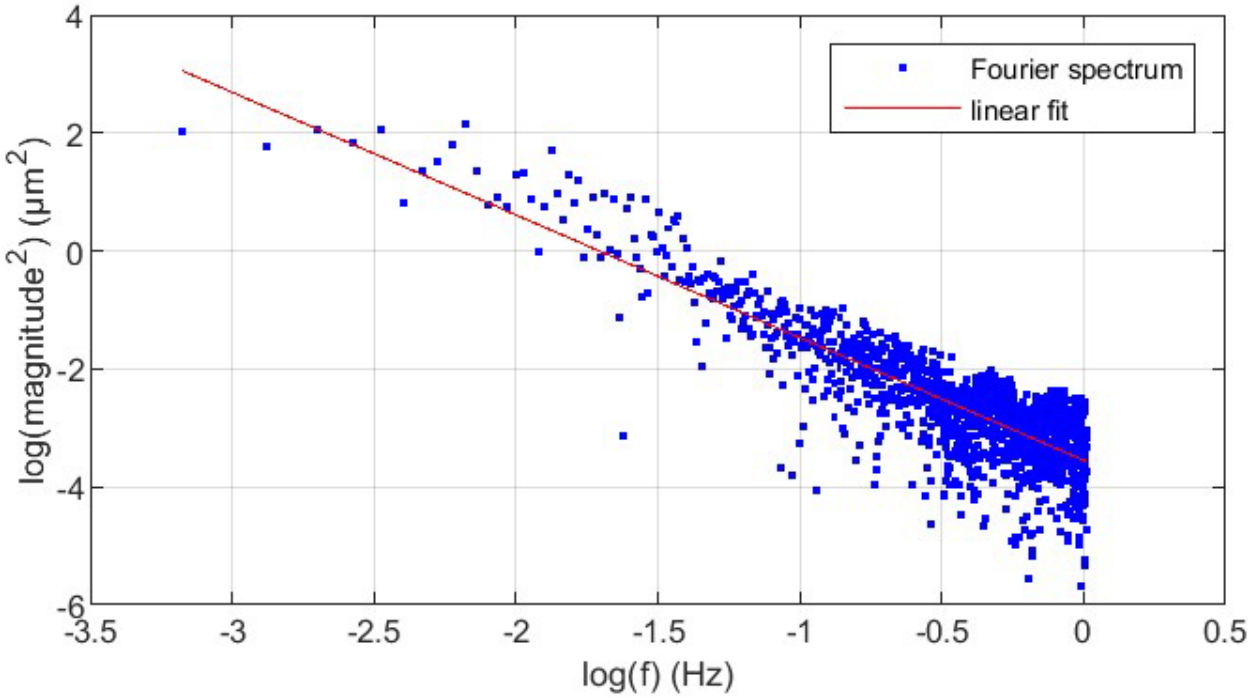

Figure H7: Fourier analysis and linear fit from SWLI line profile data sample 001

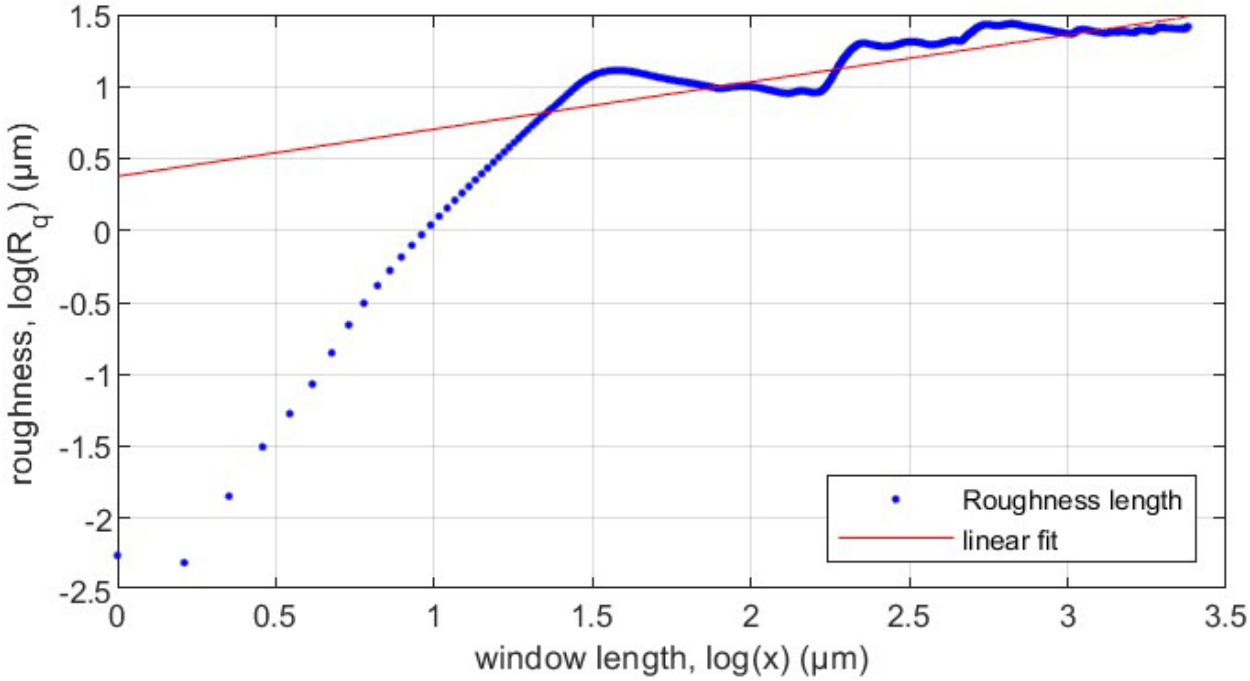

Figure H8: Roughness length method for calculating *H* from Dektak profile sample 001

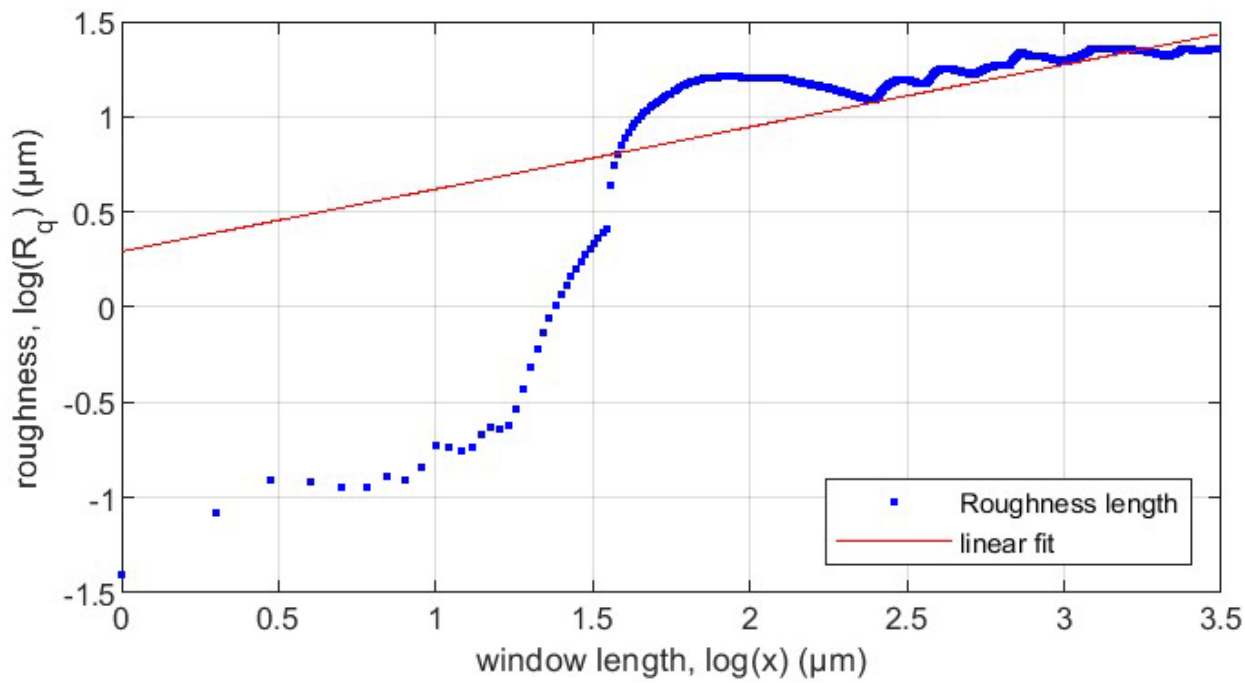

Figure H9: Roughness length method for calculating *H* from SWLI profile sample 001

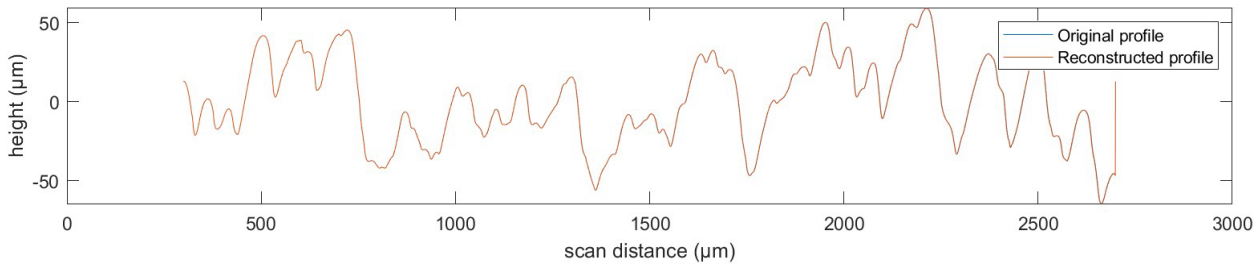

Figure H10: Original Dektak line scan profile and Fourier recreated line profile for sample 001

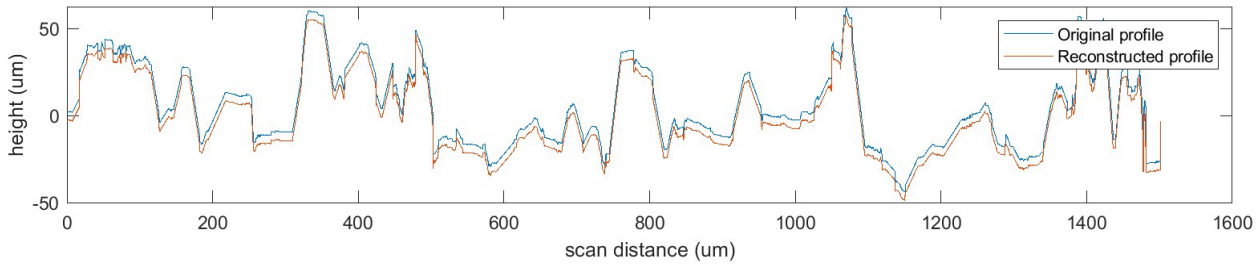

Figure H11: Original SWLI line profile and Fourier recreated line profile for sample 001

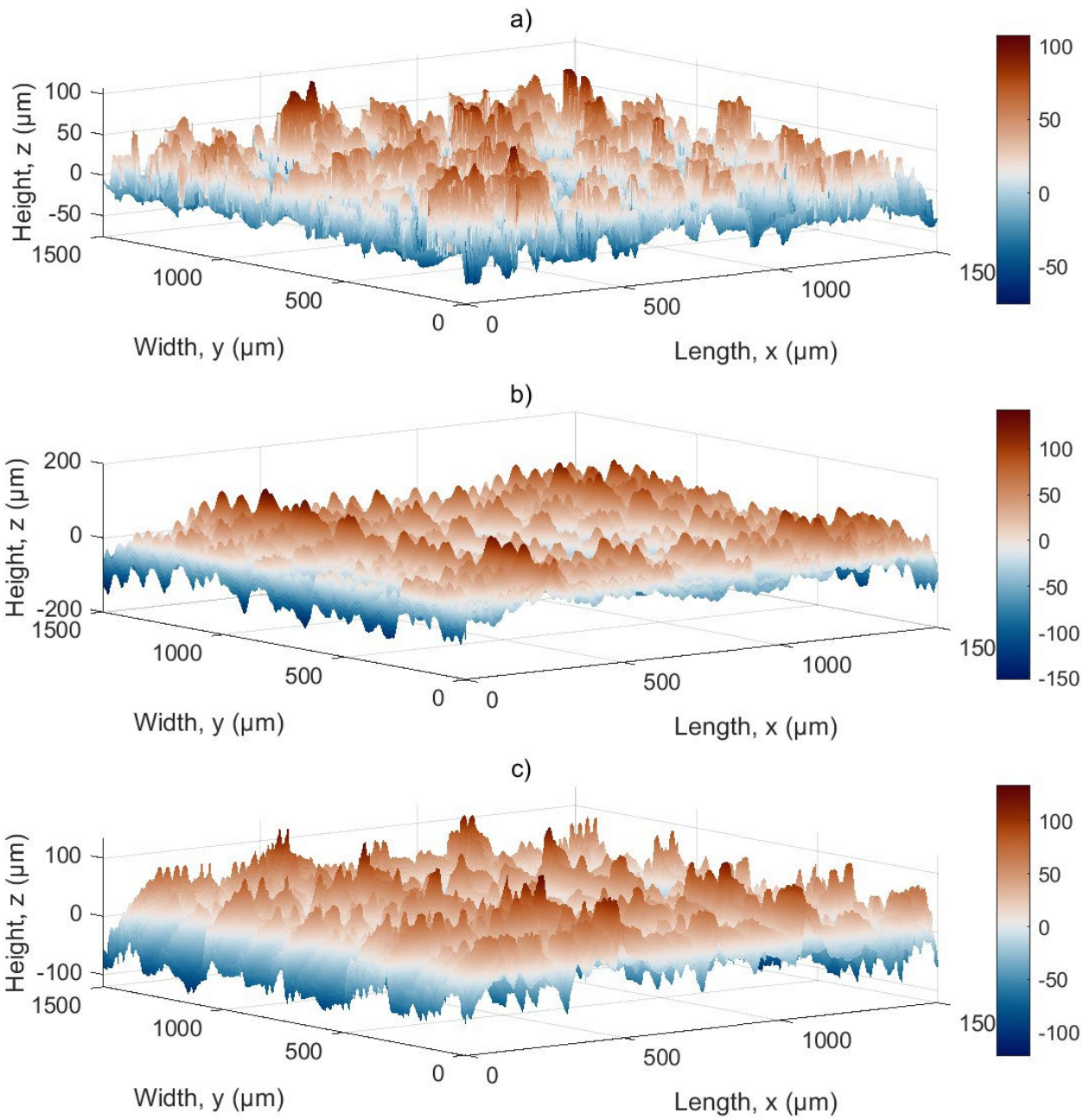

Figure H12: SWLI scan of sample 001 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 001 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 001

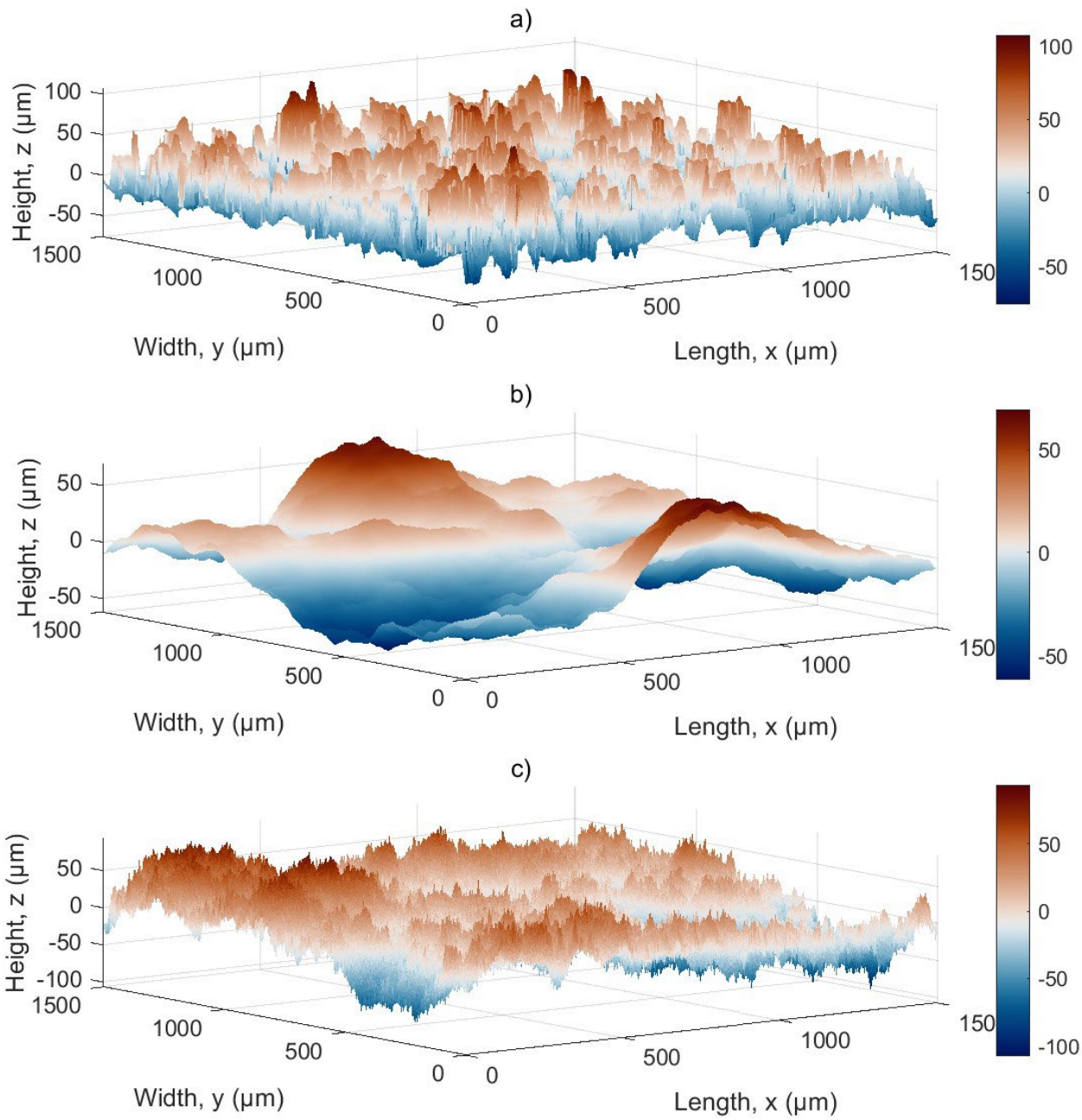

Figure H13: SWLI scan of sample 001 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 001 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 001

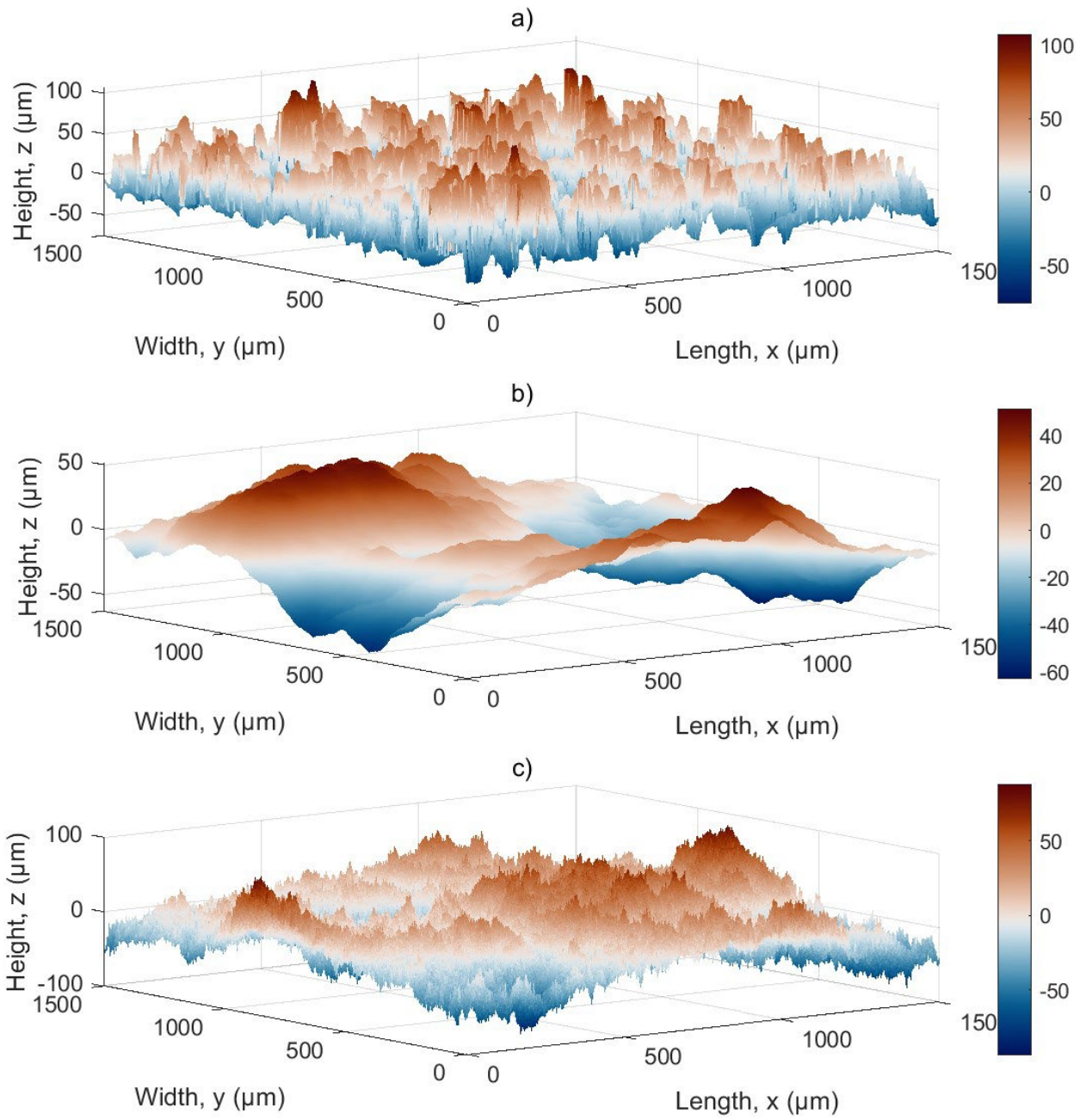

Figure H14: SWLI scan of sample 001 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 001 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 001

Appendix I: Sample 002 supplementary figures

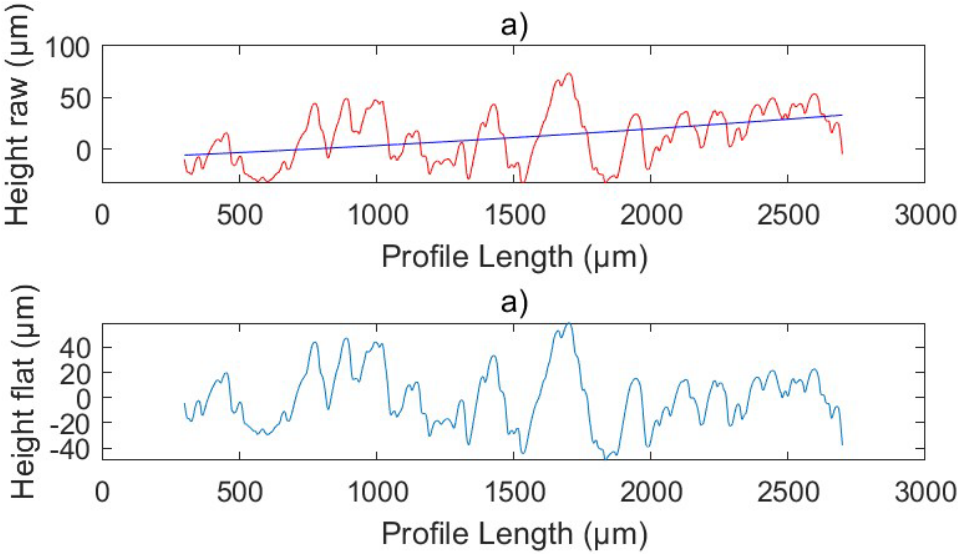

Figure I1: a) Raw profile measurement and fit curve sample 002 b) flattened line profile from sample 002

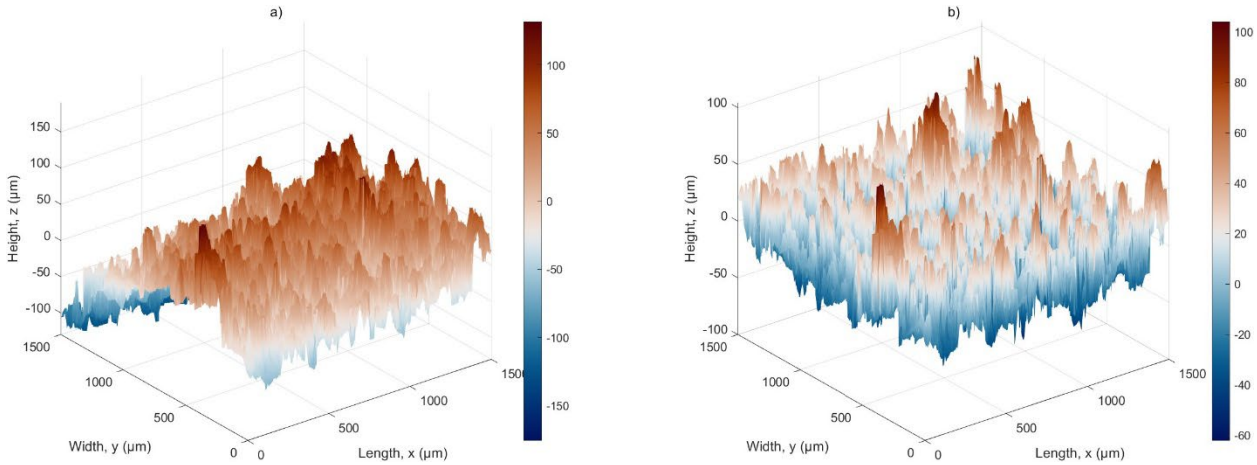

Figure I2: a) Unflattened SWLI raw surface scan sample 002 b) plane flattened SWLI surface 002

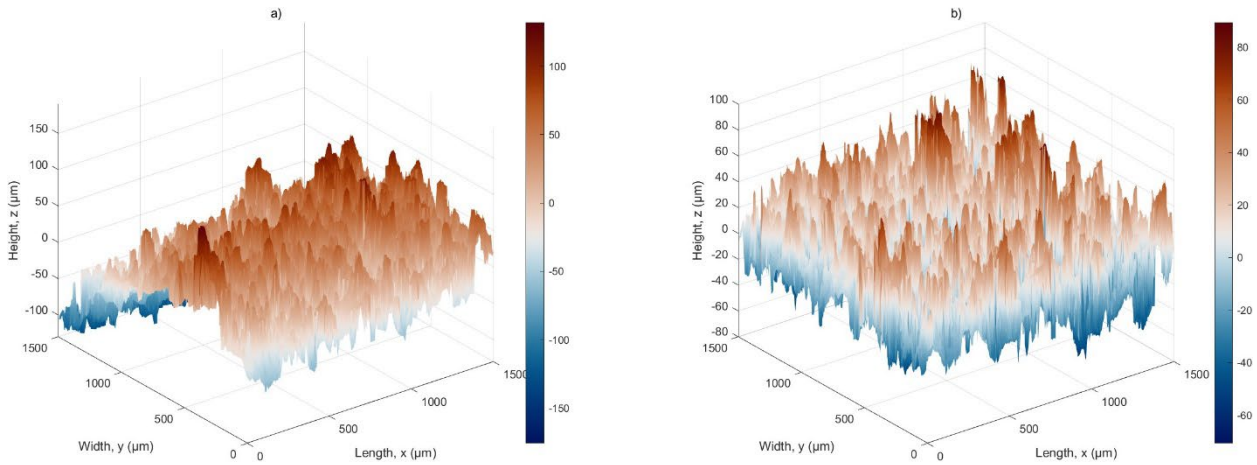

Figure I3: a) Unflattened SWLI raw surface scan sample 002 b) line by line flattened SWLI surface 002

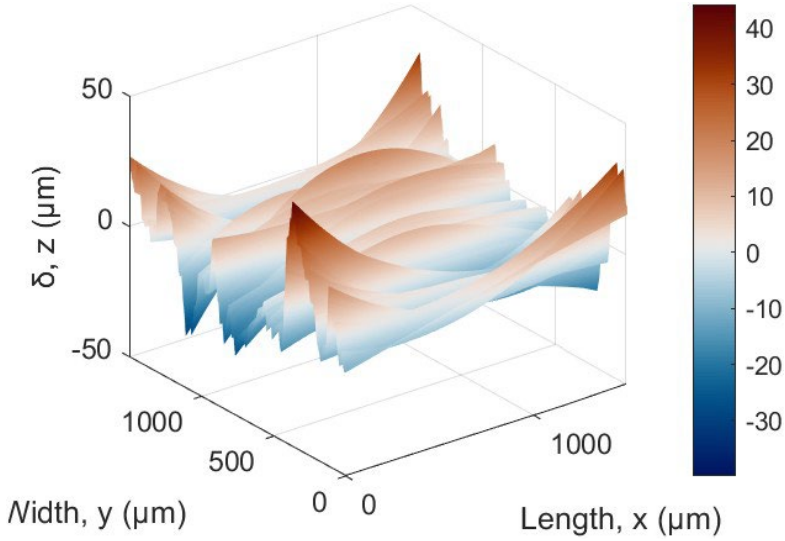

Figure I4: Difference between plane flattening and line by line flattening for sample 002

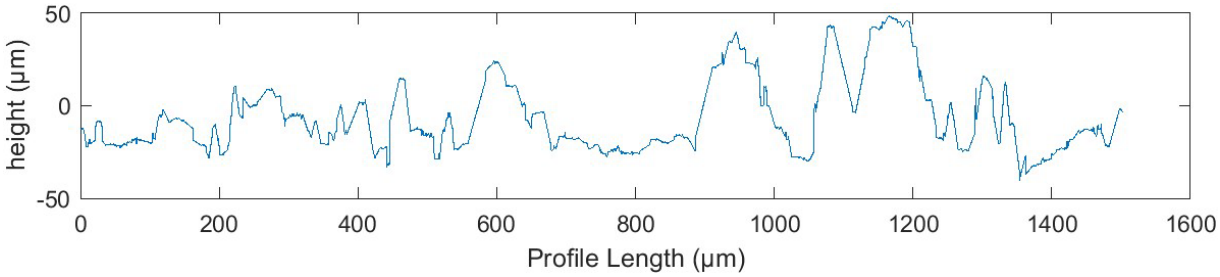

Figure I5: Flattened selected line profile from SWLI dataset sample 002

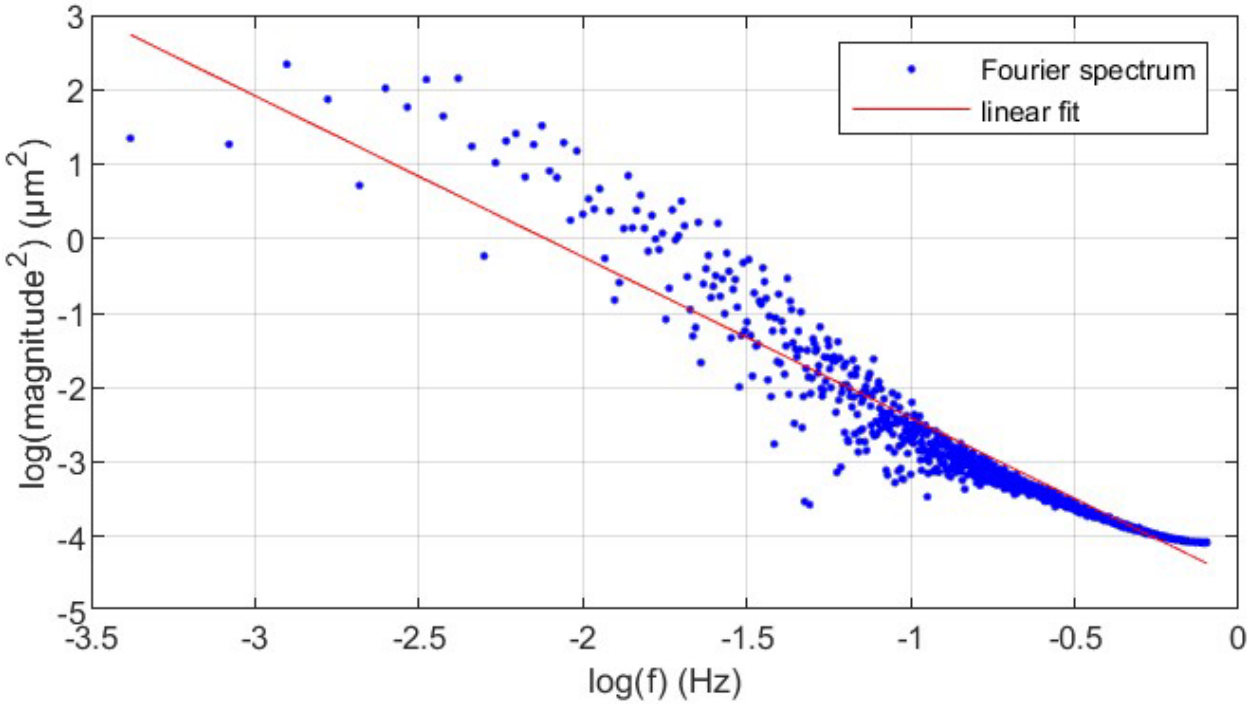

Figure I6: Fourier analysis and linear fit from Dektak line profile data sample 002

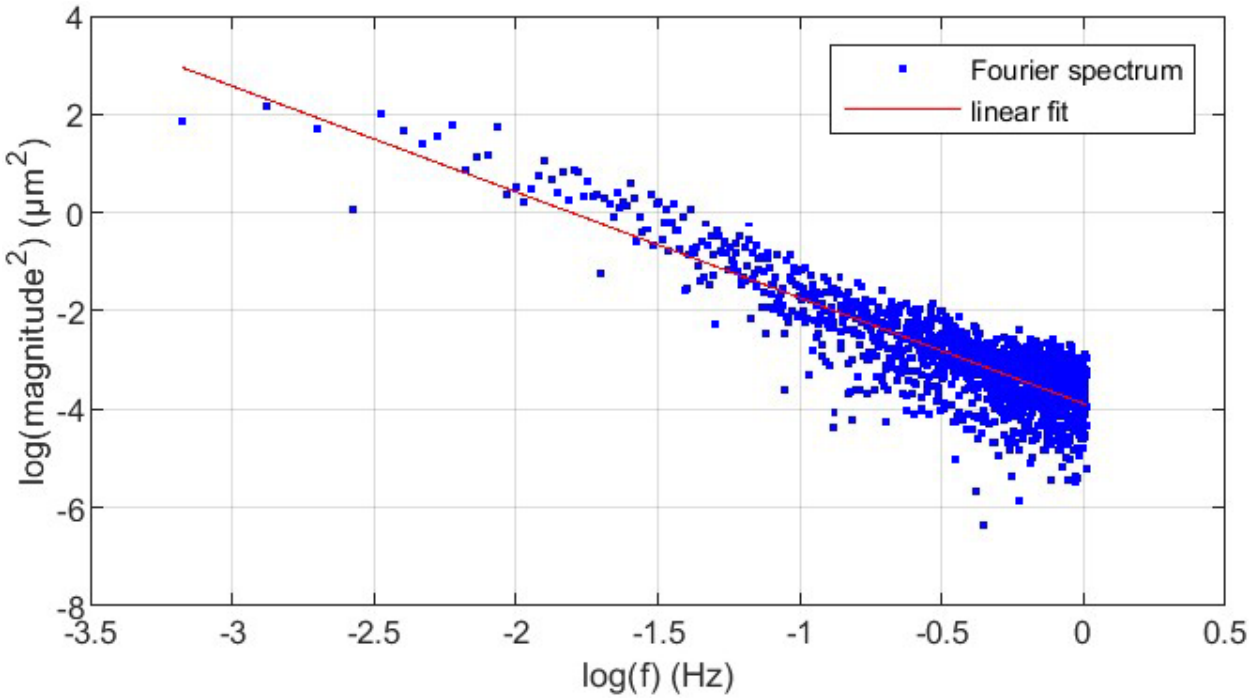

Figure I7: Fourier analysis and linear fit from SWLI line profile data sample 002

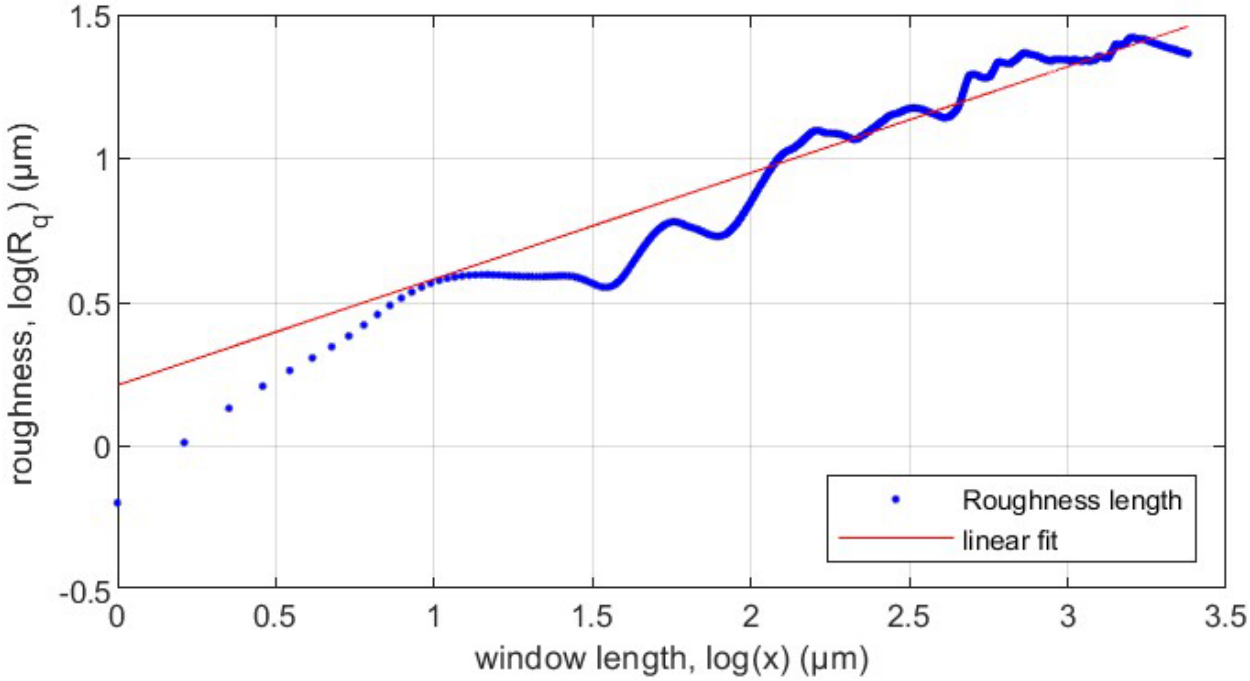

Figure I8: Roughness length method for calculating *H* from Dektak profile sample 002

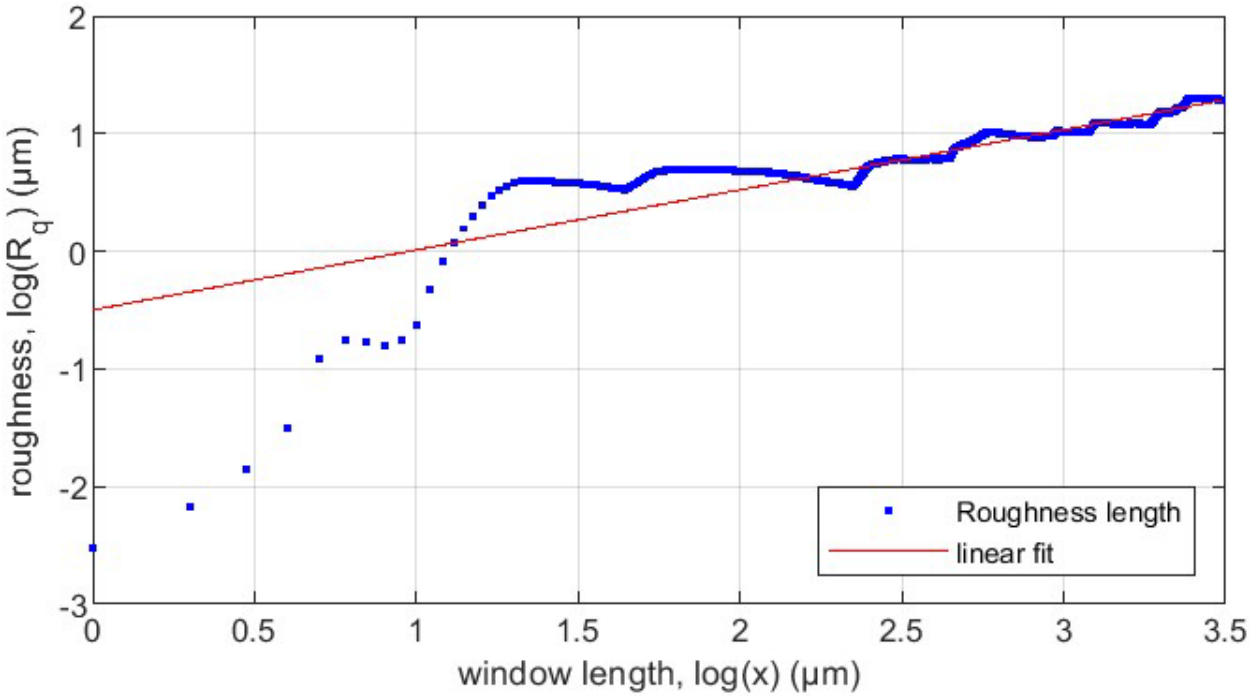

Figure I9: Roughness length method for calculating *H* from SWLI profile sample 002

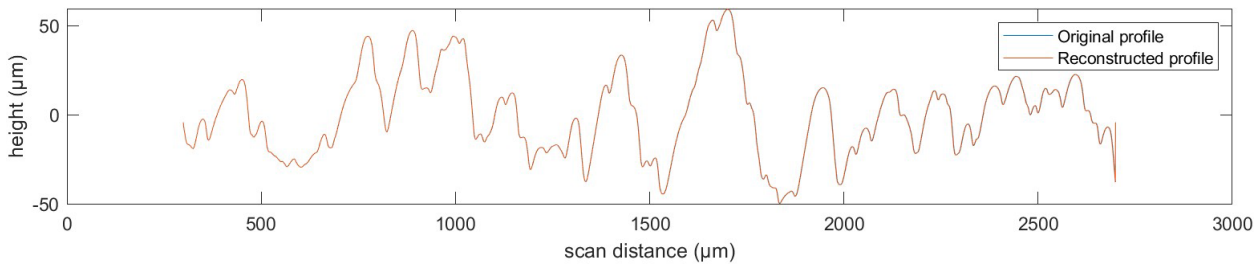

Figure I10: Original Dektak line scan profile and Fourier recreated line profile for sample 002

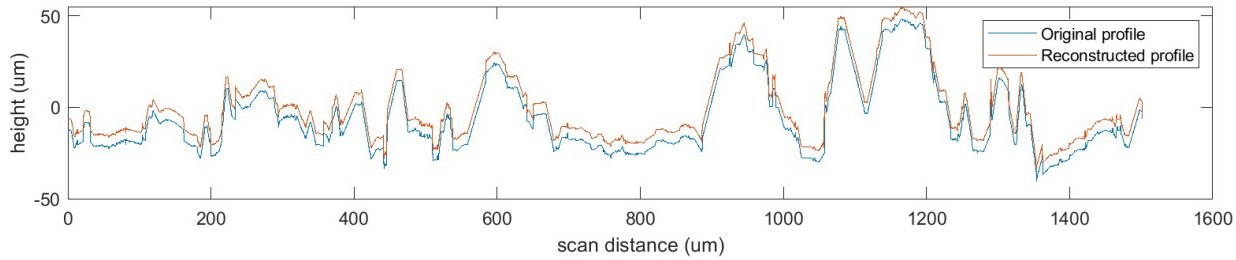

Figure I11: Original SWLI line profile and Fourier recreated line profile for sample 002

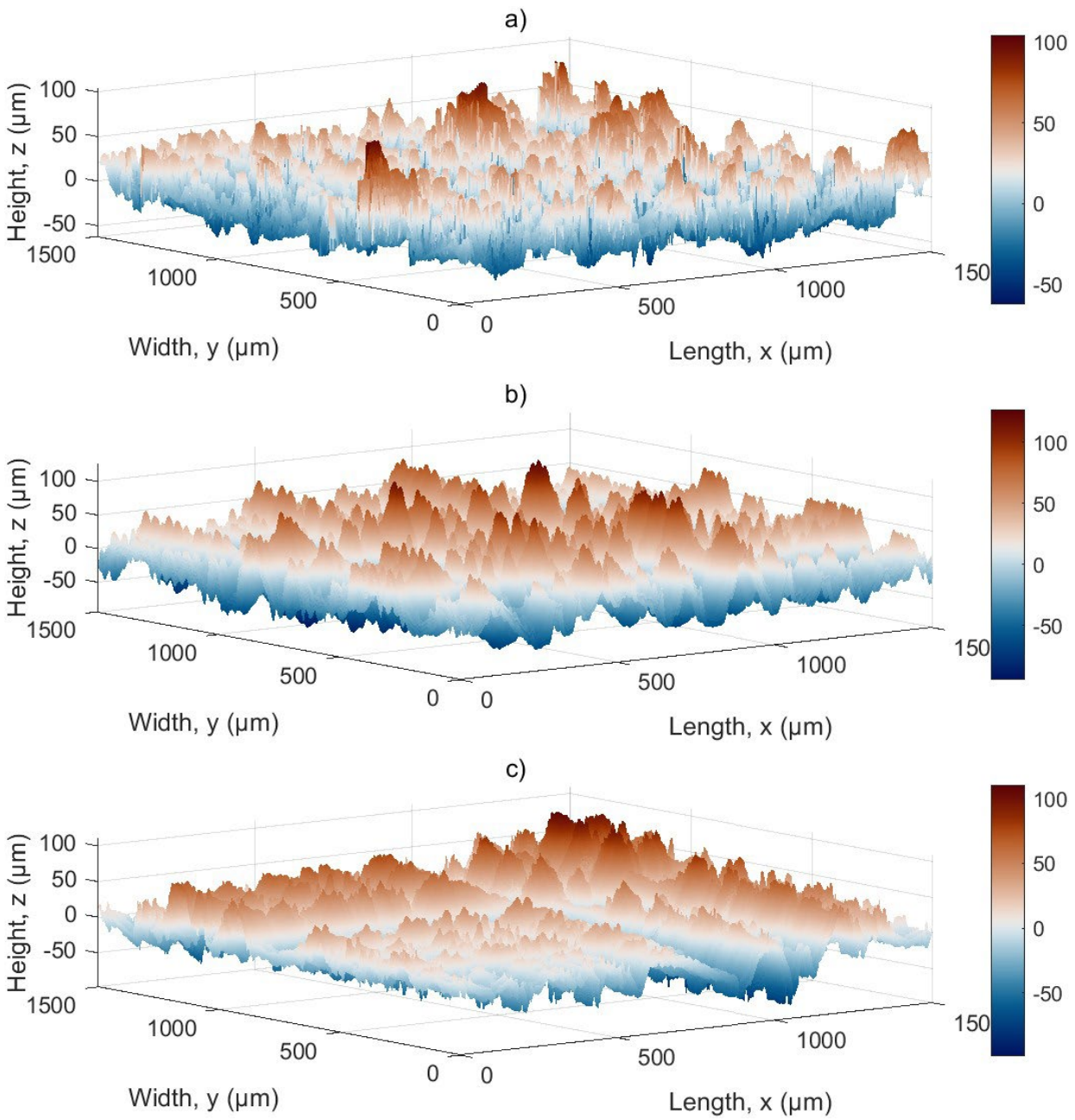

Figure I12: SWLI scan of sample 002 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 002 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 002

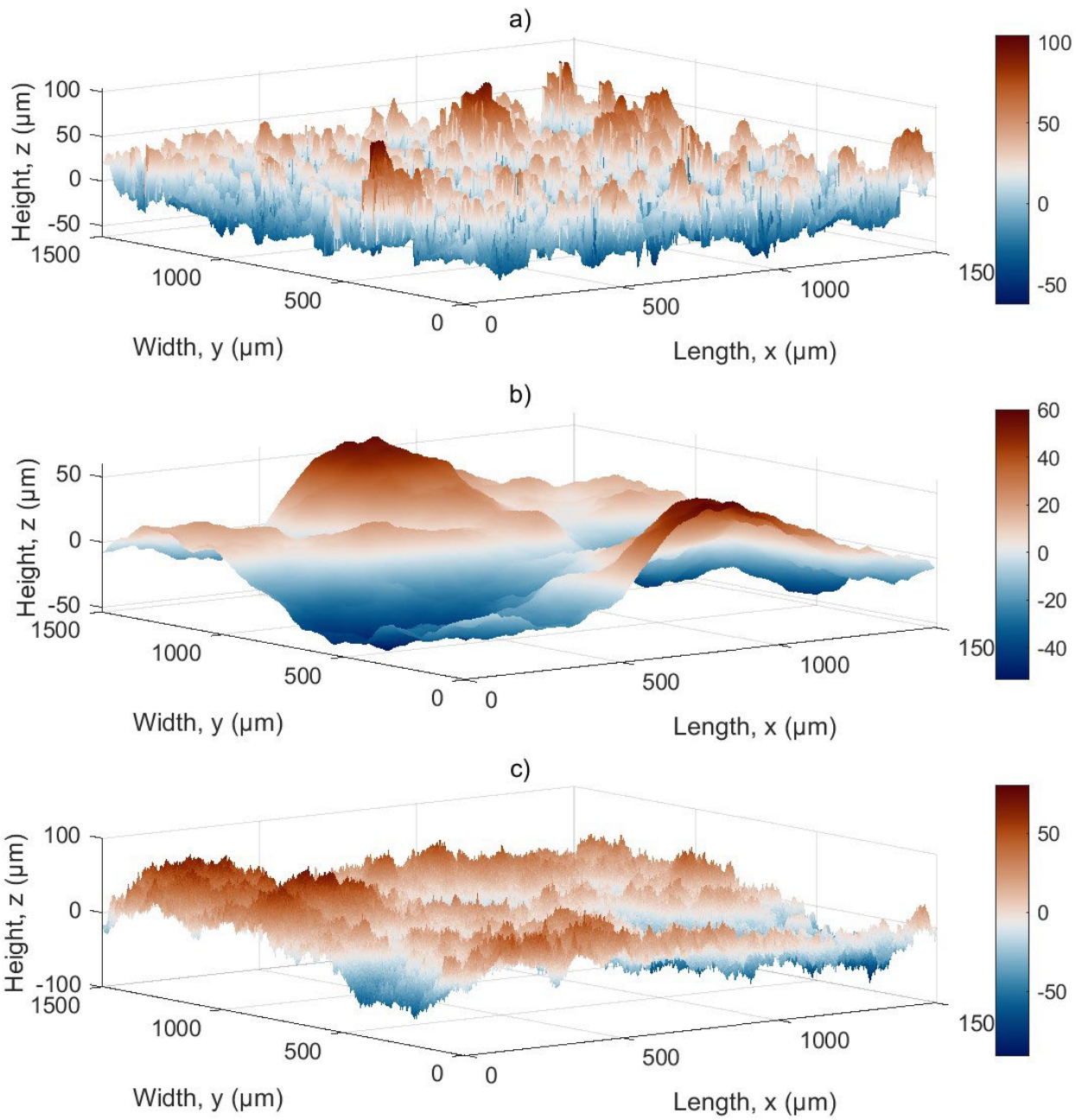

Figure I13: SWLI scan of sample 002 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 002 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 002

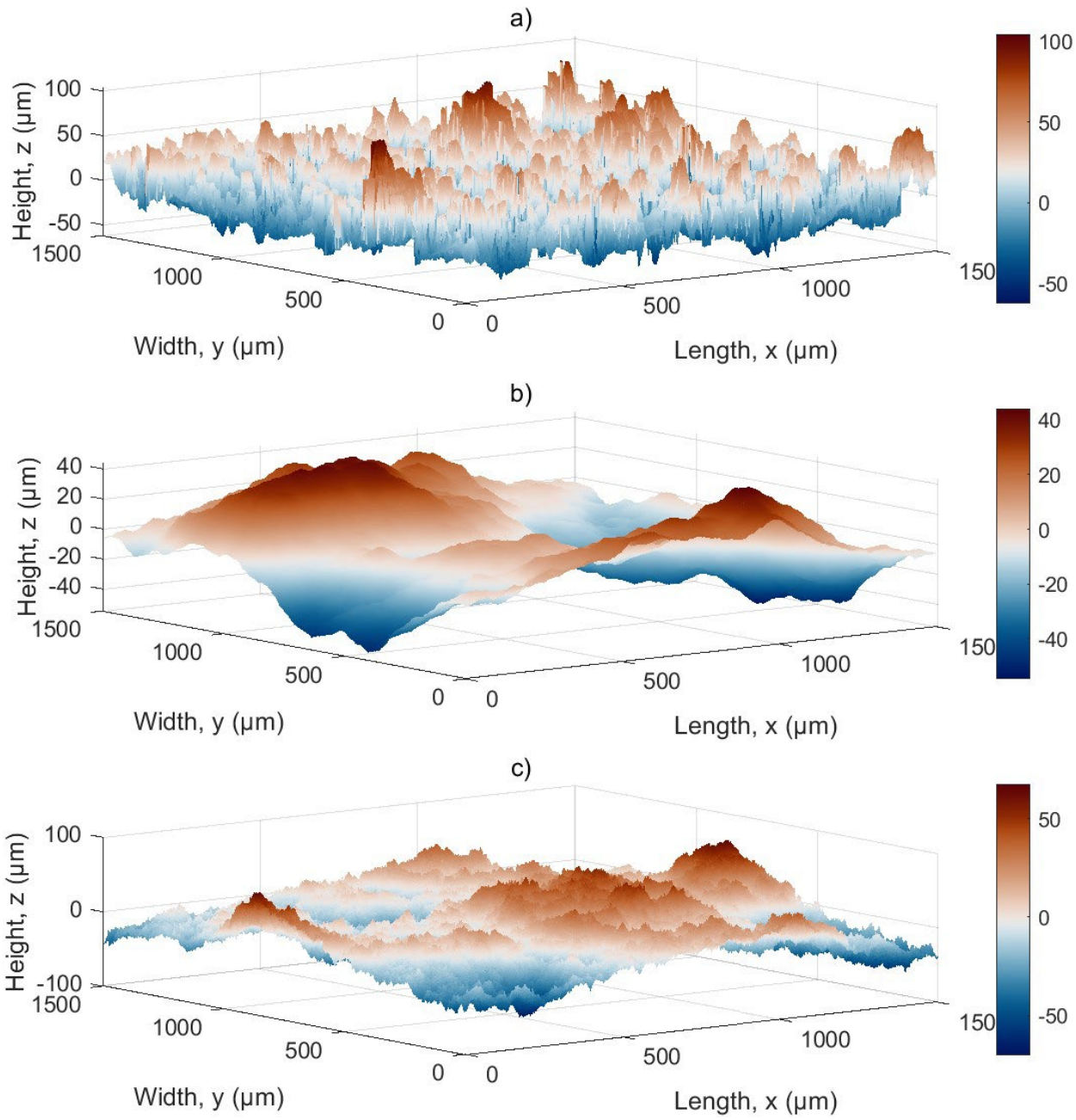

Figure I14: SWLI scan of sample 002 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 002 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 002

Appendix J: Sample 003 supplementary figures

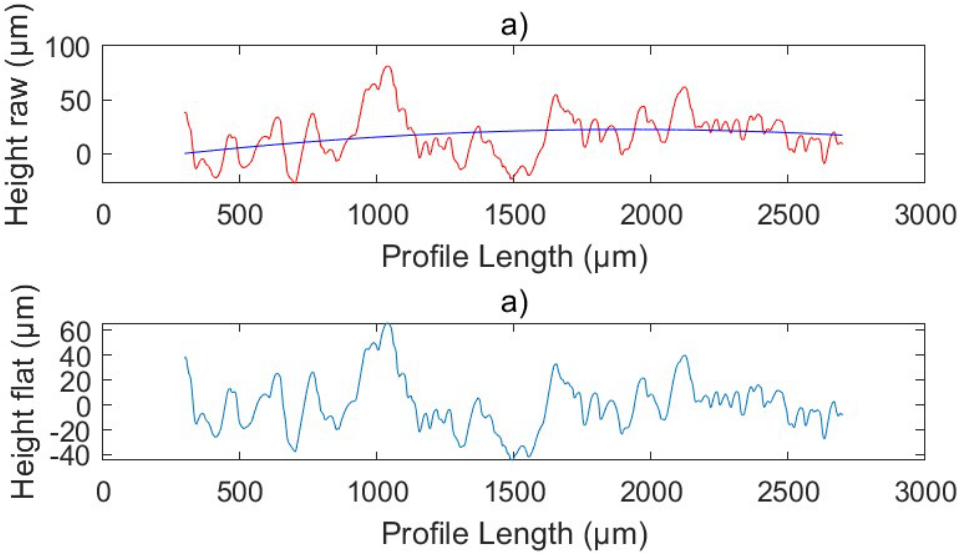

Figure J1: a) Raw profile measurement and fit curve sample 003 b) flattened line profile from sample 003

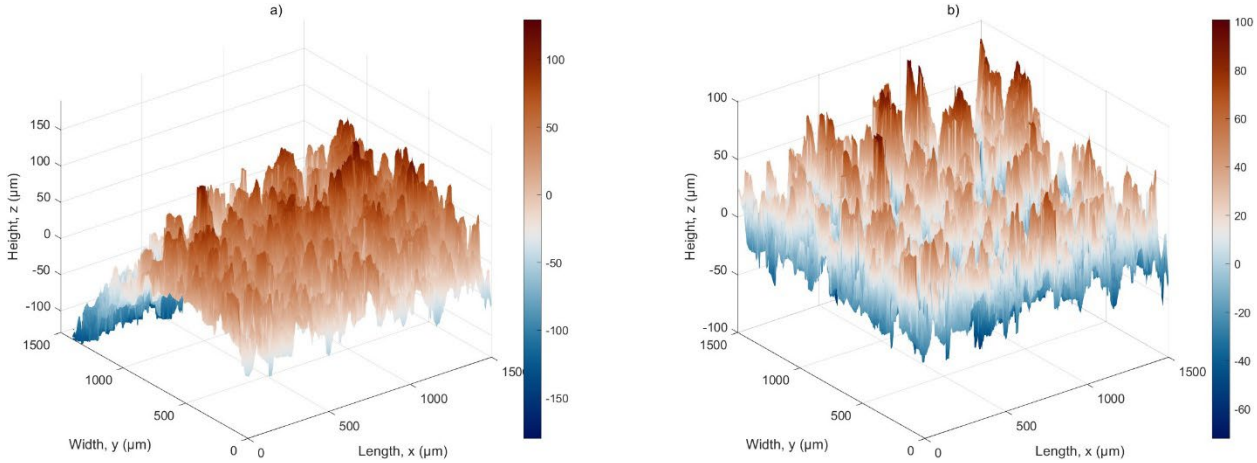

Figure J2: a) Unflattened SWLI raw surface scan sample 003 b) plane flattened SWLI surface 003

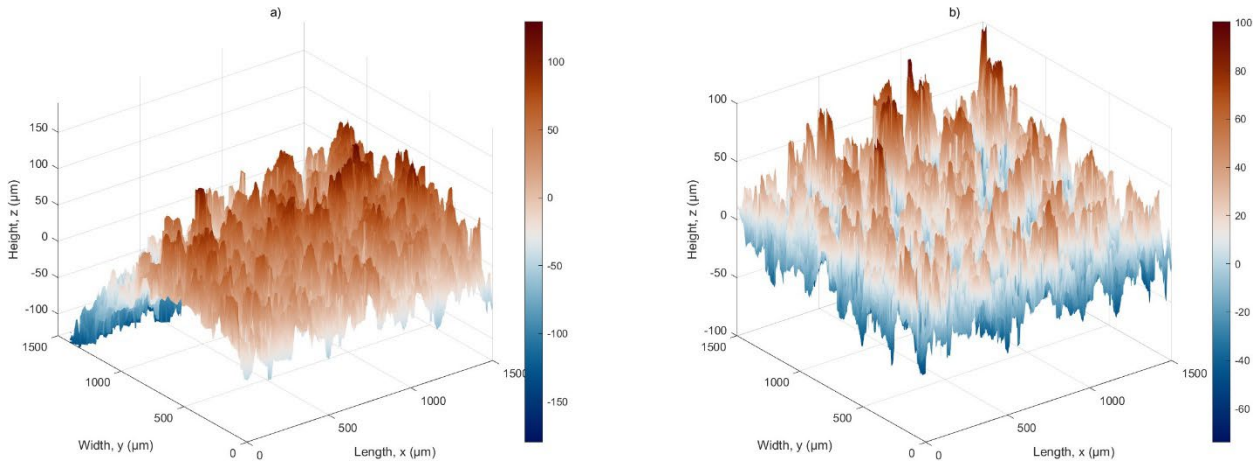

Figure J3: a) Unflattened SWLI raw surface scan sample 003 b) line by line flattened SWLI surface 003

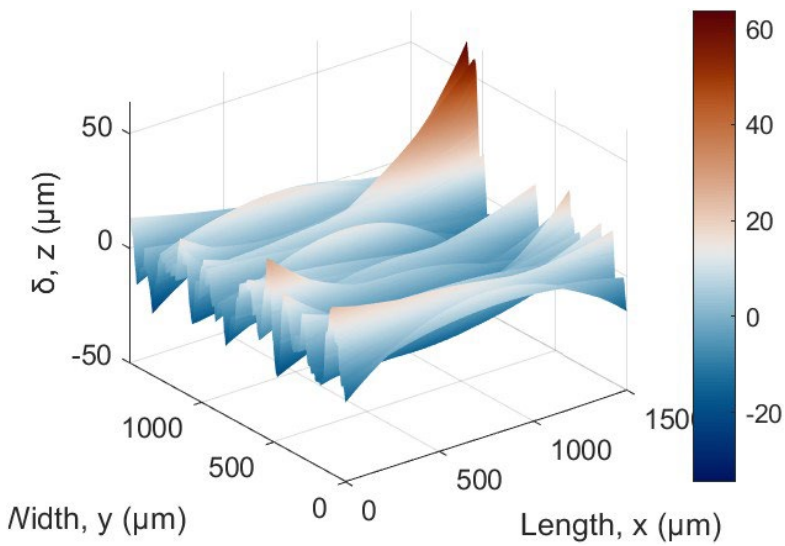

Figure J4: Difference between plane flattening and line by line flattening for sample 003

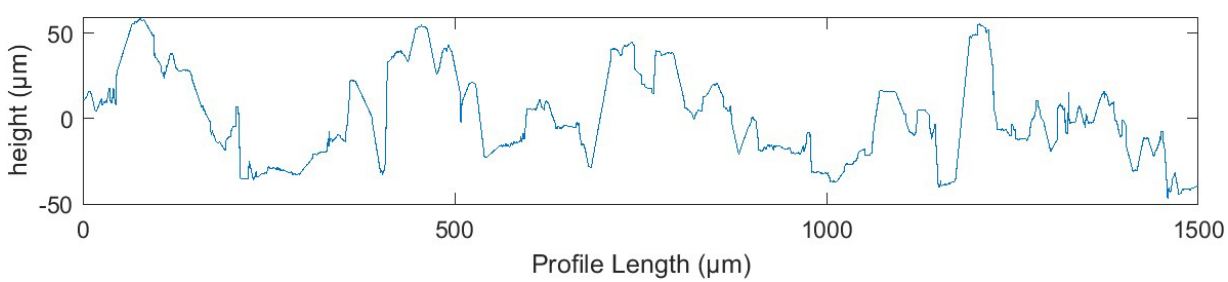

Figure J5: Flattened selected line profile from SWLI dataset sample 003

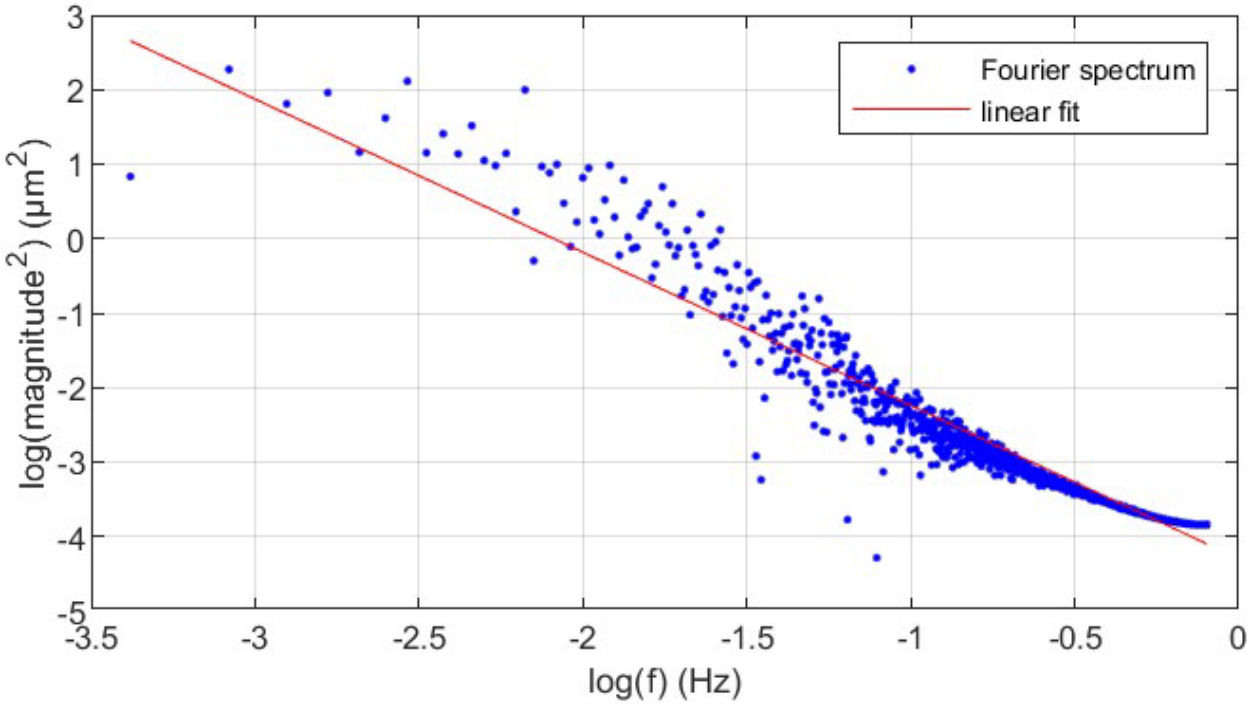

Figure J6: Fourier analysis and linear fit from Dektak line profile data sample 003

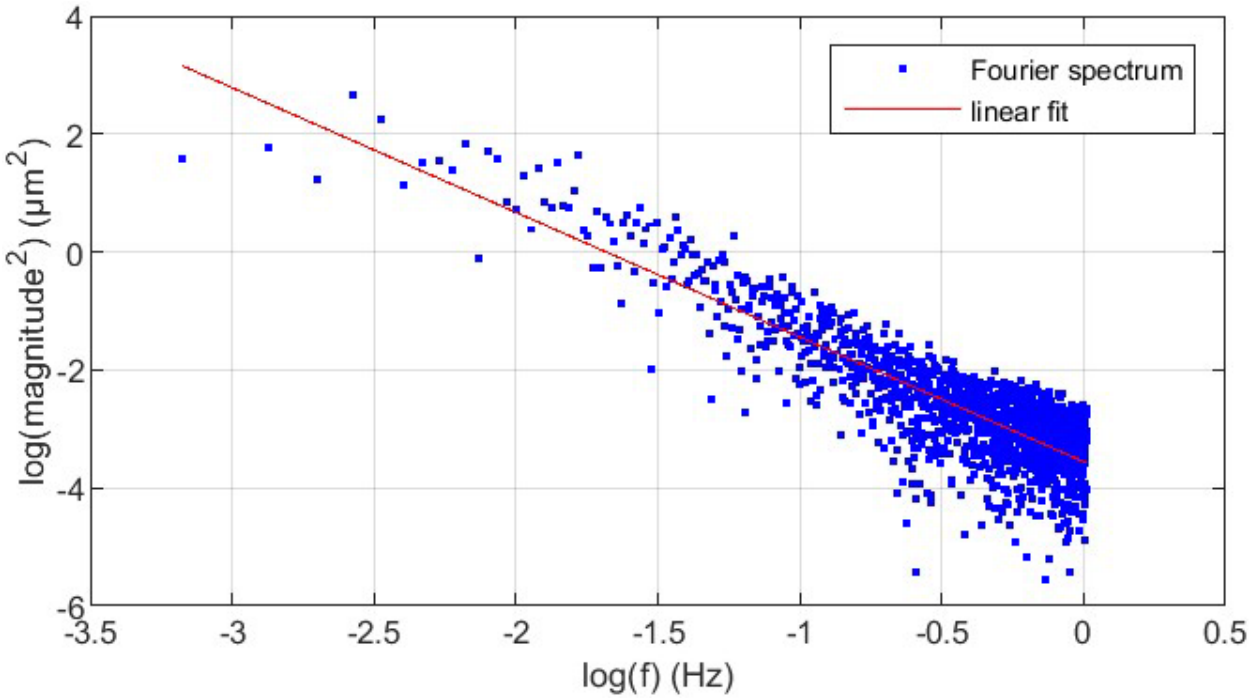

Figure J7: Fourier analysis and linear fit from SWLI line profile data sample 003

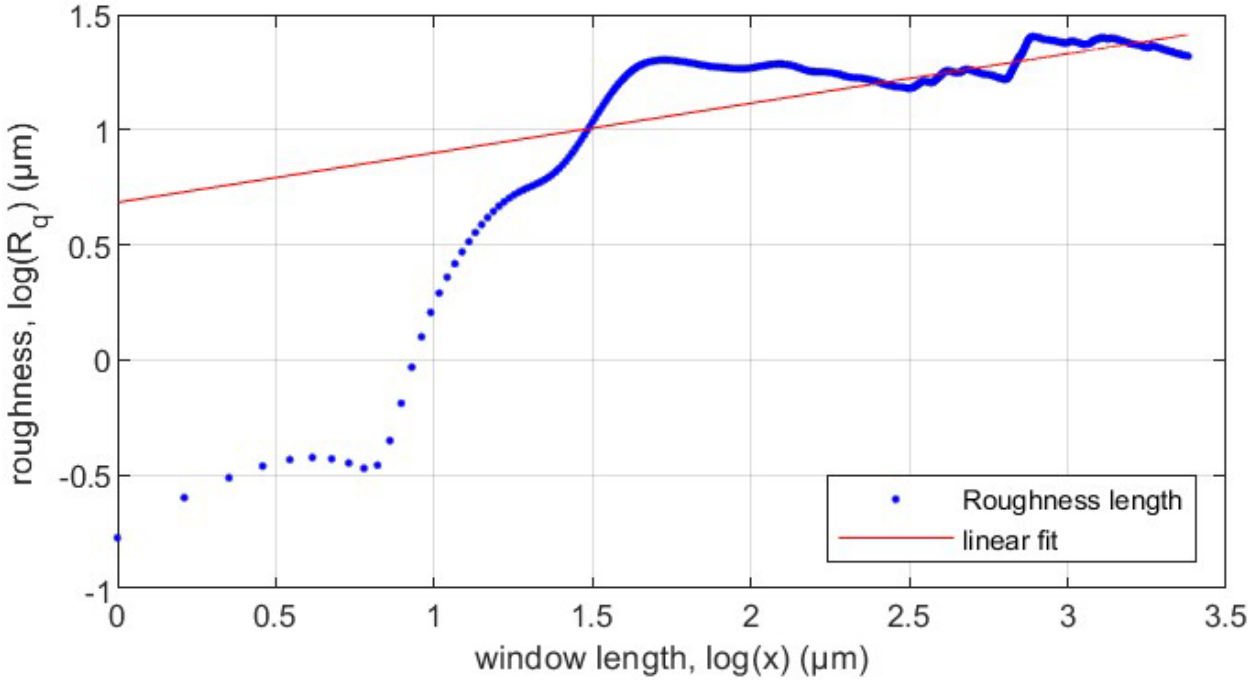

Figure J8: Roughness length method for calculating *H* from Dektak profile sample 003

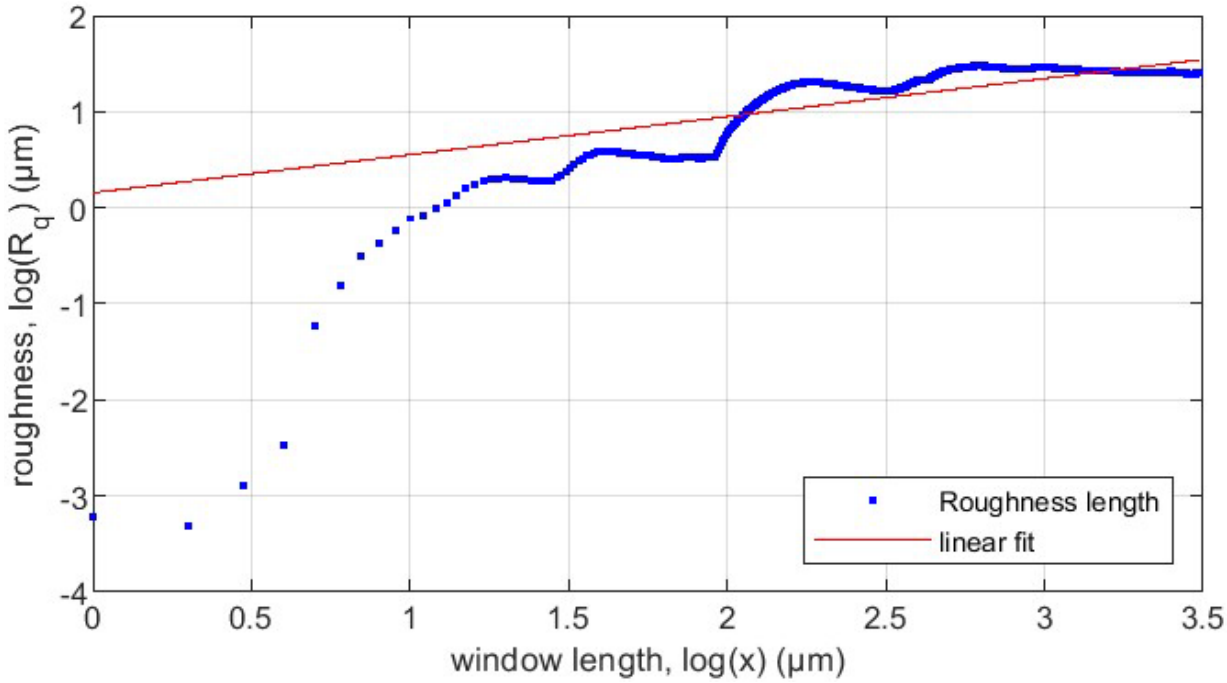

Figure J9: Roughness length method for calculating *H* from SWLI profile sample 003

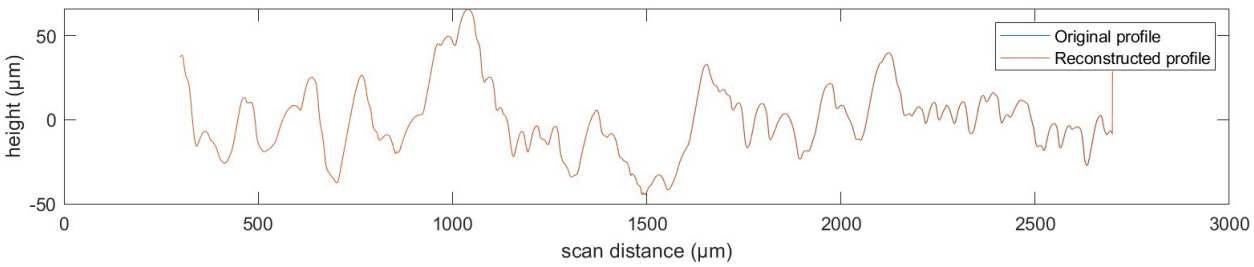

Figure J10: Original Dektak line scan profile and Fourier recreated line profile for sample 003

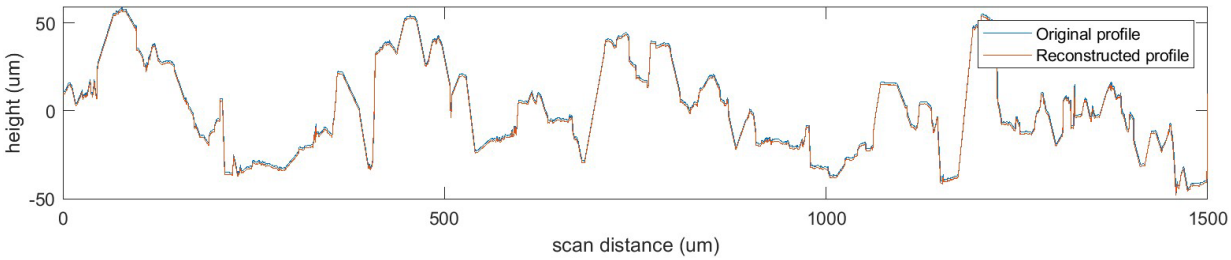

Figure J11: Original SWLI line profile and Fourier recreated line profile for sample 003

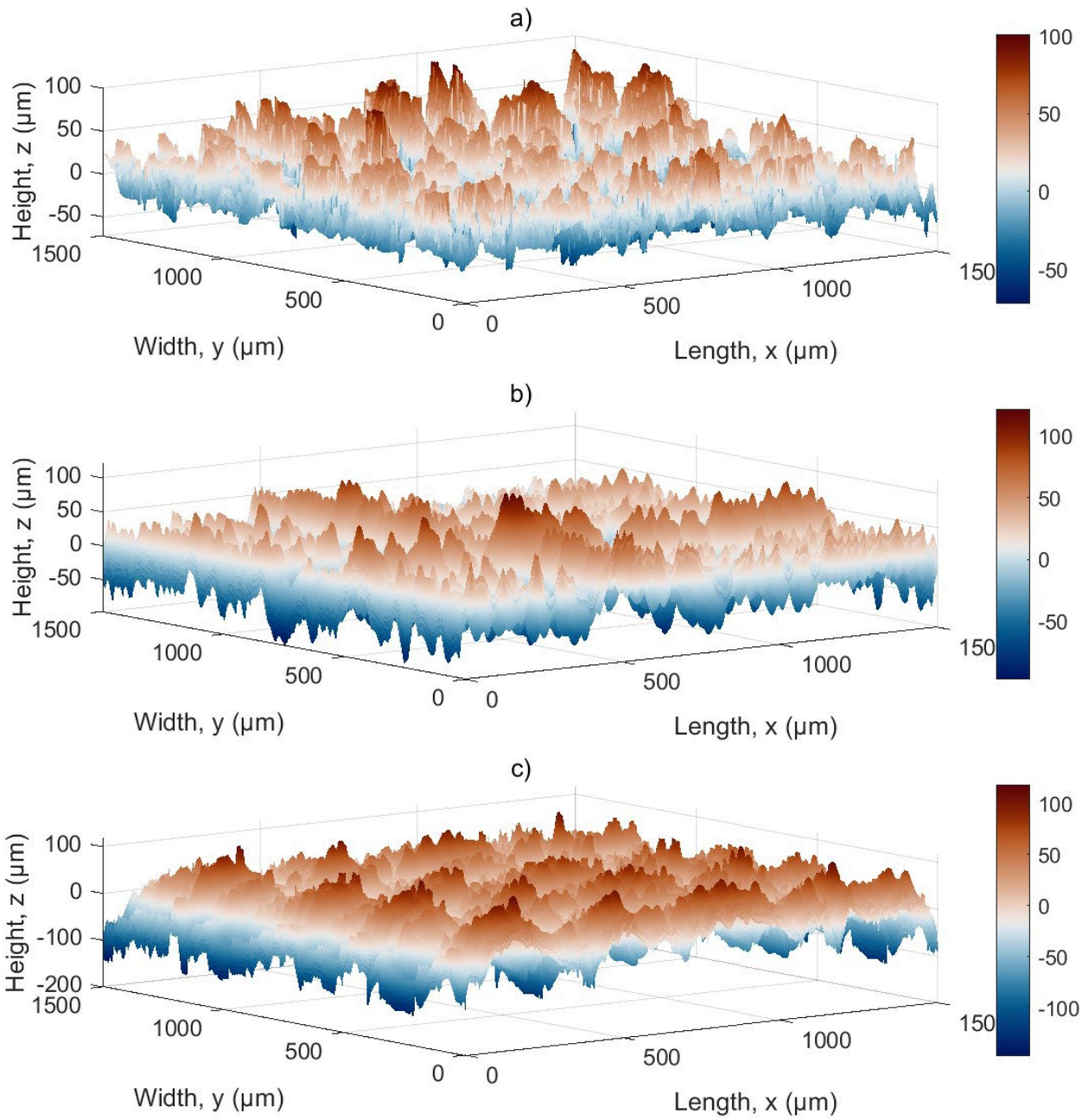

Figure J12: SWLI scan of sample 003 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 003 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 003

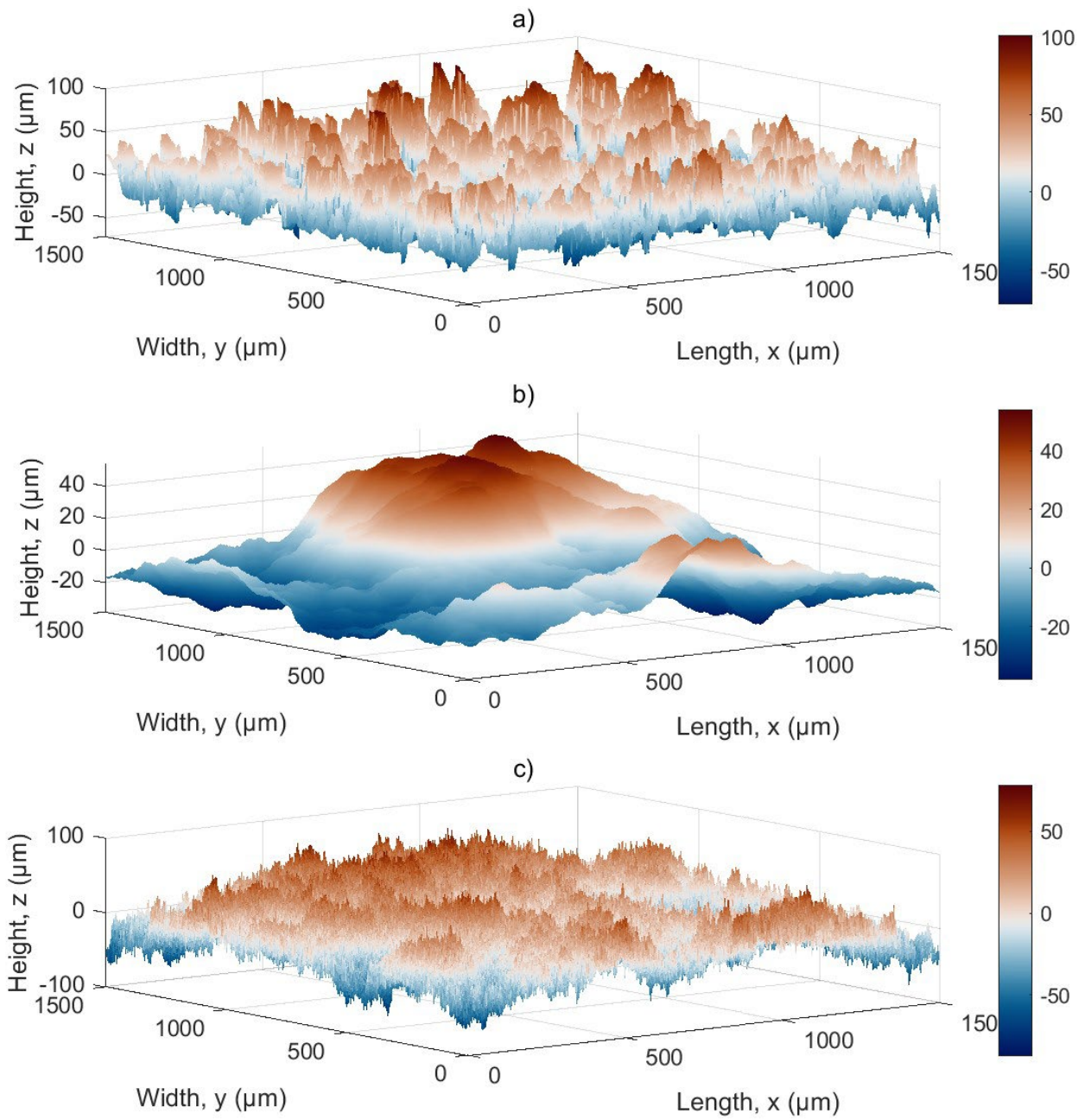

Figure J13: SWLI scan of sample 003 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 003 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 003

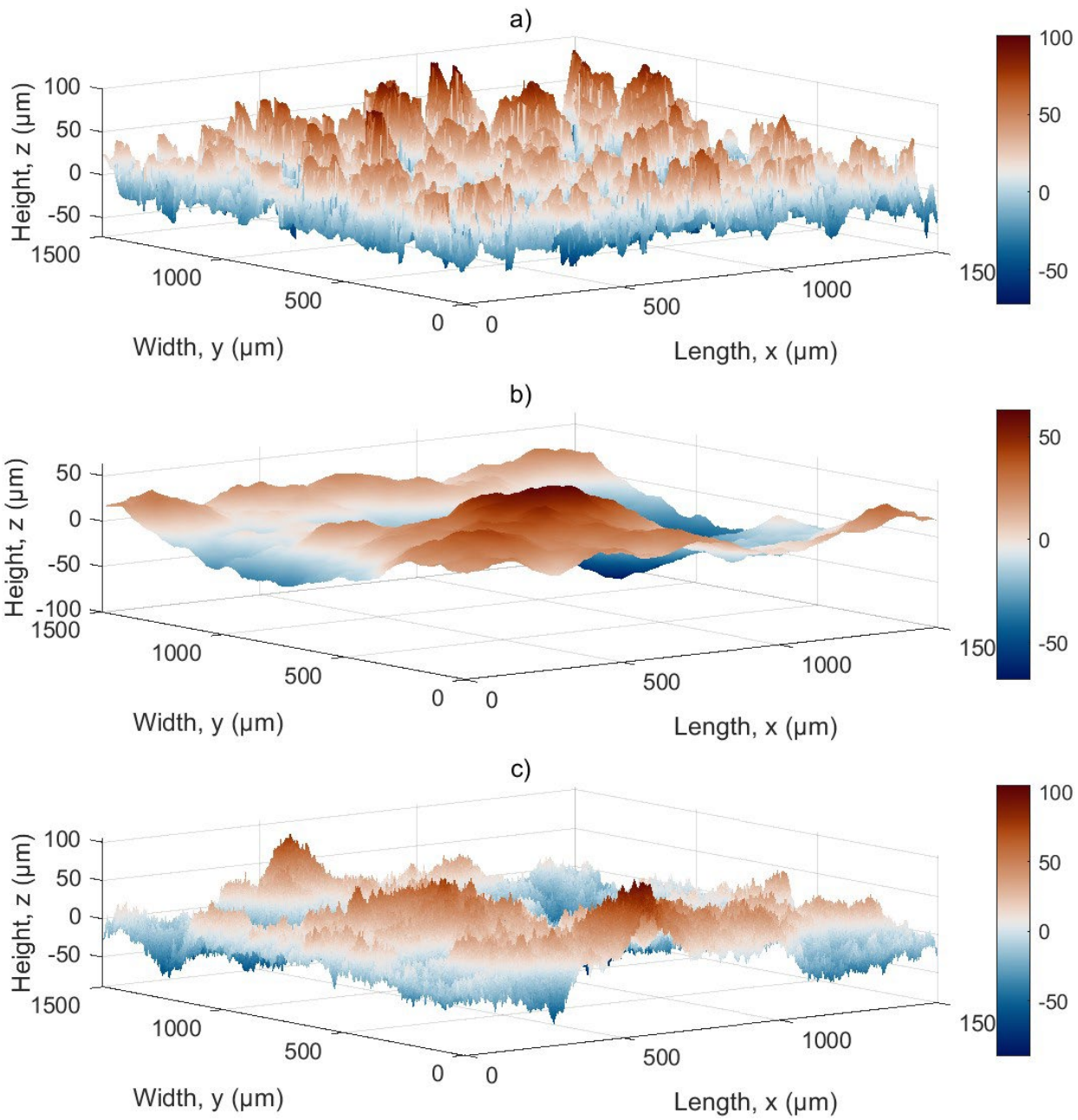

Figure J14: SWLI scan of sample 003 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 003 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 003

Appendix K: Sample 004 supplementary figures

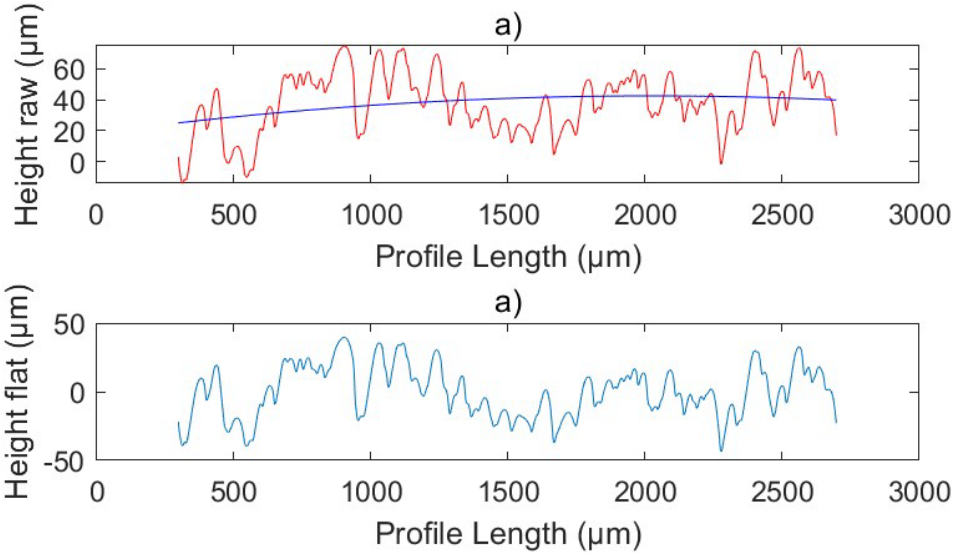

Figure K1: a) Raw profile measurement and fit curve sample 004 b) flattened line profile from sample 004

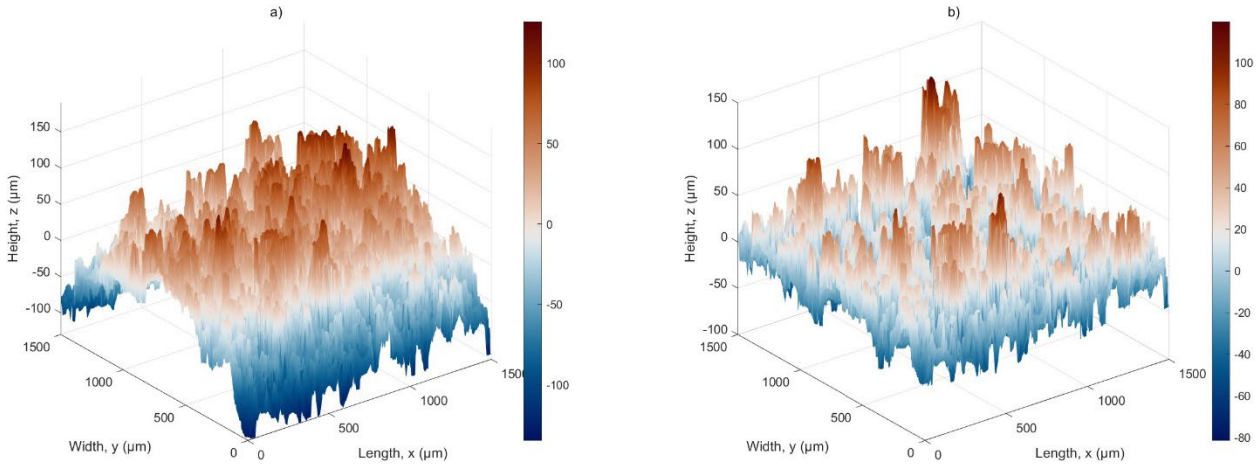

Figure K2: a) Unflattened SWLI raw surface scan sample 004 b) plane flattened SWLI surface 004

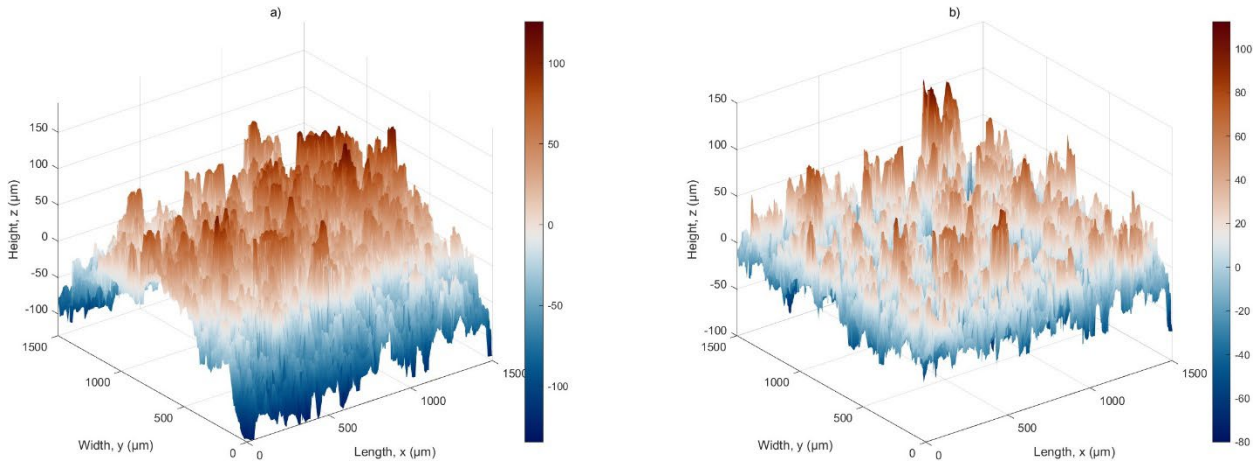

Figure K3: a) Unflattened SWLI raw surface scan sample 004 b) line by line flattened SWLI surface 004

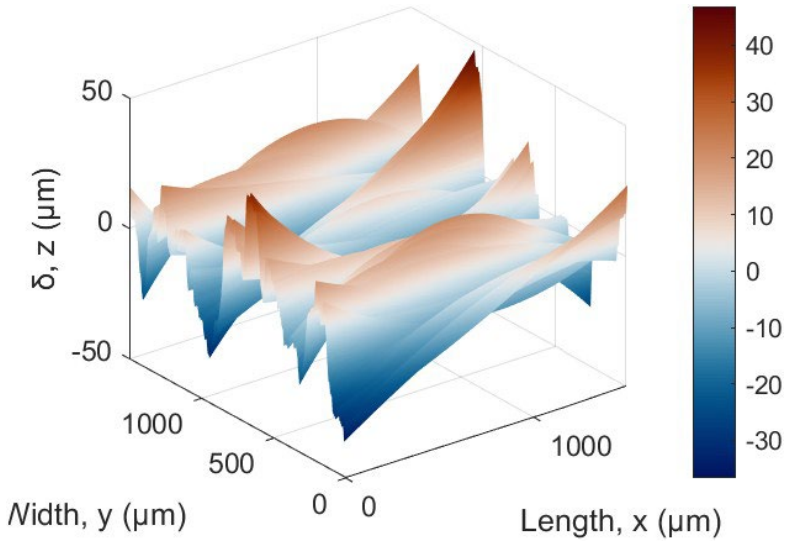

Figure K4: Difference between plane flattening and line by line flattening for sample 004

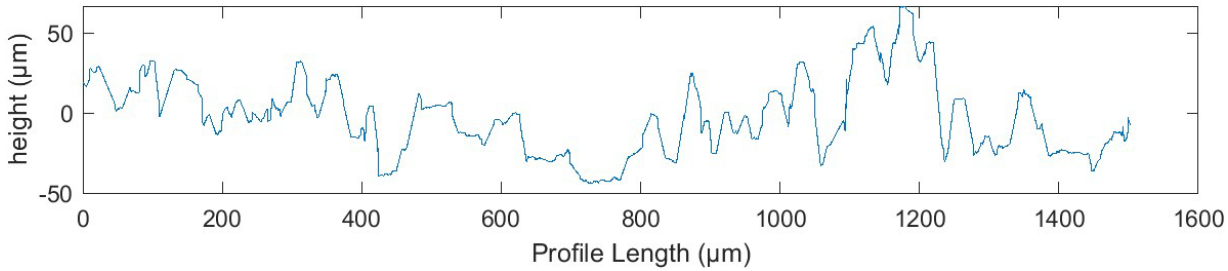

Figure K5: Flattened selected line profile from SWLI dataset sample 004

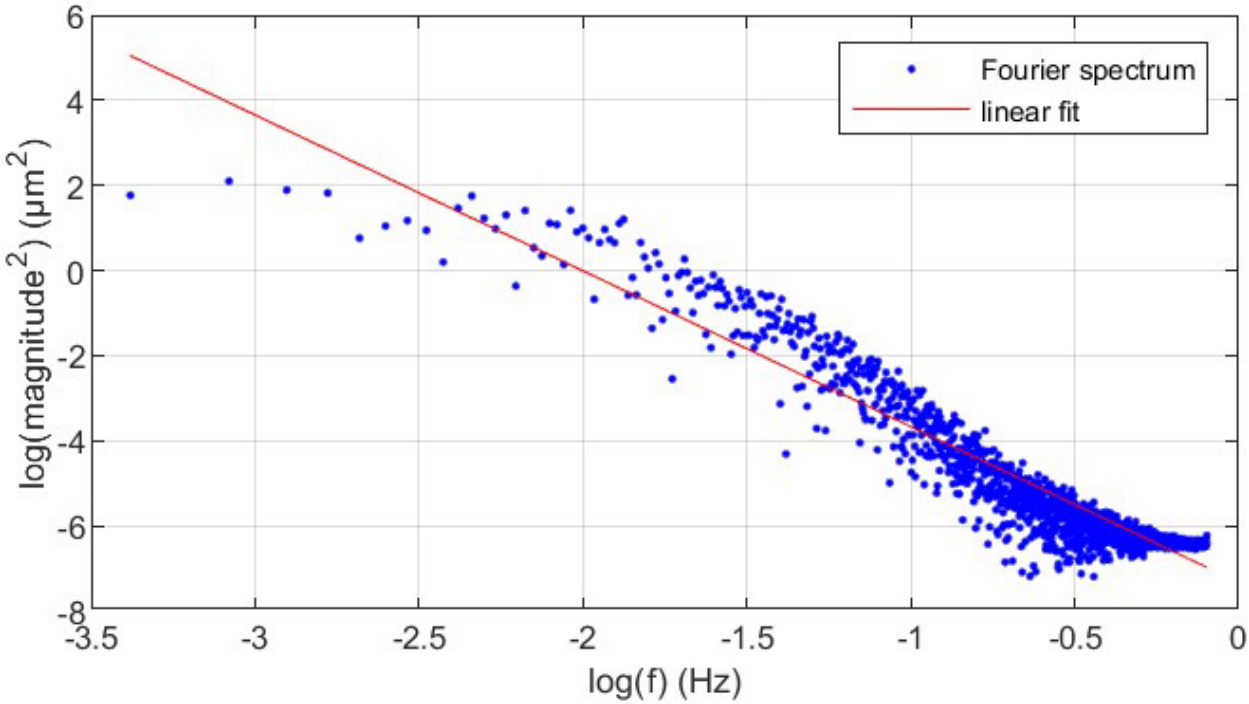

Figure K6: Fourier analysis and linear fit from Dektak line profile data sample 004

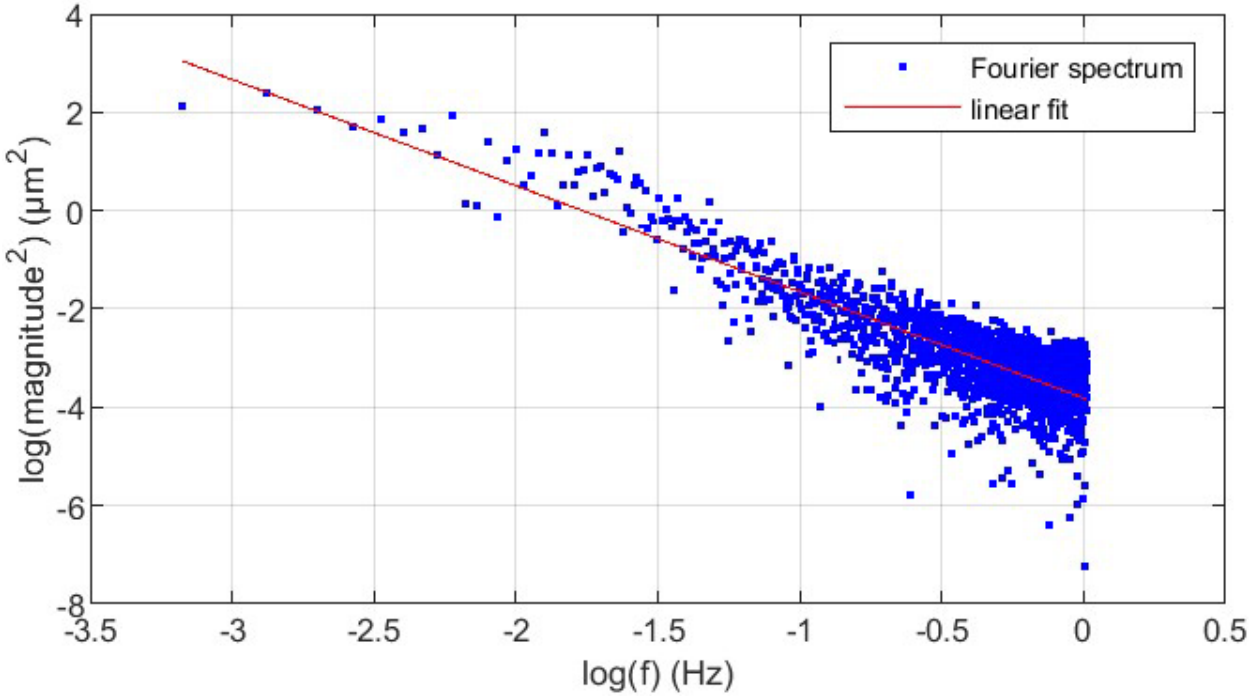

Figure K7: Fourier analysis and linear fit from SWLI line profile data sample 004

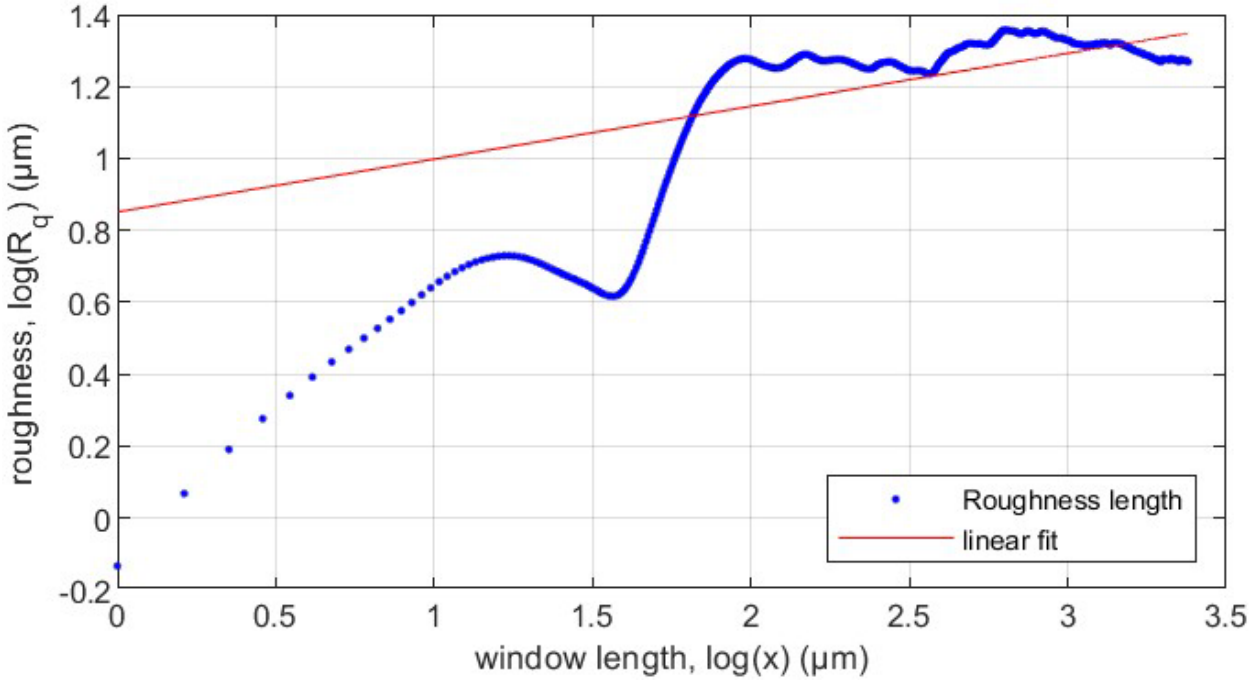

Figure K8: Roughness length method for calculating *H* from Dektak profile sample 004

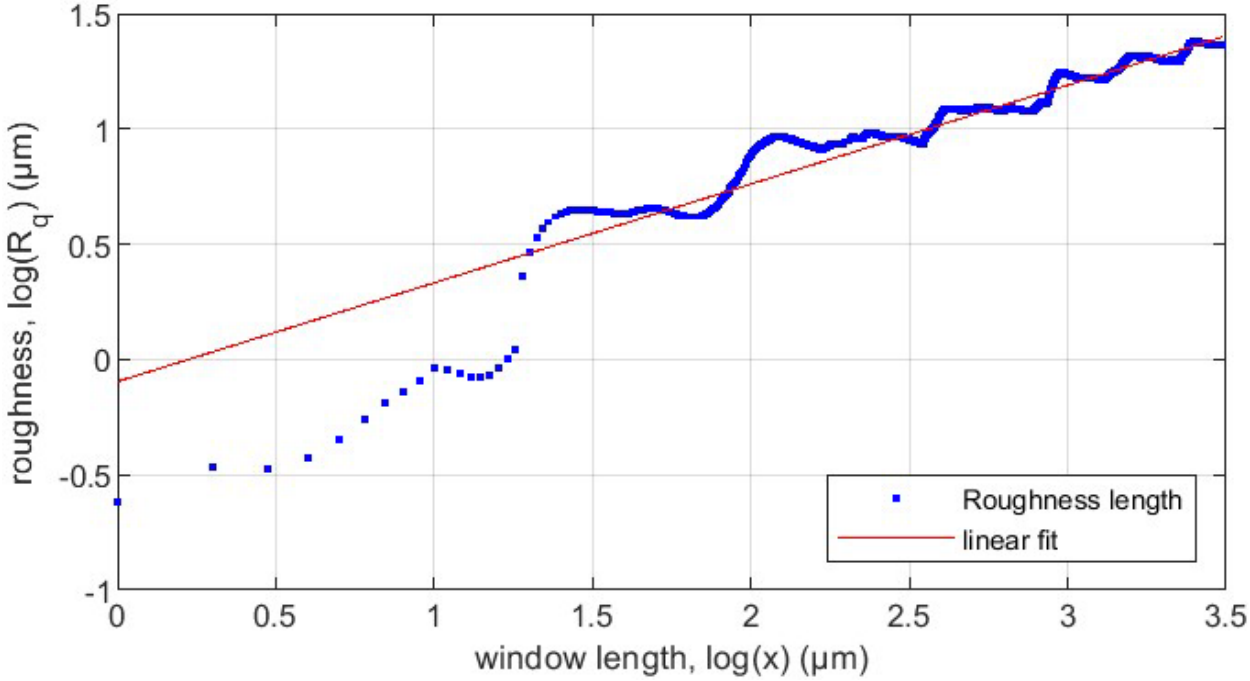

Figure K9: Roughness length method for calculating *H* from SWLI profile sample 004

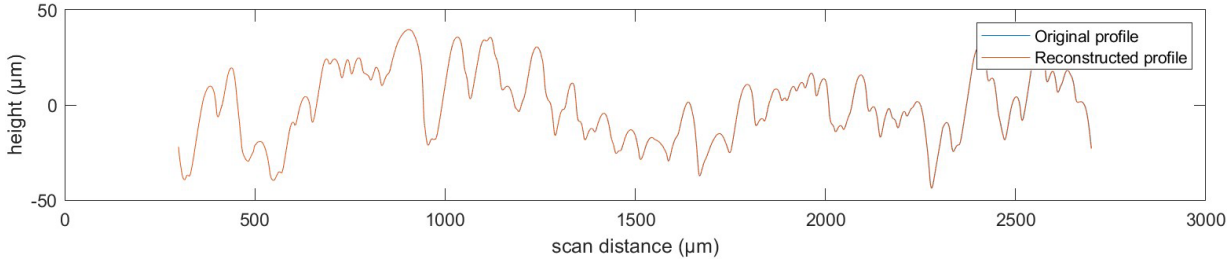

Figure K10: Original Dektak line scan profile and Fourier recreated line profile for sample 004

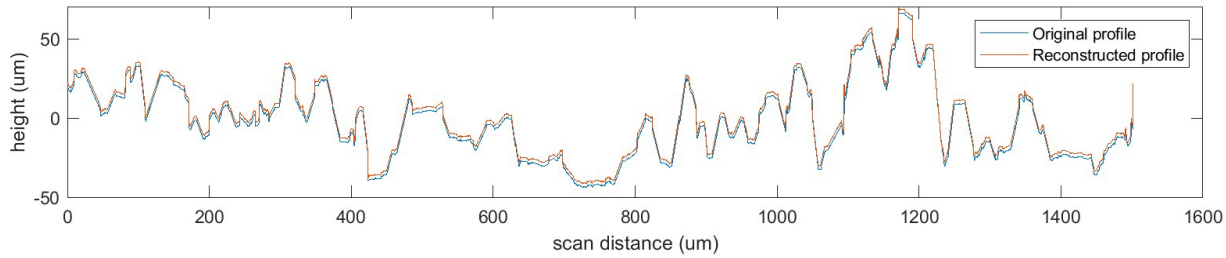

Figure K11: Original SWLI line profile and Fourier recreated line profile for sample 004

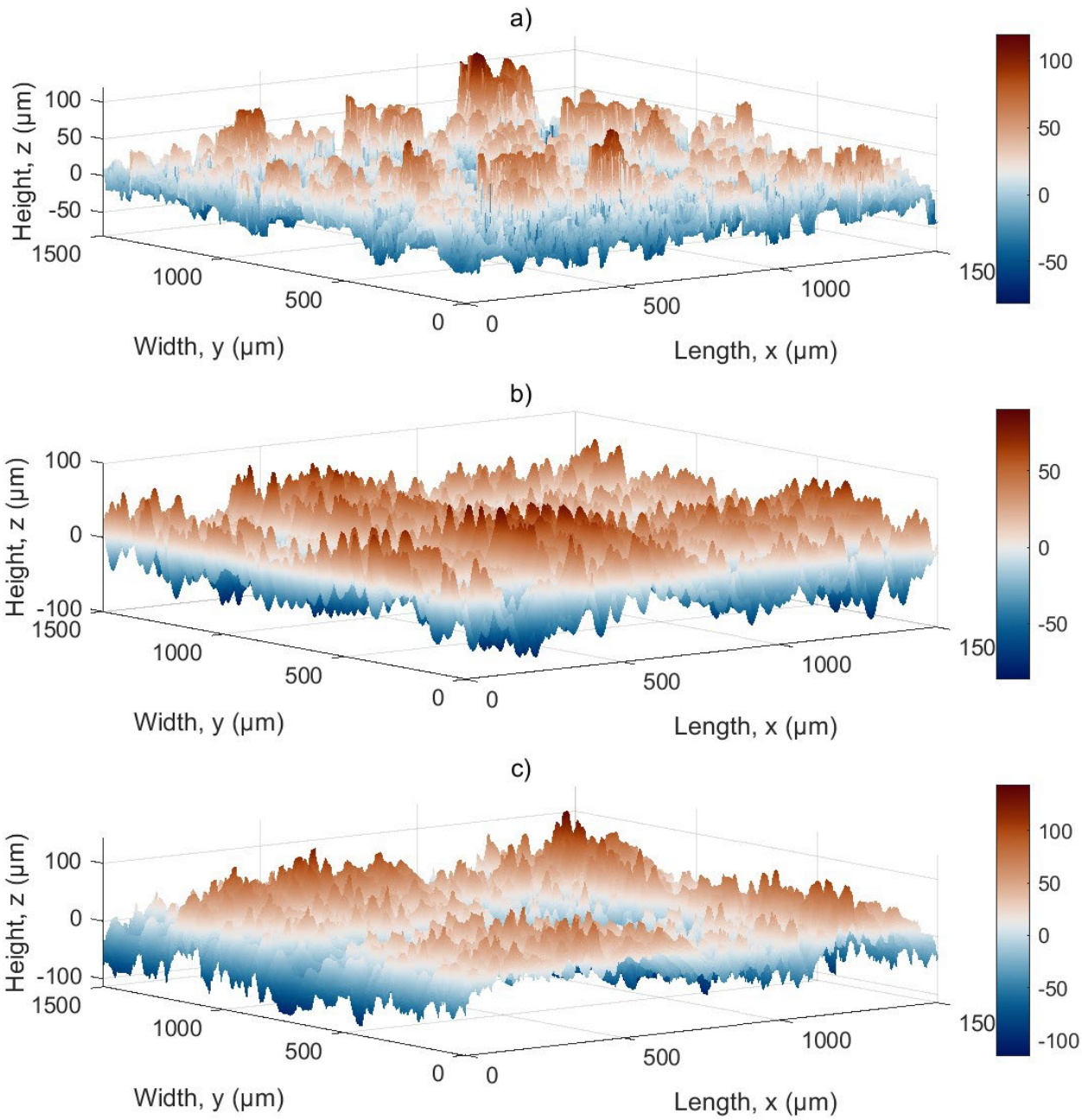

Figure K12: SWLI scan of sample 004 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 004 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 004

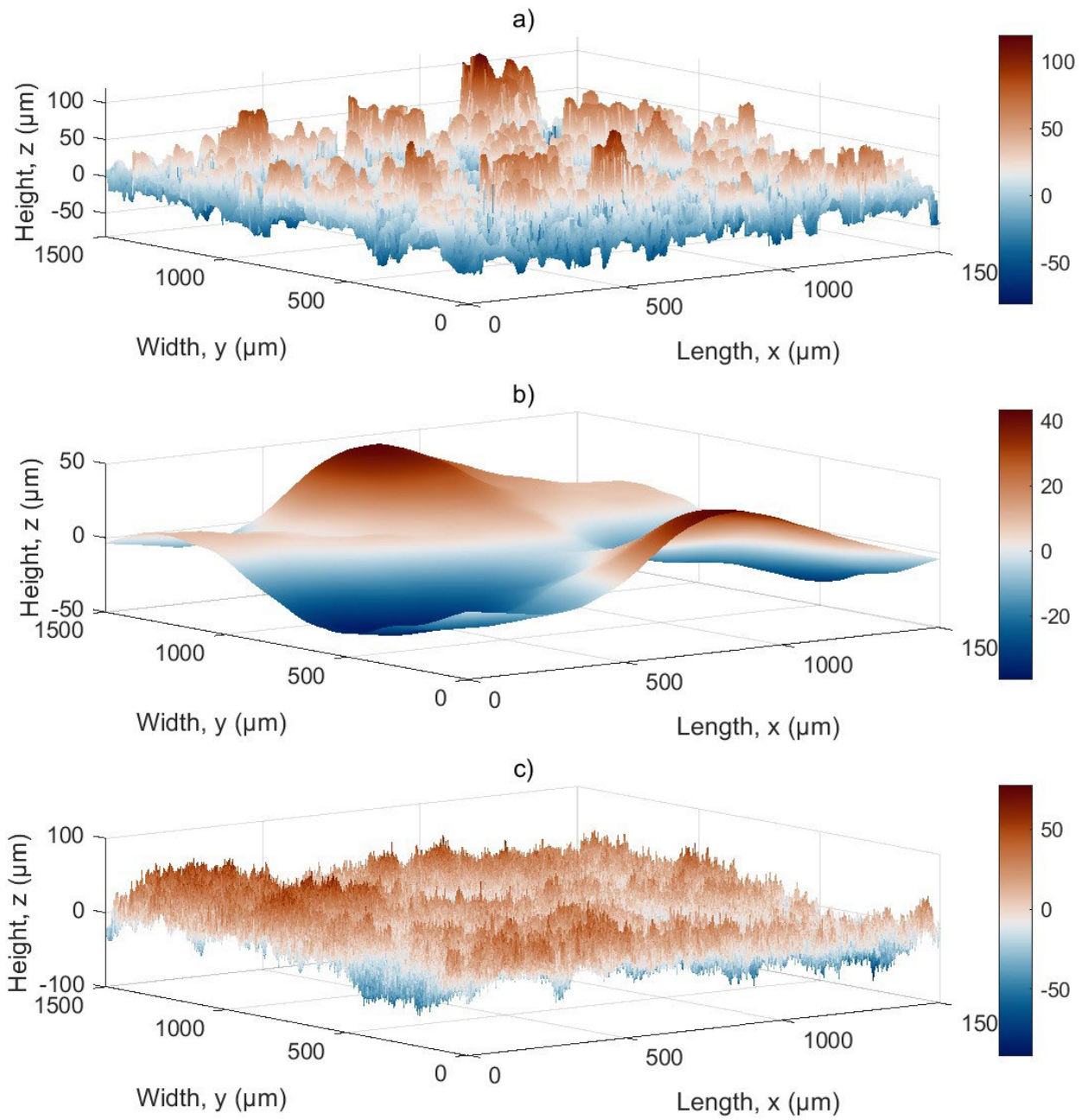

Figure K13: SWLI scan of sample 004 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 004 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 004

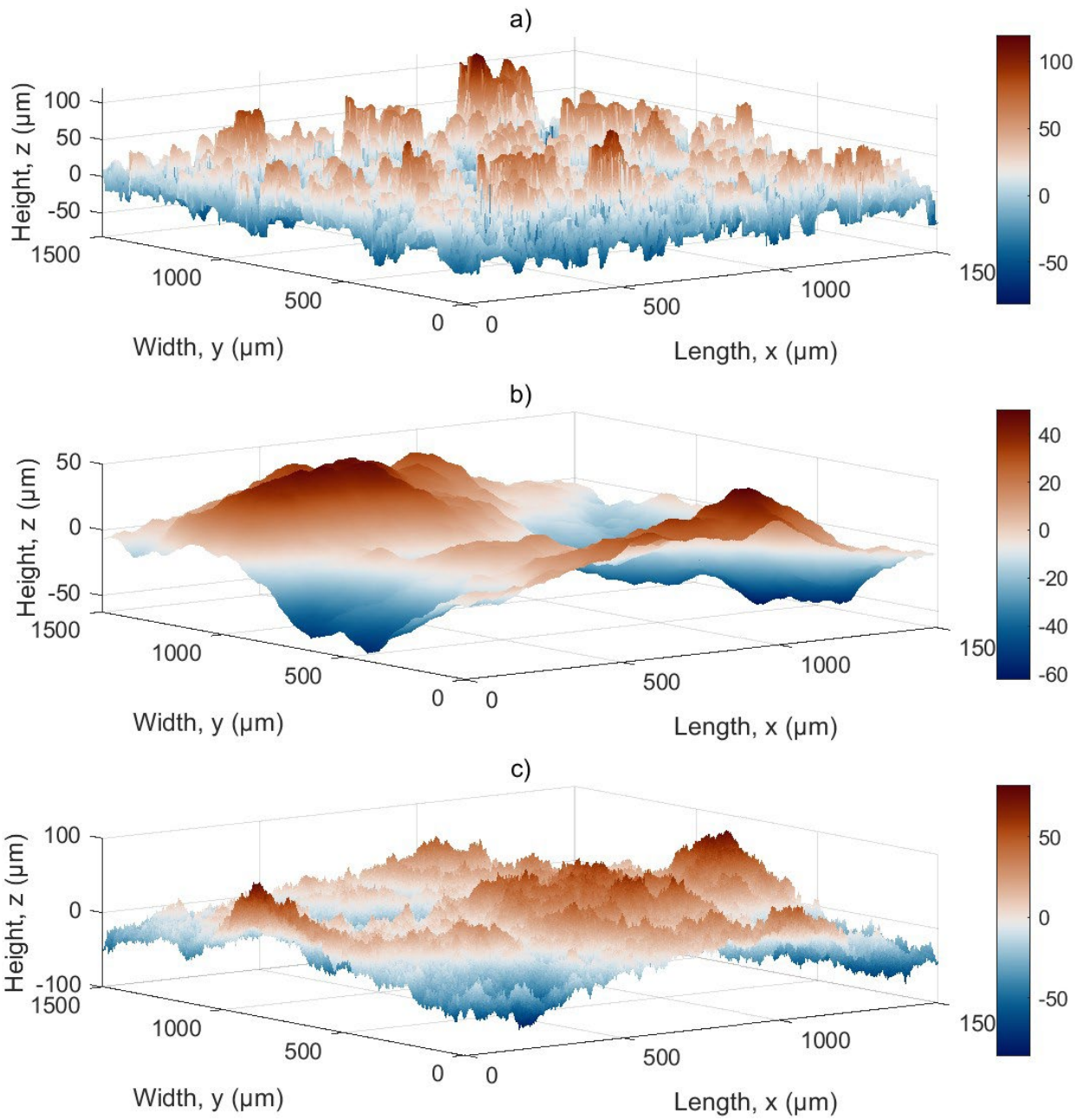

Figure K14: SWLI scan of sample 004 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 004 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 004

Appendix L: Sample 009 supplementary figures

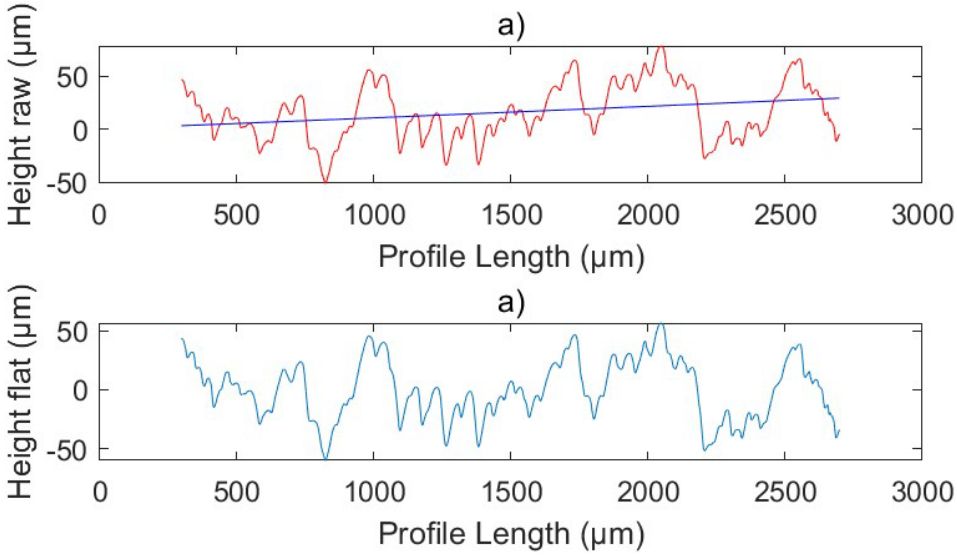

Figure L1: a) Raw profile measurement and fit curve sample 009 b) flattened line profile from sample 009

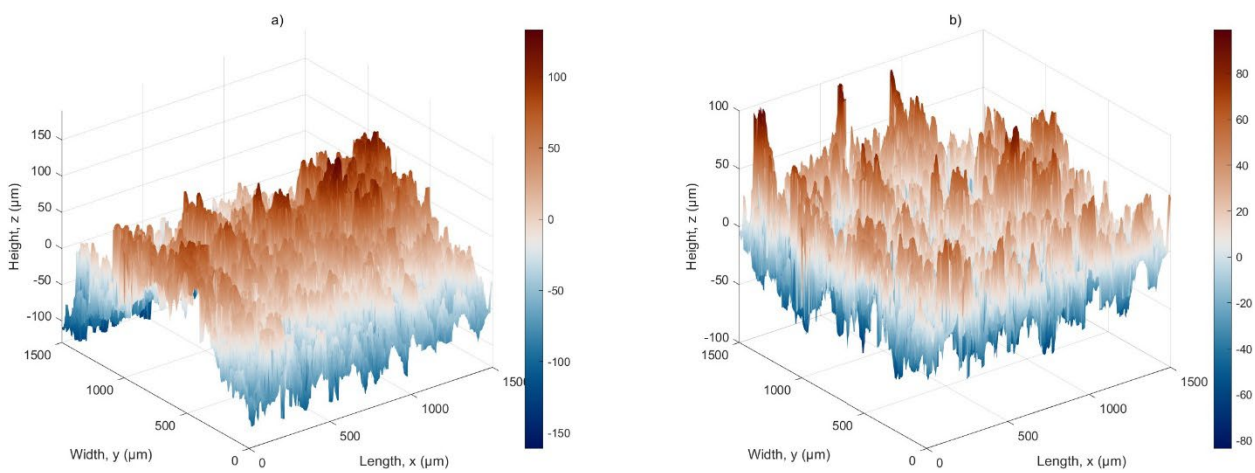

Figure L2: a) Unflattened SWLI raw surface scan sample 009 b) plane flattened SWLI surface 009

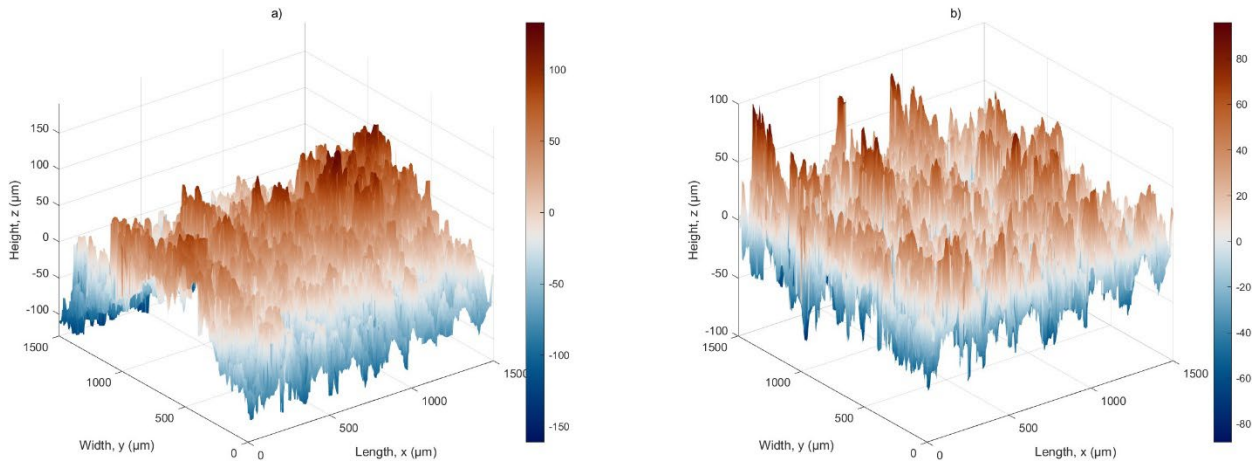

Figure L3: a) Unflattened SWLI raw surface scan sample 009 b) line by line flattened SWLI surface 009

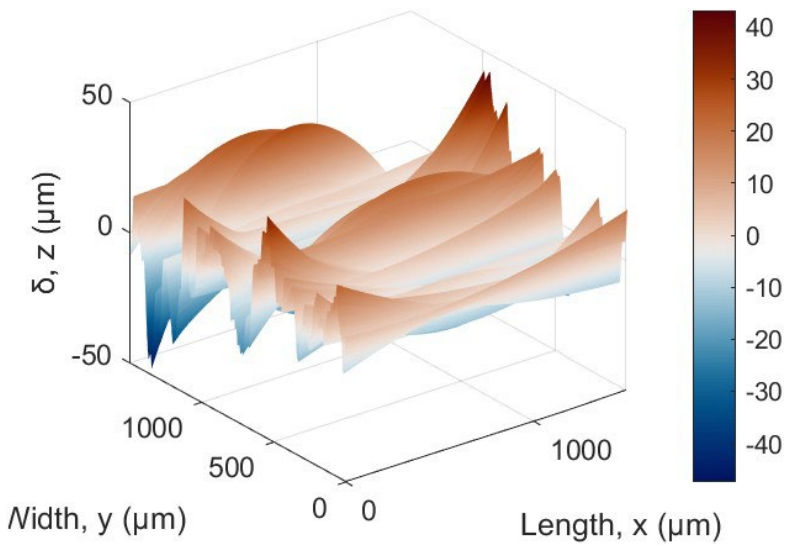

Figure L4: Difference between plane flattening and line by line flattening for sample 009

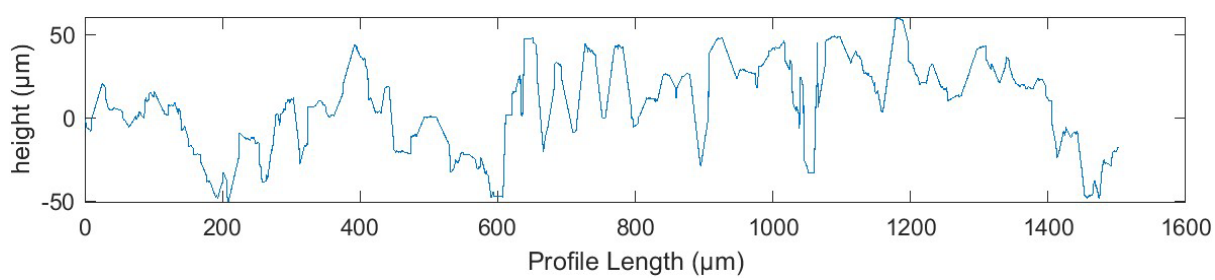

Figure L5: Flattened selected line profile from SWLI dataset sample 009

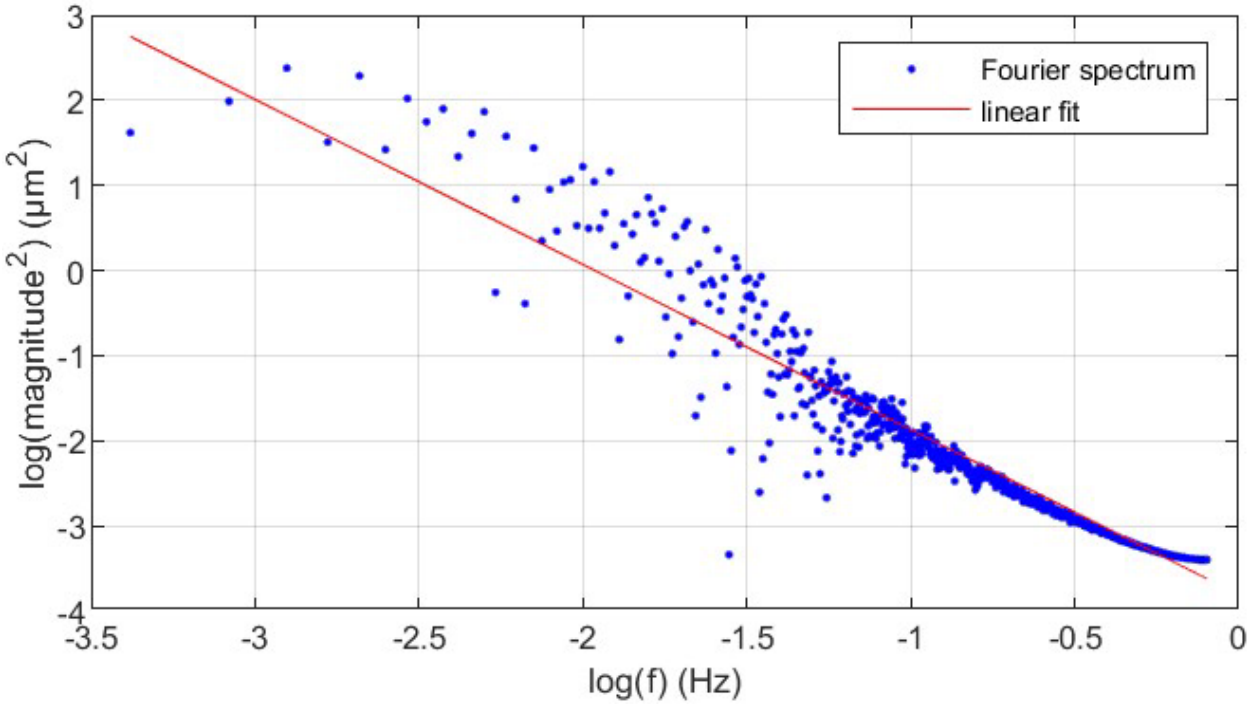

Figure L6: Fourier analysis and linear fit from Dektak line profile data sample 009

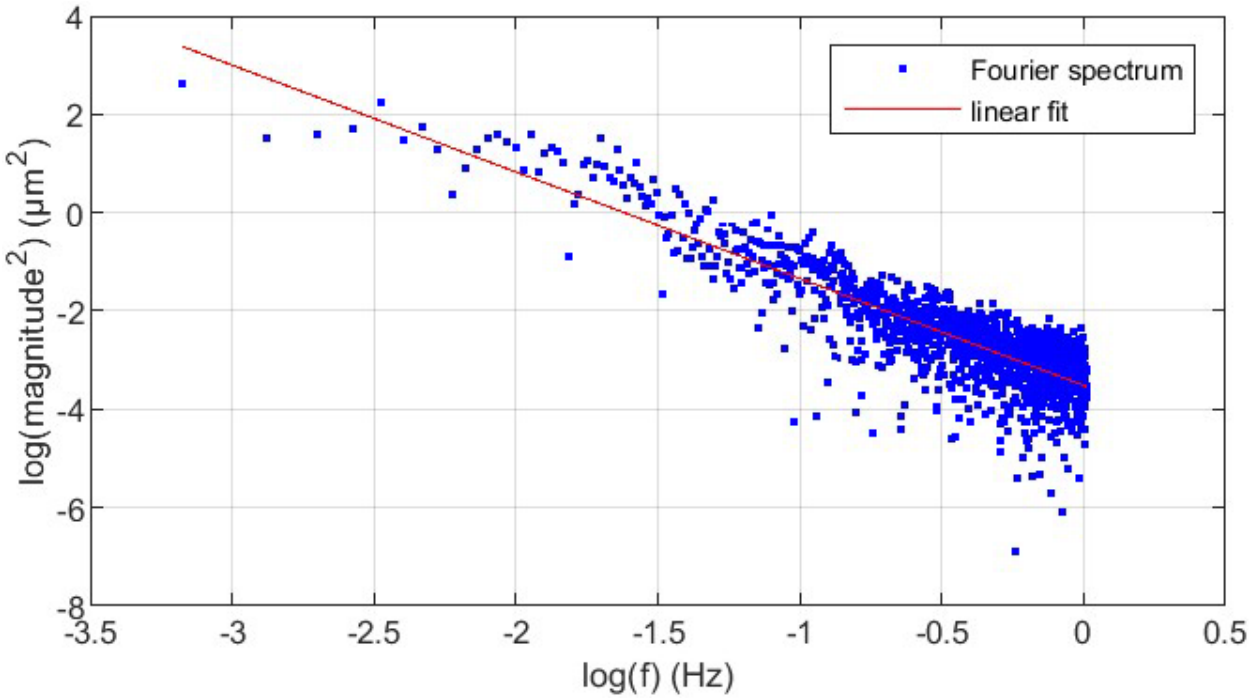

Figure L7: Fourier analysis and linear fit from SWLI line profile data sample 009

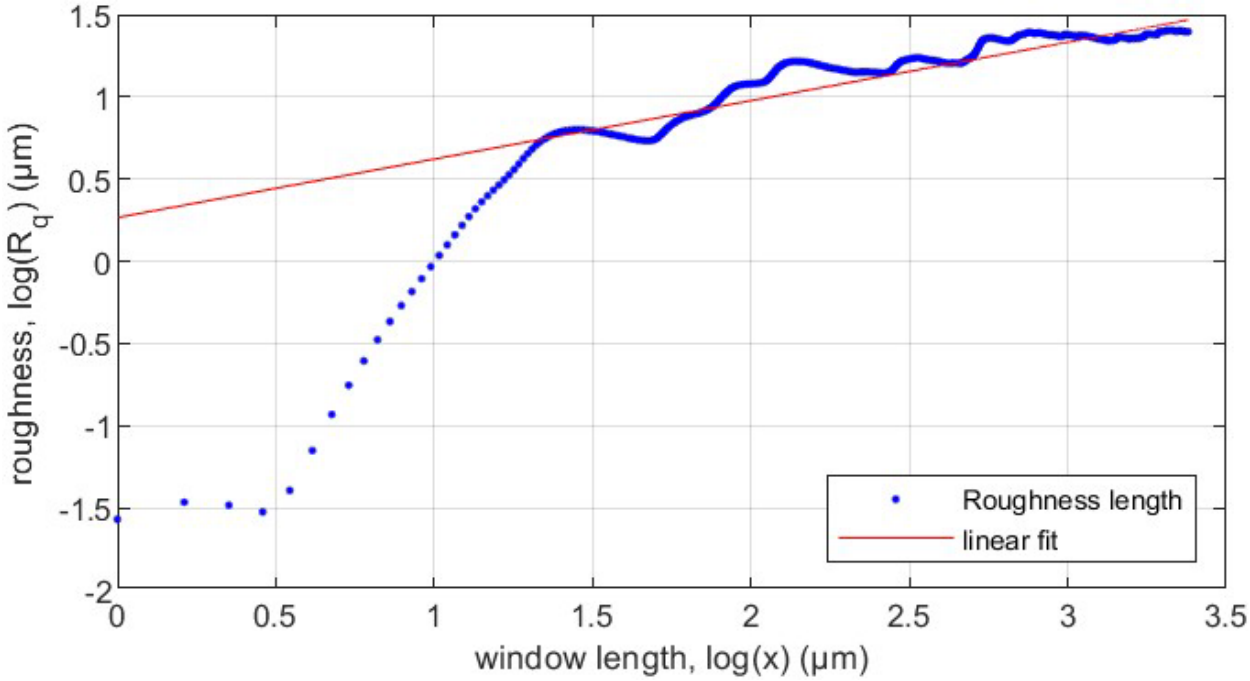

Figure L8: Roughness length method for calculating *H* from Dektak profile sample 009

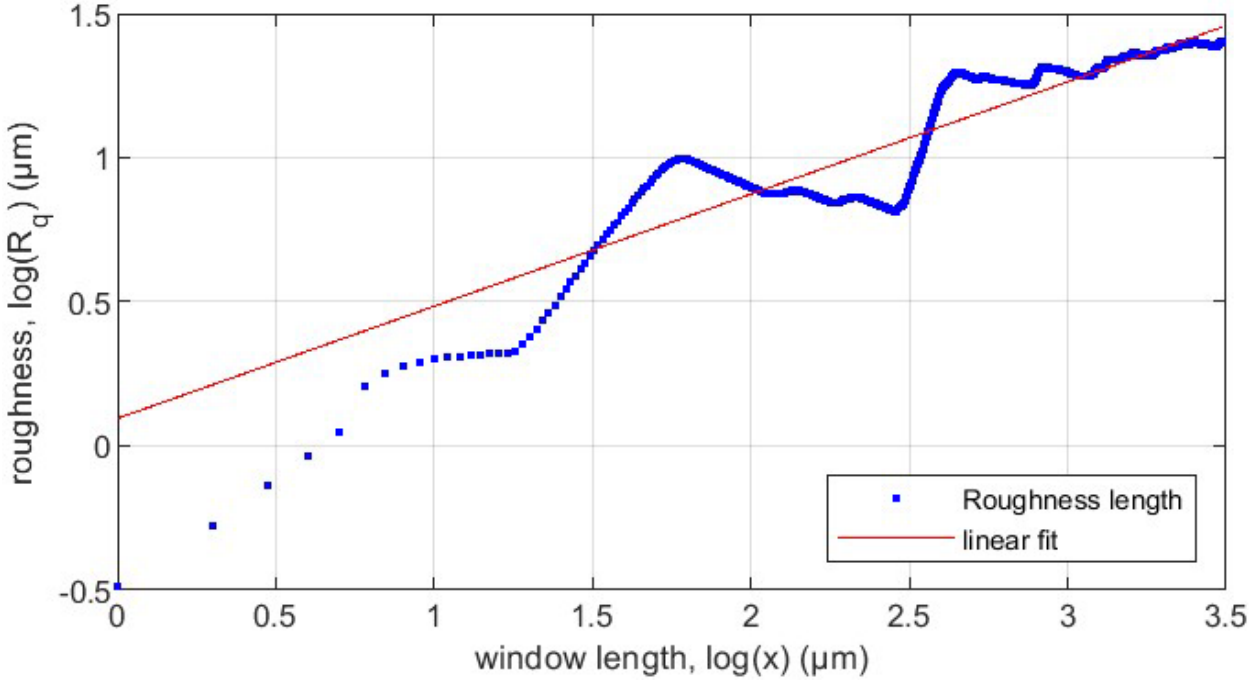

Figure L9: Roughness length method for calculating *H* from SWLI profile sample 009

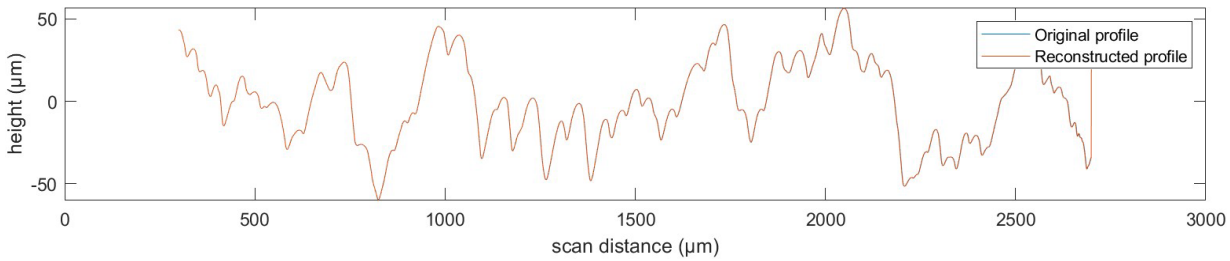

Figure L10: Original Dektak line scan profile and Fourier recreated line profile for sample 009

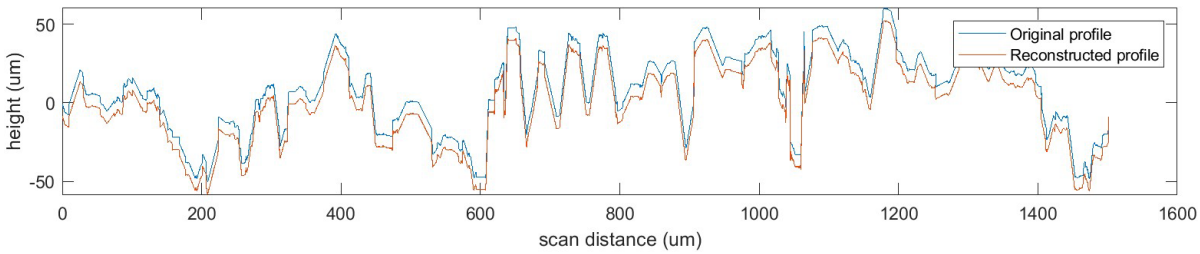

Figure L11: Original SWLI line profile and Fourier recreated line profile for sample 009

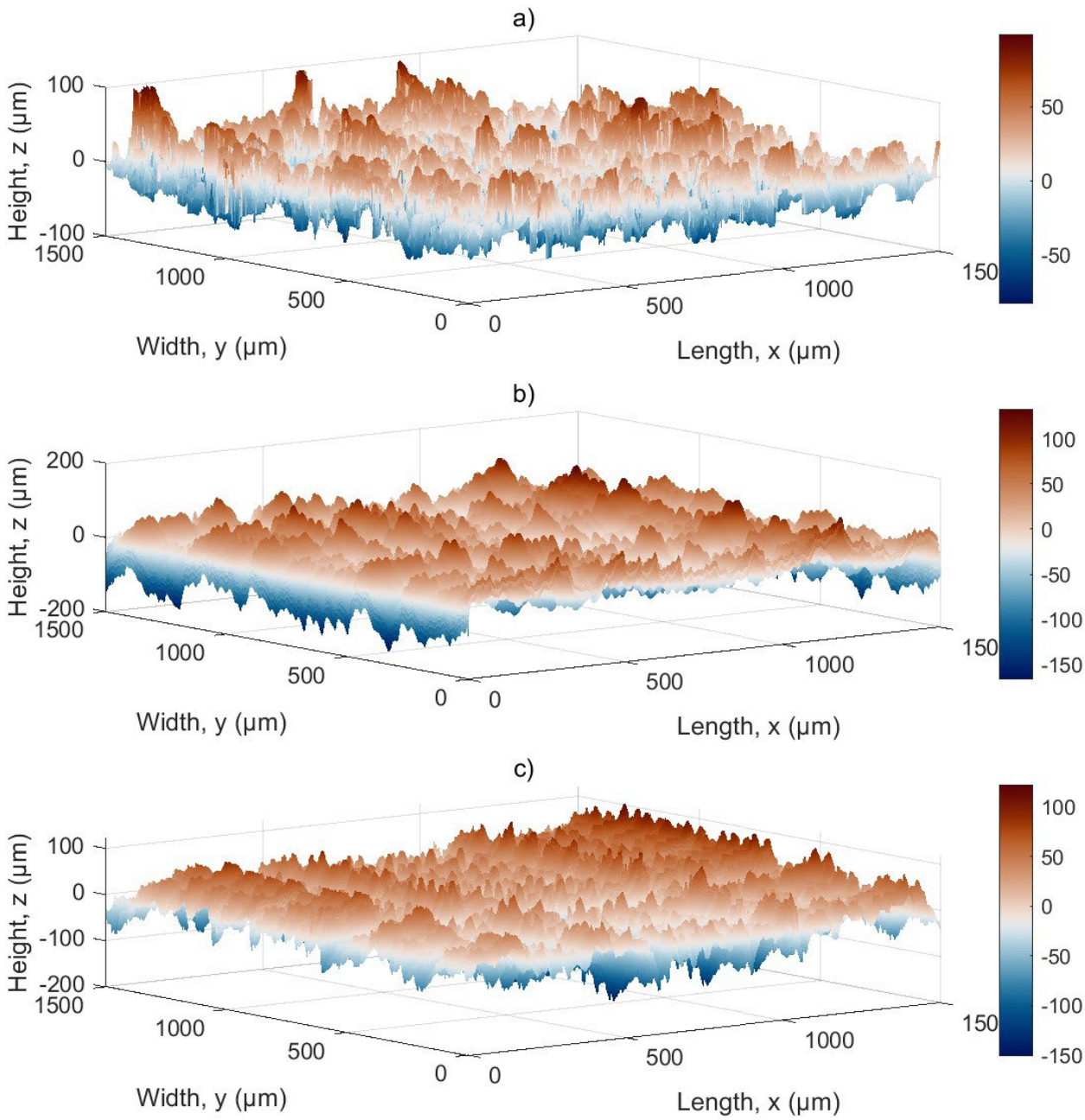

Figure L12: SWLI scan of sample 009 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 009 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 009

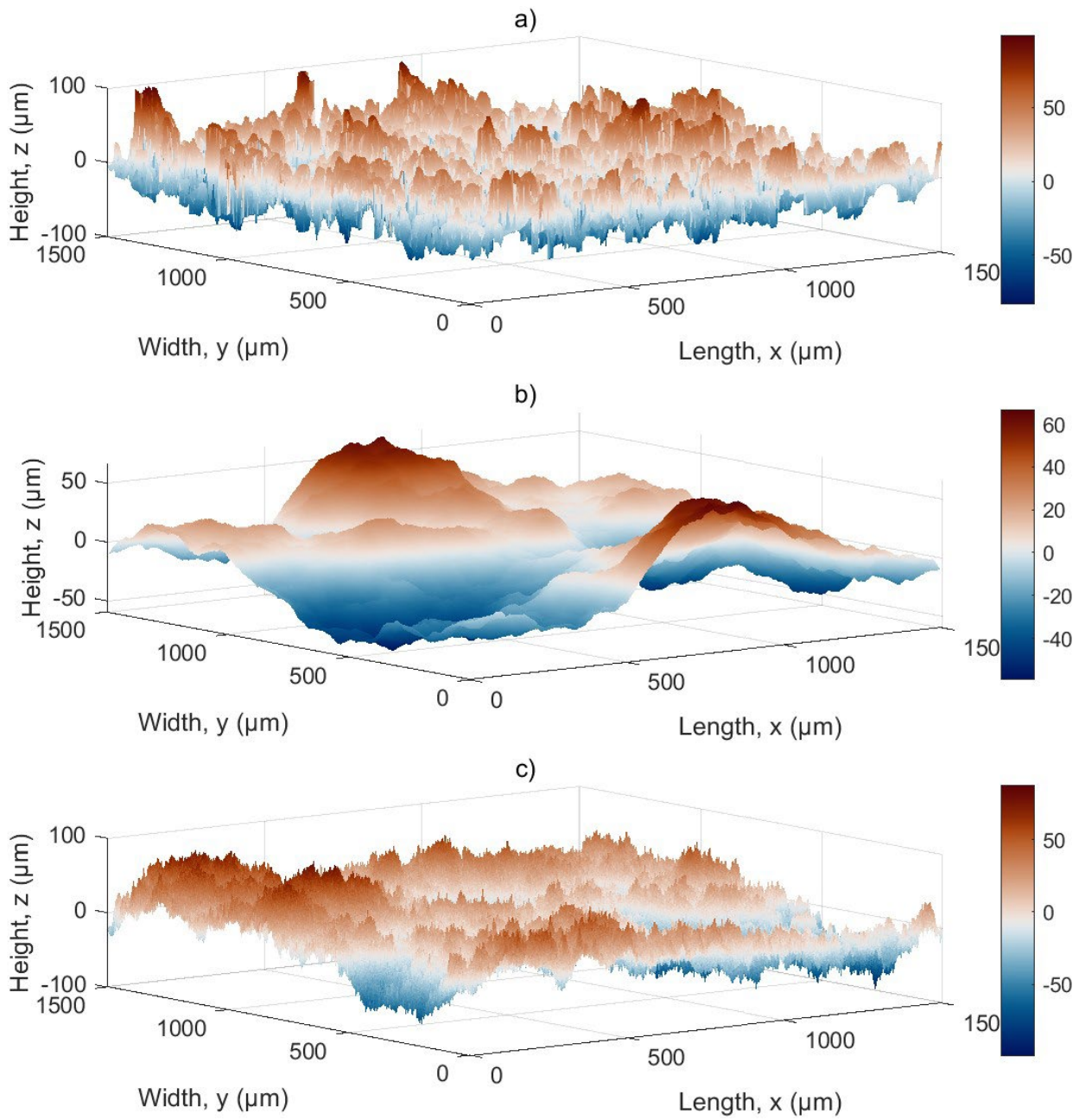

Figure L13: SWLI scan of sample 009 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 009 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 009

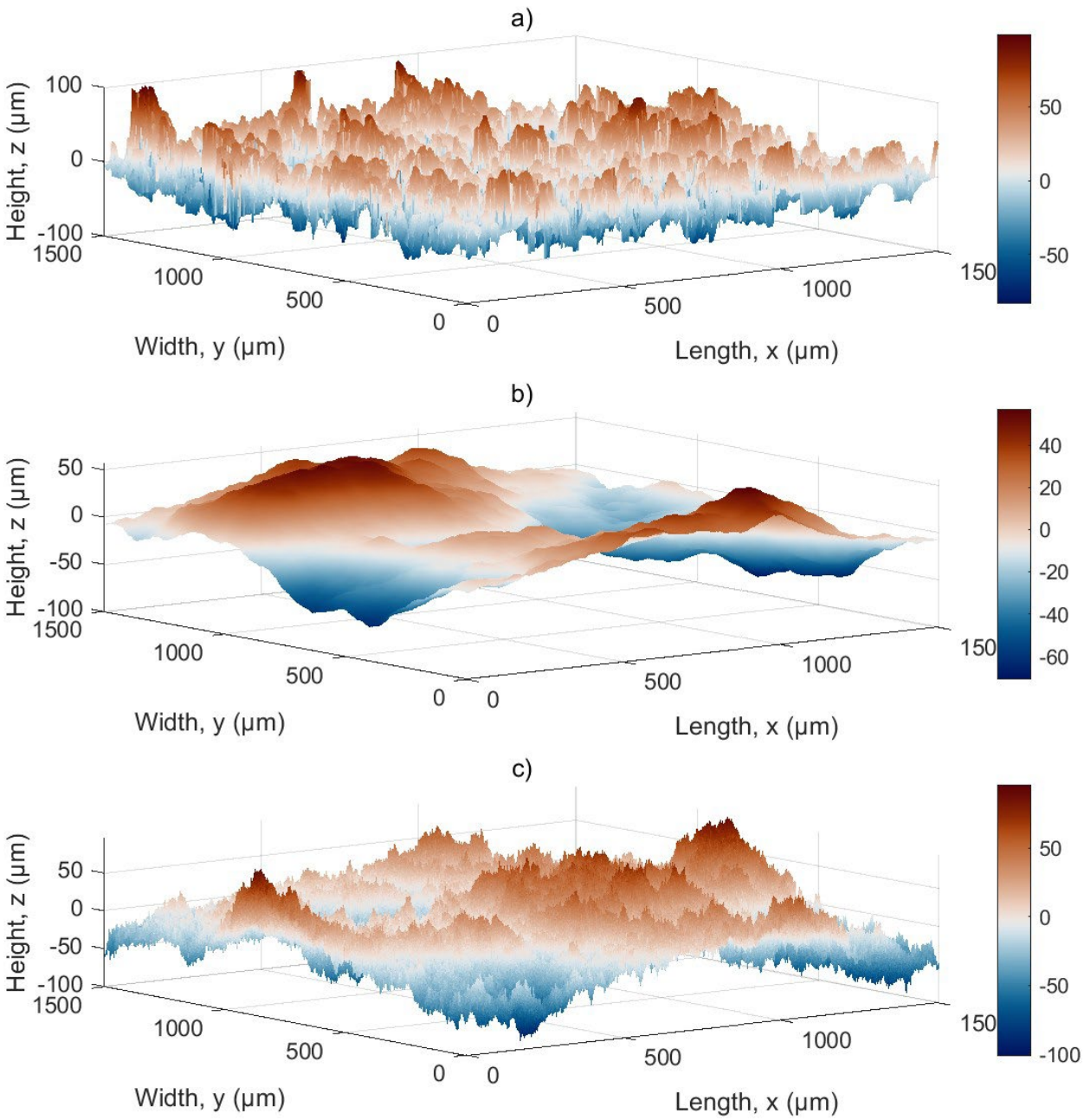

Figure L14: SWLI scan of sample 009 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 009 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 009

Appendix M: Sample 010 supplementary figures

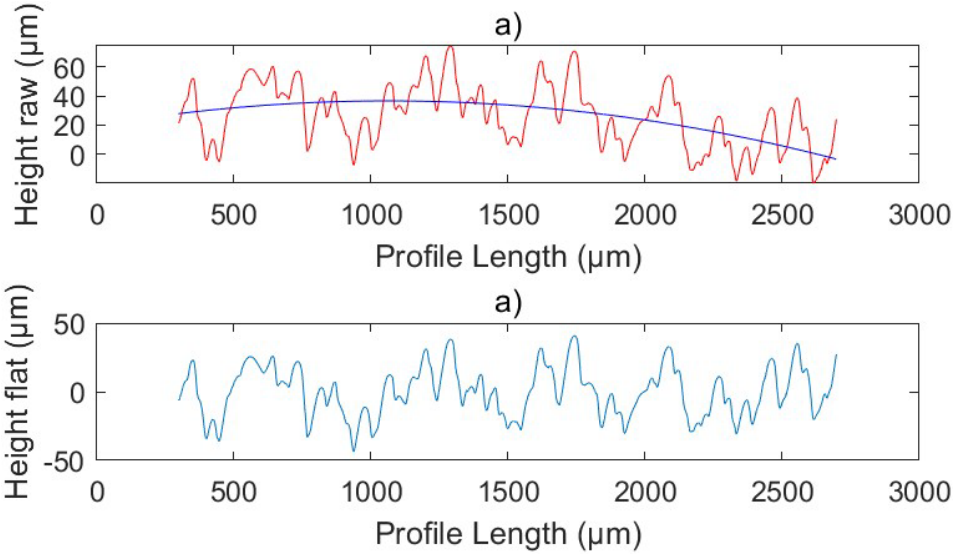

Figure M1: a) Raw profile measurement and fit curve sample 010 b) flattened line profile from sample 010

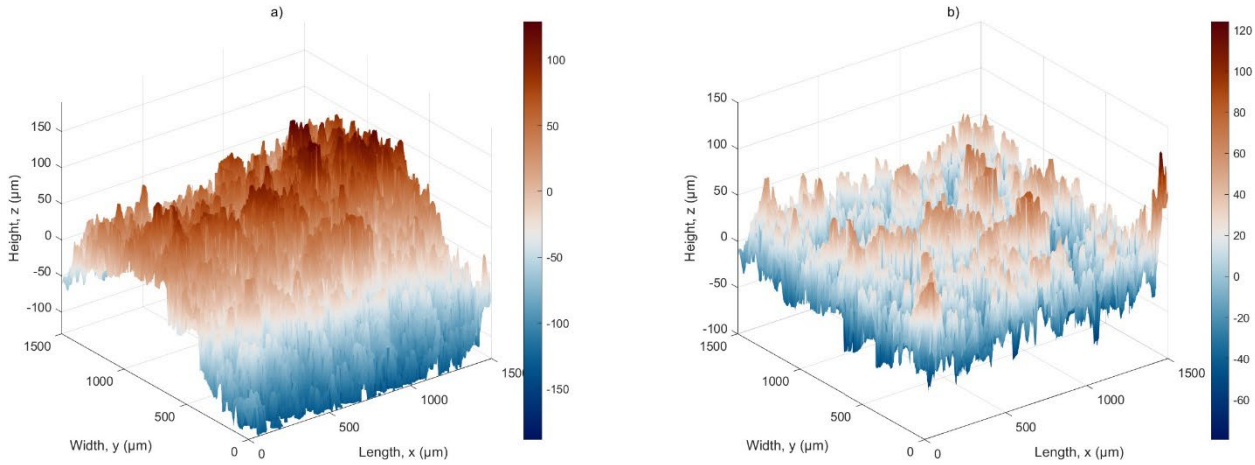

Figure M2: a) Unflattened SWLI raw surface scan sample 010 b) plane flattened SWLI surface 010

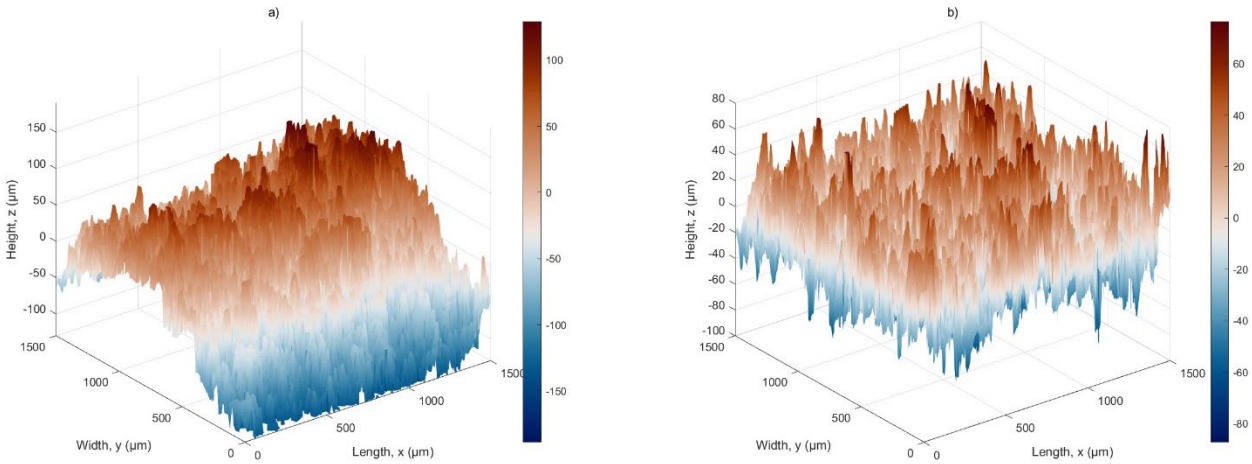

Figure M3: a) Unflattened SWLI raw surface scan sample 010 b) line by line flattened SWLI surface 010

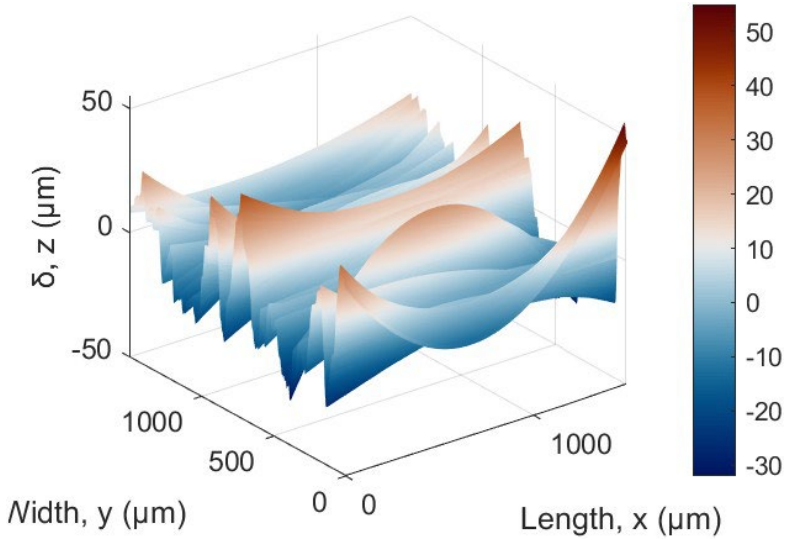

Figure M4: Difference between plane flattening and line by line flattening for sample 010

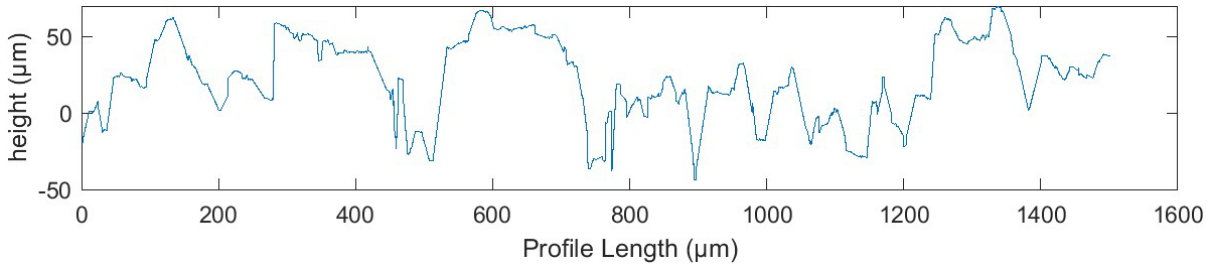

Figure M5: Flattened selected line profile from SWLI dataset sample 010

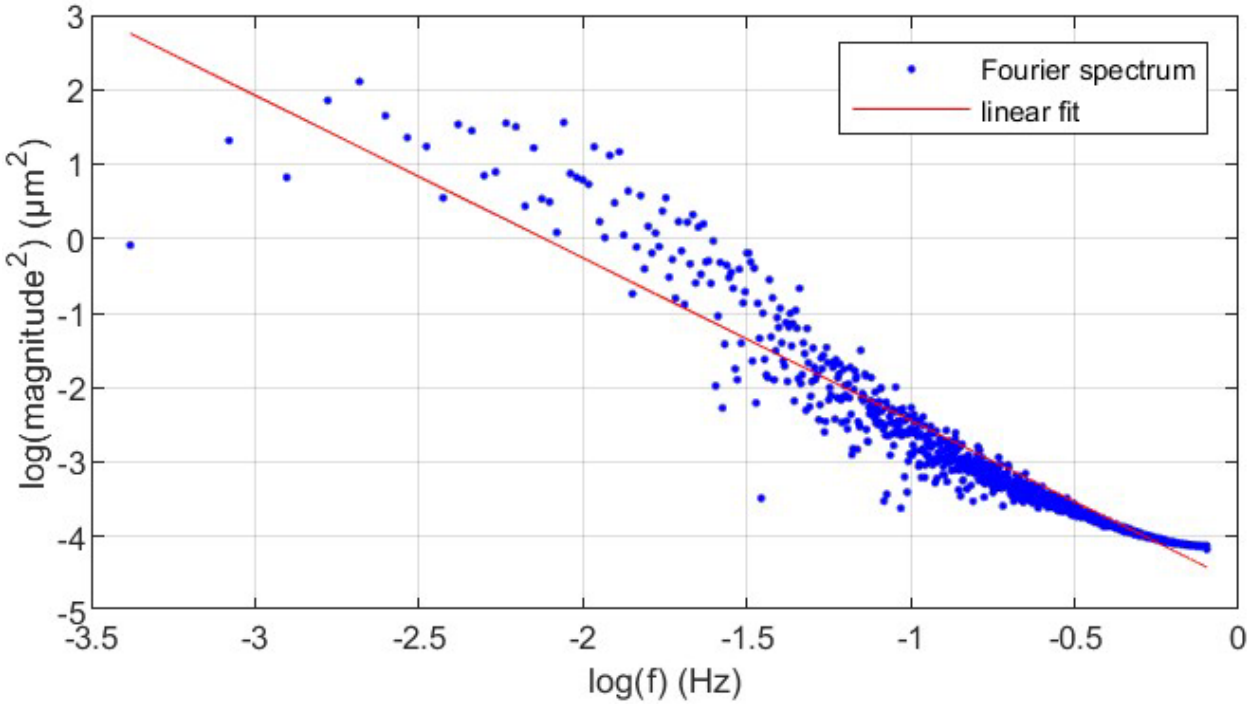

Figure M6: Fourier analysis and linear fit from Dektak line profile data sample 010

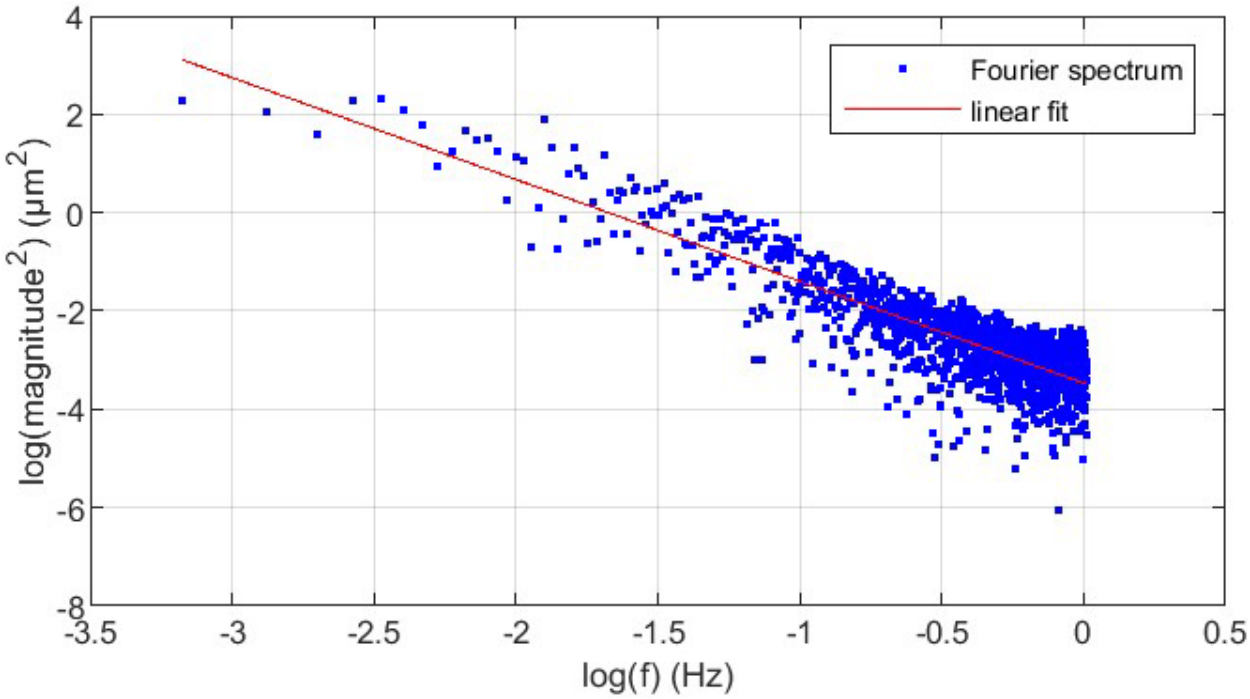

Figure M7: Fourier analysis and linear fit from SWLI line profile data sample 010

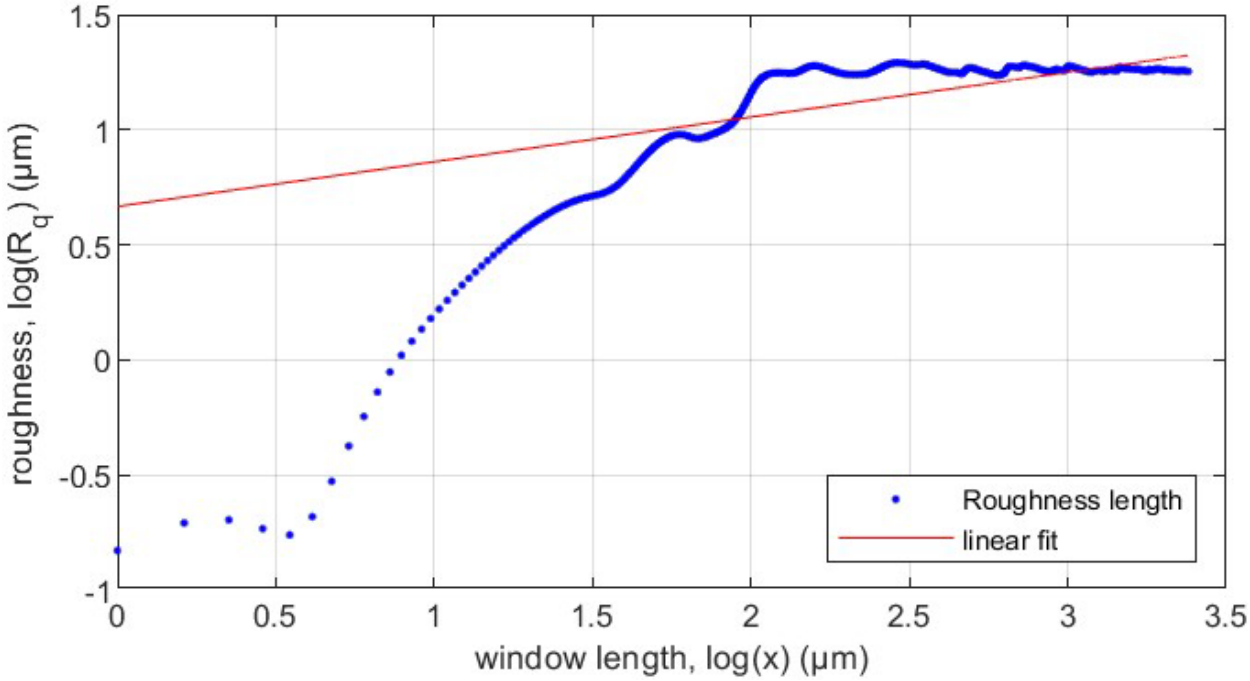

Figure M8: Roughness length method for calculating *H* from Dektak profile sample 010

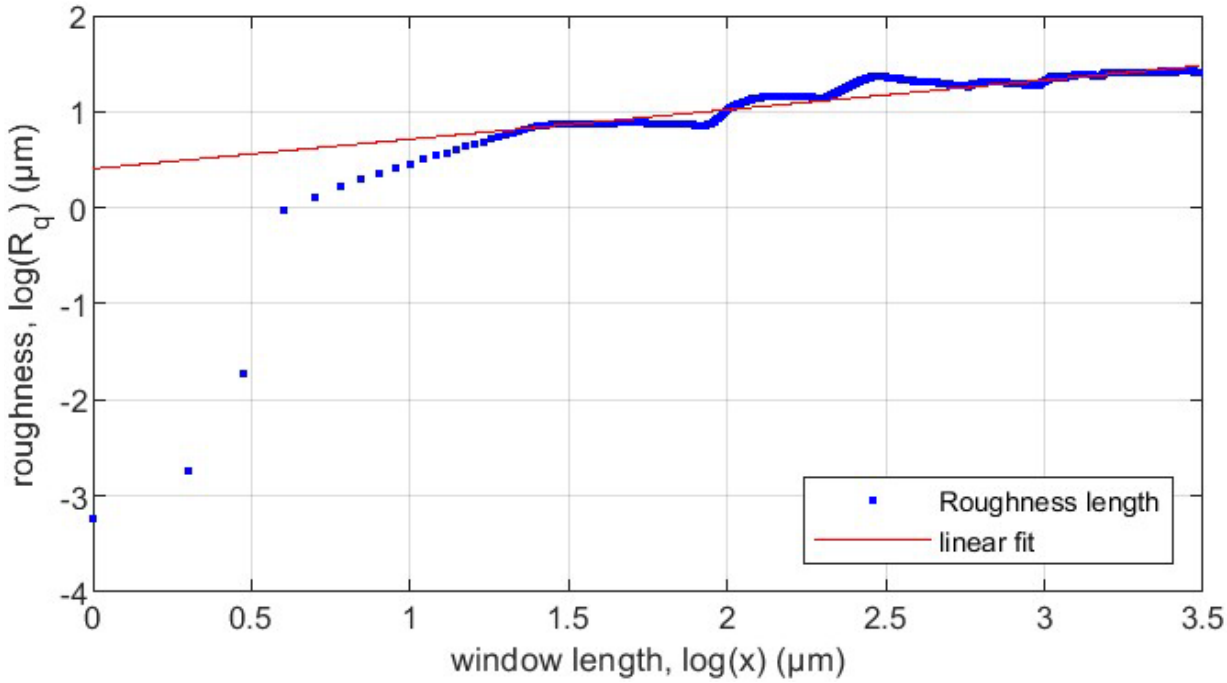

Figure M9: Roughness length method for calculating *H* from SWLI profile sample 010

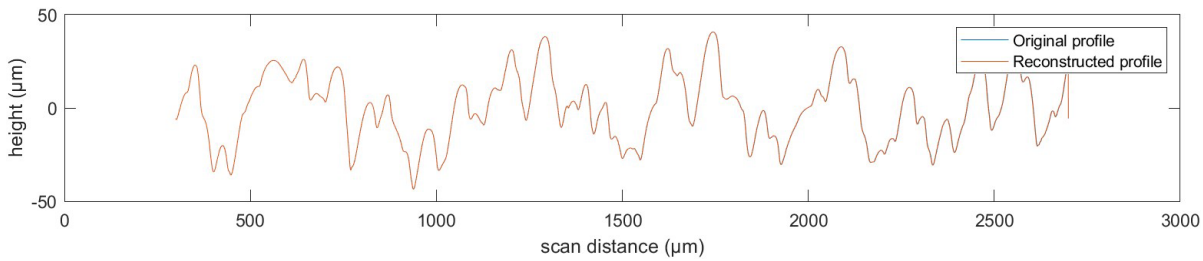

Figure M10: Original Dektak line scan profile and Fourier recreated line profile for sample 010

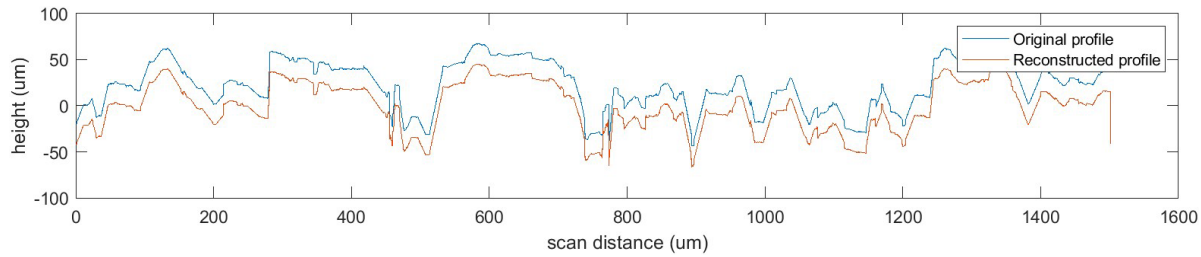

Figure M11: Original SWLI line profile and Fourier recreated line profile for sample 010

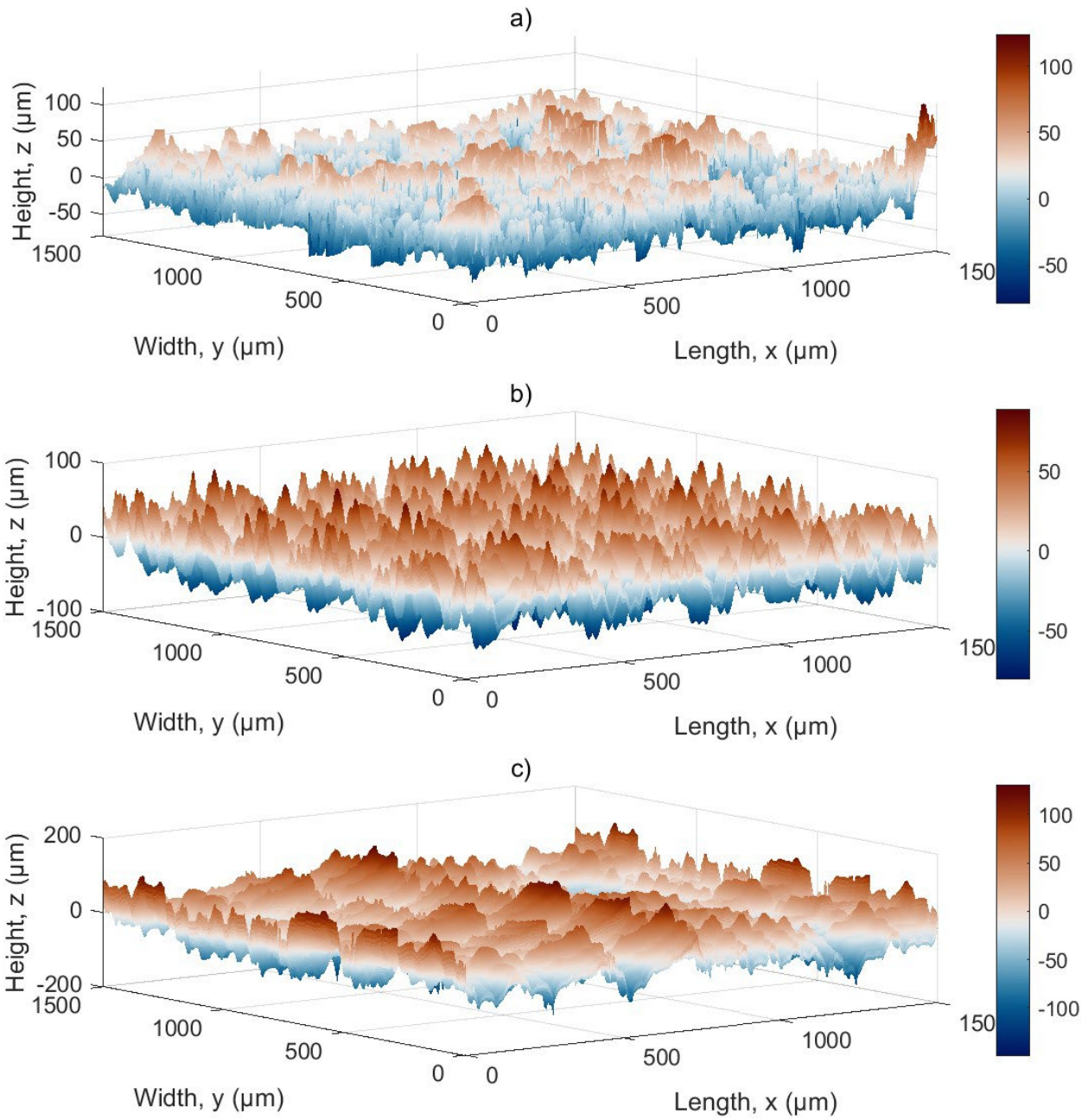

Figure M12: SWLI scan of sample 010 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 010 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 010

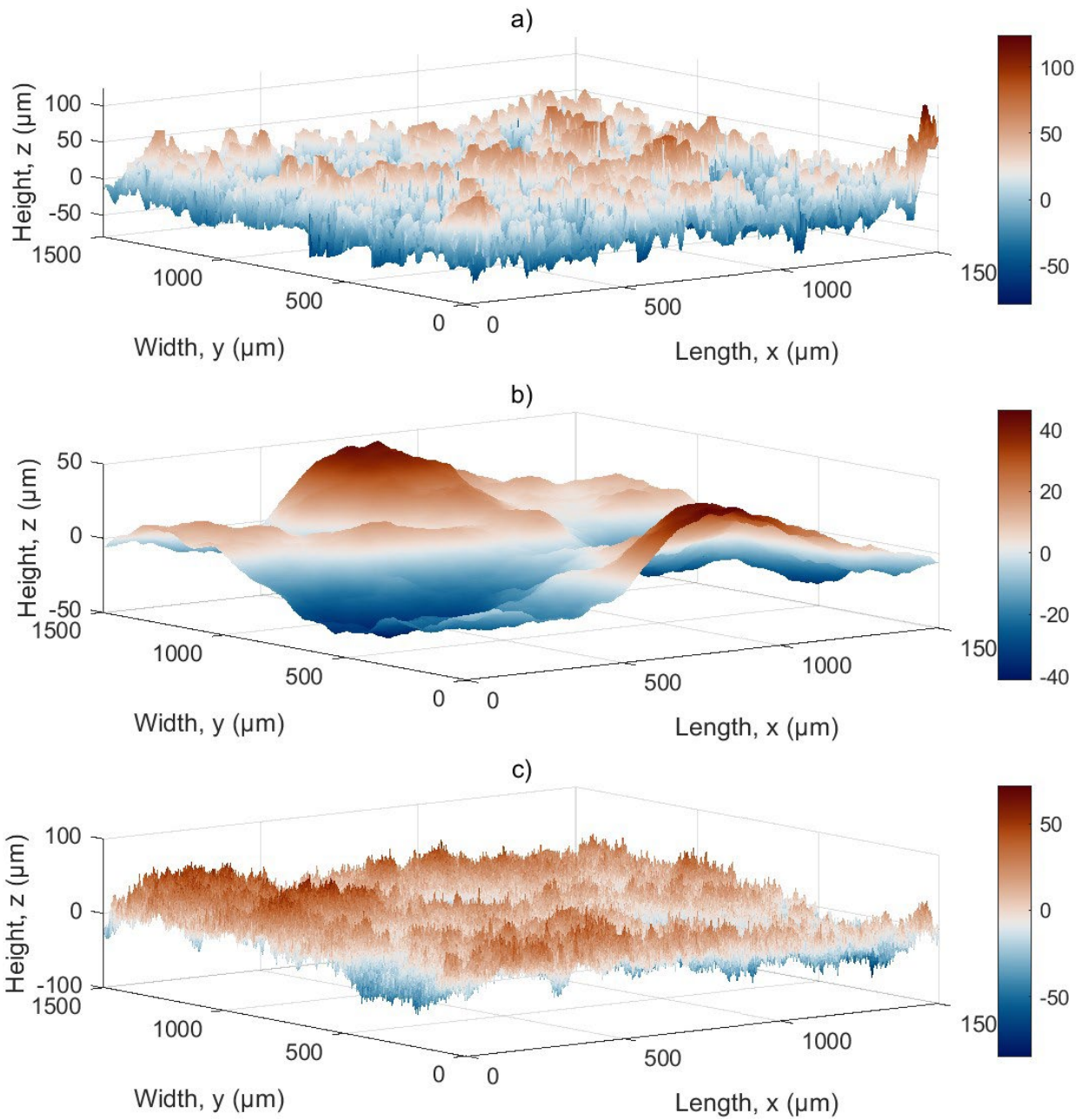

Figure M13: SWLI scan of sample 010 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 010 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 010

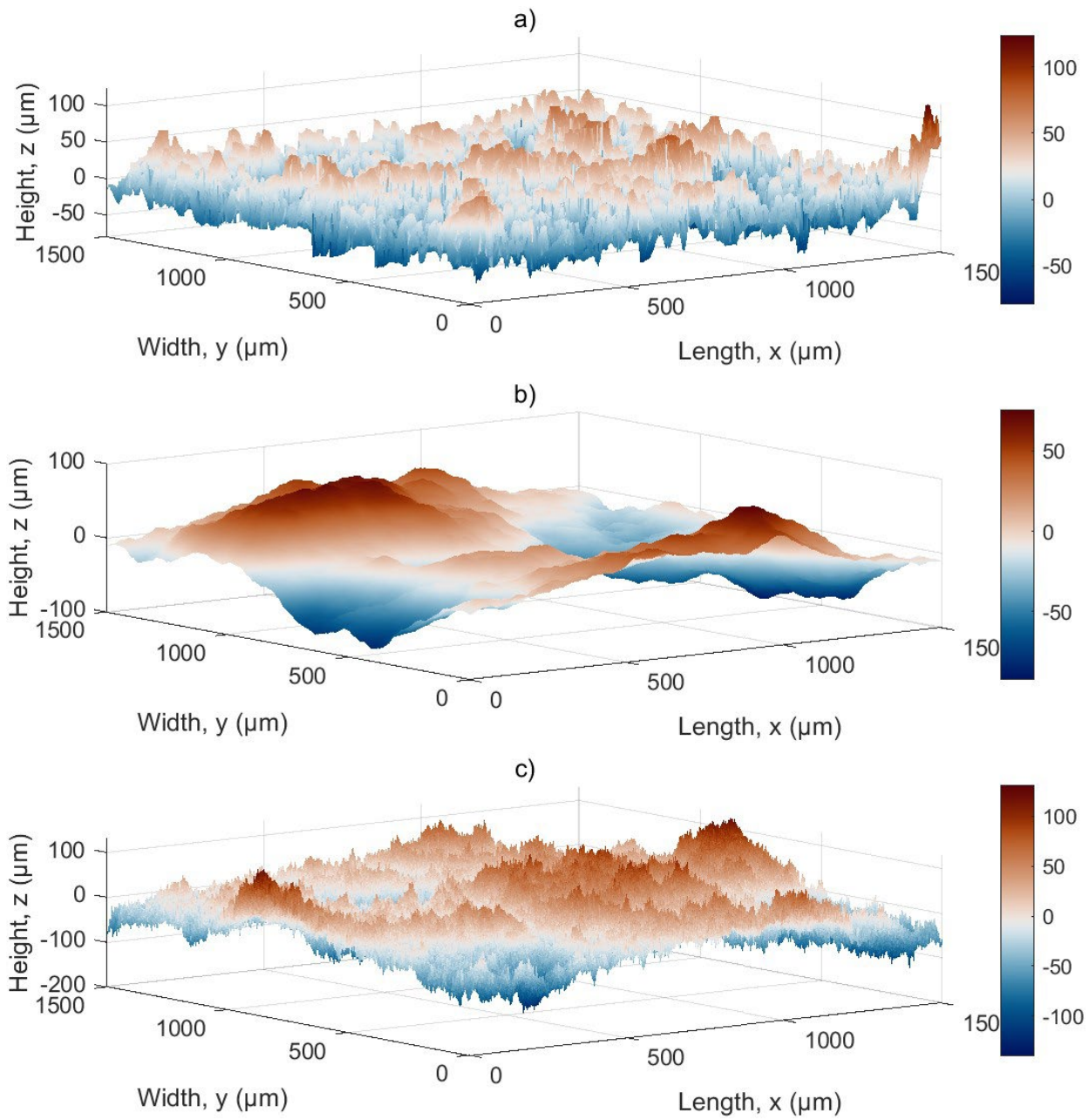

Figure M14: SWLI scan of sample 010 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 010 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 010

Appendix N: Sample 016 supplementary figures

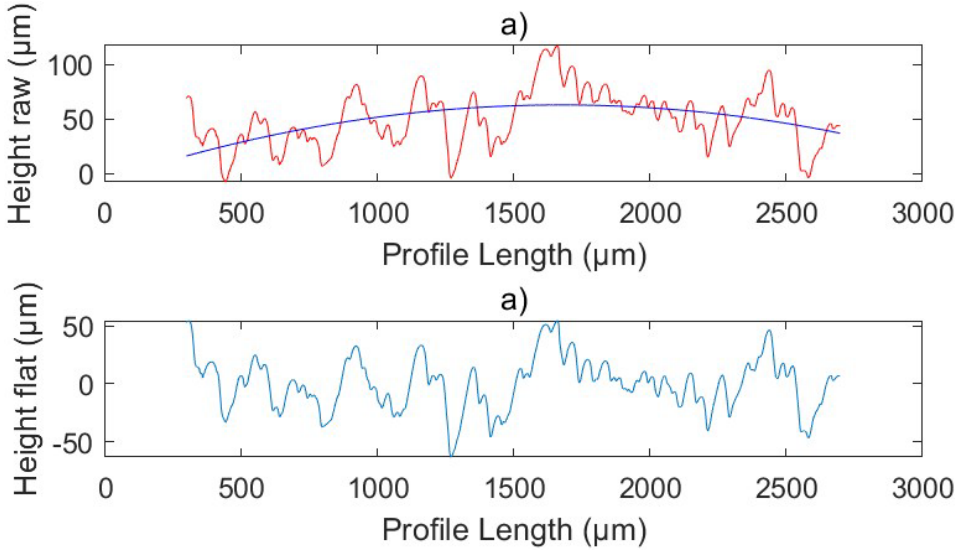

Figure N1: a) Raw profile measurement and fit curve sample 016 b) flattened line profile from sample 016

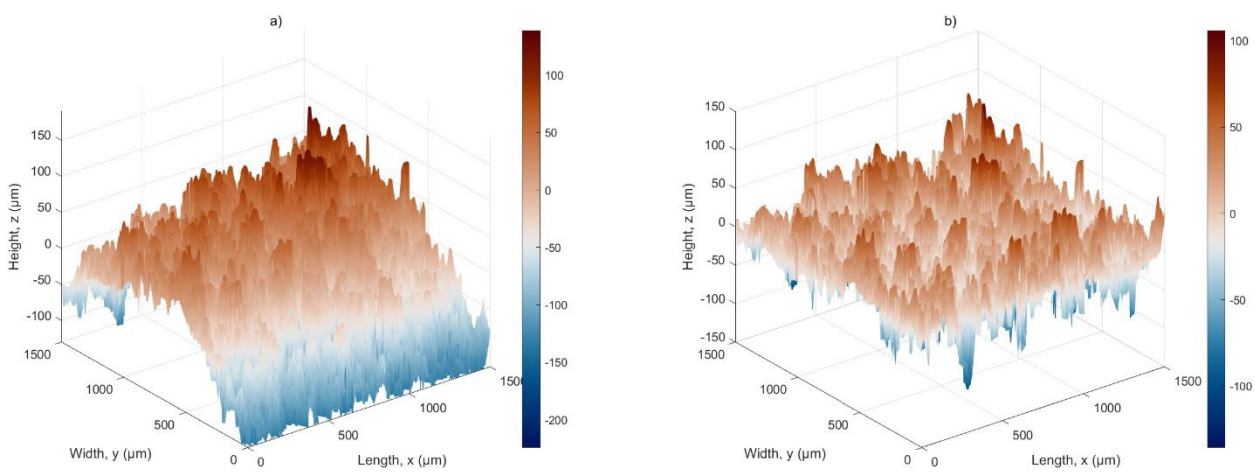

Figure N2: a) Unflattened SWLI raw surface scan sample 016 b) plane flattened SWLI surface 016

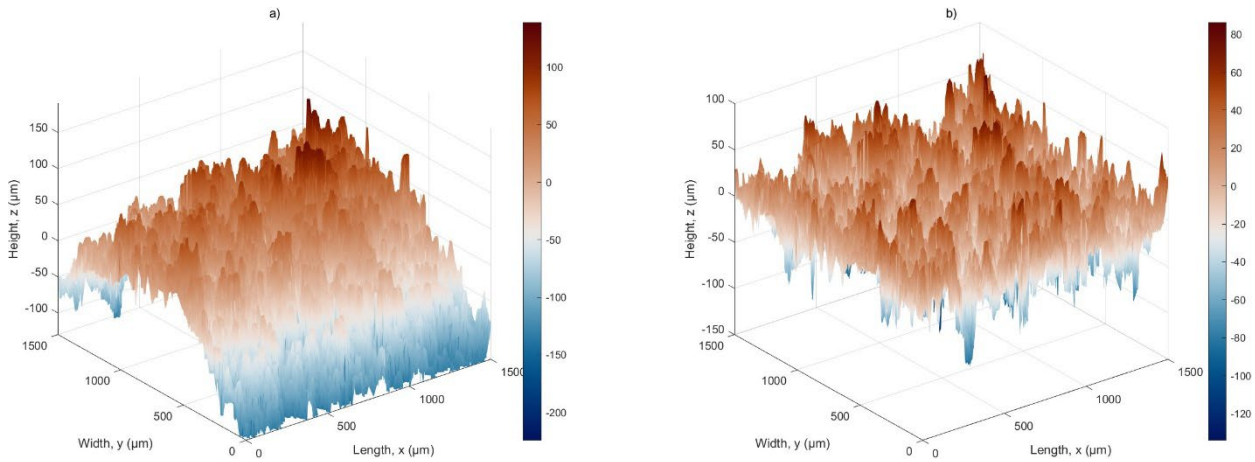

Figure N3: a) Unflattened SWLI raw surface scan sample 016 b) line by line flattened SWLI surface 016

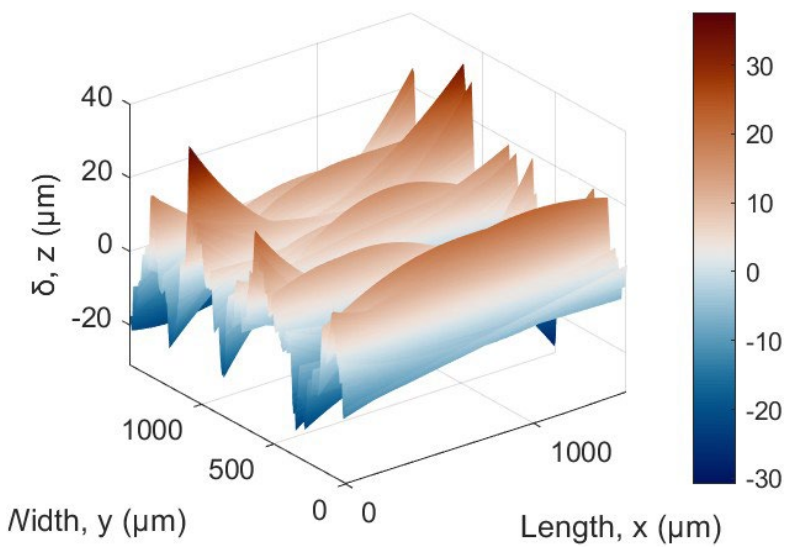

Figure N4: Difference between plane flattening and line by line flattening for sample 016

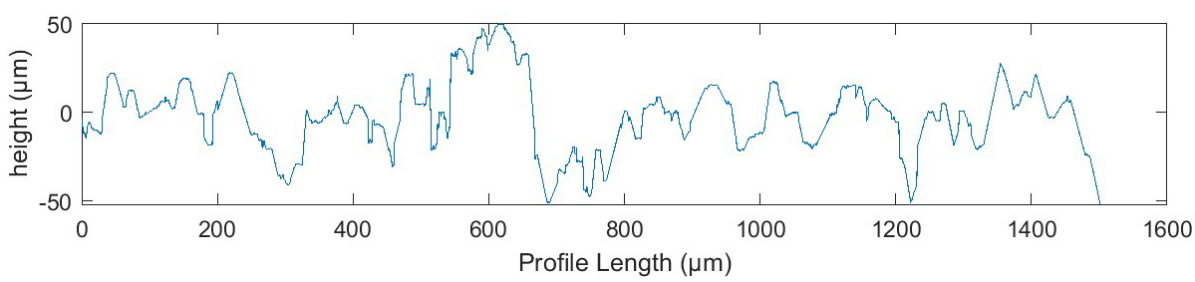

Figure N5: Flattened selected line profile from SWLI dataset sample 016

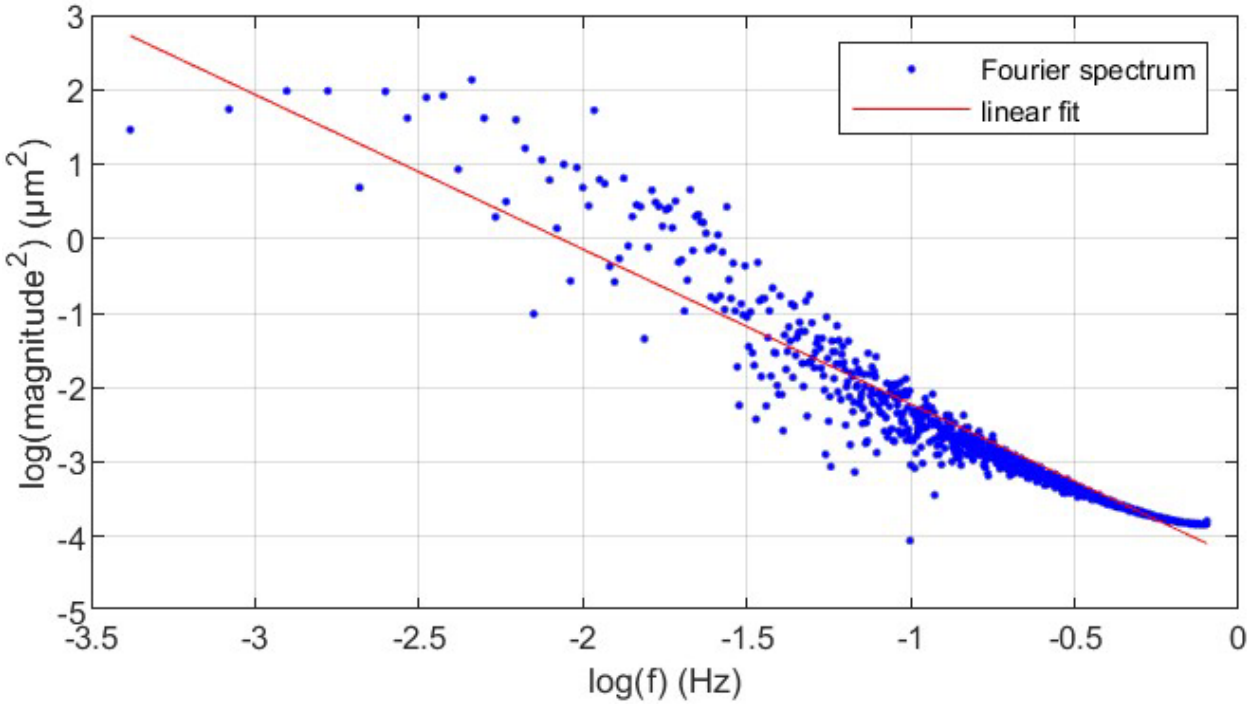

Figure N6: Fourier analysis and linear fit from Dektak line profile data sample 016

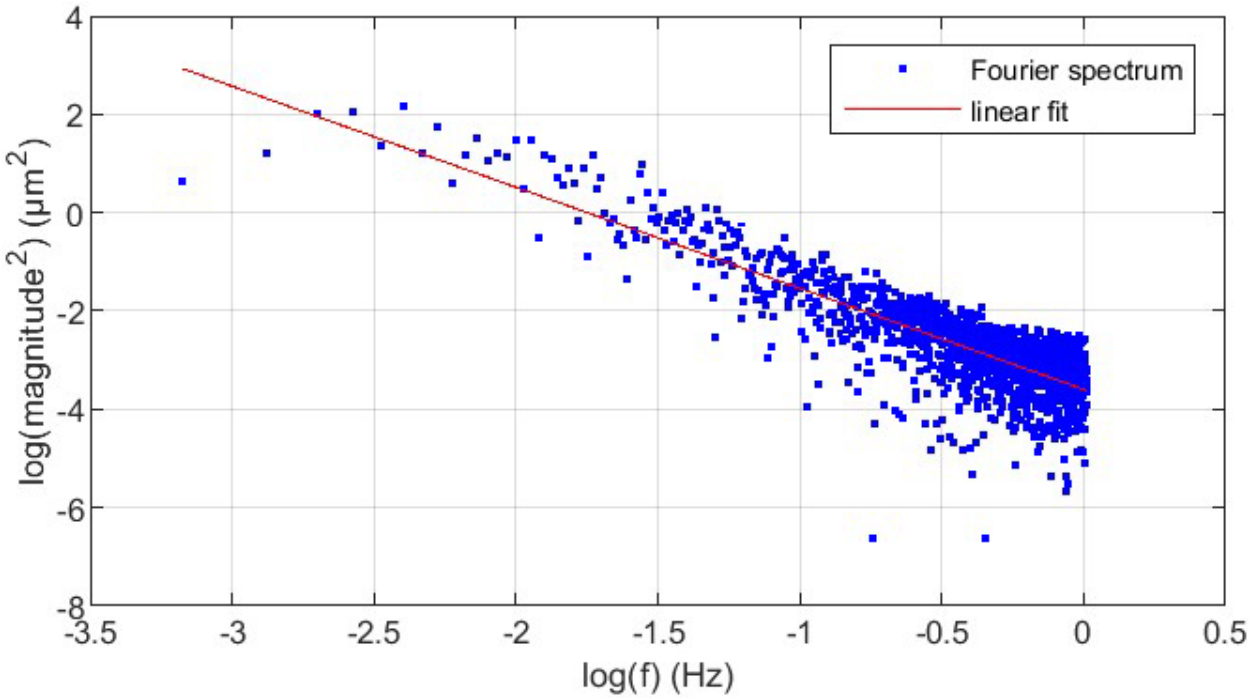

Figure N7: Fourier analysis and linear fit from SWLI line profile data sample 016

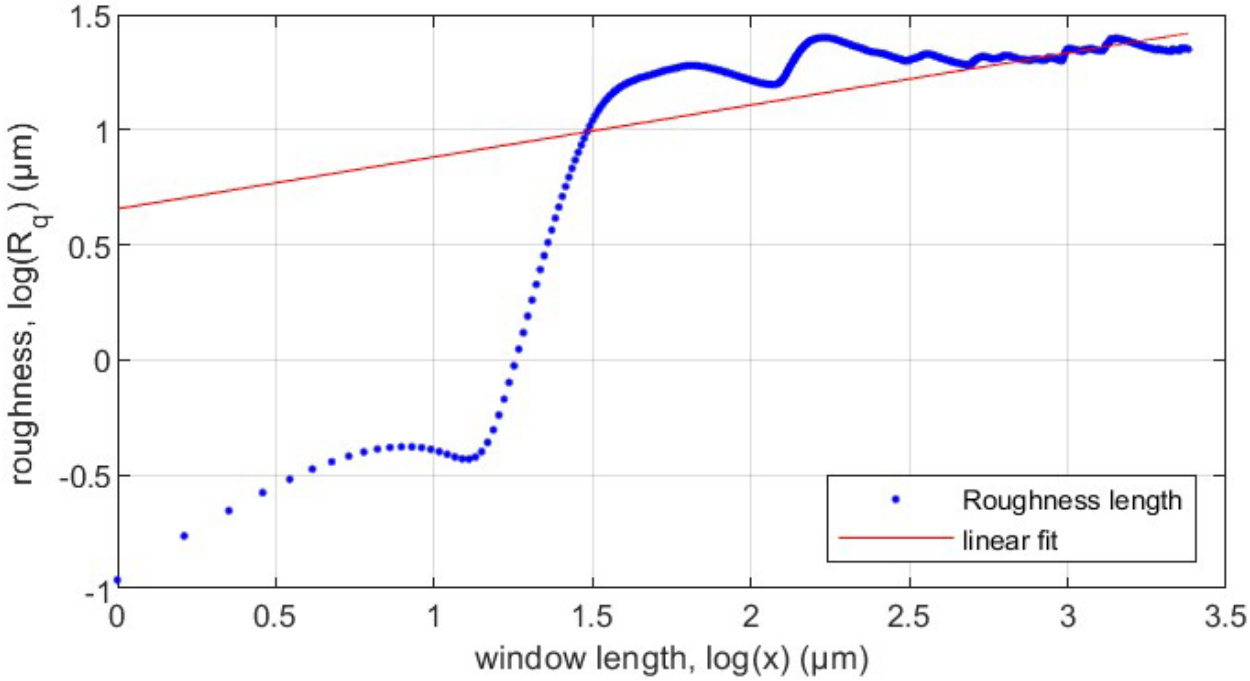

Figure N8: Roughness length method for calculating *H* from Dektak profile sample 016

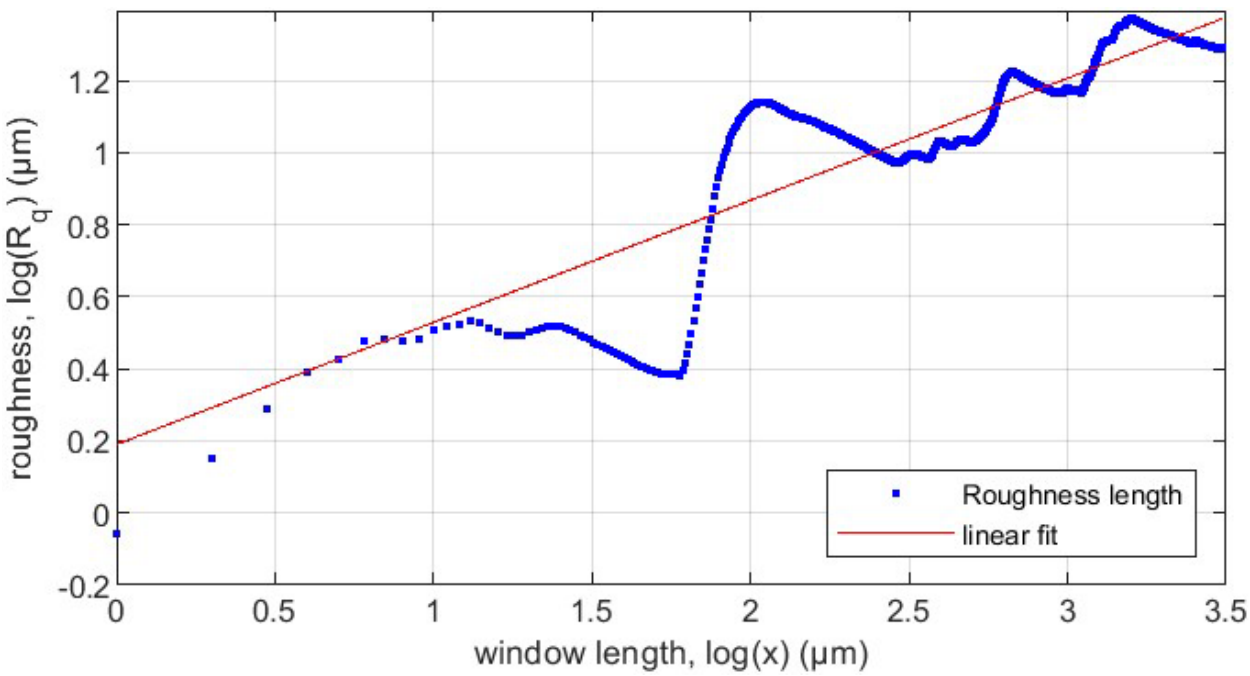

Figure N9: Roughness length method for calculating *H* from SWLI profile sample 016

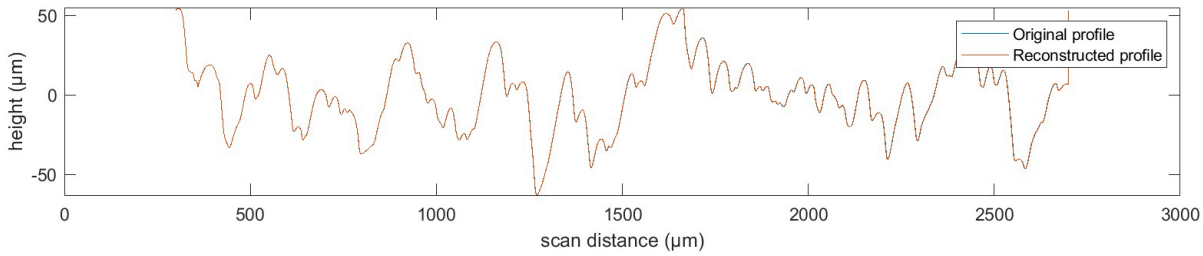

Figure N10: Original Dektak line scan profile and Fourier recreated line profile for sample 016

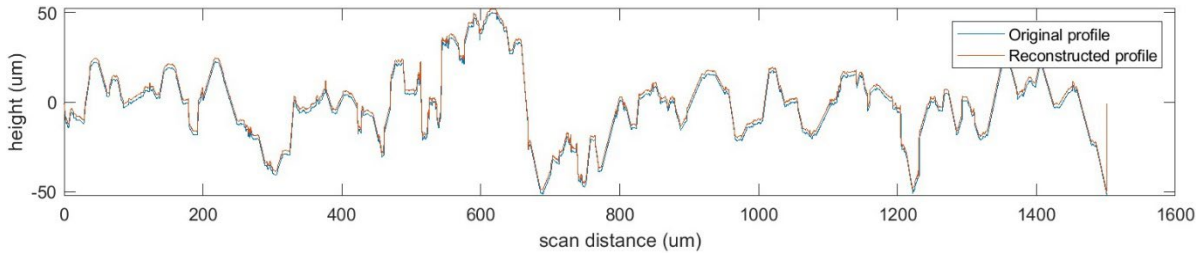

Figure N11: Original SWLI line profile and Fourier recreated line profile for sample 016

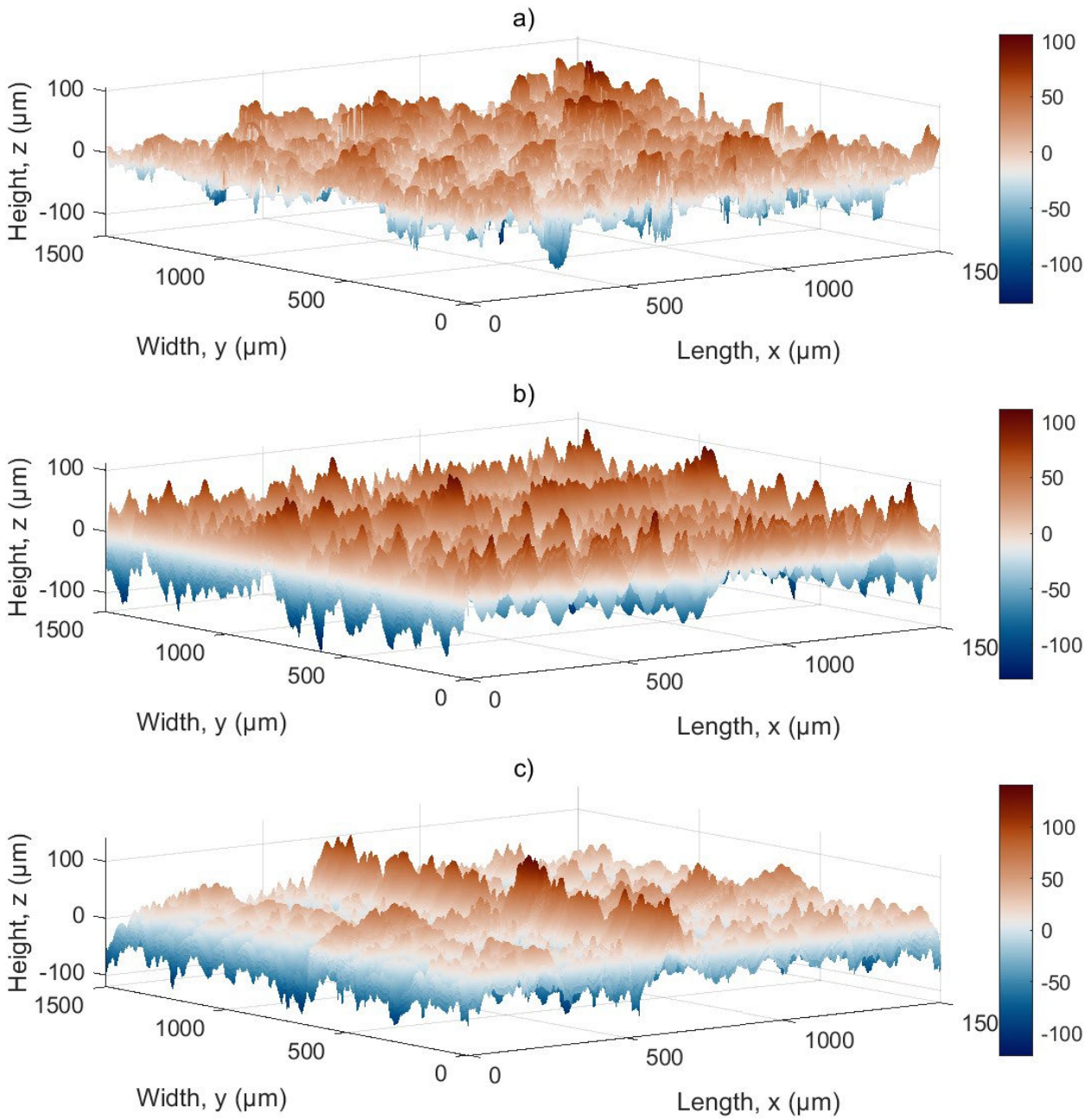

Figure N12: SWLI scan of sample 016 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 016 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 016

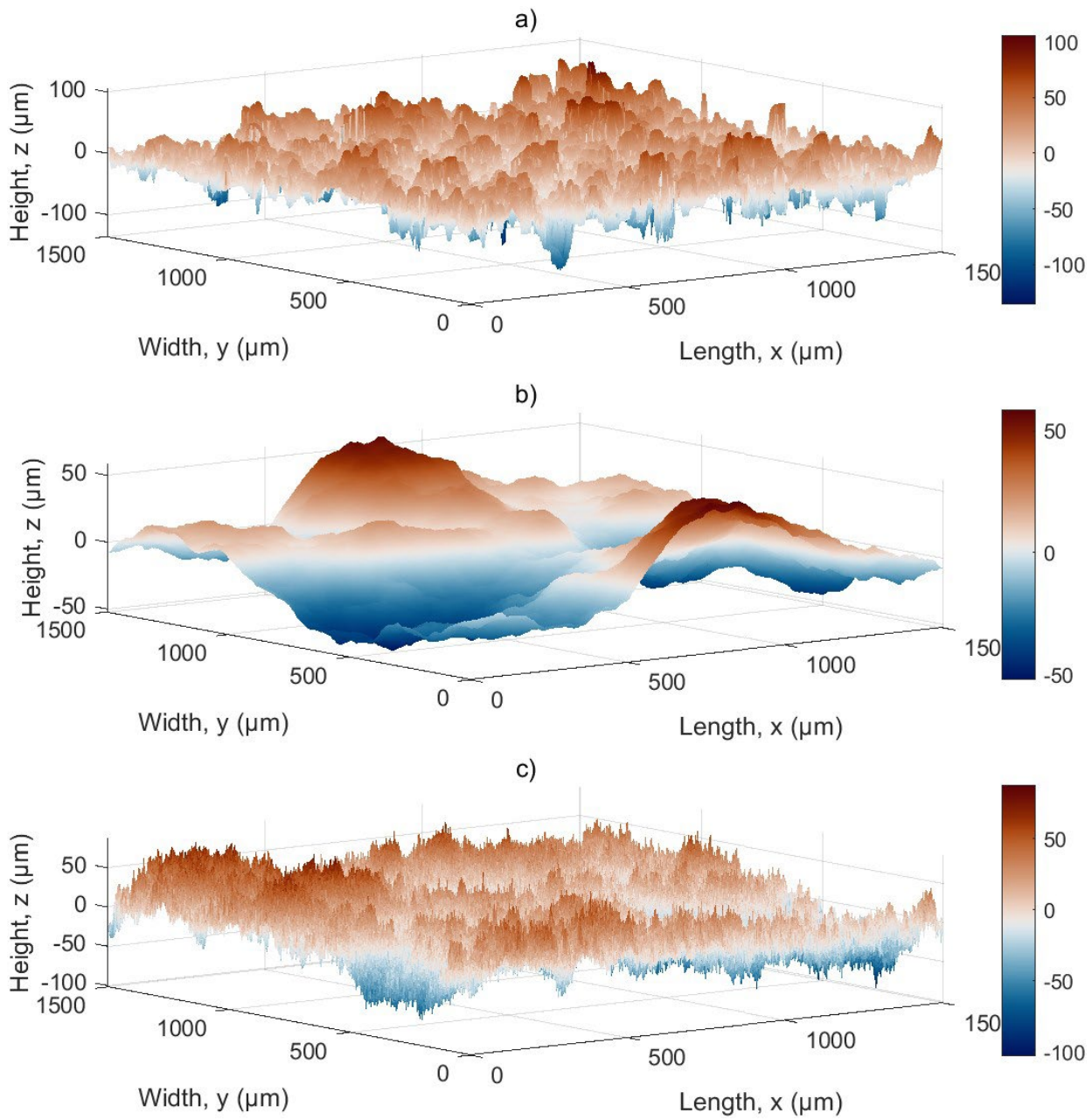

Figure N13: SWLI scan of sample 016 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 016 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 016

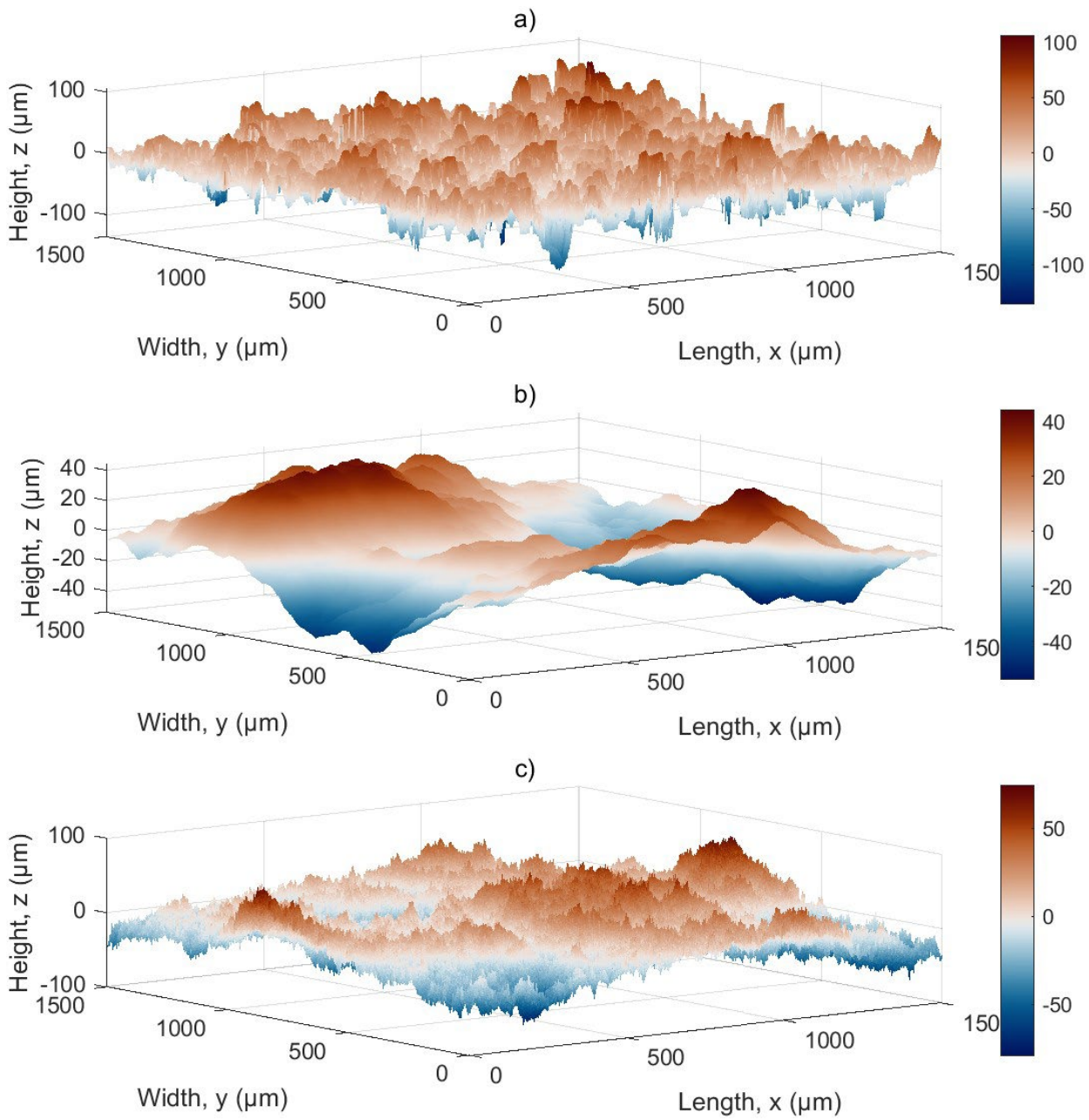

Figure N14: SWLI scan of sample 016 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 016 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 016

Appendix O: Sample 017 supplementary figures

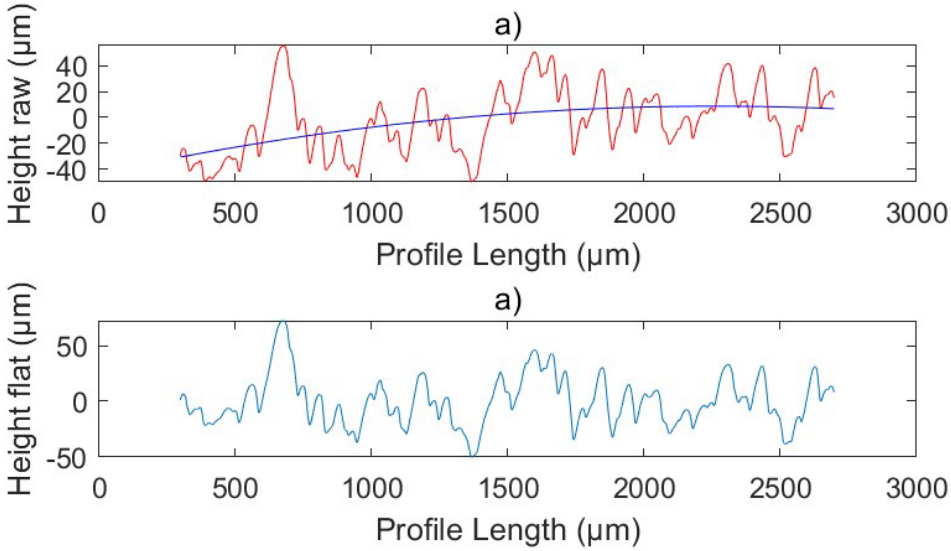

Figure O1: a) Raw profile measurement and fit curve sample 017 b) flattened line profile from sample 017

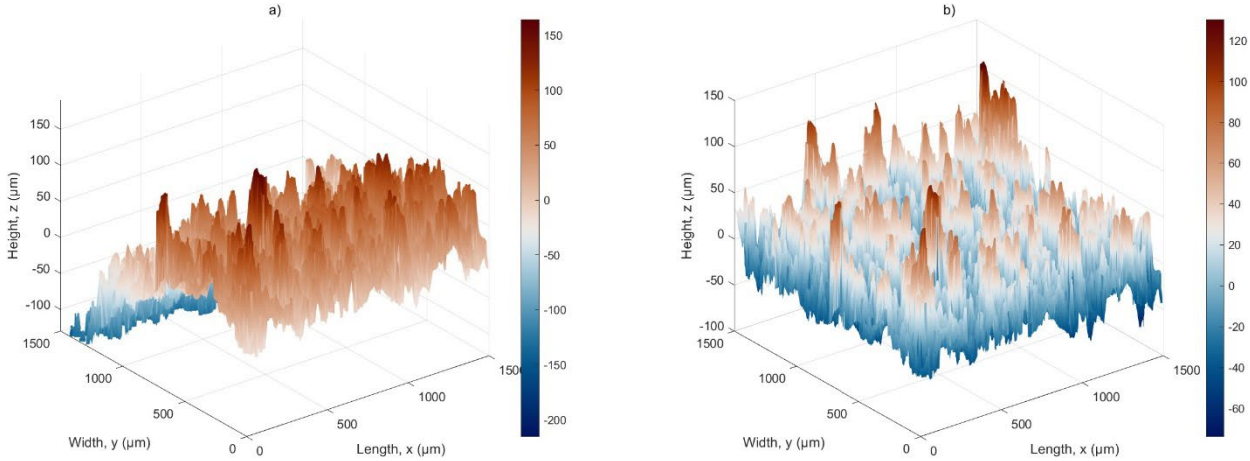

Figure O2: a) Unflattened SWLI raw surface scan sample 017 b) plane flattened SWLI surface 017

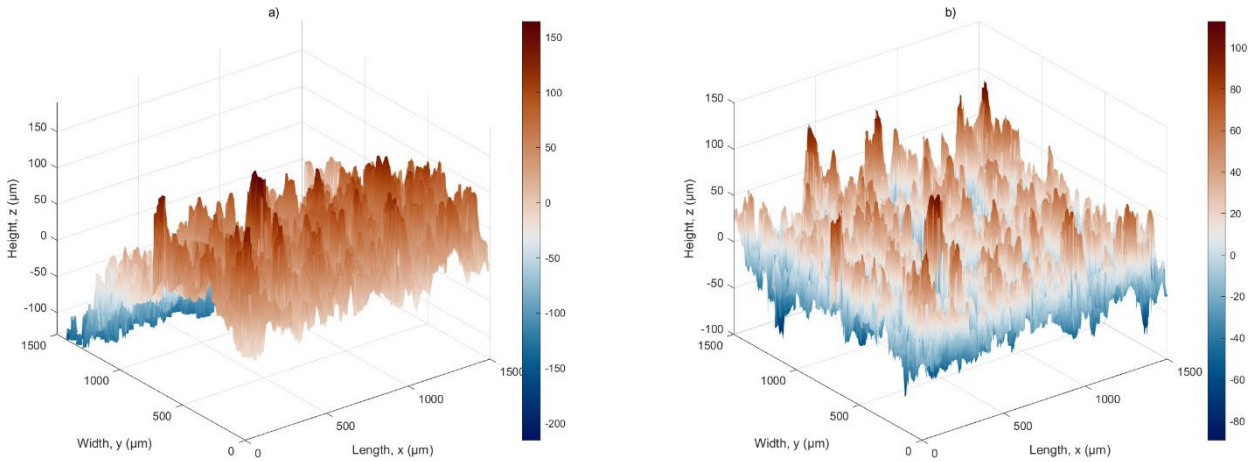

Figure O3: a) Unflattened SWLI raw surface scan sample 017 b) line by line flattened SWLI surface 017

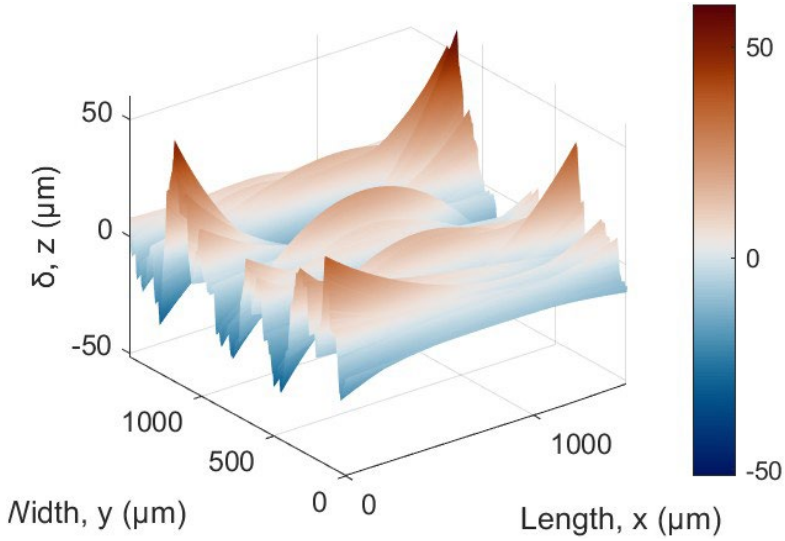

Figure O4: Difference between plane flattening and line by line flattening for sample 017

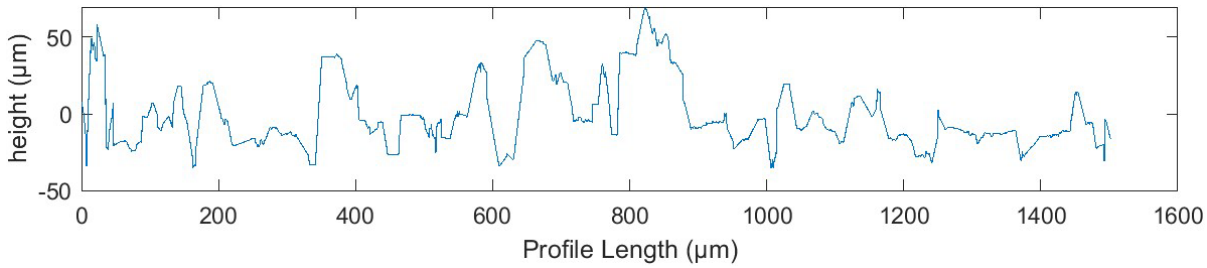

Figure O5: Flattened selected line profile from SWLI dataset sample 017

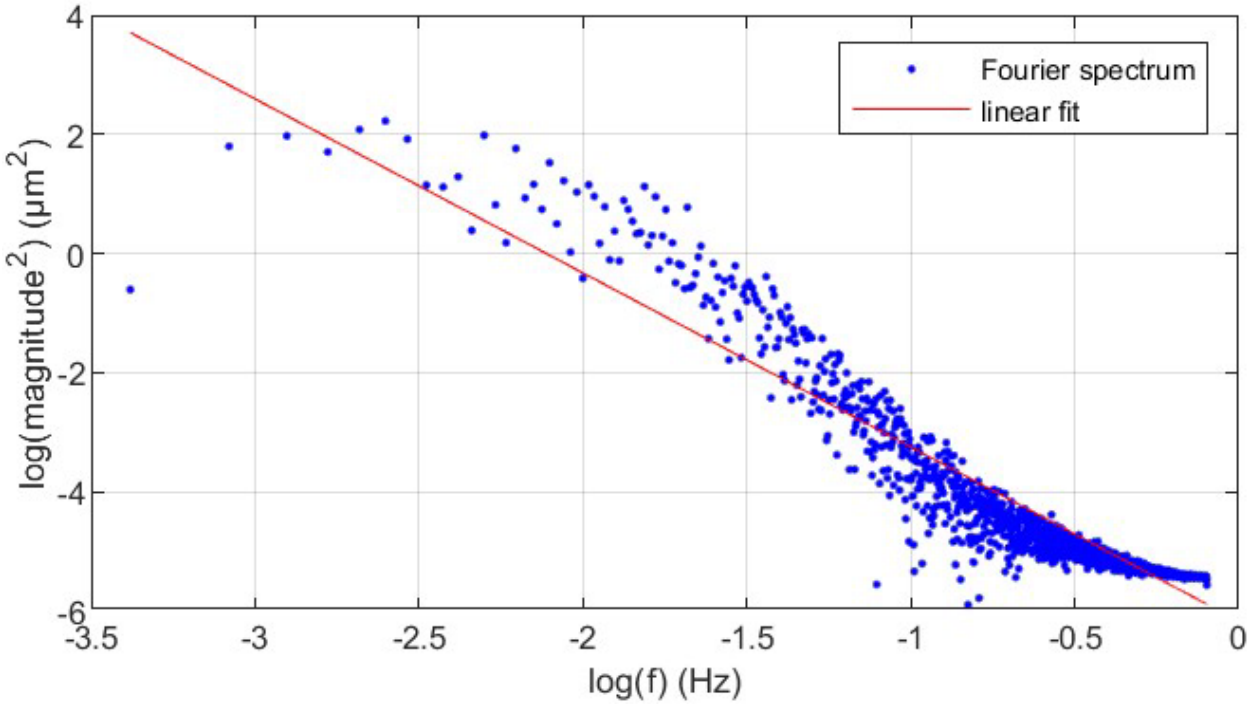

Figure O6: Fourier analysis and linear fit from Dektak line profile data sample 017

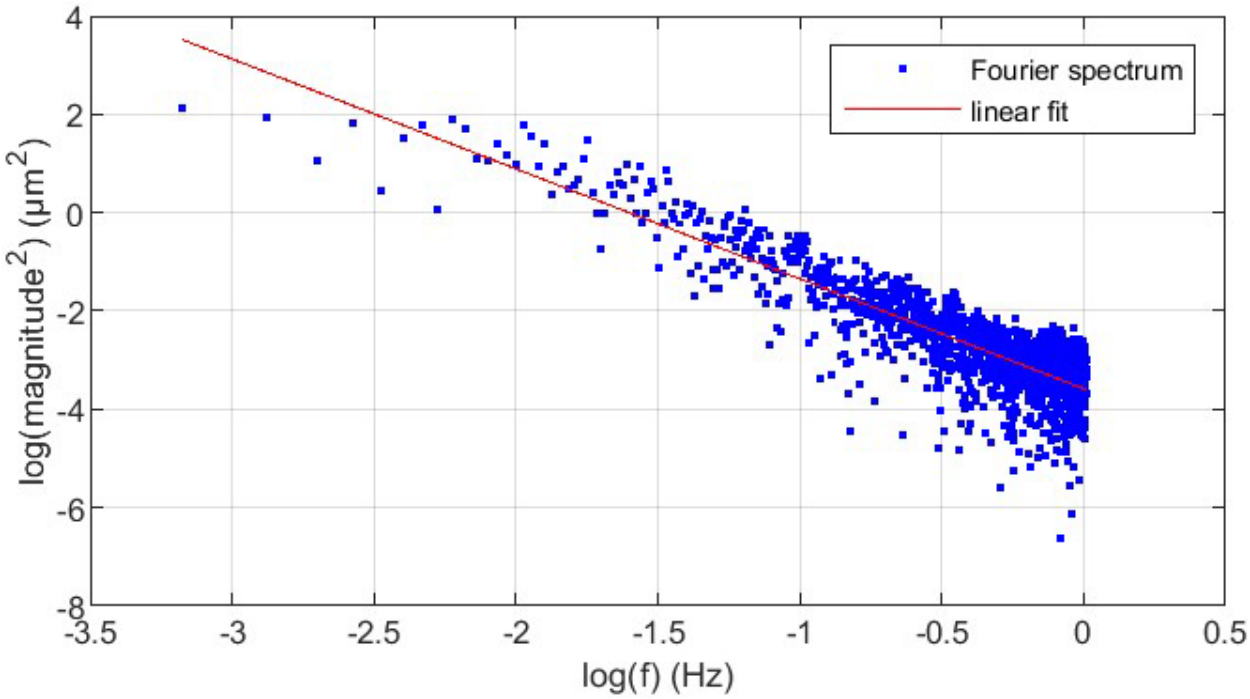

Figure O7: Fourier analysis and linear fit from SWLI line profile data sample 017

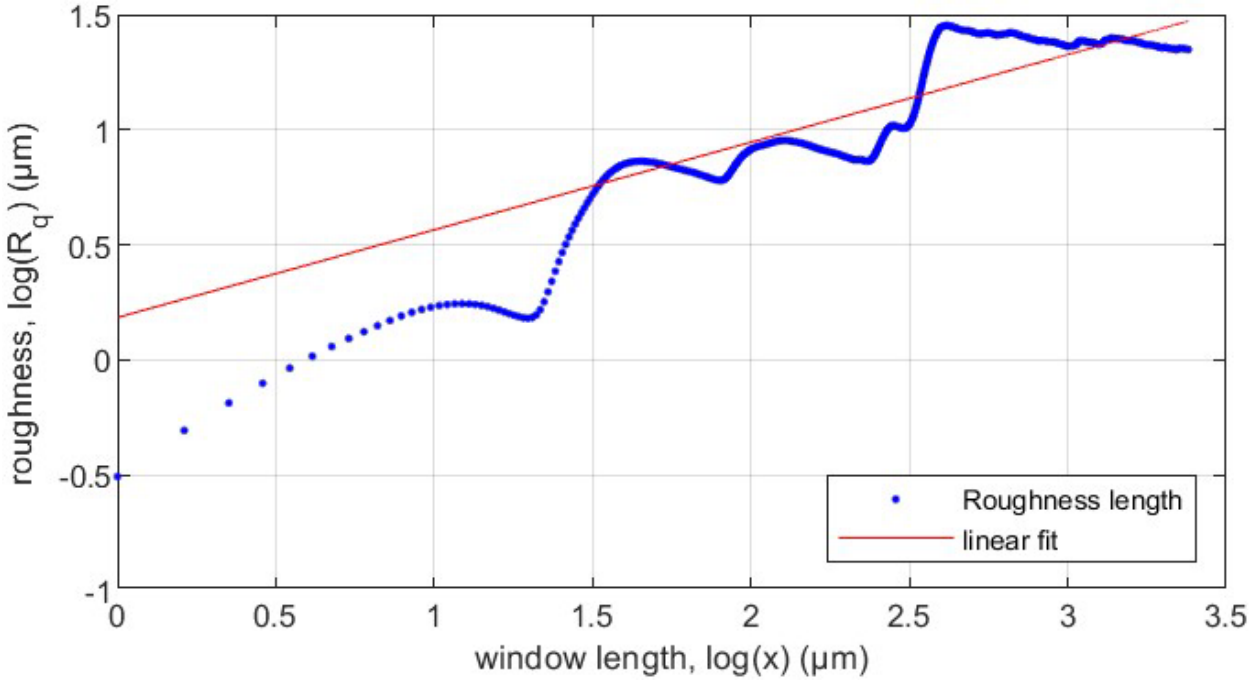

Figure O8: Roughness length method for calculating *H* from Dektak profile sample 017

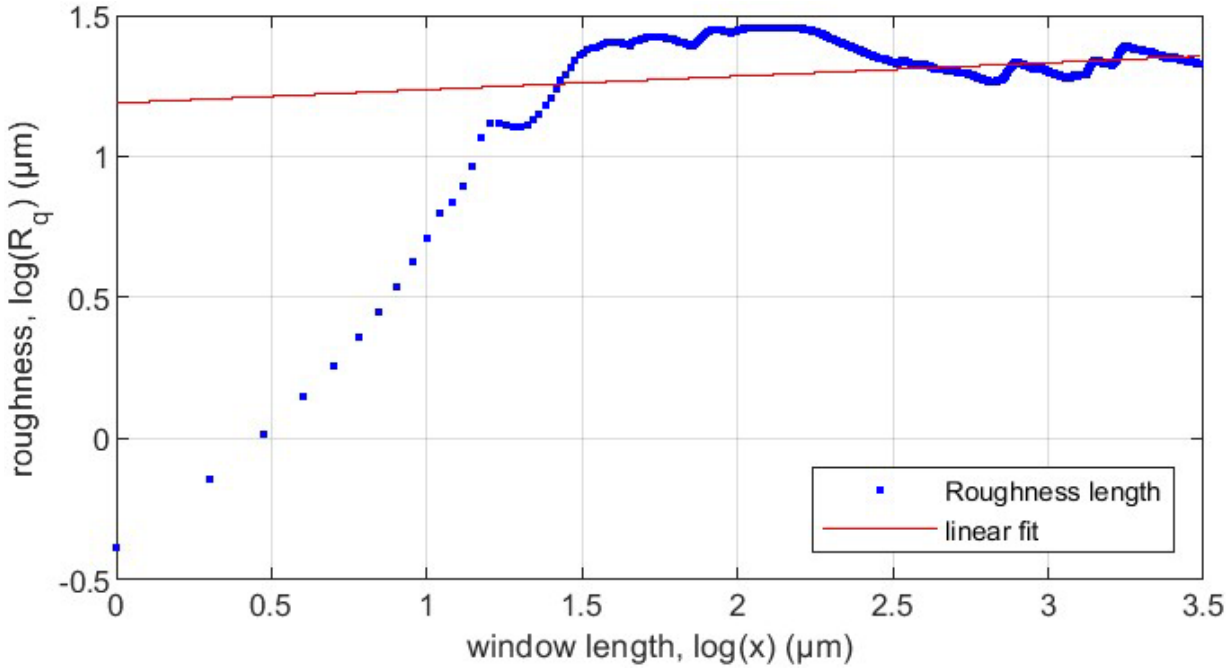

Figure O9: Roughness length method for calculating *H* from SWLI profile sample 017

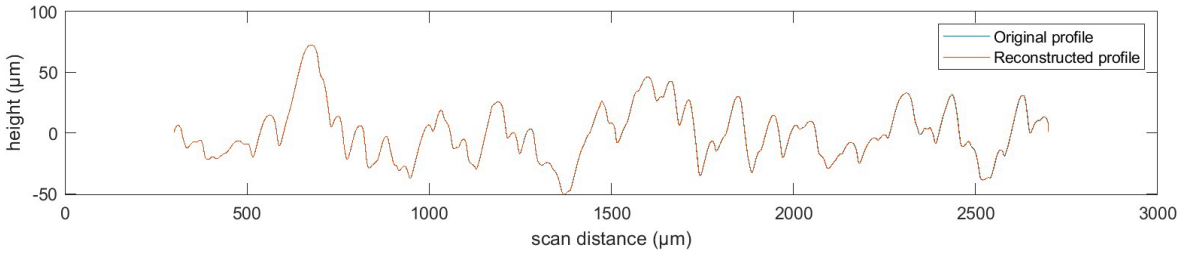

Figure O10: Original Dektak line scan profile and Fourier recreated line profile for sample 017

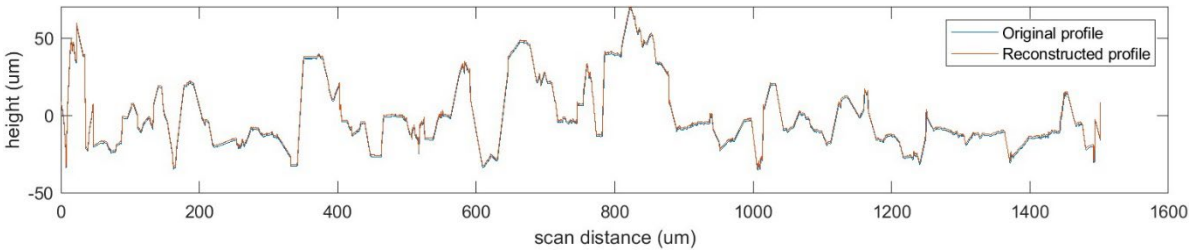

Figure O11: Original SWLI line profile and Fourier recreated line profile for sample 017

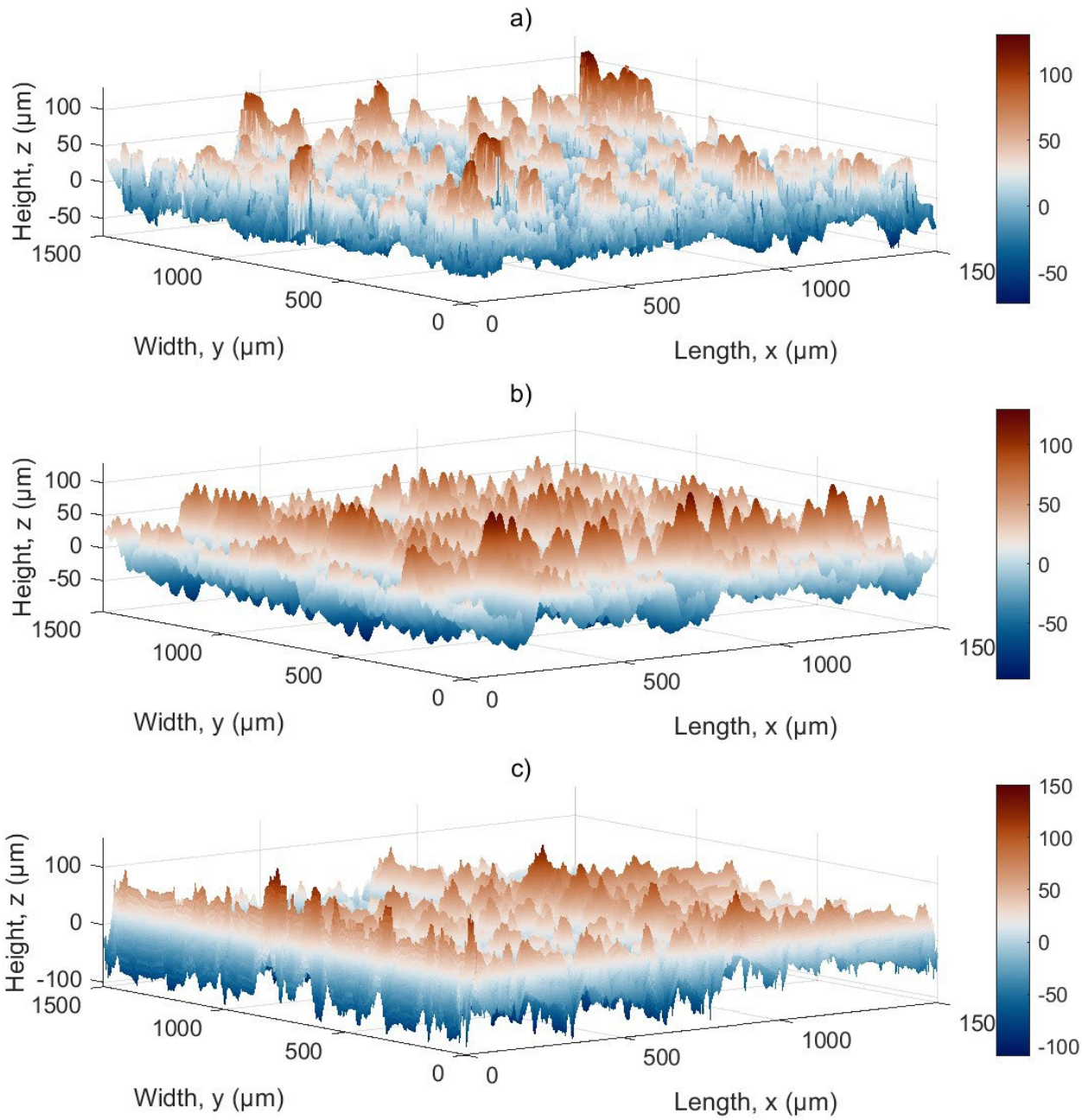

Figure O12: SWLI scan of sample 017 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 017 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 017

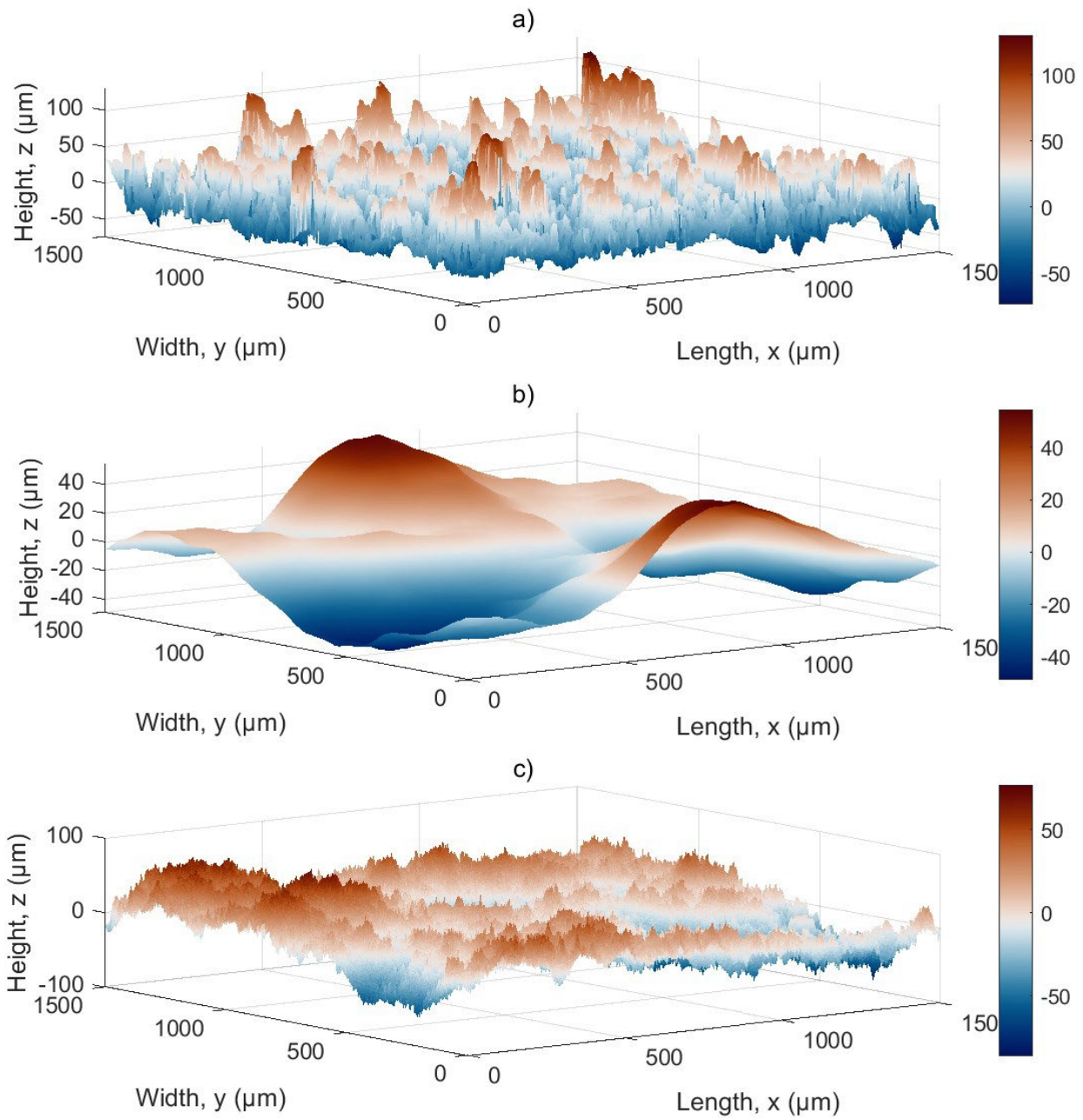

Figure O13: SWLI scan of sample 017 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 017 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 017

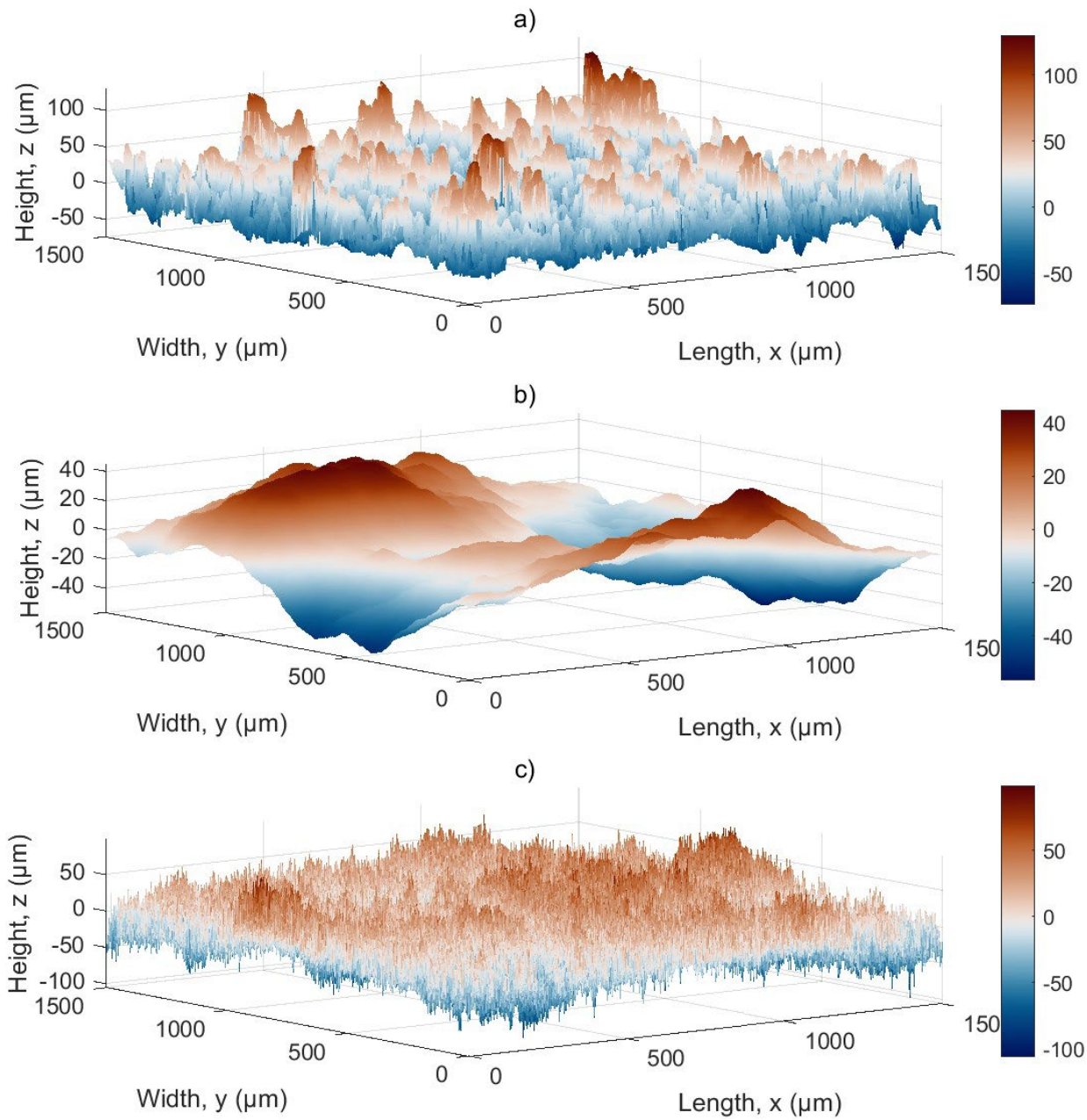

Figure O14: SWLI scan of sample 017 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 017 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 017

Appendix P: Sample 018 supplementary figures

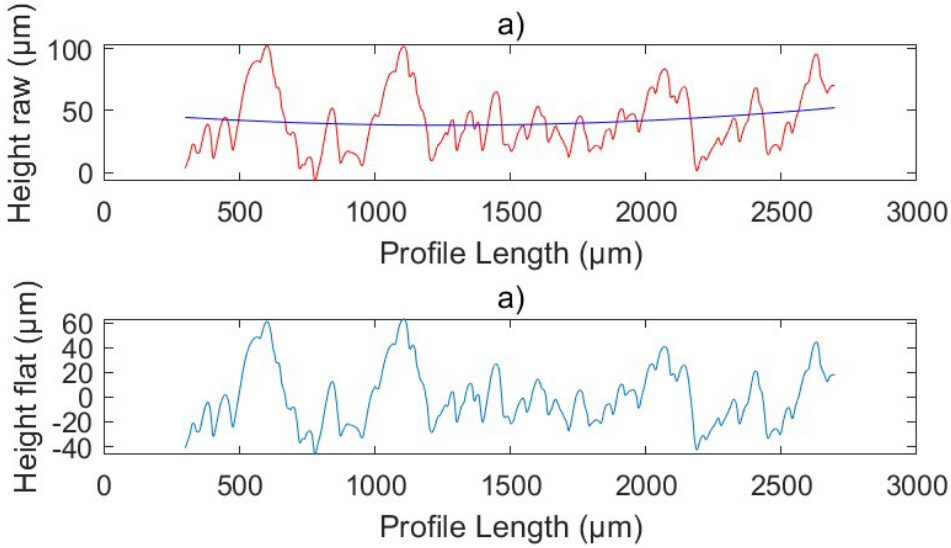

Figure P1: a) Raw profile measurement and fit curve sample 018 b) flattened line profile from sample 018

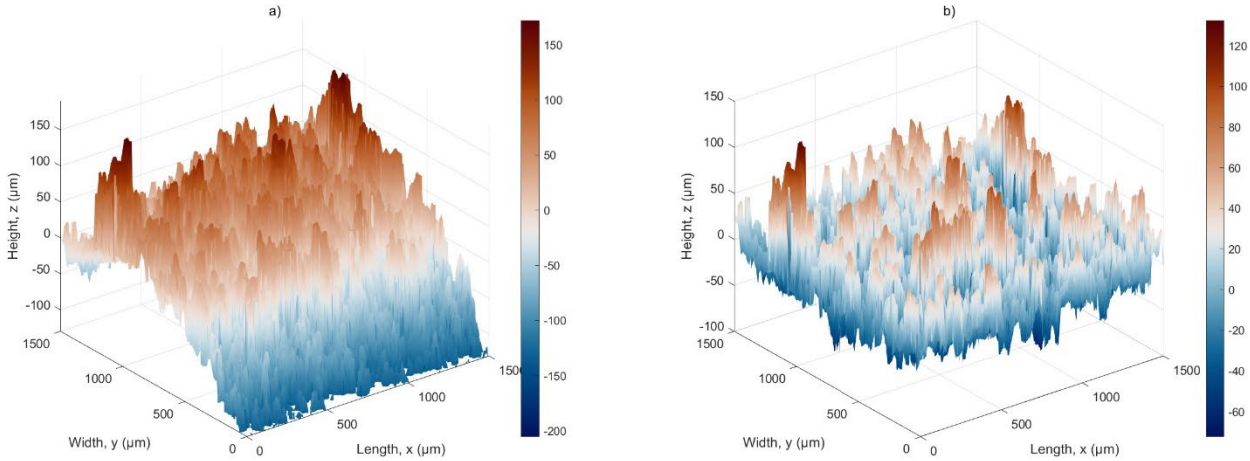

Figure P2: a) Unflattened SWLI raw surface scan sample 018 b) plane flattened SWLI surface 018

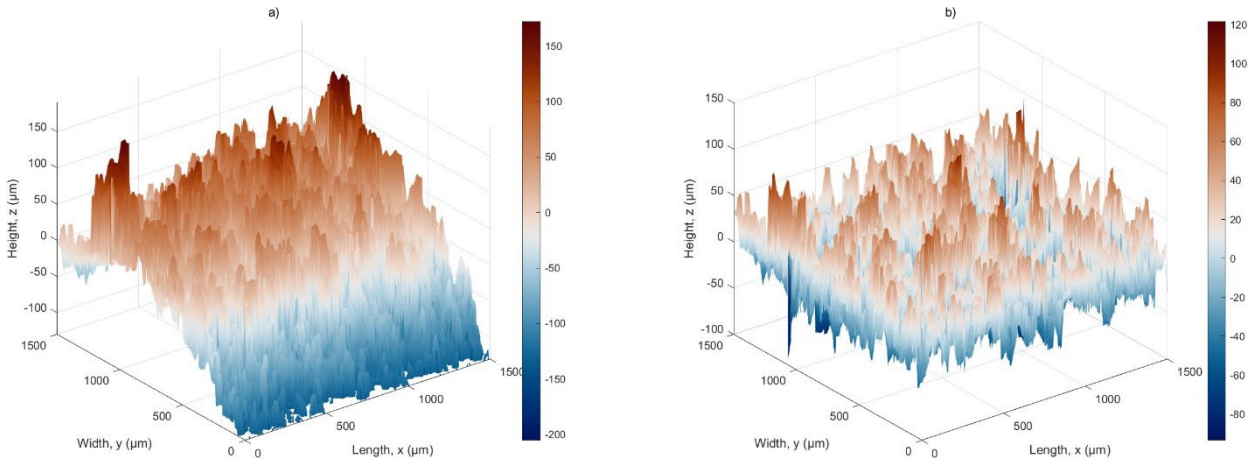

Figure P3: a) Unflattened SWLI raw surface scan sample 018 b) line by line flattened SWLI surface 018

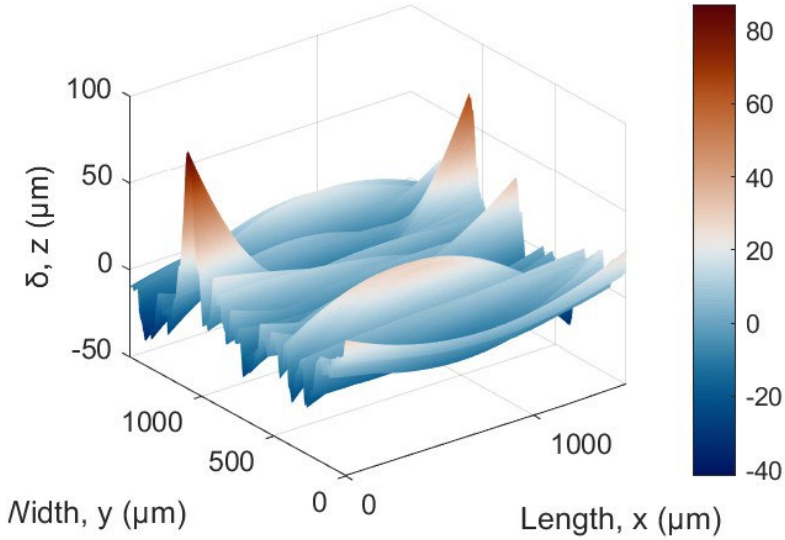

Figure P4: Difference between plane flattening and line by line flattening for sample 018

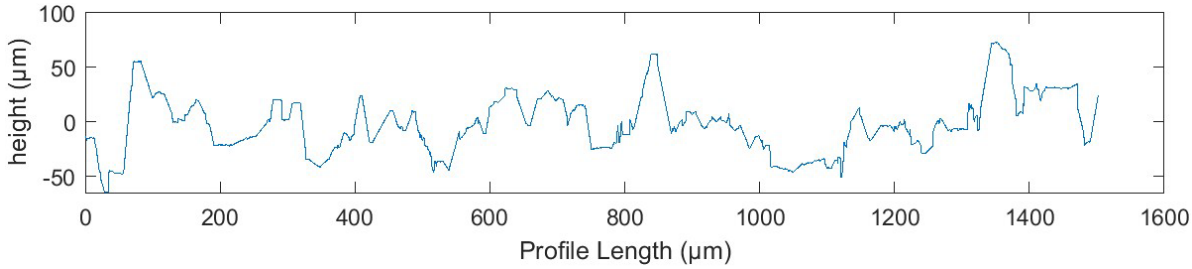

Figure P5: Flattened selected line profile from SWLI dataset sample 018

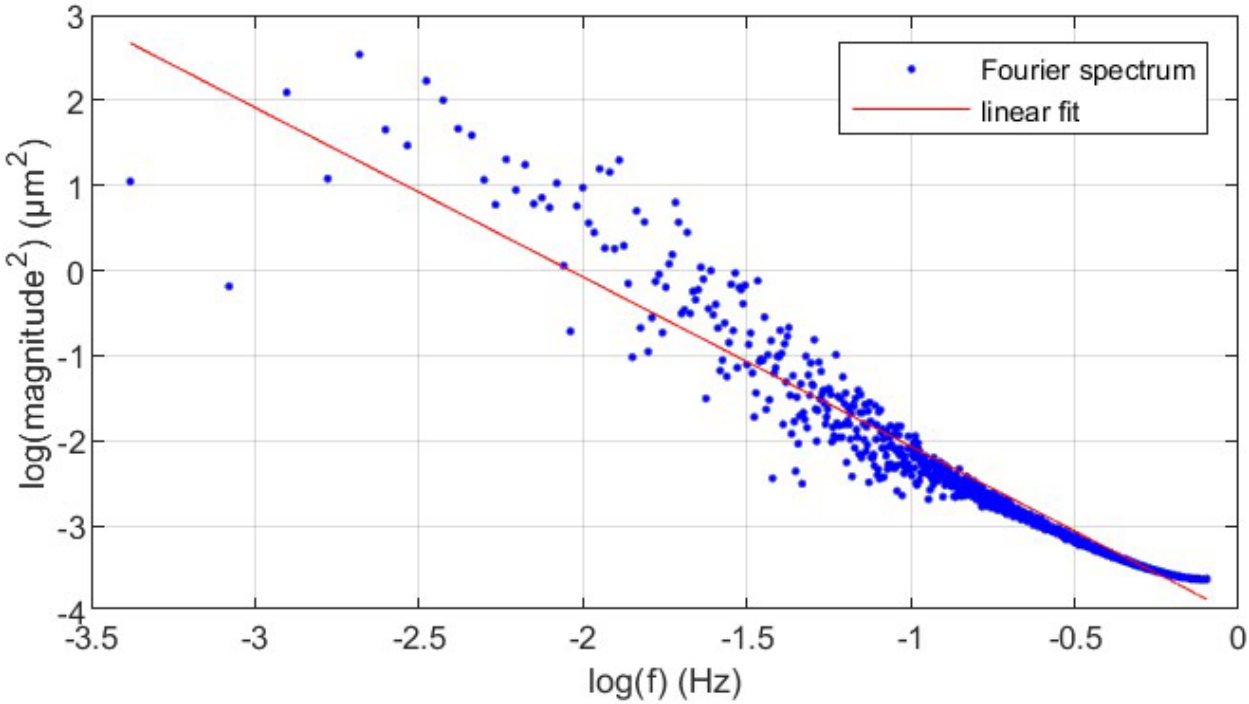

Figure P6: Fourier analysis and linear fit from Dektak line profile data sample 018

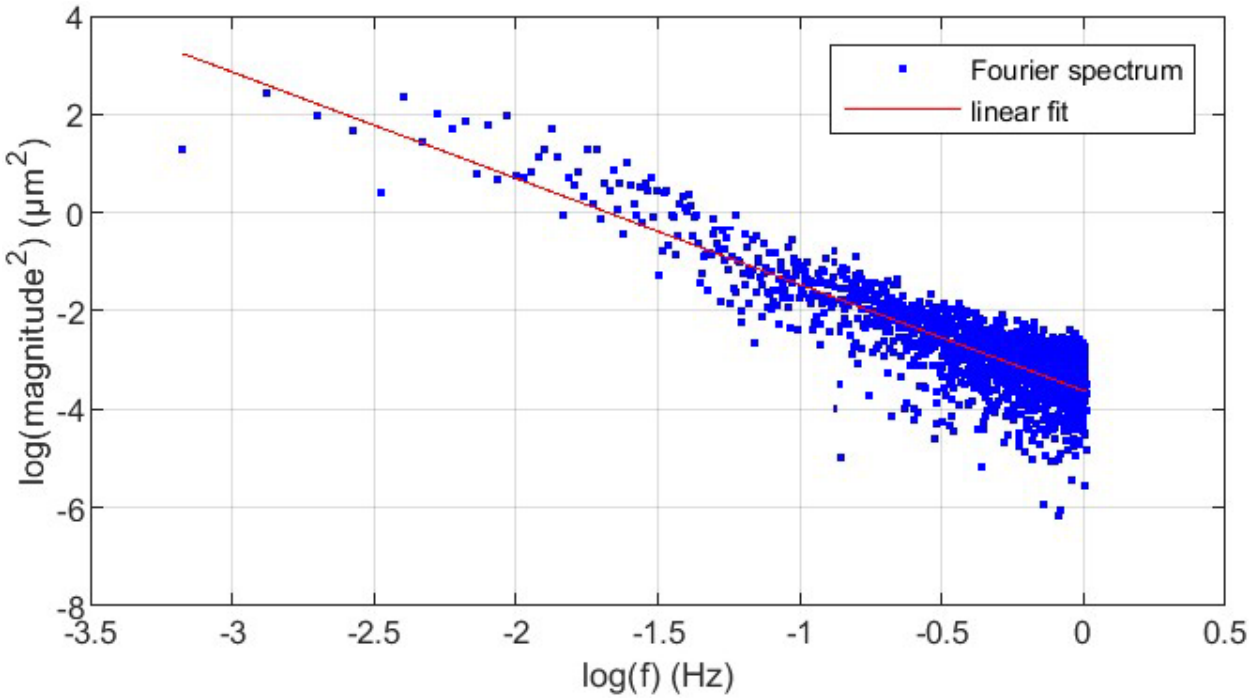

Figure P7: Fourier analysis and linear fit from SWLI line profile data sample 018

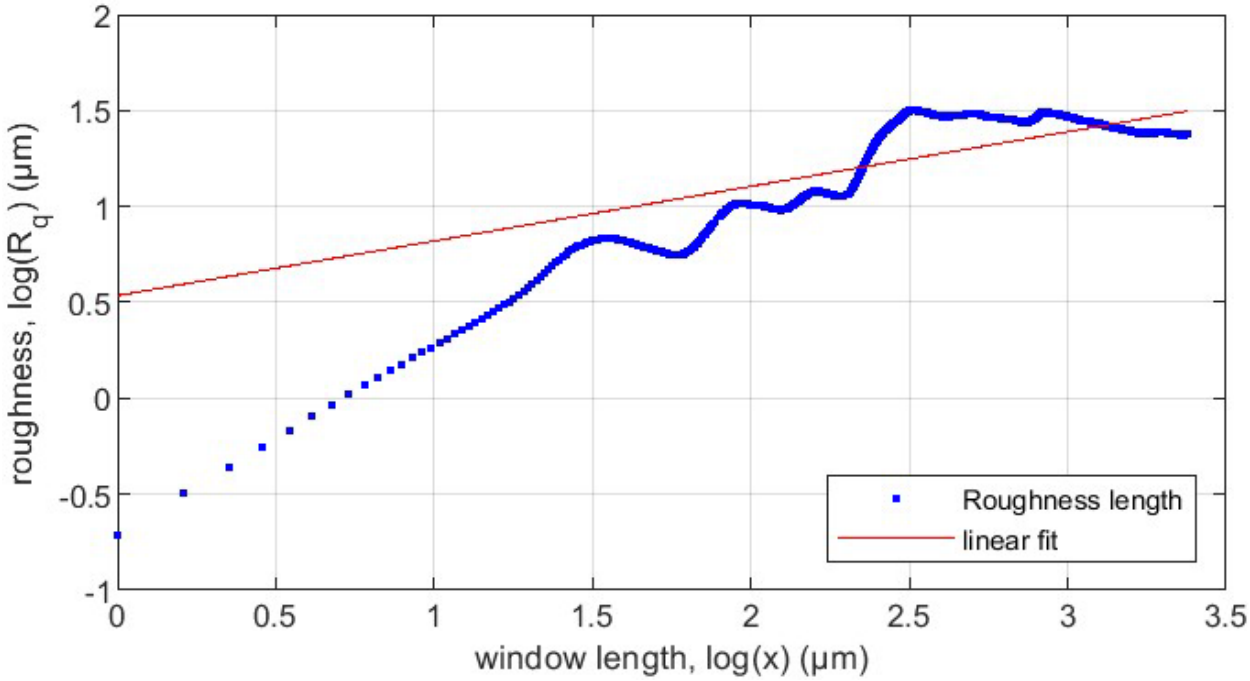

Figure P8: Roughness length method for calculating *H* from Dektak profile sample 018

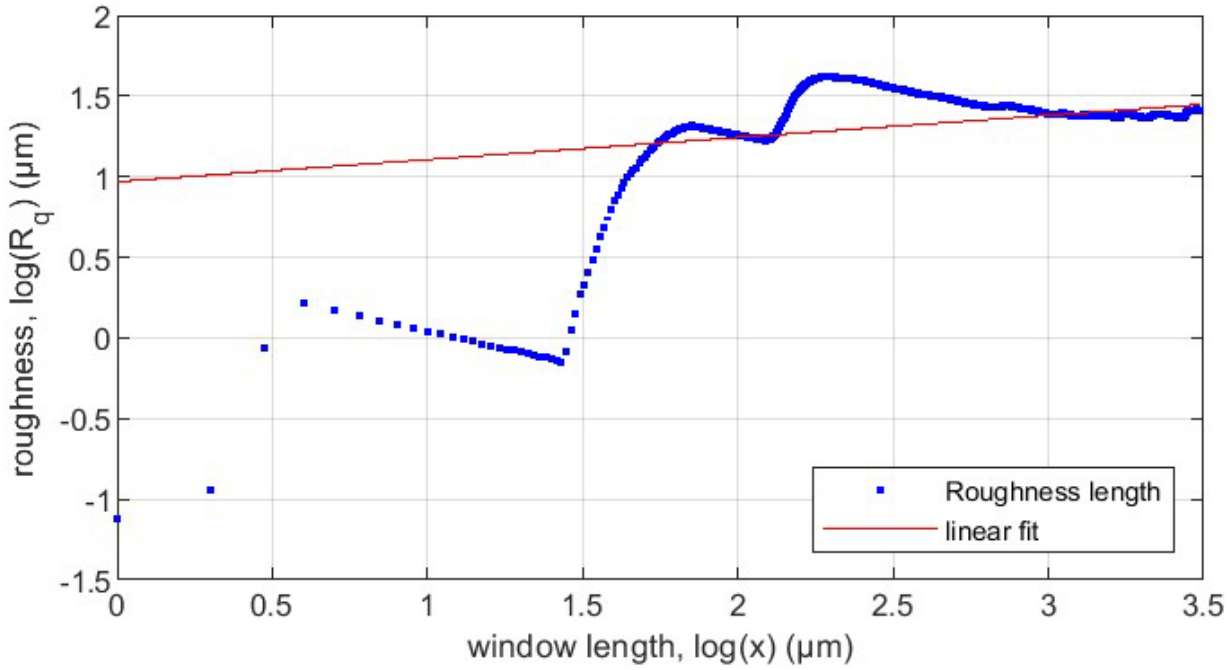

Figure P9: Roughness length method for calculating *H* from SWLI profile sample 018

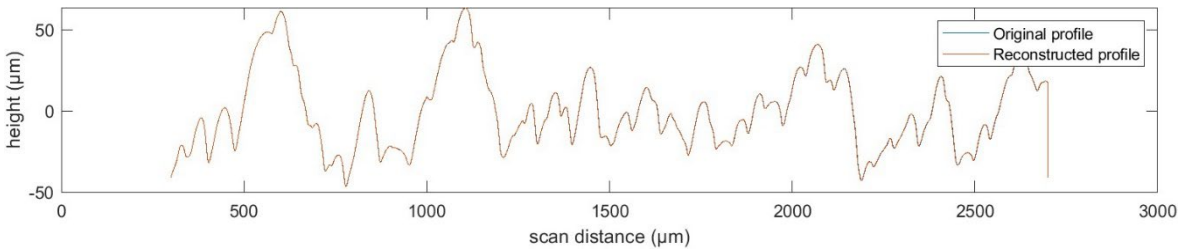

Figure P10: Original Dektak line scan profile and Fourier recreated line profile for sample 018

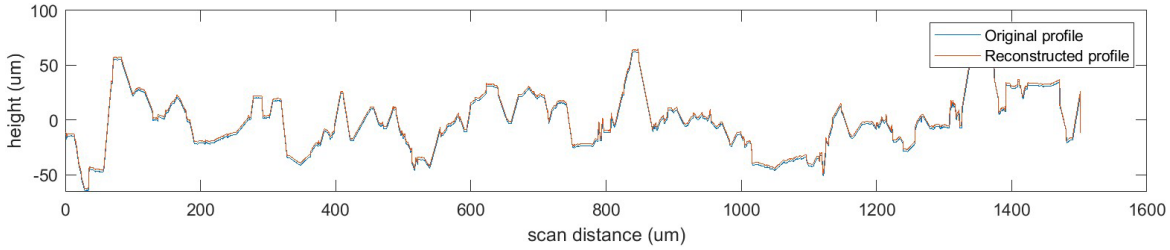

Figure P11: Original SWLI line profile and Fourier recreated line profile for sample 018

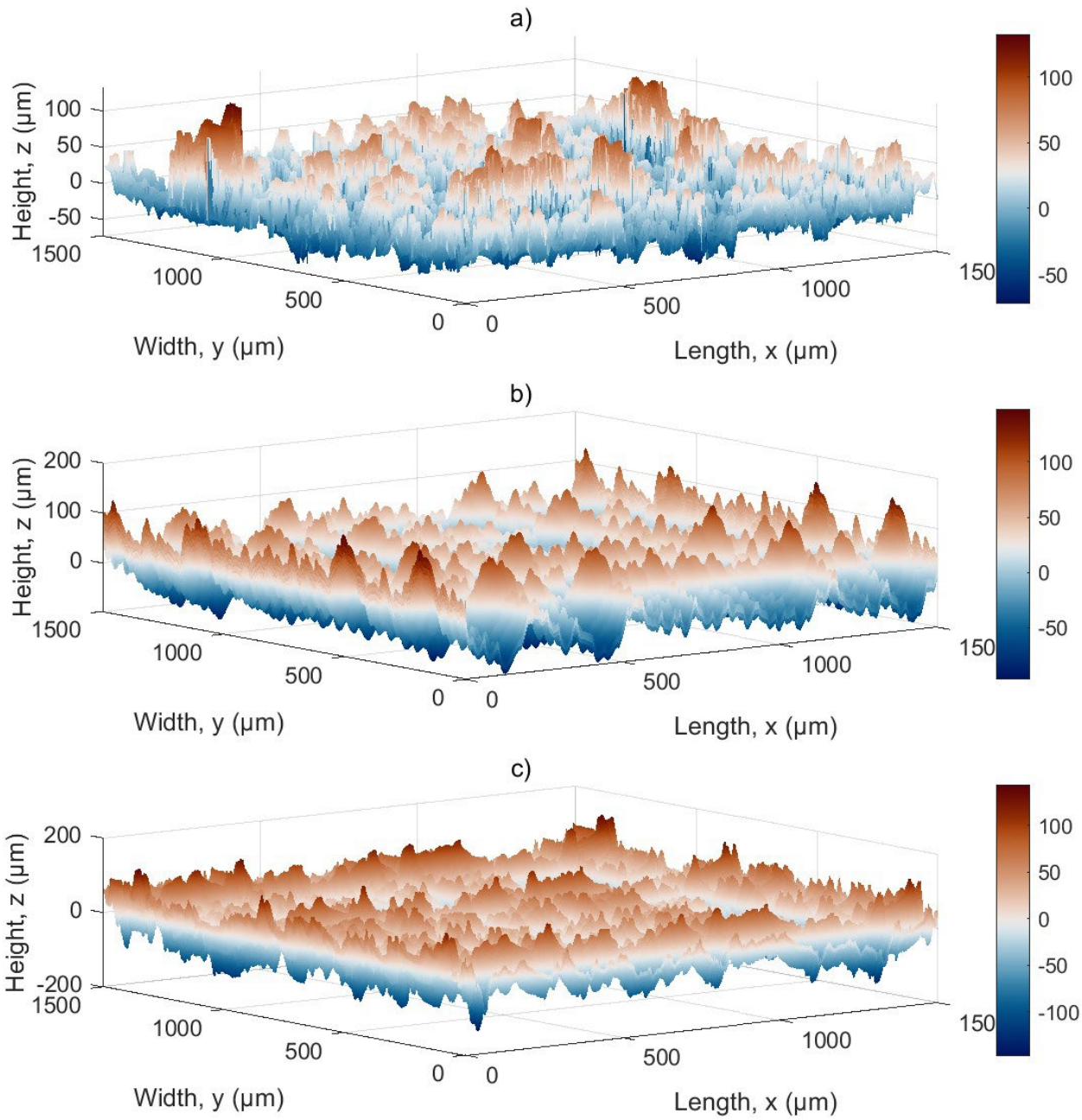

Figure P12: SWLI scan of sample 018 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 018 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 018

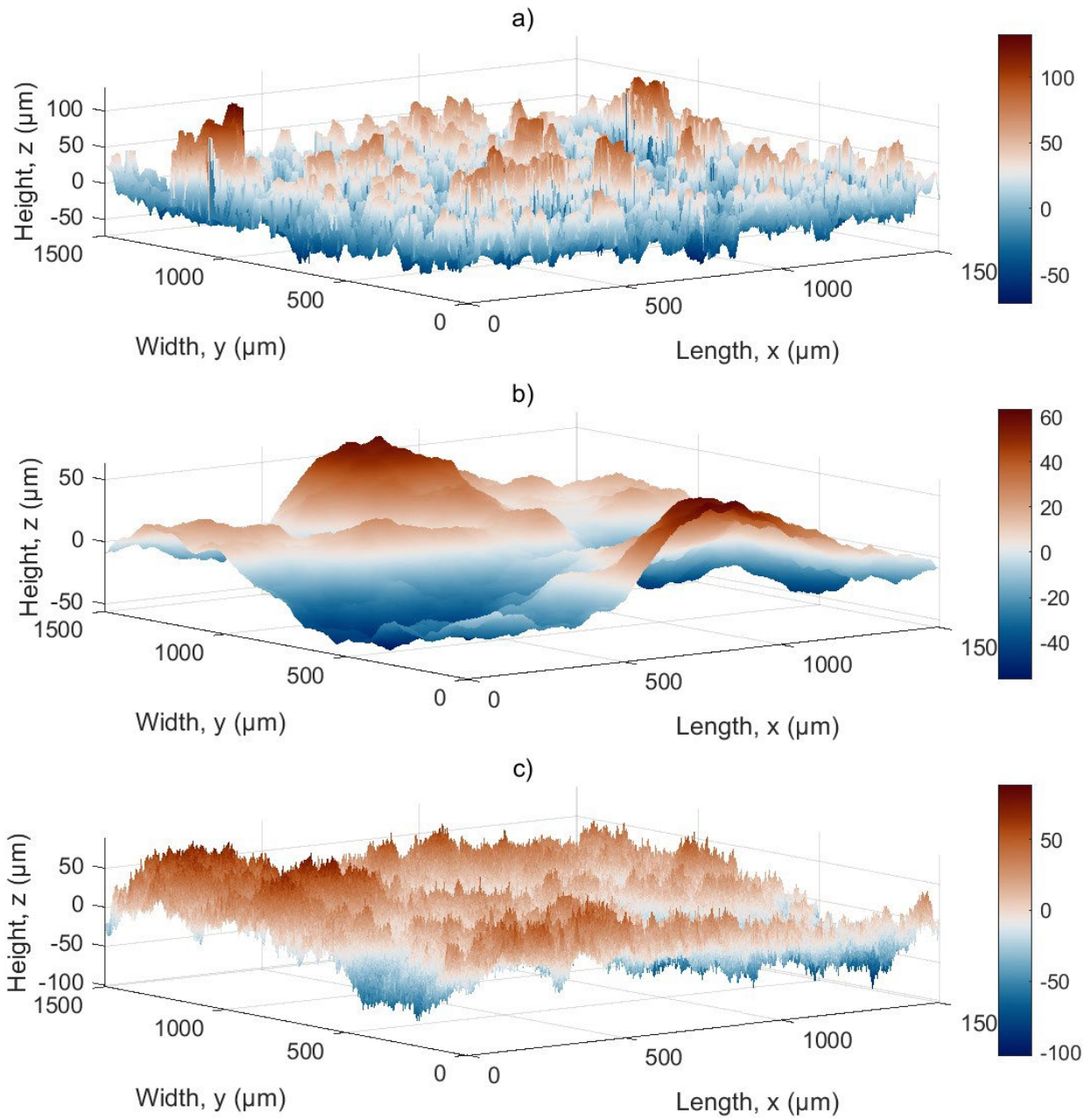

Figure P13: SWLI scan of sample 018 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 018 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 018

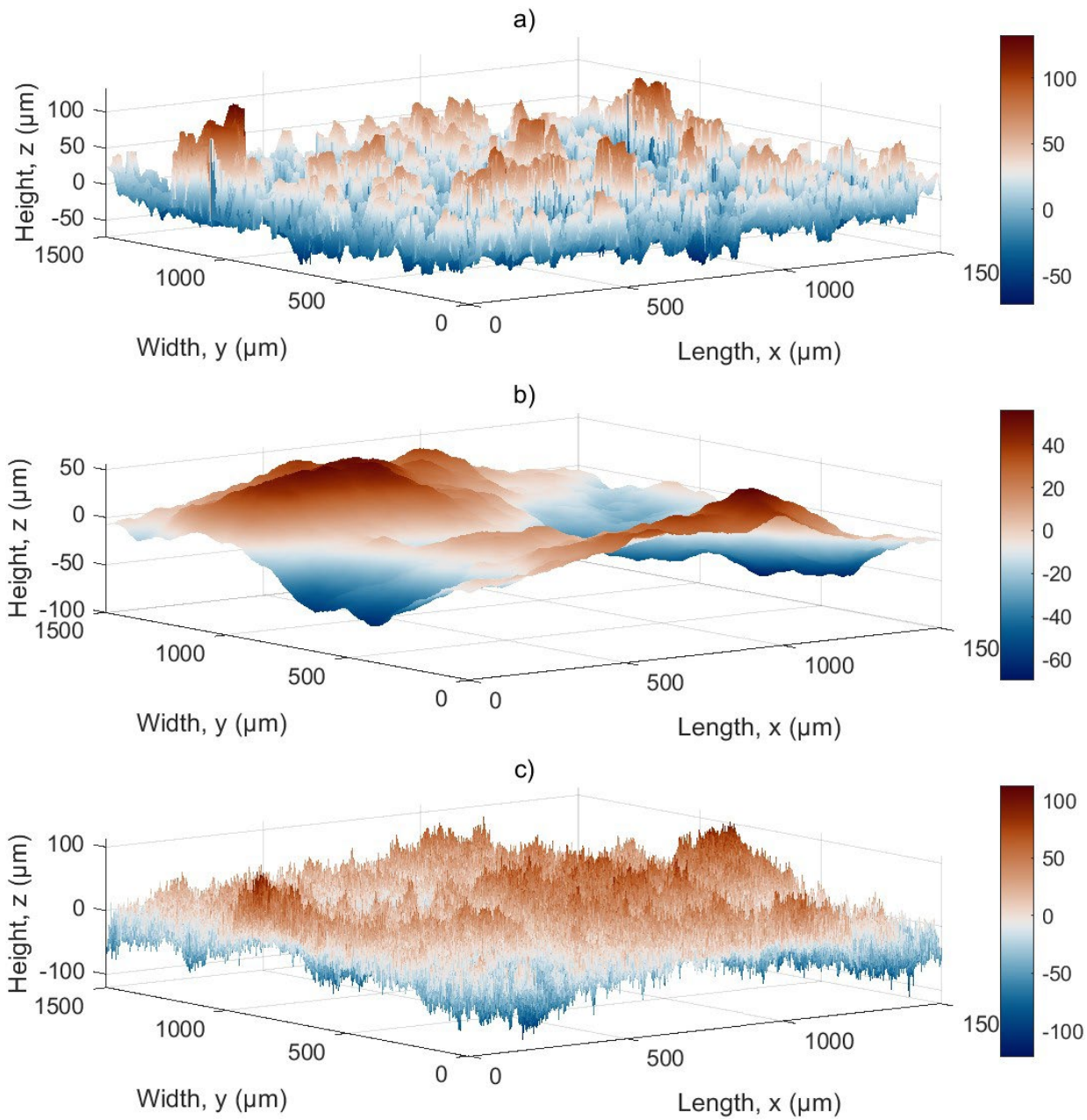

Figure P14: SWLI scan of sample 018 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 018 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 018

Appendix Q: Sample 024 supplementary figures

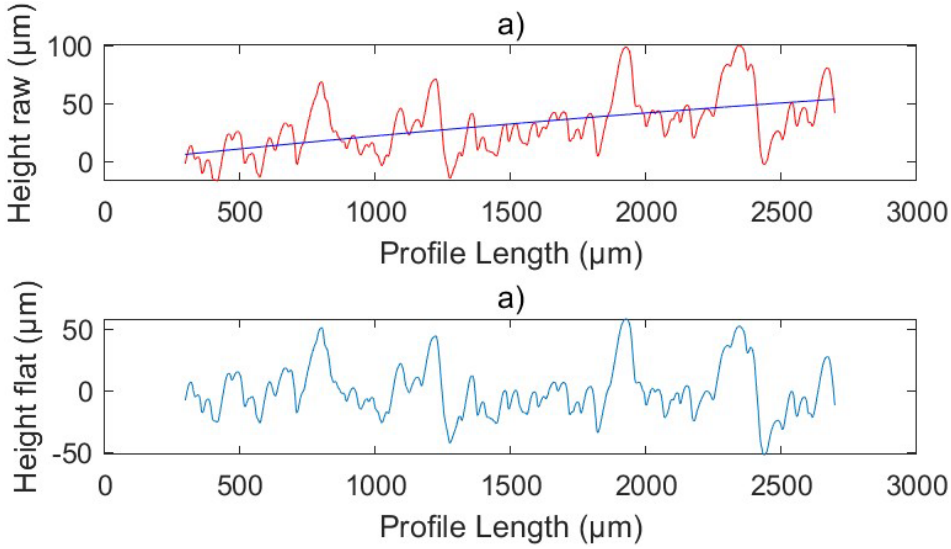

Figure Q1: a) Raw profile measurement and fit curve sample 024 b) flattened line profile from sample 024

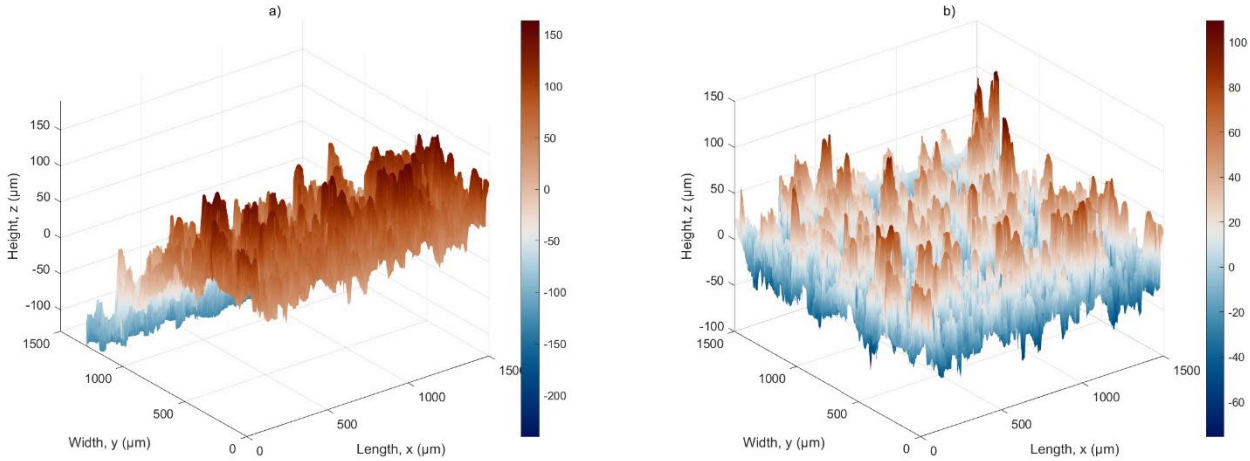

Figure Q2: a) Unflattened SWLI raw surface scan sample 024 b) plane flattened SWLI surface 024

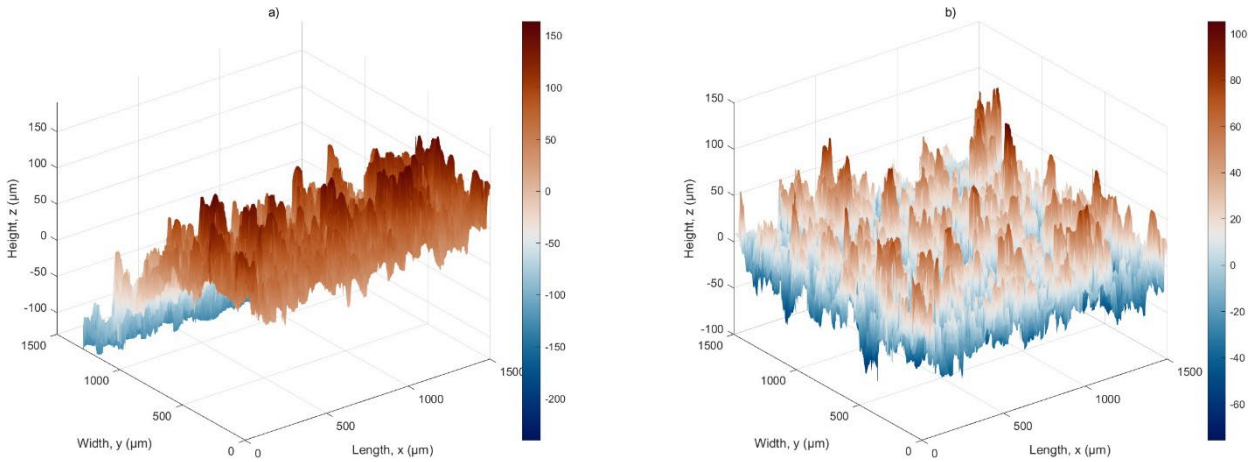

Figure Q3: a) Unflattened SWLI raw surface scan sample 024 b) line by line flattened SWLI surface 024

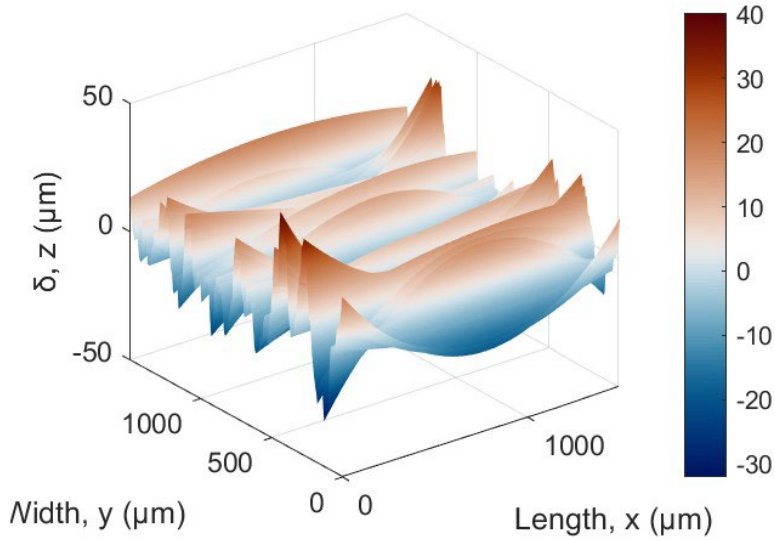

Figure Q4: Difference between plane flattening and line by line flattening for sample 024

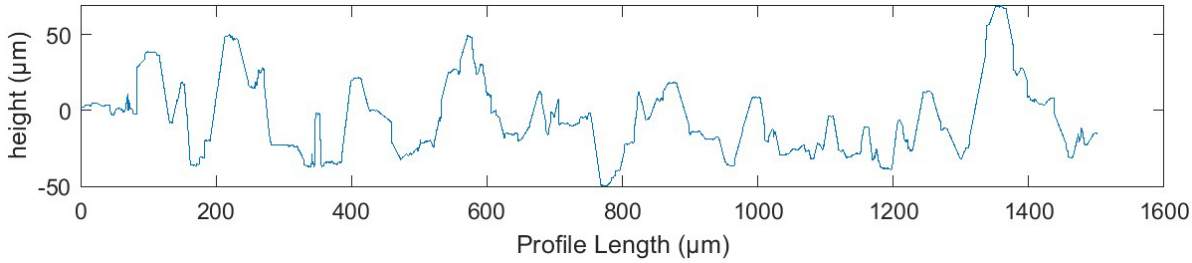

Figure Q5: Flattened selected line profile from SWLI dataset sample 024

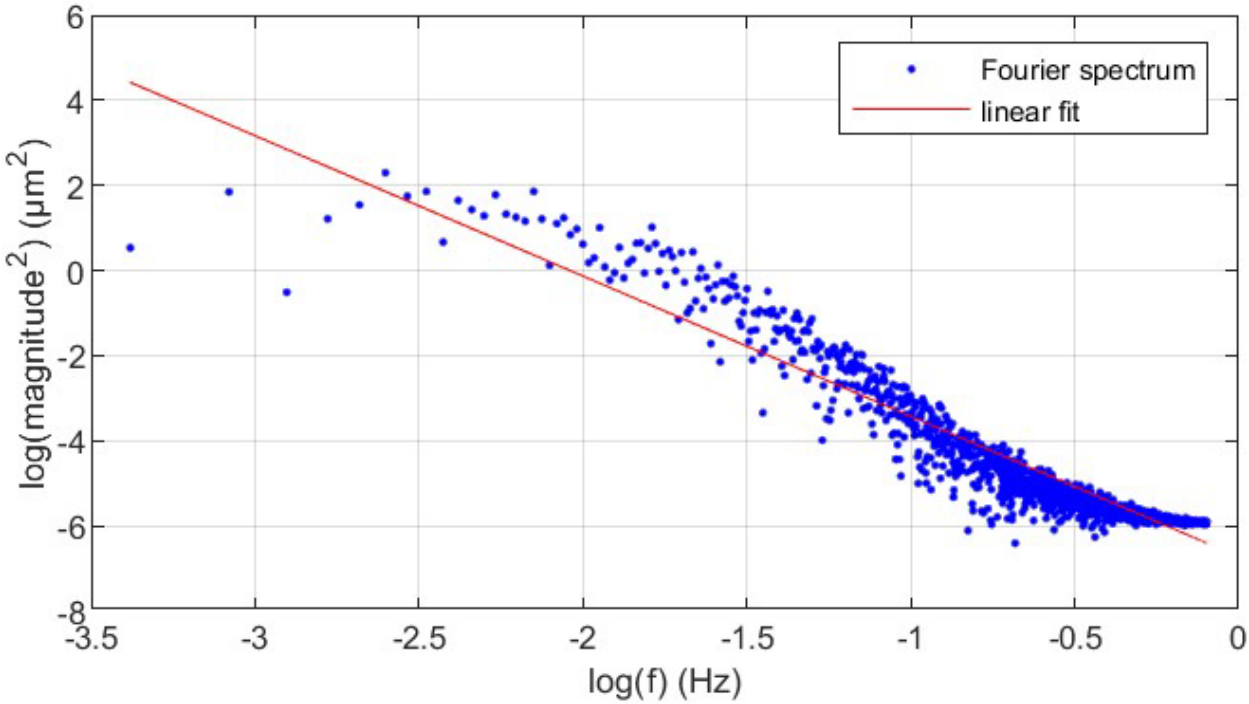

Figure Q6: Fourier analysis and linear fit from Dektak line profile data sample 024

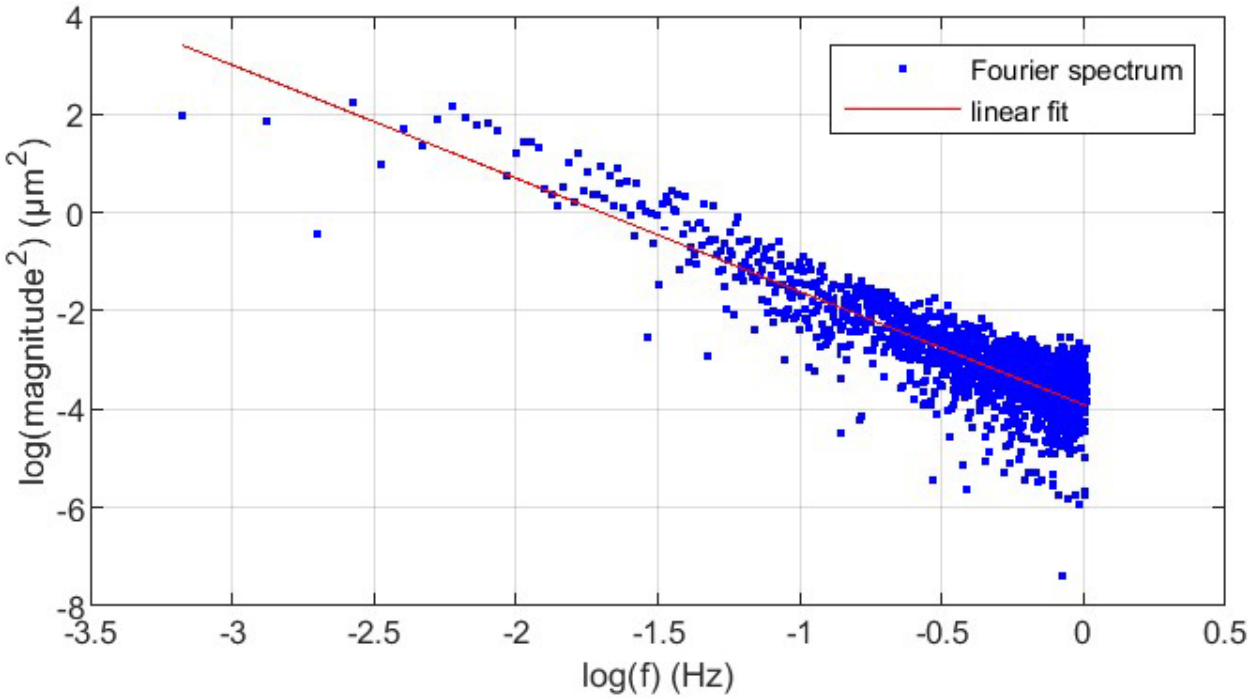

Figure Q7: Fourier analysis and linear fit from SWLI line profile data sample 024

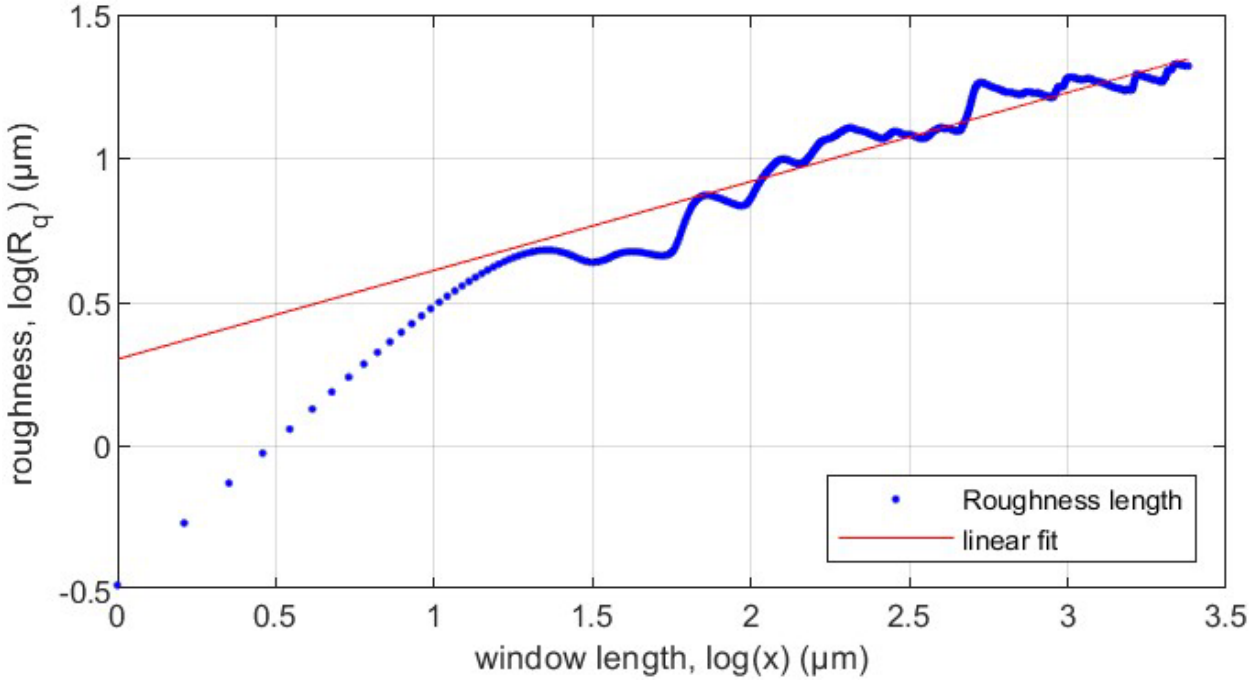

Figure Q8: Roughness length method for calculating *H* from Dektak profile sample 024

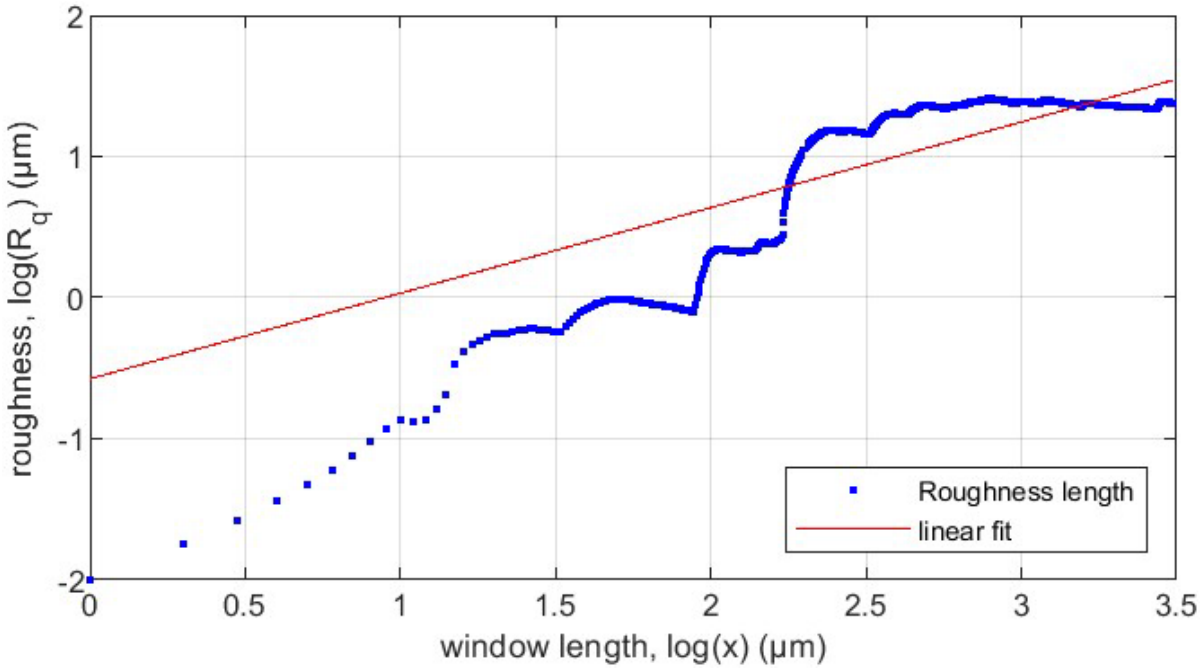

Figure Q9: Roughness length method for calculating *H* from SWLI profile sample 024

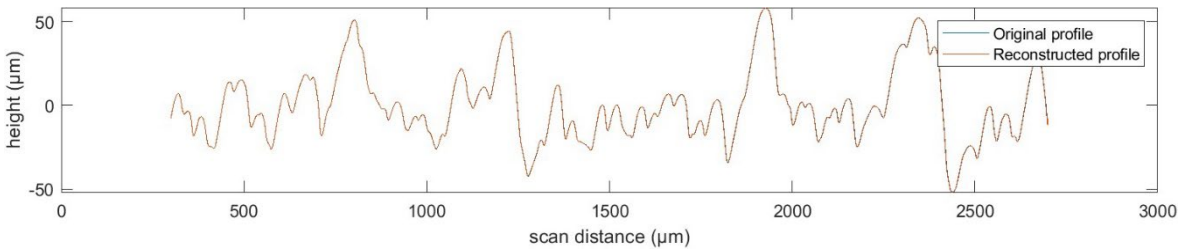

Figure Q10: Original Dektak line scan profile and Fourier recreated line profile for sample 024

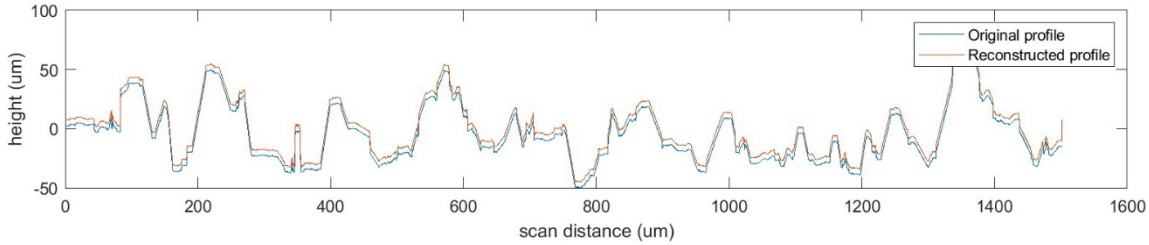

Figure Q11: Original SWLI line profile and Fourier recreated line profile for sample 024

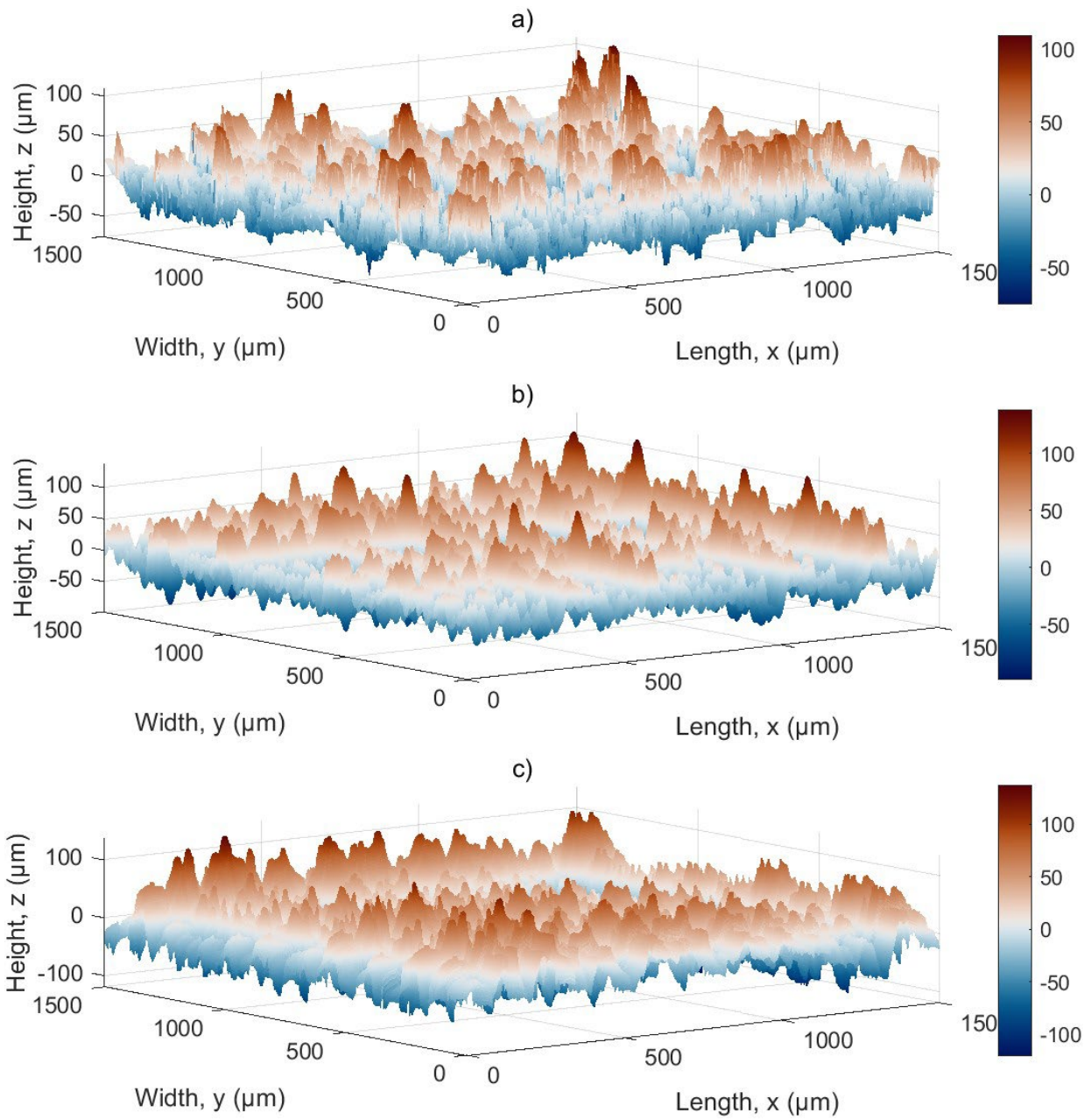

Figure Q12: SWLI scan of sample 024 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 024 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 024

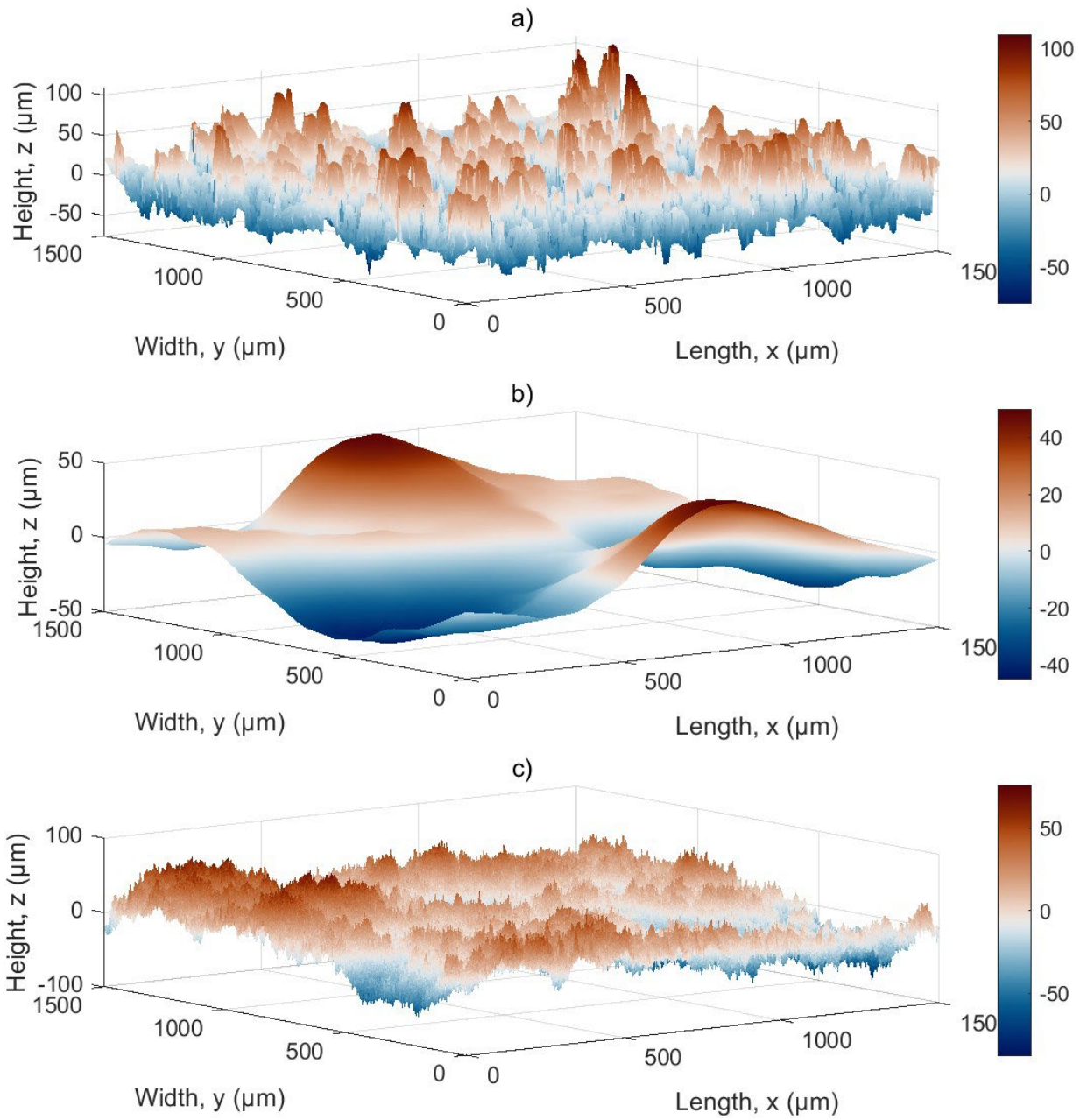

Figure Q13: SWLI scan of sample 024 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 024 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 024

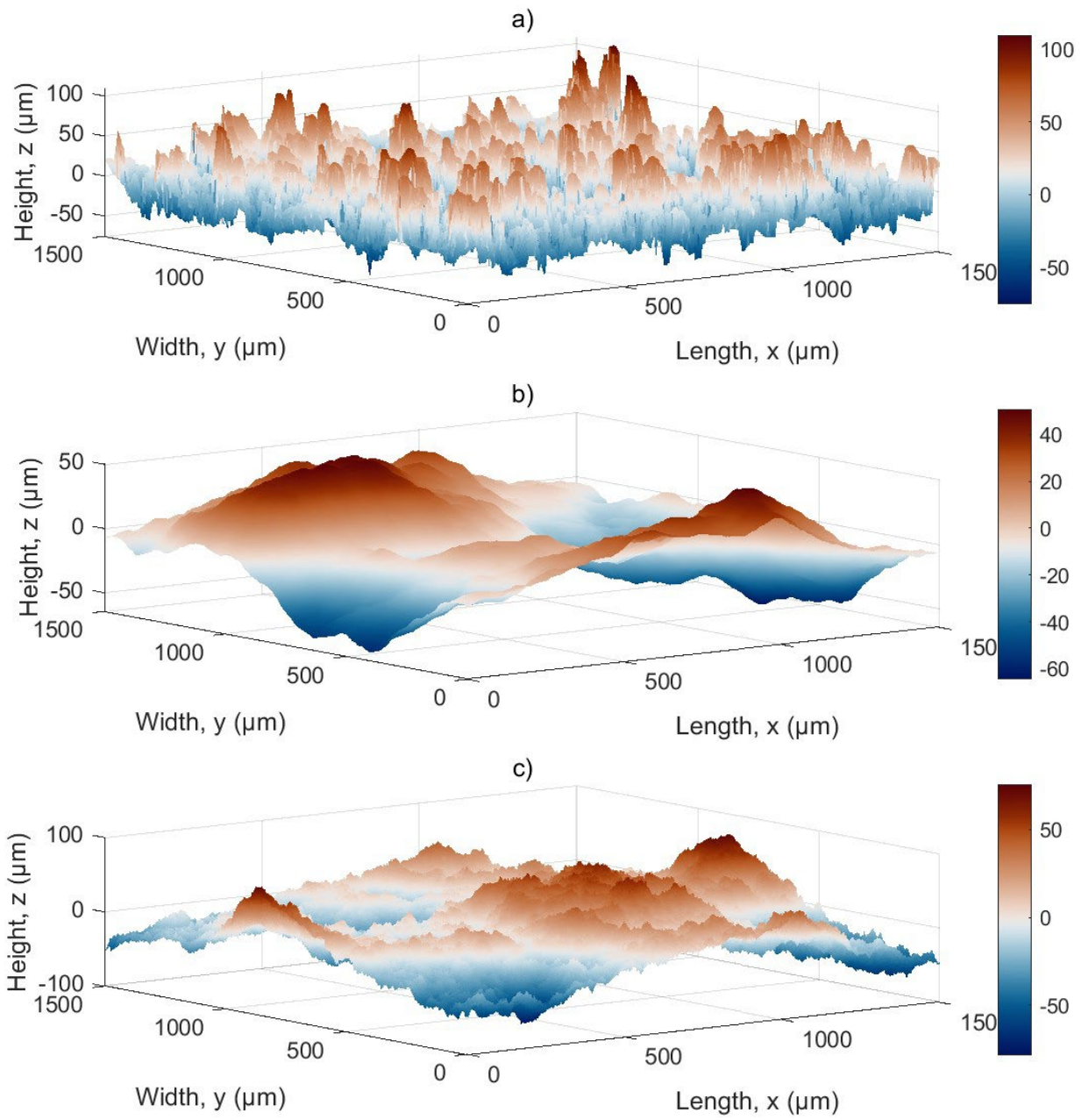

Figure Q14: SWLI scan of sample 024 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 024 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 024

Appendix R: Sample 025 supplementary figures

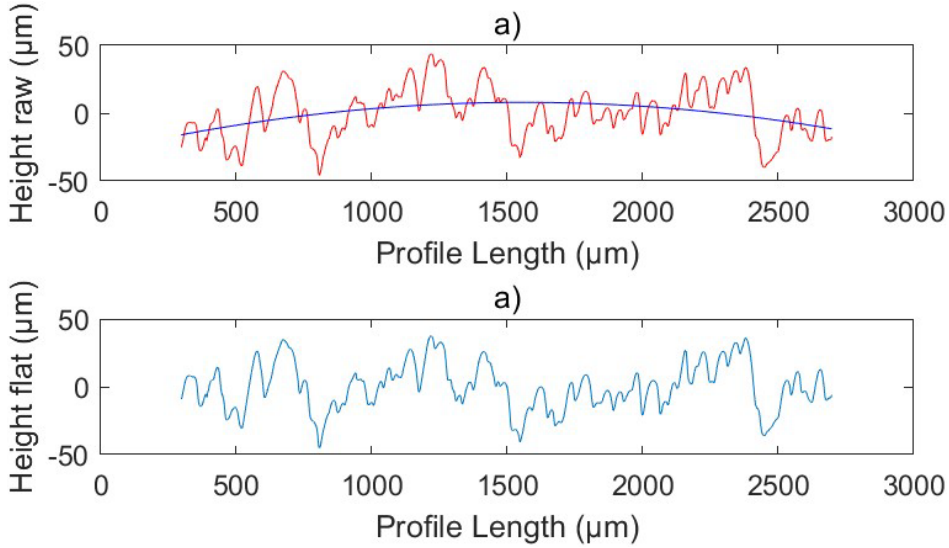

Figure R1: a) Raw profile measurement and fit curve sample 025 b) flattened line profile from sample 025

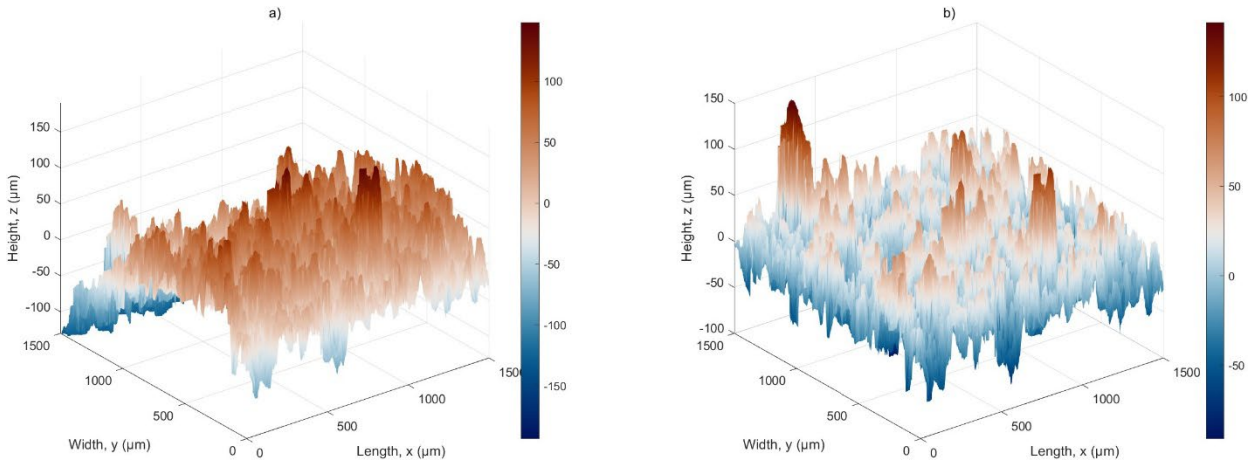

Figure R2: a) Unflattened SWLI raw surface scan sample 025 b) plane flattened SWLI surface 025

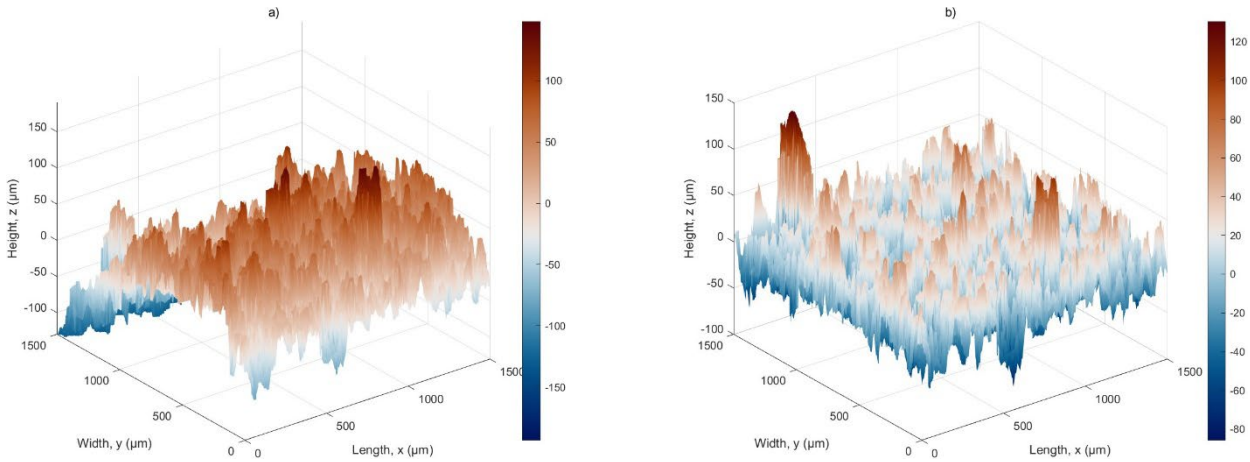

Figure R3: a) Unflattened SWLI raw surface scan sample 025 b) line by line flattened SWLI surface 025

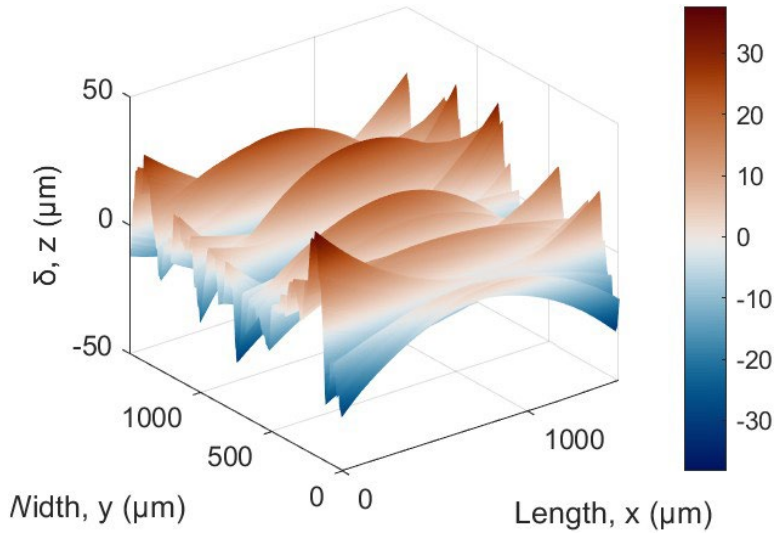

Figure R4: Difference between plane flattening and line by line flattening for sample 025

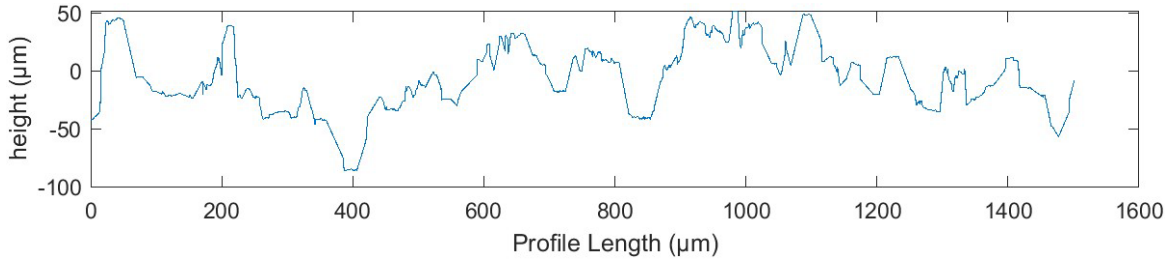

Figure R5: Flattened selected line profile from SWLI dataset sample 025

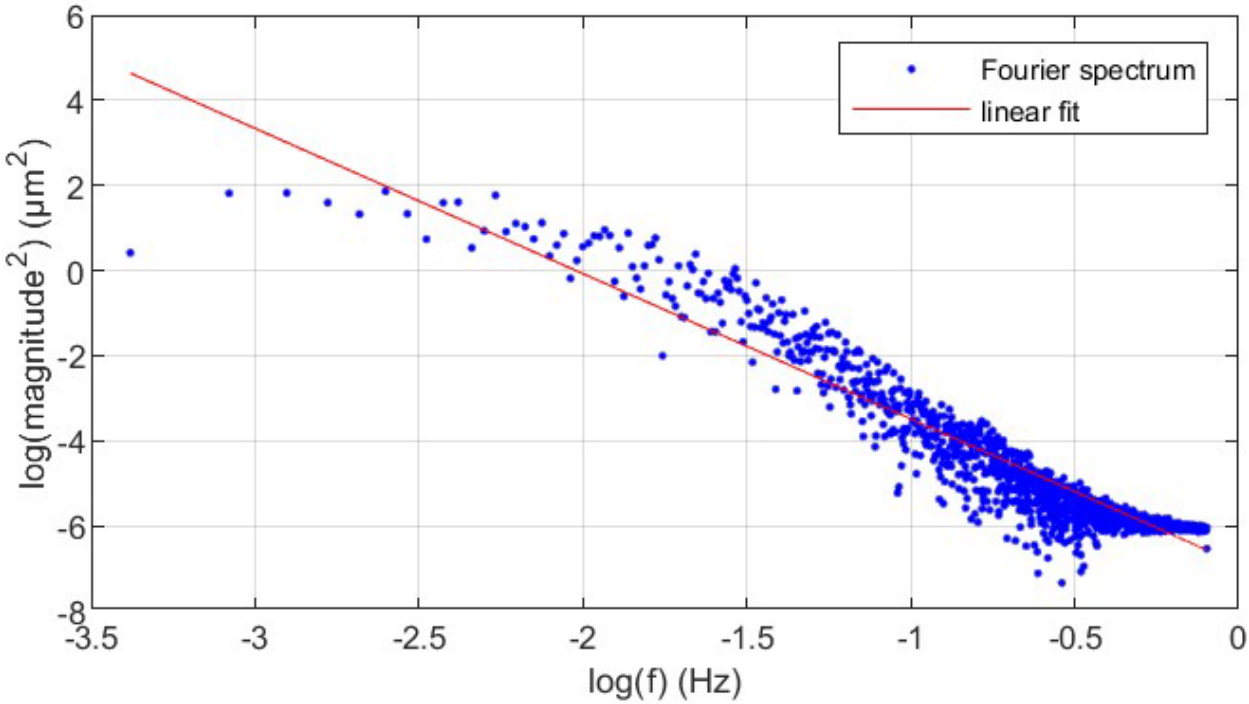

Figure R6: Fourier analysis and linear fit from Dektak line profile data sample 025

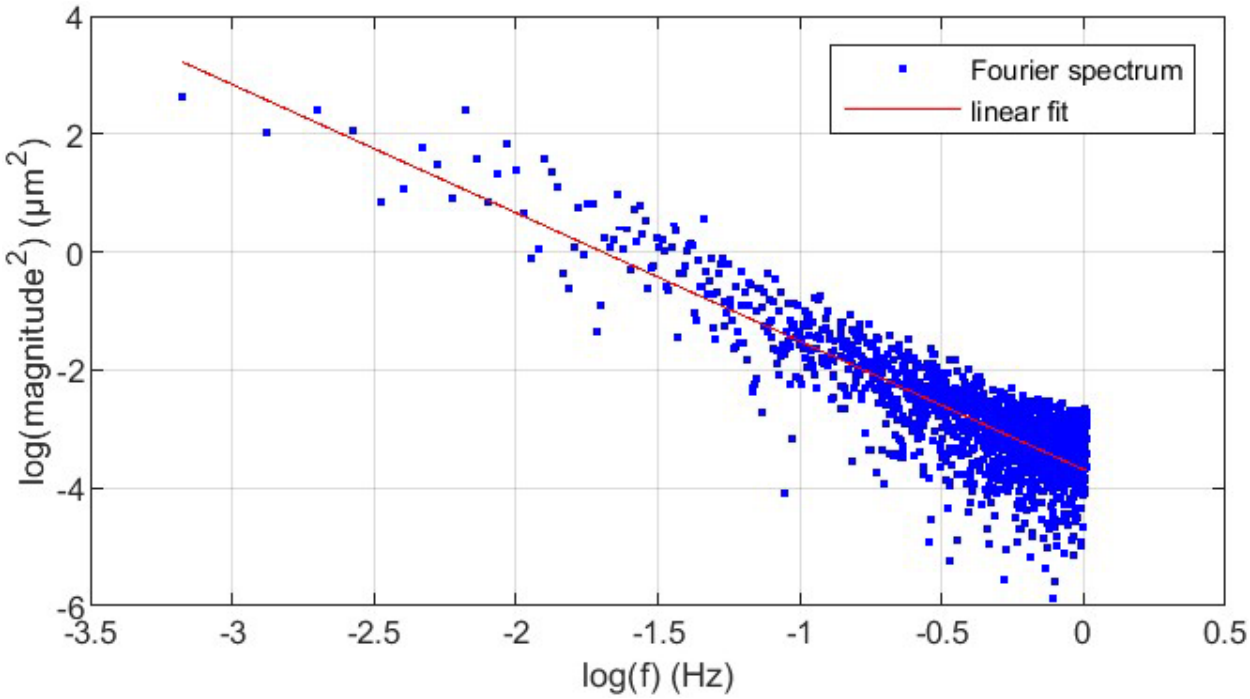

Figure R7: Fourier analysis and linear fit from SWLI line profile data sample 025

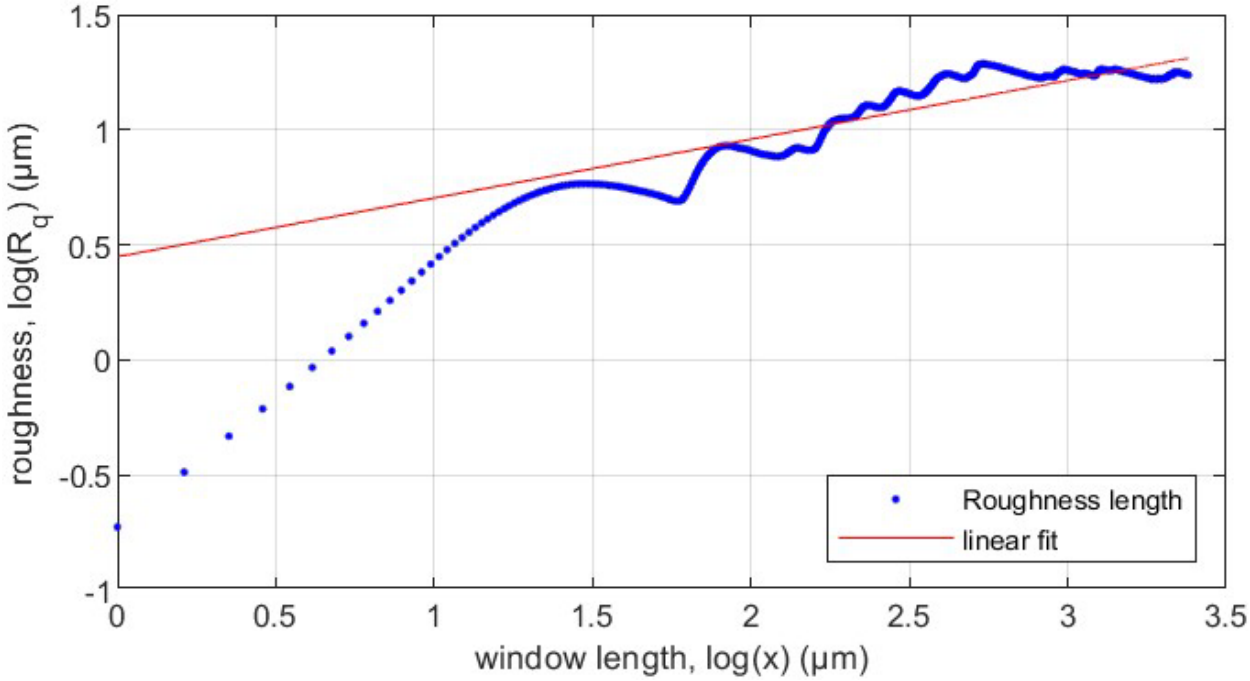

Figure R8: Roughness length method for calculating *H* from Dektak profile sample 025

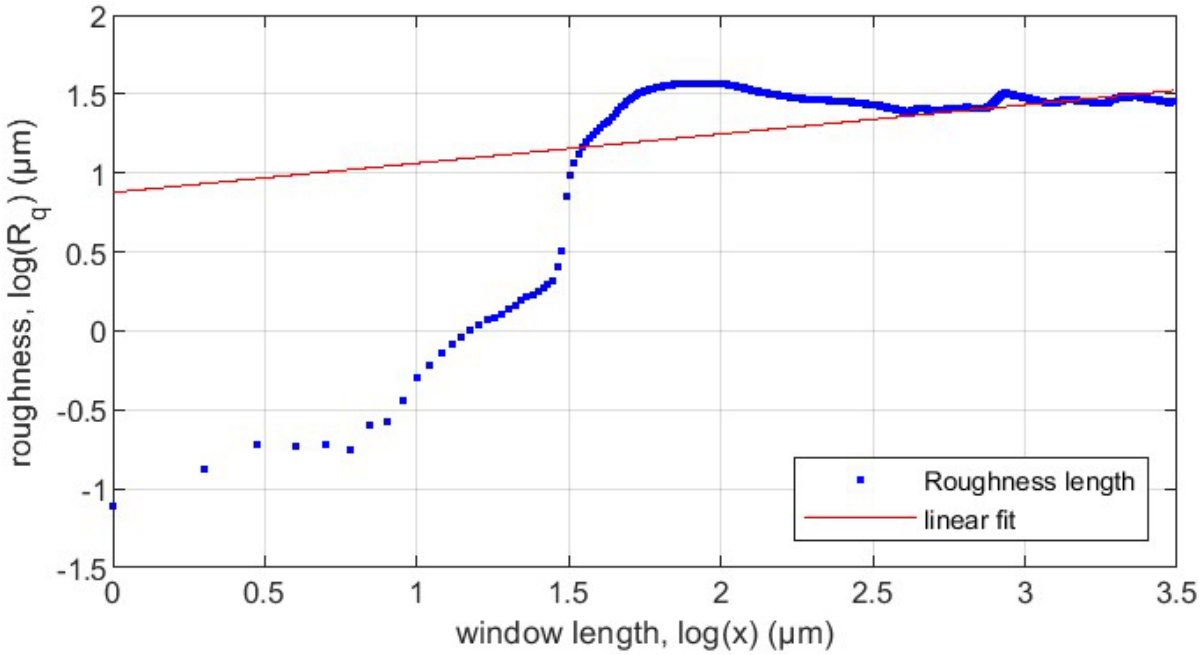

Figure R9: Roughness length method for calculating *H* from SWLI profile sample 025

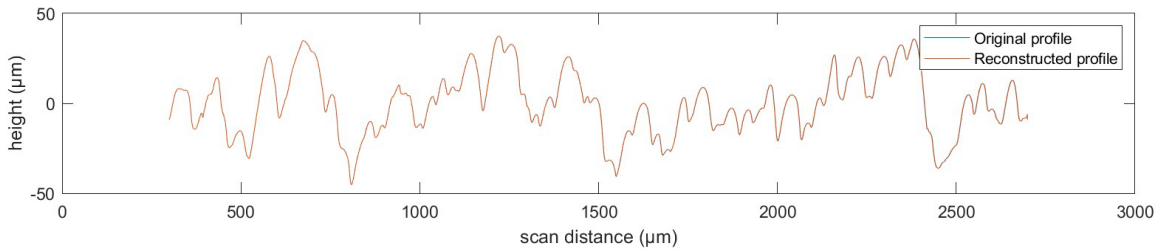

Figure R10: Original Dektak line scan profile and Fourier recreated line profile for sample 025

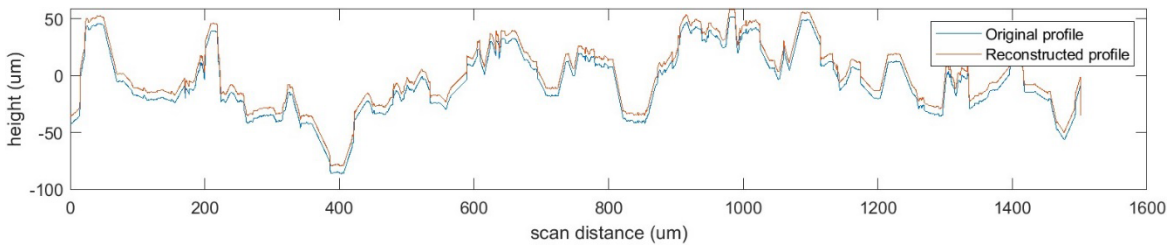

Figure R11: Original SWLI line profile and Fourier recreated line profile for sample 025

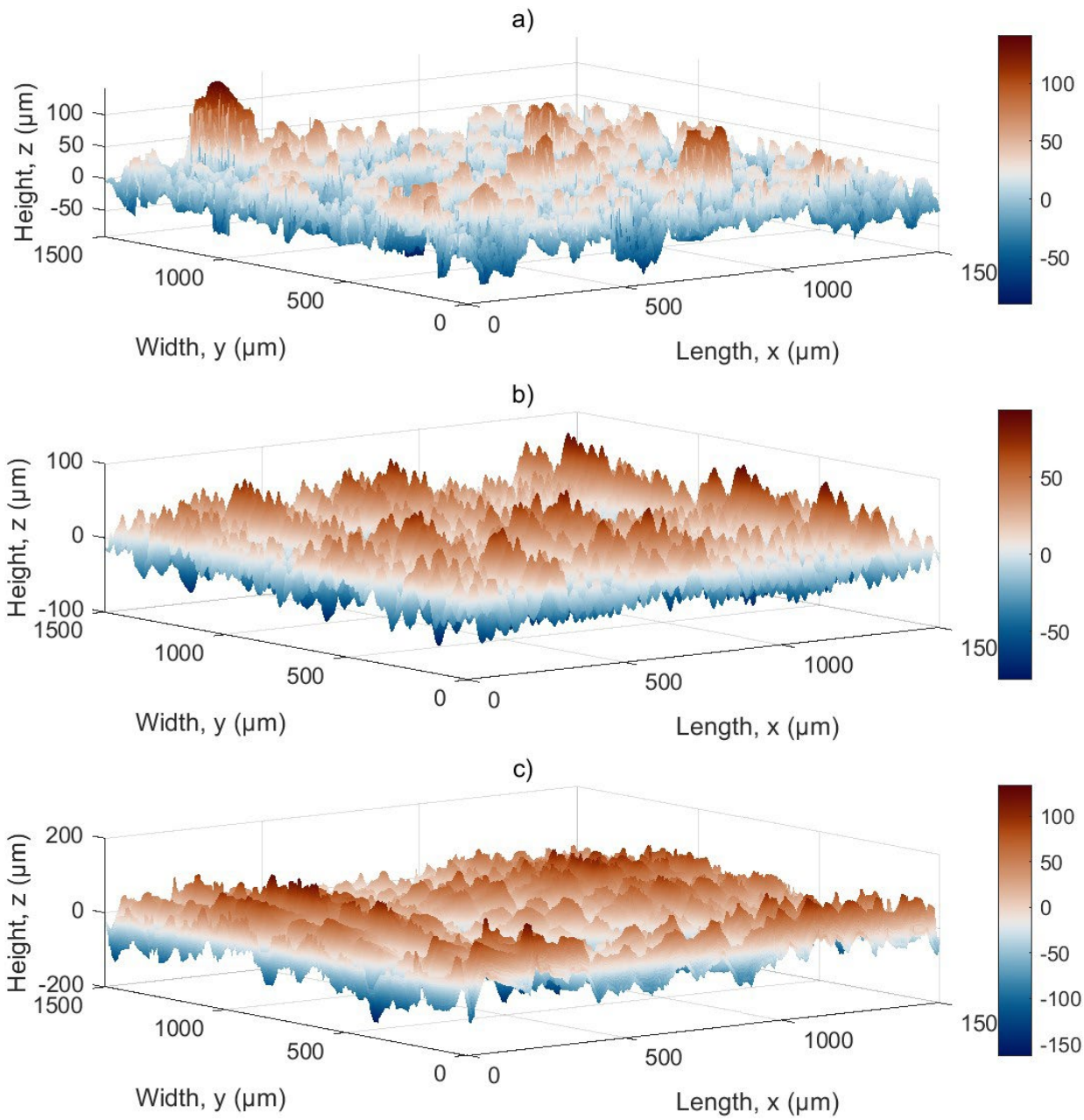

Figure R12: SWLI scan of sample 025 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 025 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 025

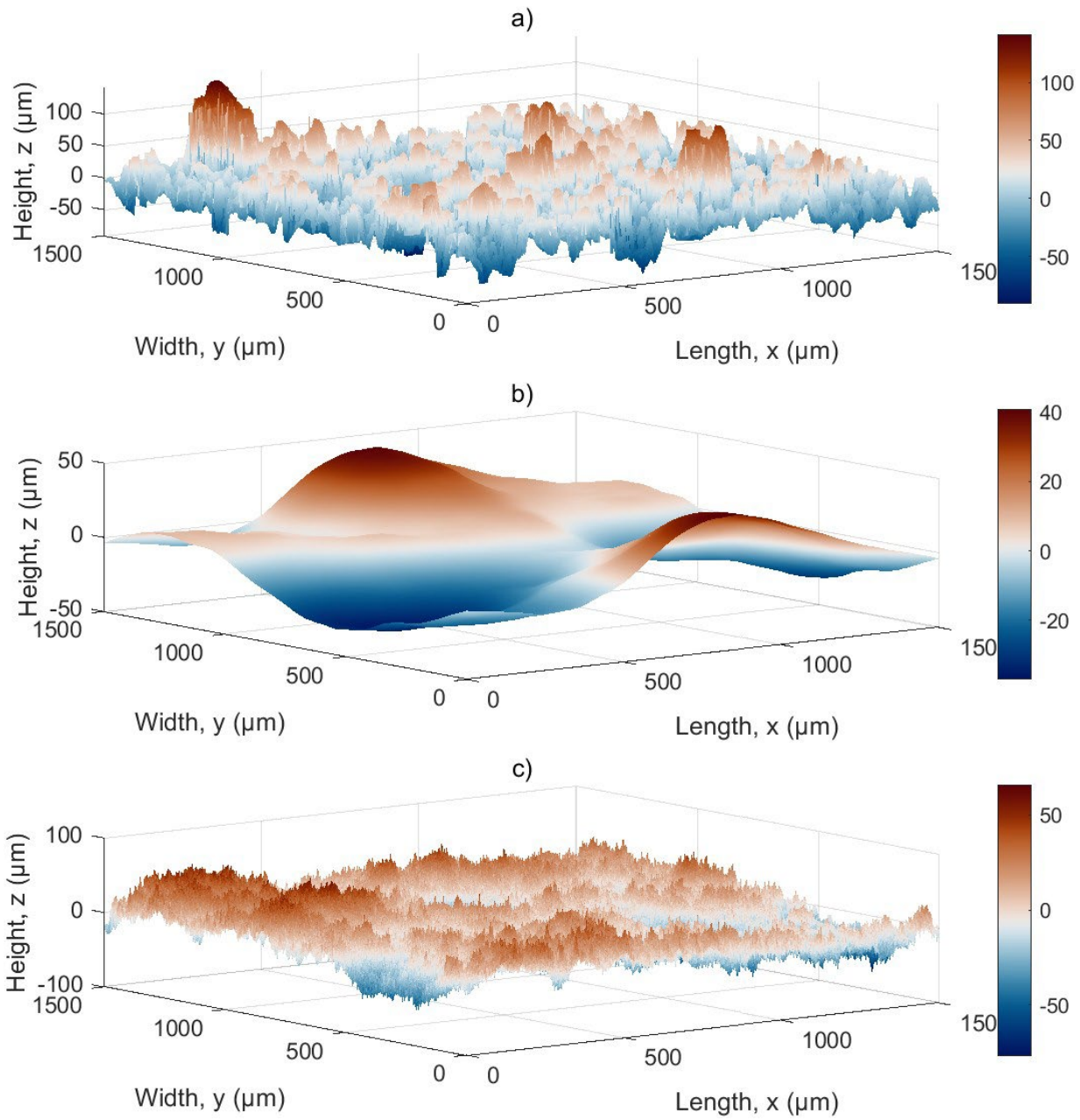

Figure R13: SWLI scan of sample 025 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 025 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 025

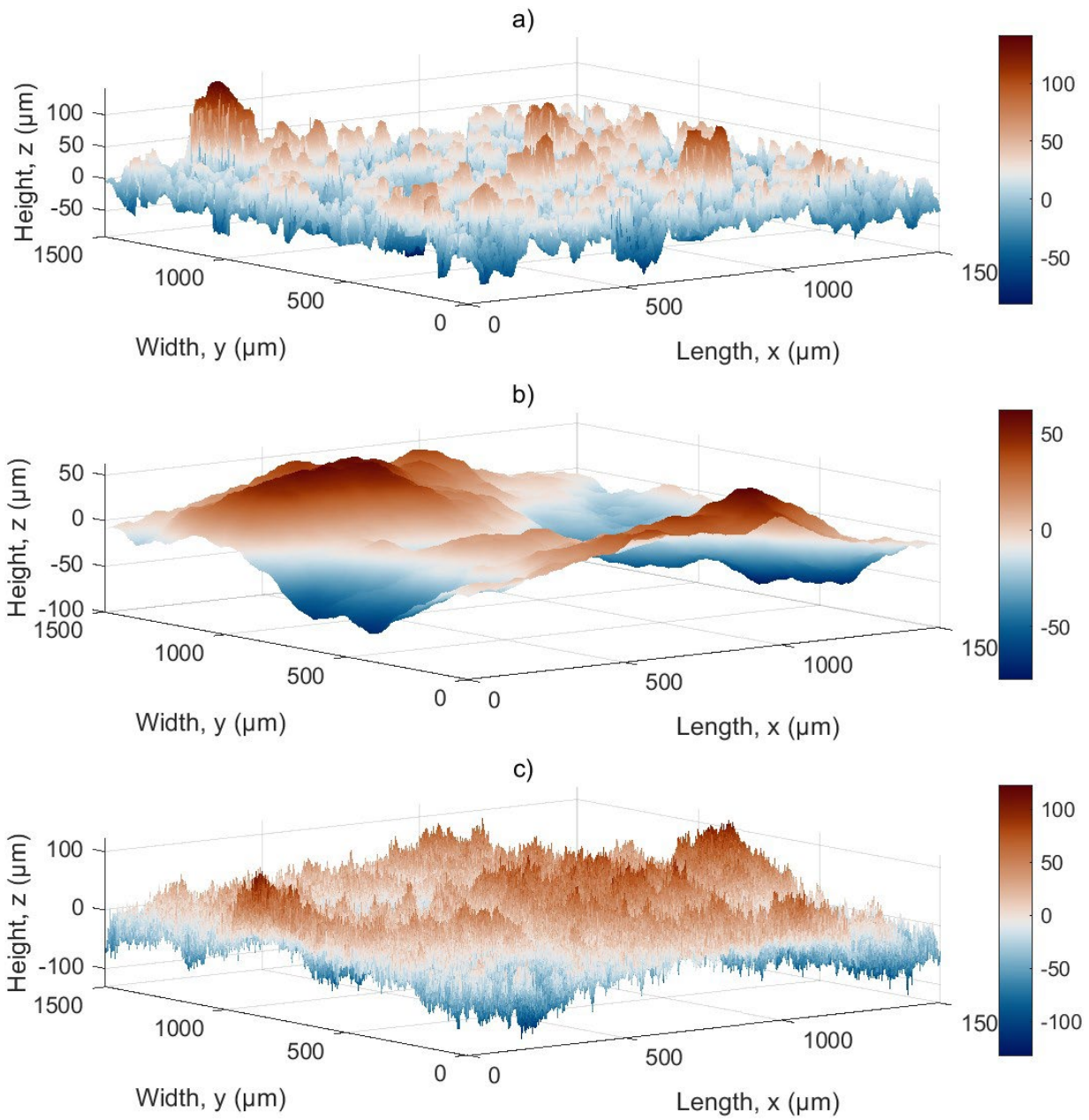

Figure R14: SWLI scan of sample 025 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 025 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 025

Appendix S: Sample 031 supplementary figures

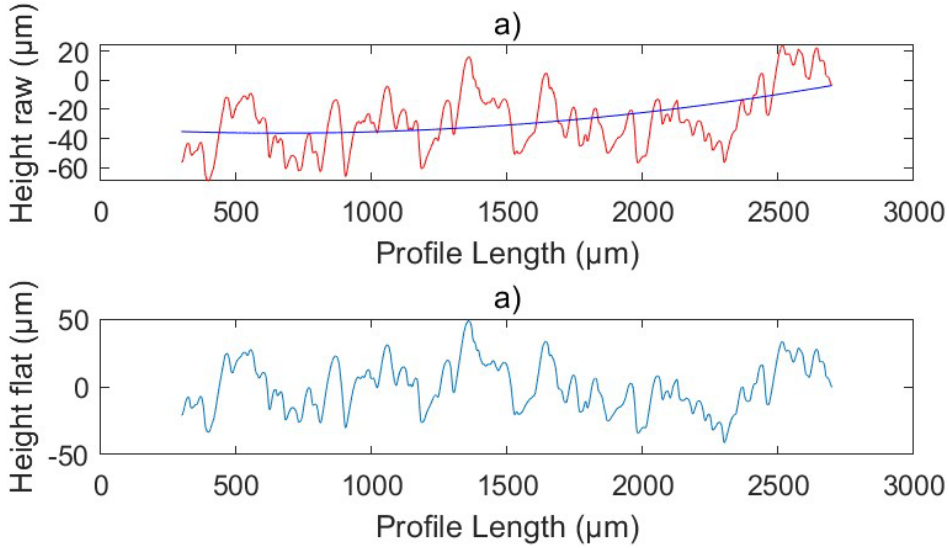

Figure S1: a) Raw profile measurement and fit curve sample 031 b) flattened line profile from sample 031

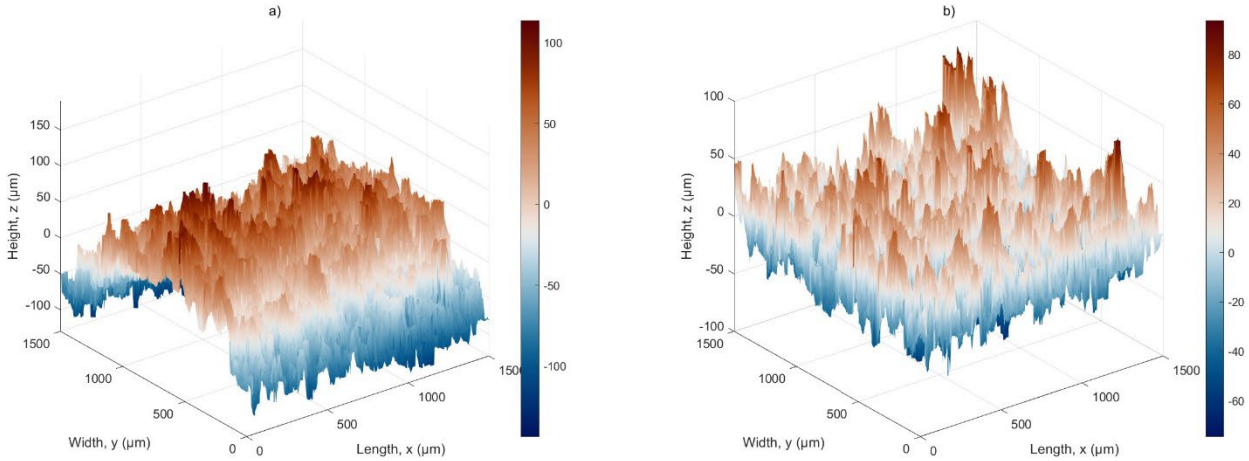

Figure S2: a) Unflattened SWLI raw surface scan sample 031 b) plane flattened SWLI surface 031

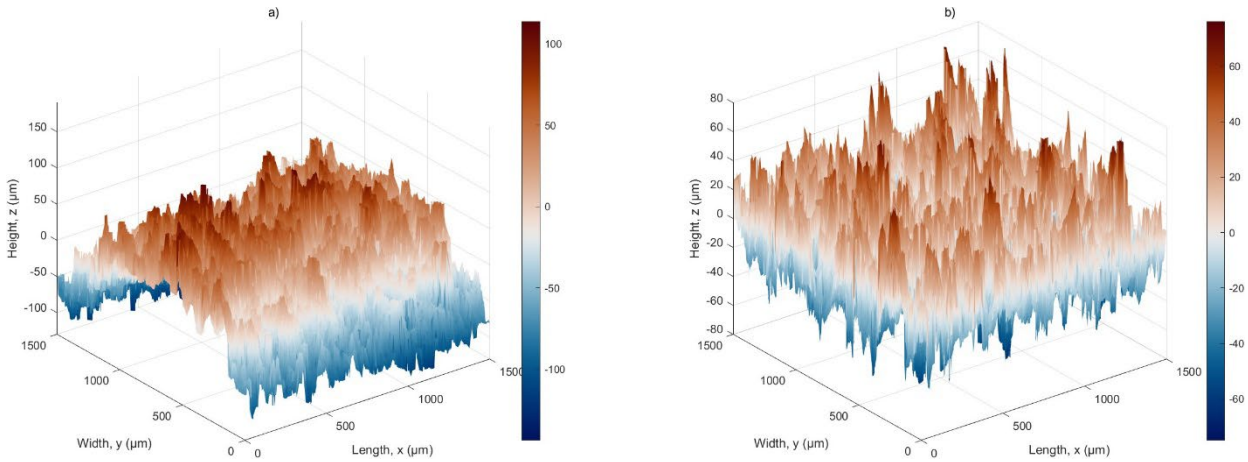

Figure S3: a) Unflattened SWLI raw surface scan sample 031 b) line by line flattened SWLI surface 031

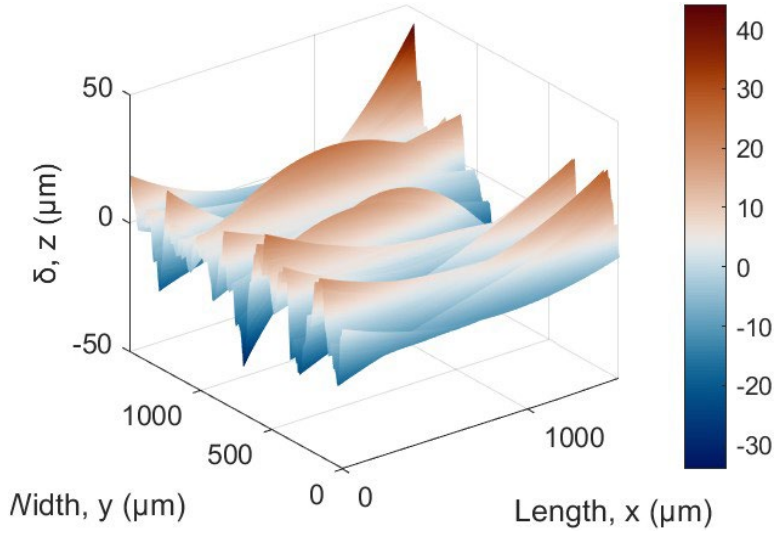

Figure S4: Difference between plane flattening and line by line flattening for sample 031

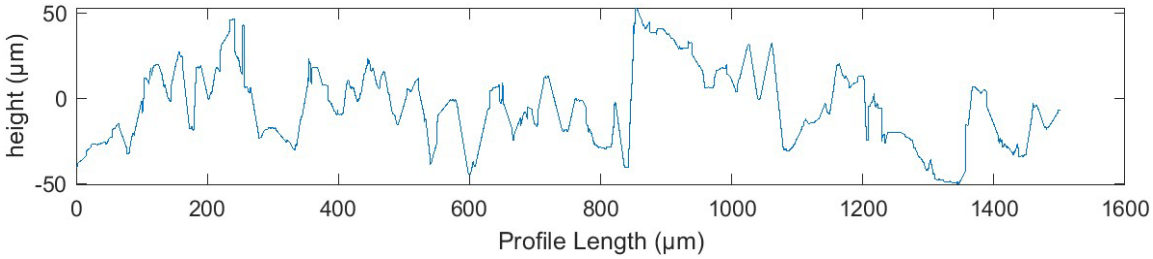

Figure S5: Flattened selected line profile from SWLI dataset sample 031

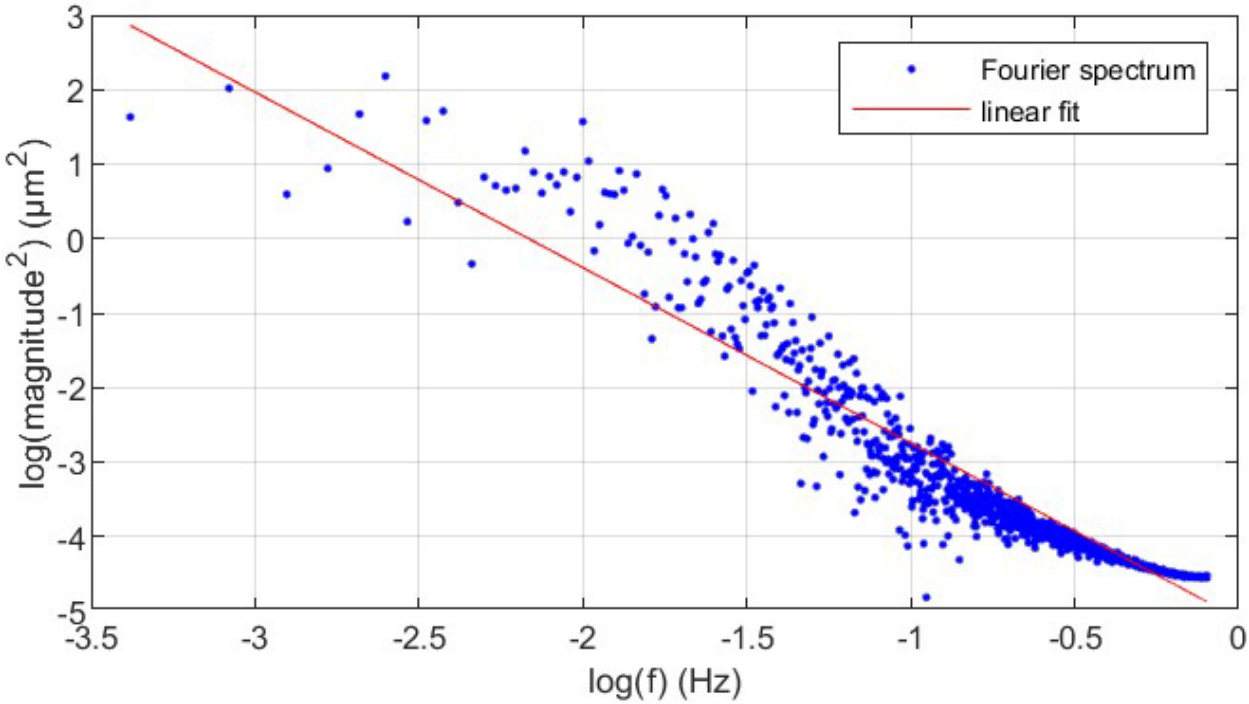

Figure S6: Fourier analysis and linear fit from Dektak line profile data sample 031

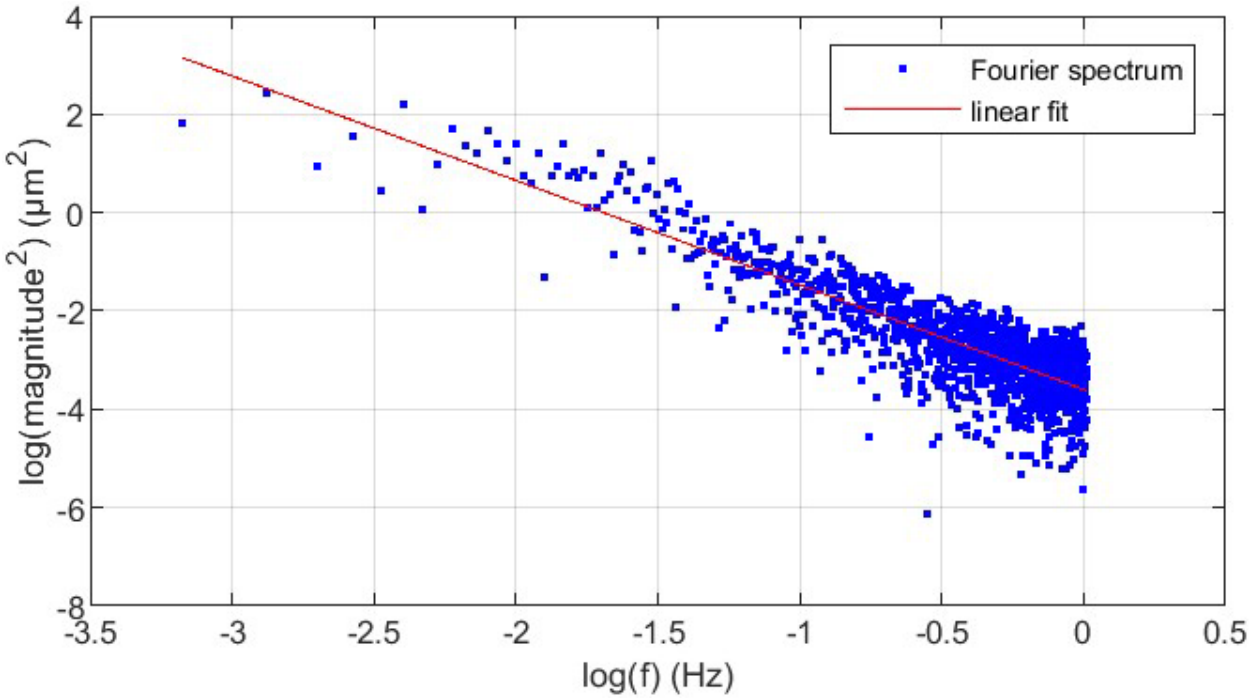

Figure S7: Fourier analysis and linear fit from SWLI line profile data sample 031

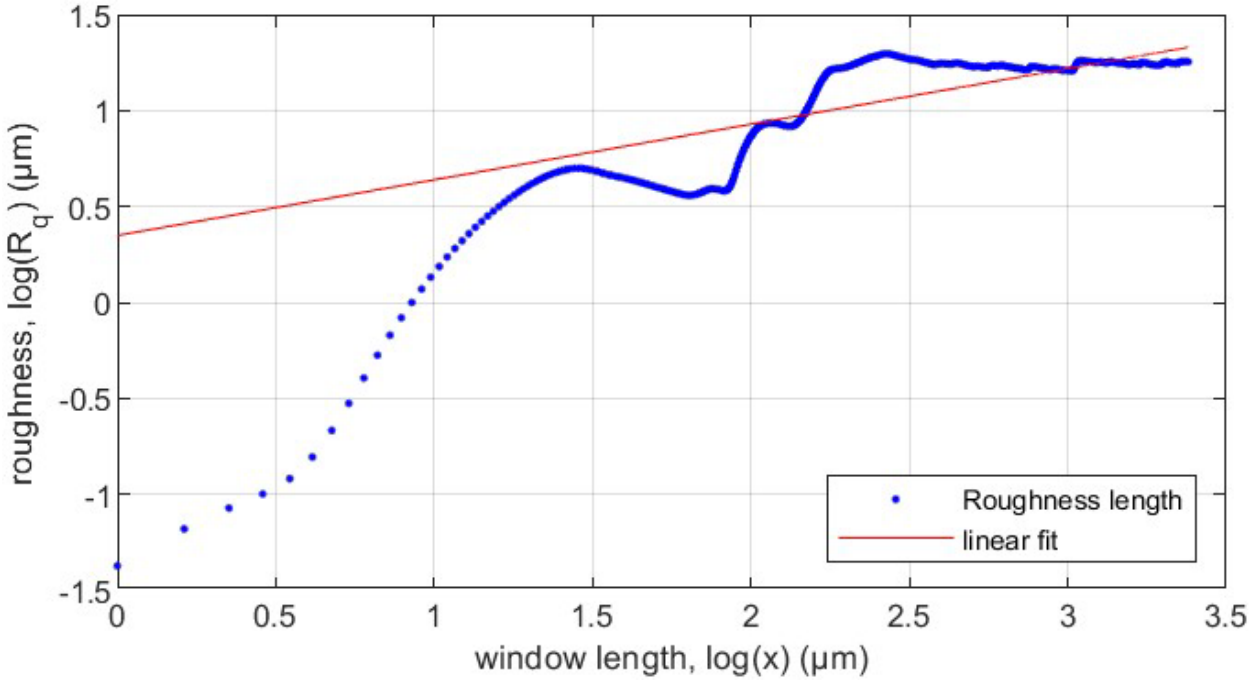

Figure S8: Roughness length method for calculating *H* from Dektak profile sample 031

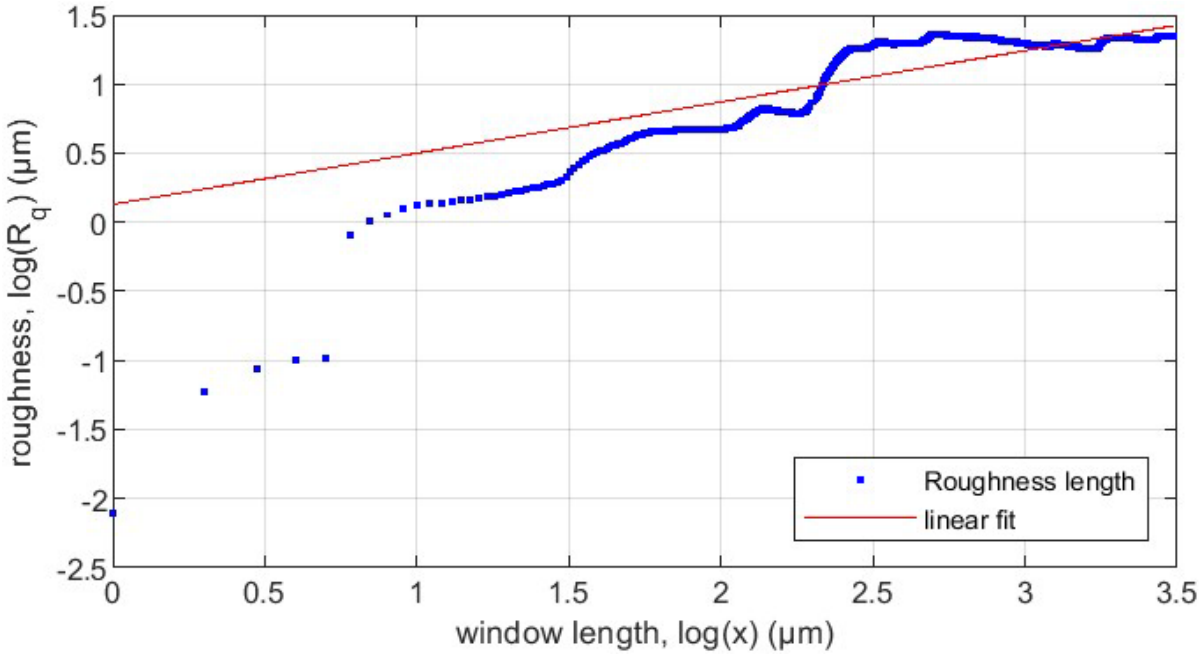

Figure S9: Roughness length method for calculating *H* from SWLI profile sample 031

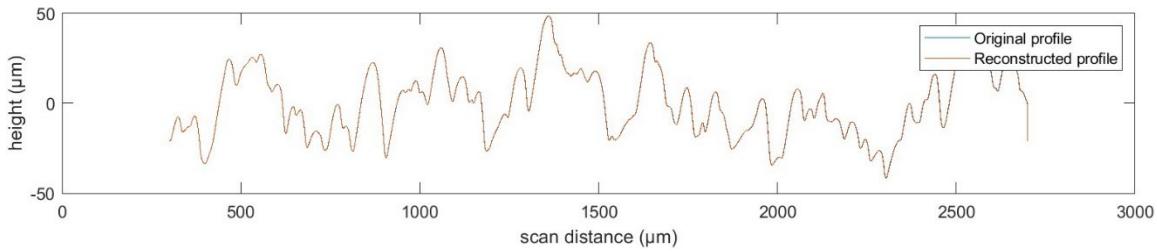

Figure S10: Original Dektak line scan profile and Fourier recreated line profile for sample 031

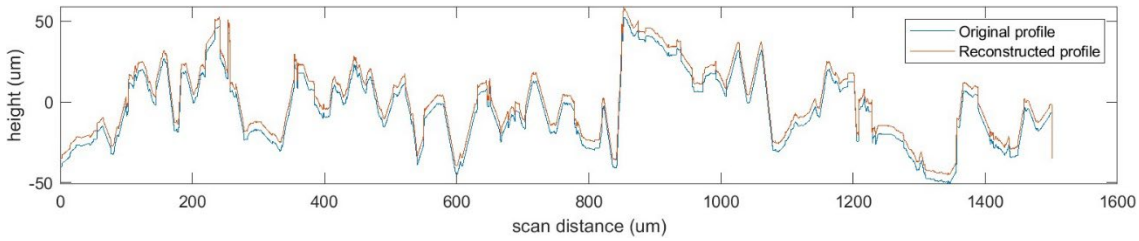

Figure S11: Original SWLI line profile and Fourier recreated line profile for sample 031

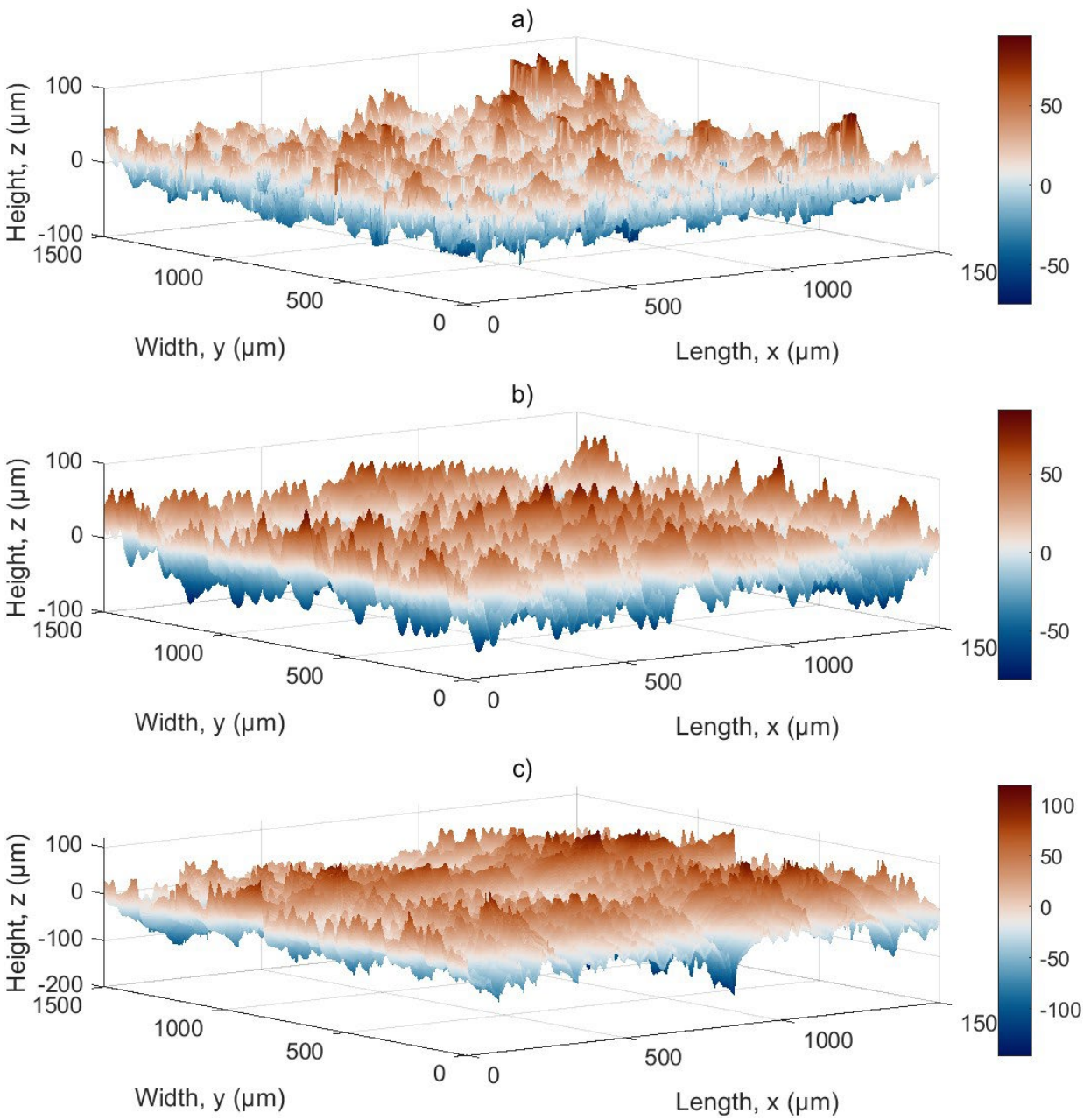

Figure S12: SWLI scan of sample 031 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 031 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 031

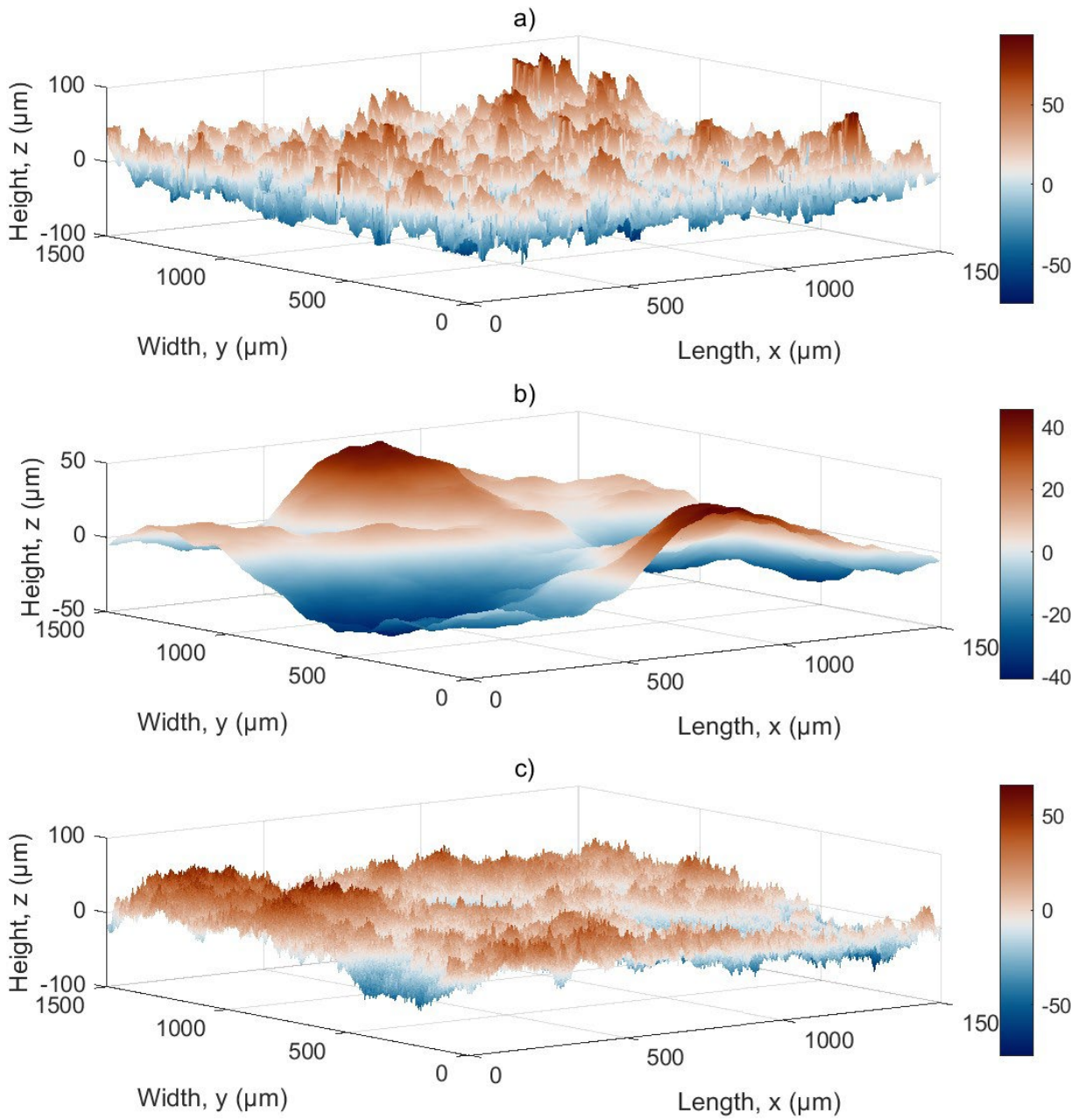

Figure S13: SWLI scan of sample 031 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 031 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 031

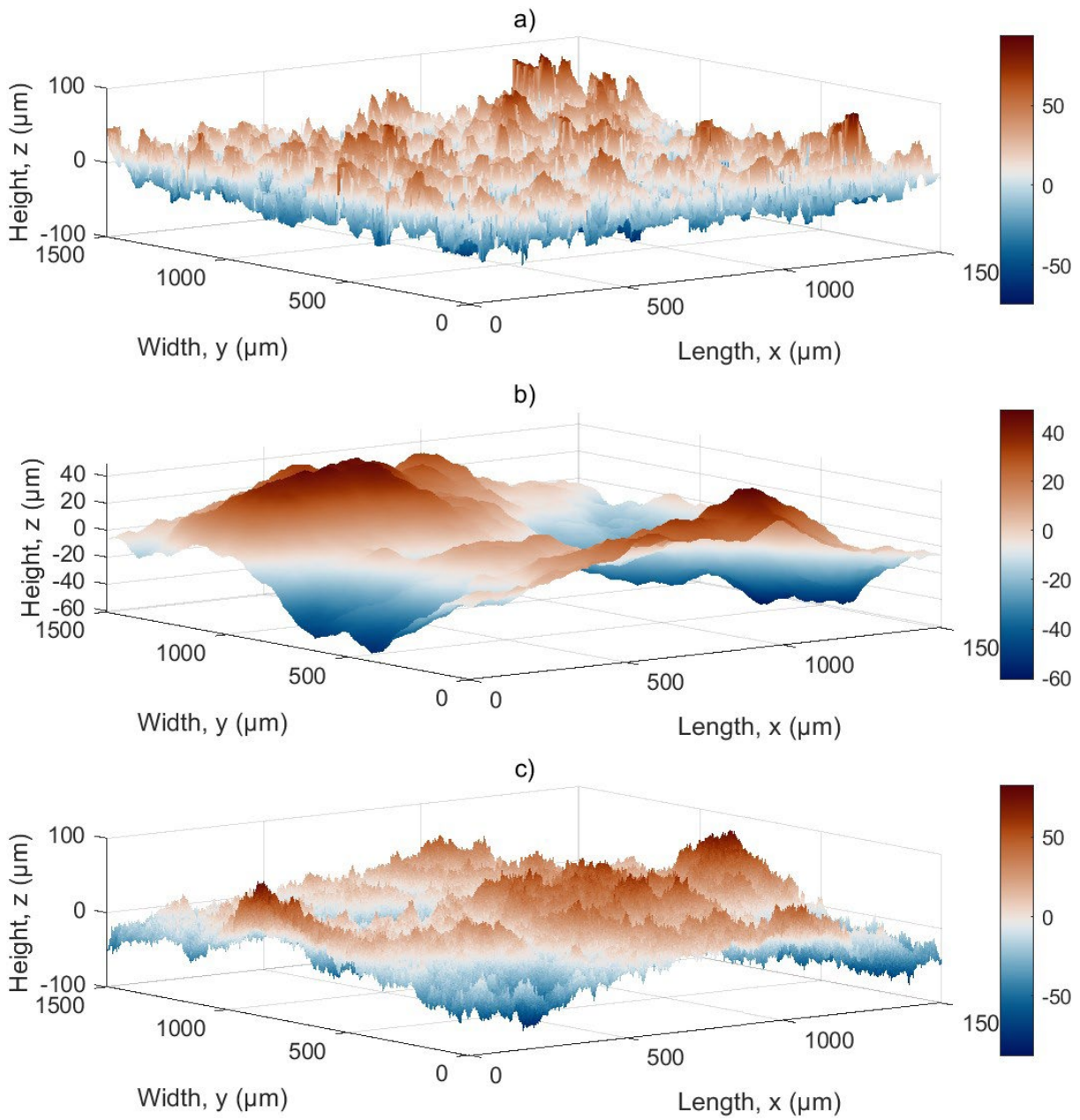

Figure S14: SWLI scan of sample 031 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 031 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 031

Appendix T: Sample 032 supplementary figures

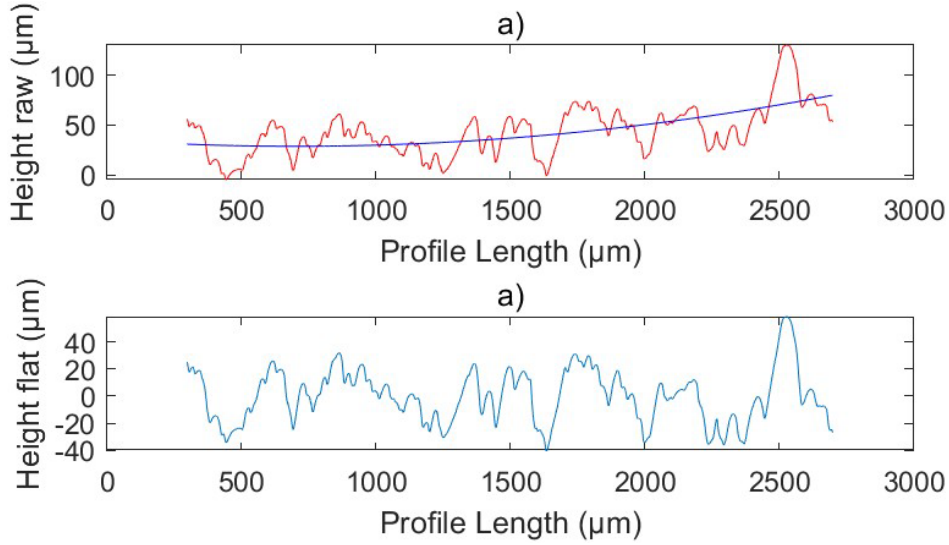

Figure T1: a) Raw profile measurement and fit curve sample 032 b) flattened line profile from sample 032

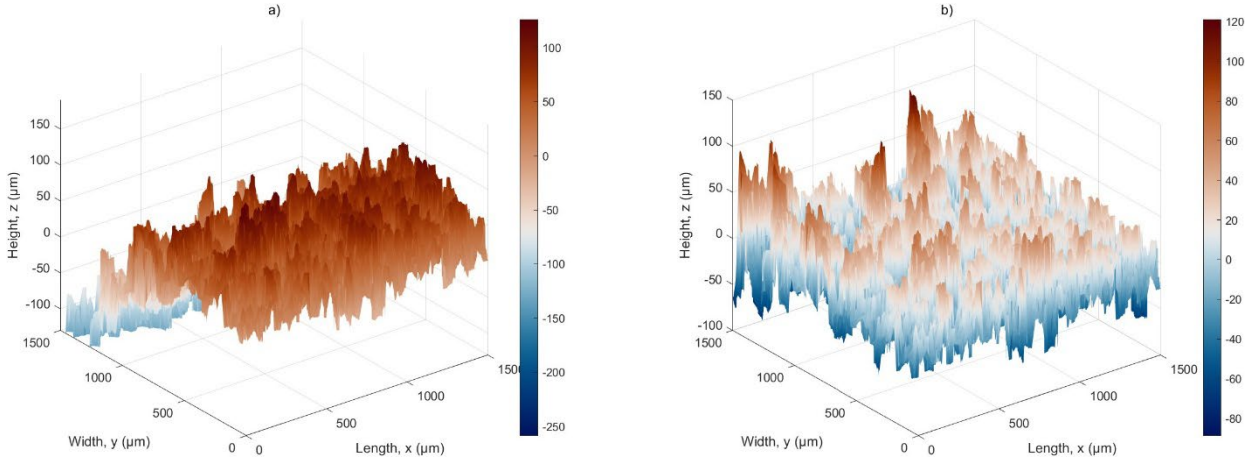

Figure T2: a) Unflattened SWLI raw surface scan sample 032 b) plane flattened SWLI surface 032

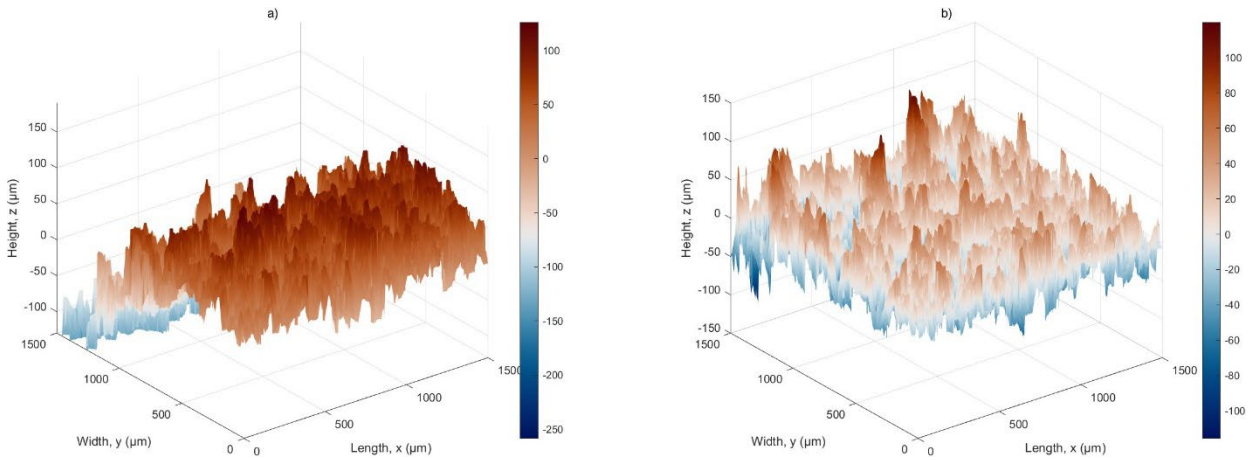

Figure T3: a) Unflattened SWLI raw surface scan sample 032 b) line by line flattened SWLI surface 032

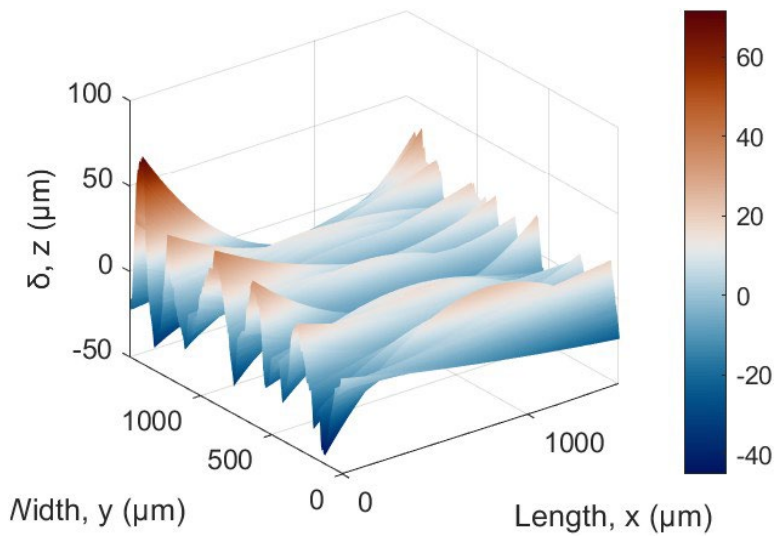

Figure T4: Difference between plane flattening and line by line flattening for sample 032

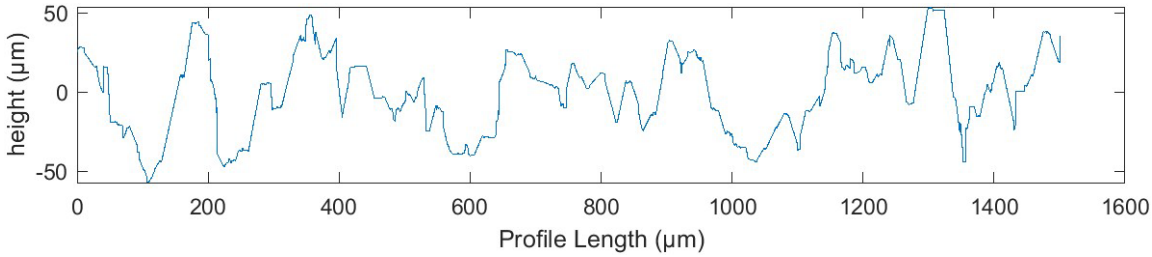

Figure T5: Flattened selected line profile from SWLI dataset sample 032

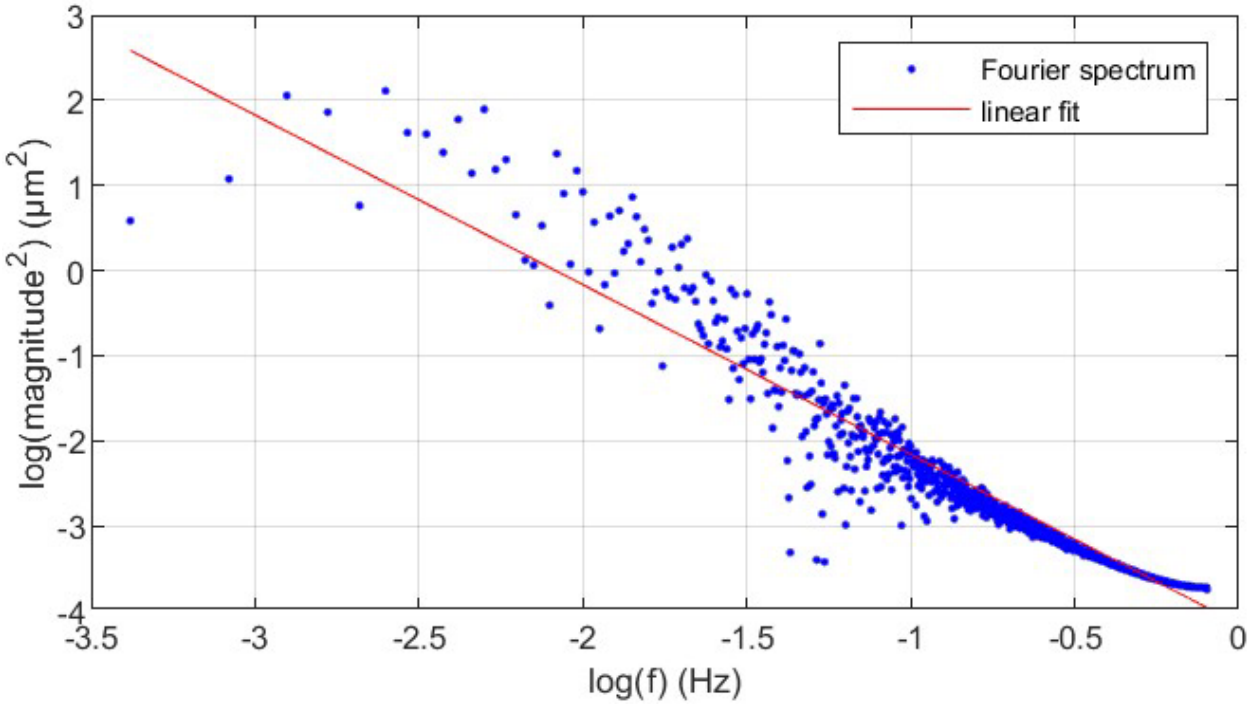

Figure T6: Fourier analysis and linear fit from Dektak line profile data sample 032

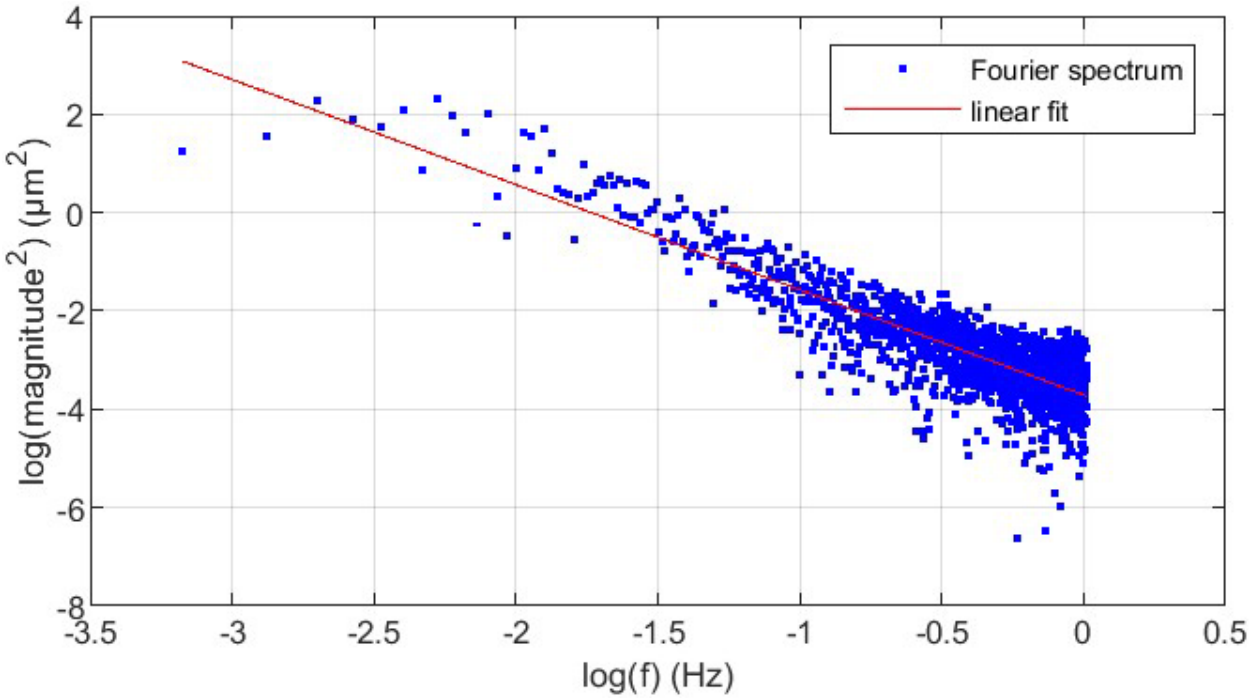

Figure T7: Fourier analysis and linear fit from SWLI line profile data sample 032

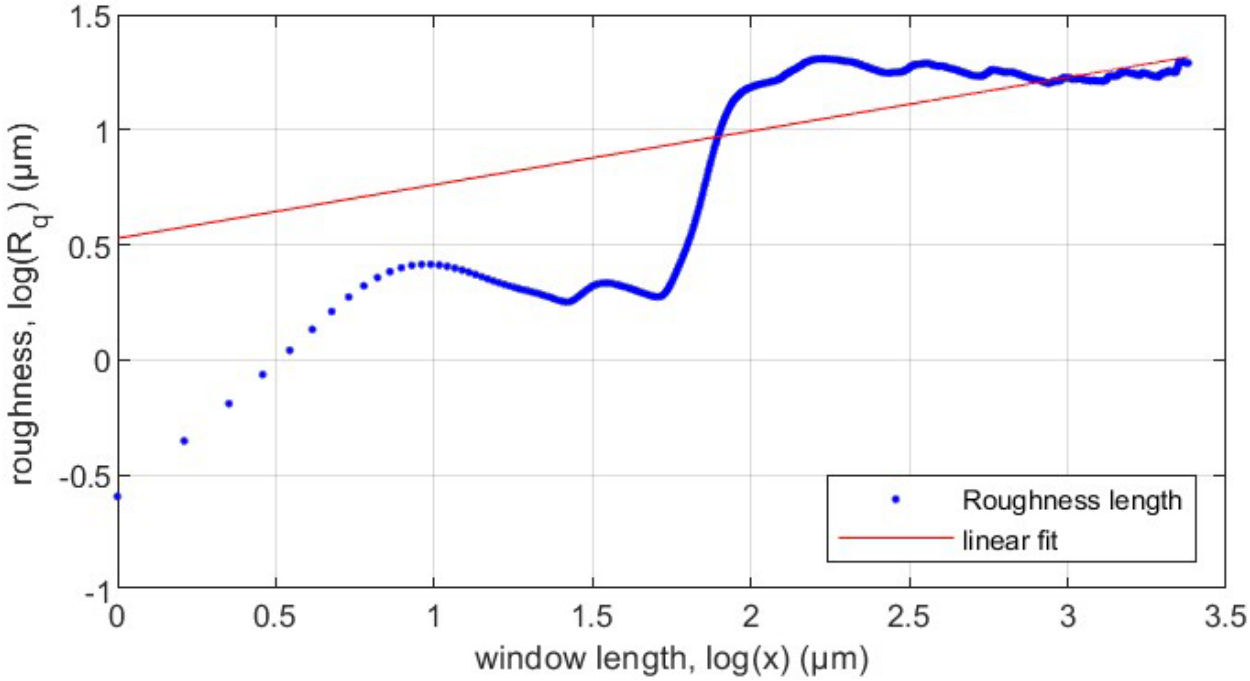

Figure T8: Roughness length method for calculating *H* from Dektak profile sample 032

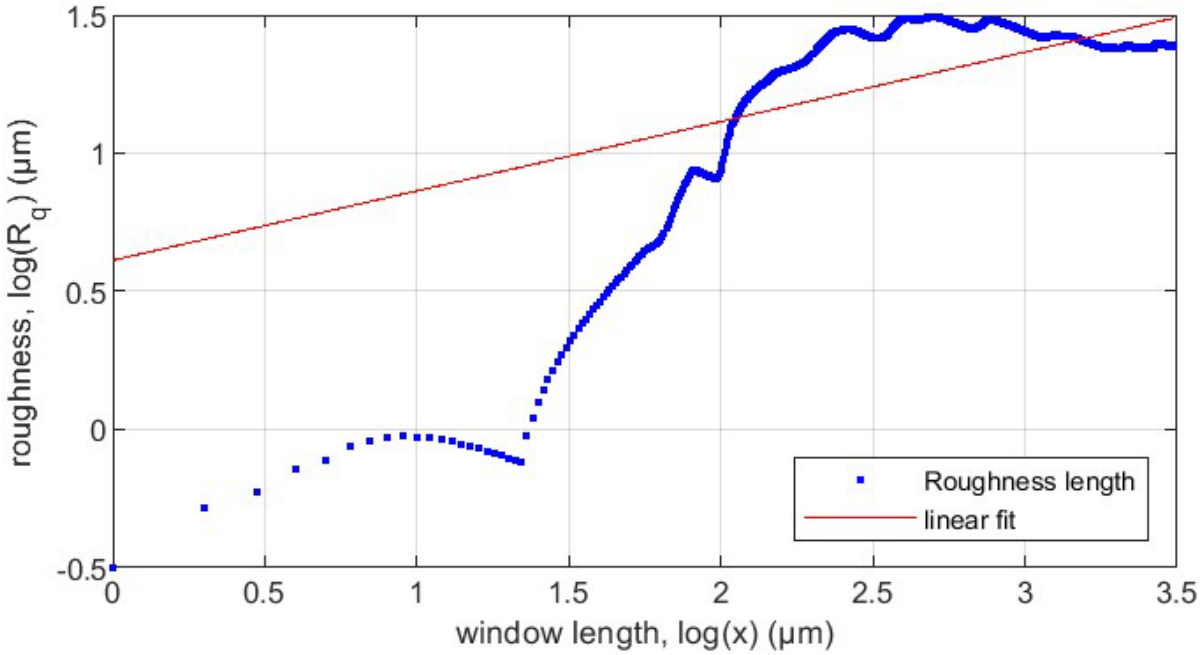

Figure T9: Roughness length method for calculating *H* from SWLI profile sample 032

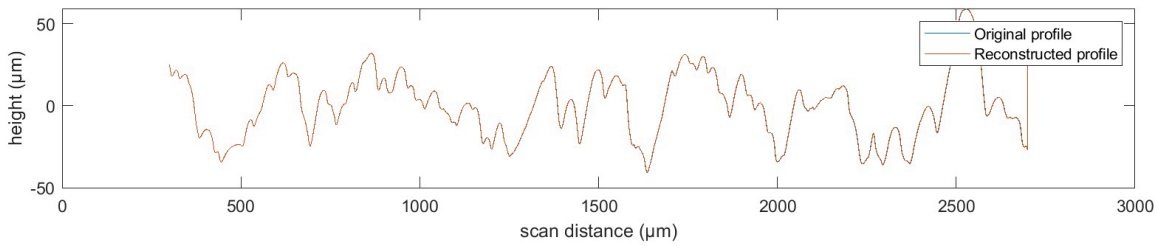

Figure T10: Original Dektak line scan profile and Fourier recreated line profile for sample 032

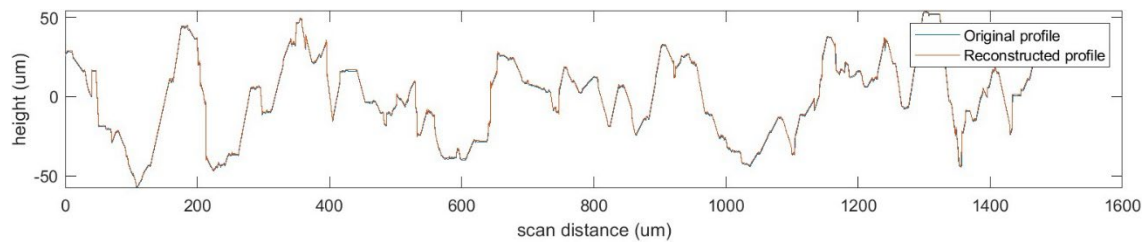

Figure T11: Original SWLI line profile and Fourier recreated line profile for sample 032

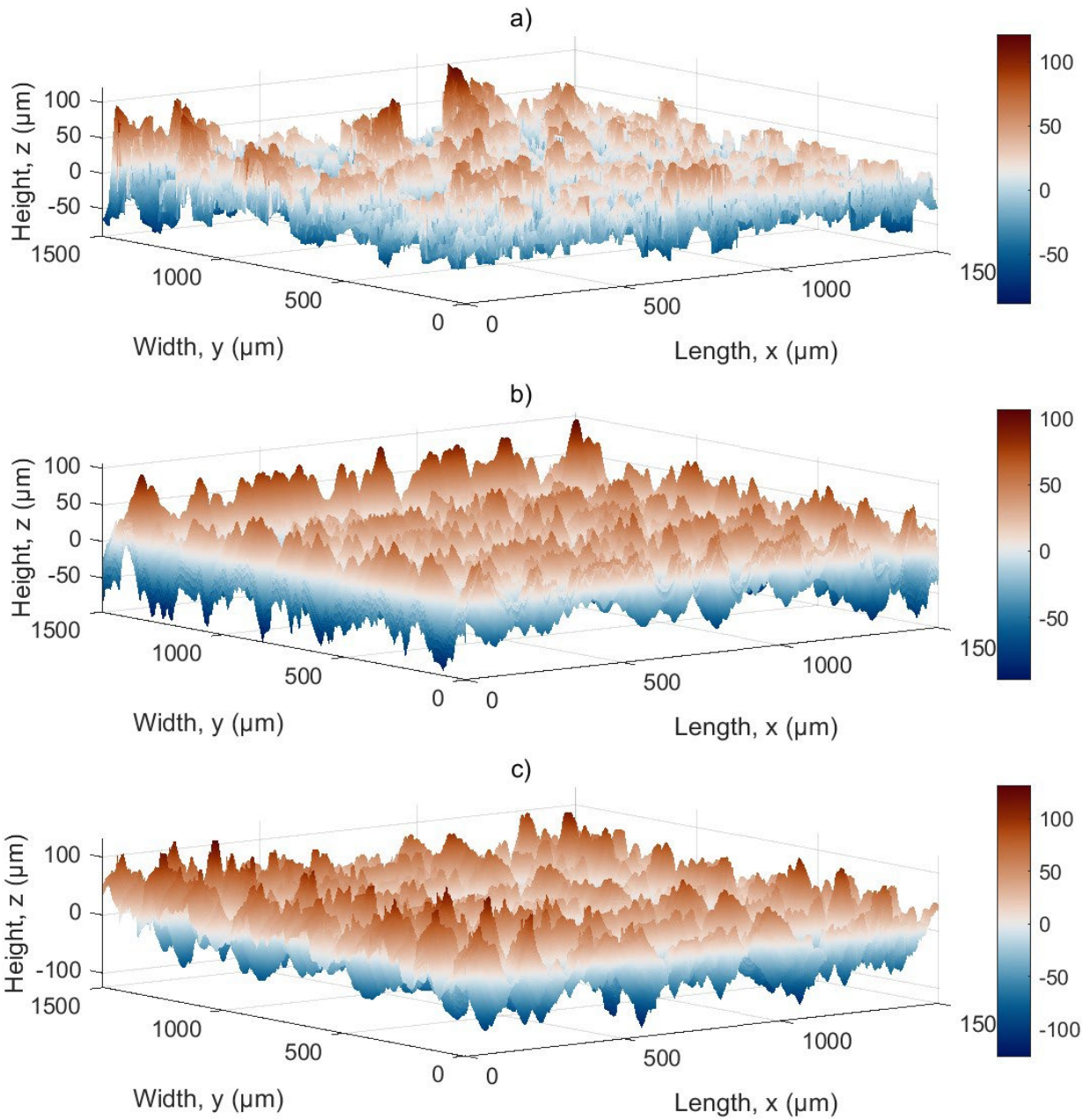

Figure T12: SWLI scan of sample 032 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 032 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 032

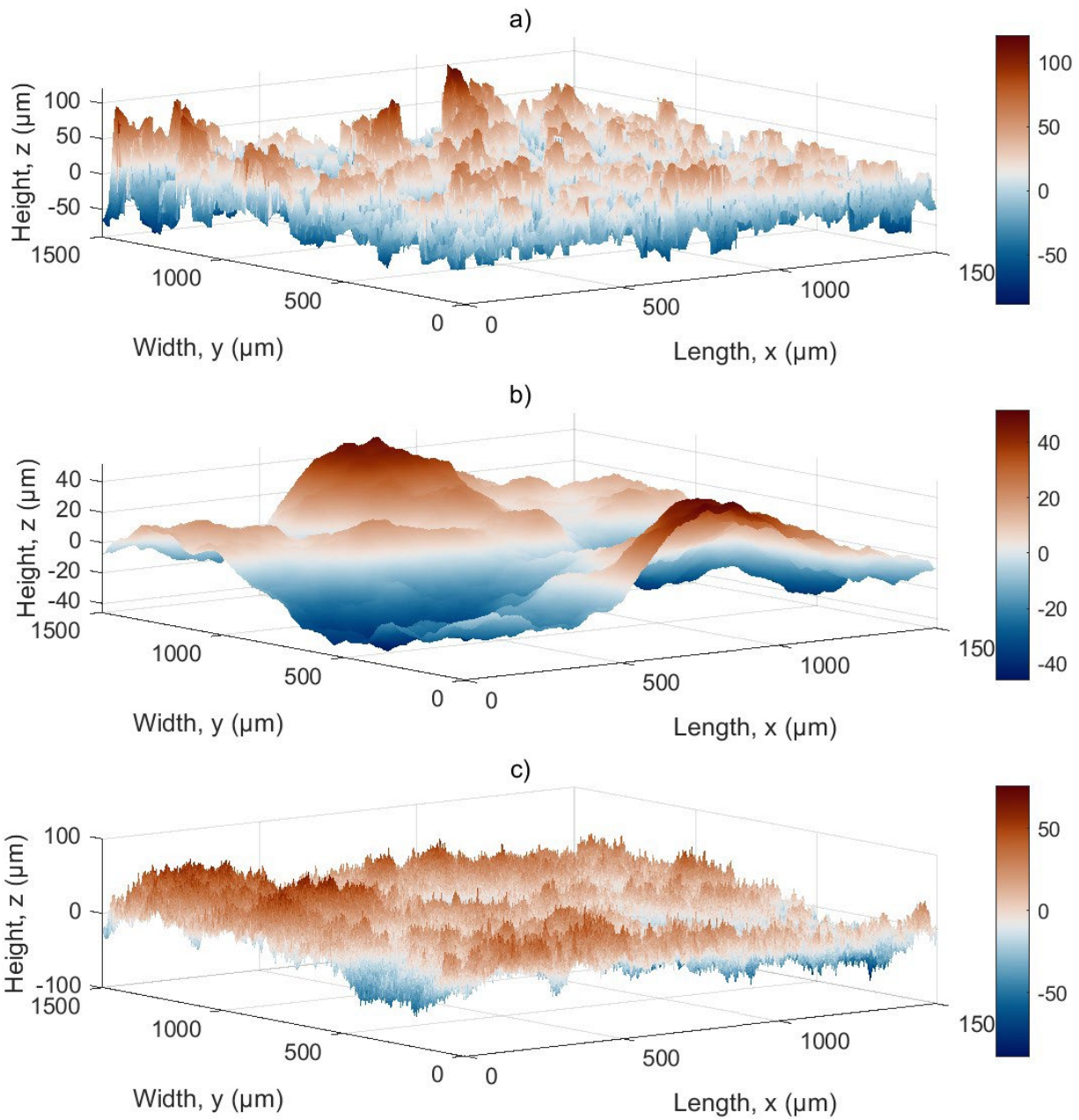

Figure T13: SWLI scan of sample 032 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 032 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 032

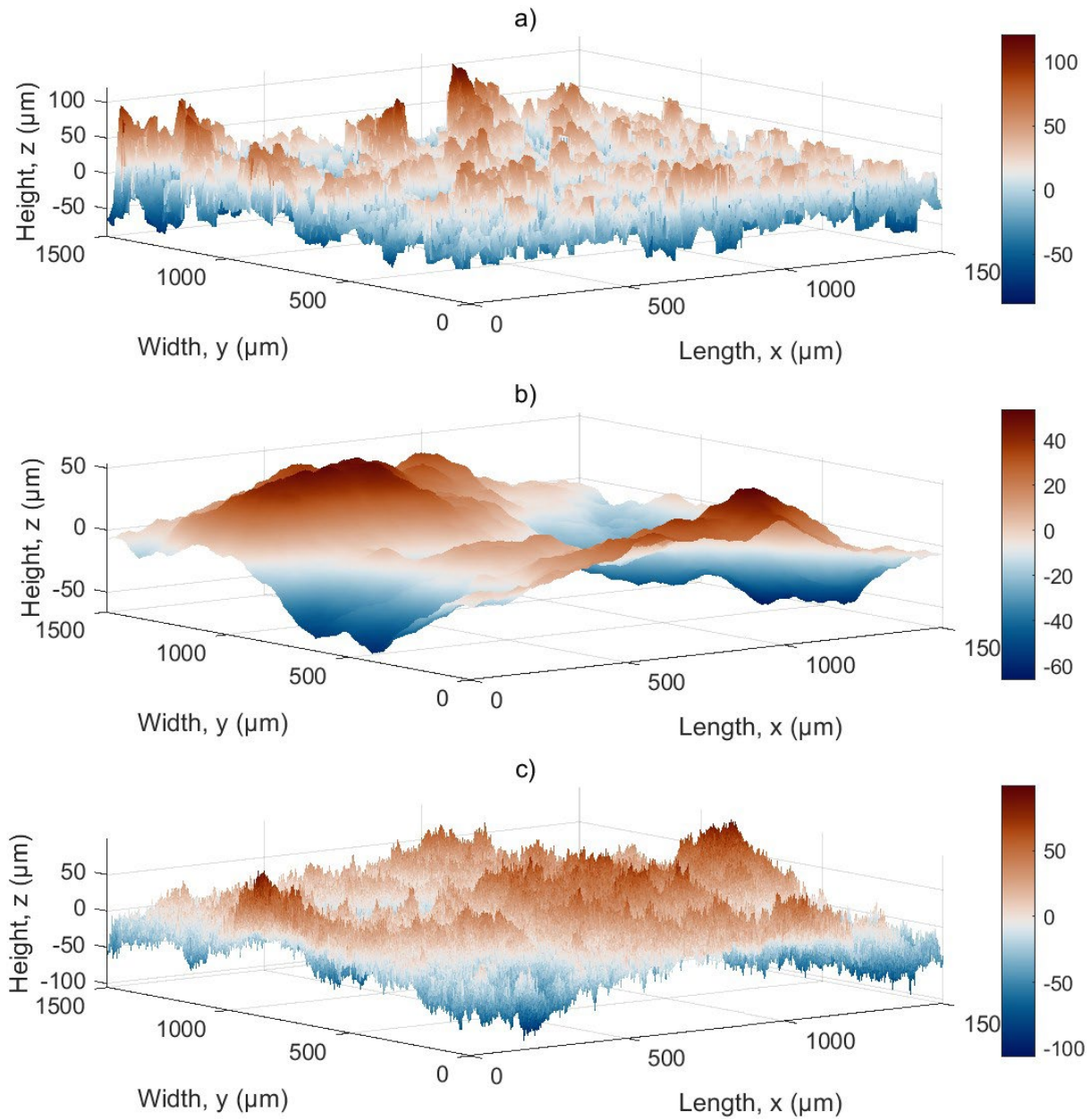

Figure T14: SWLI scan of sample 032 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 032 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 032

Appendix U: Sample 033 supplementary figures

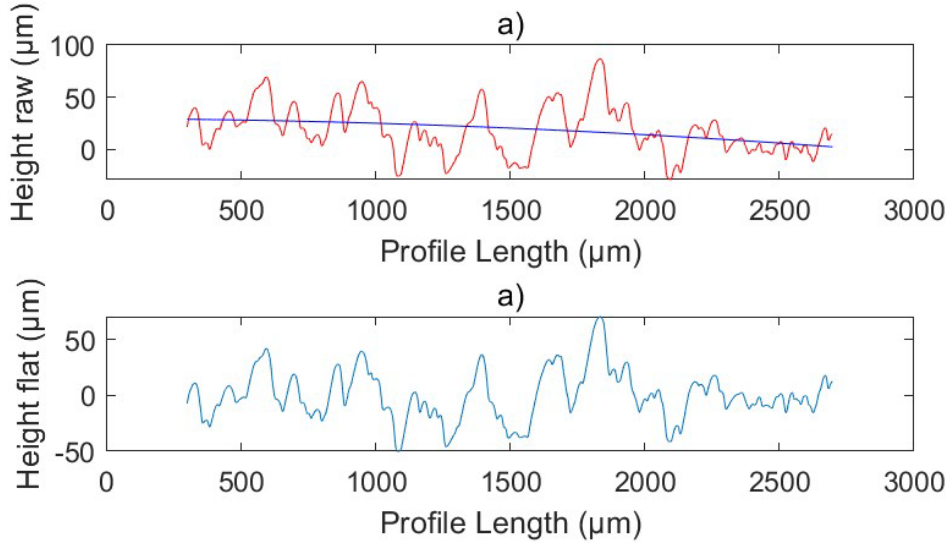

Figure U1: a) Raw profile measurement and fit curve sample 033 b) flattened line profile from sample 033

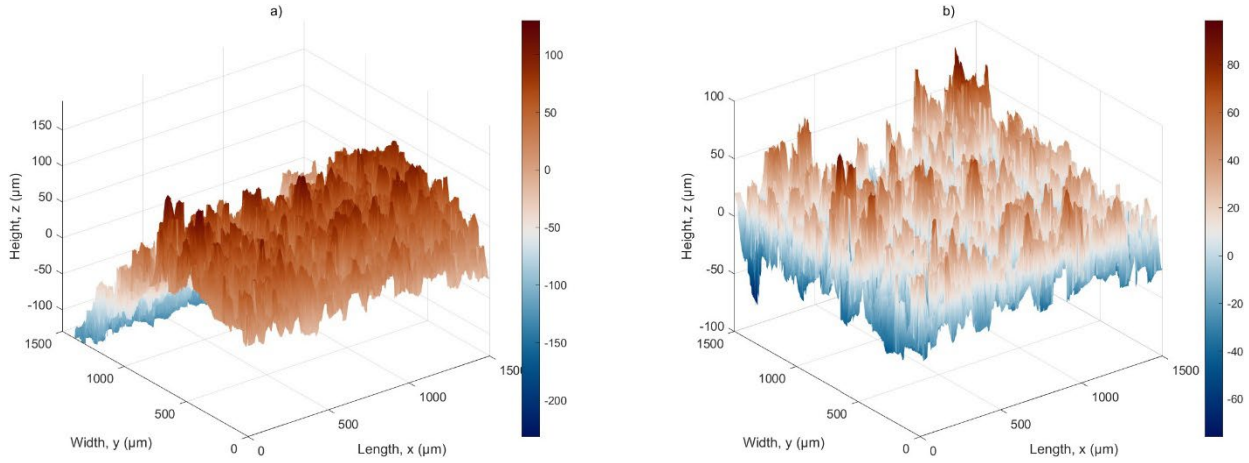

Figure U2: a) Unflattened SWLI raw surface scan sample 033 b) plane flattened SWLI surface 033

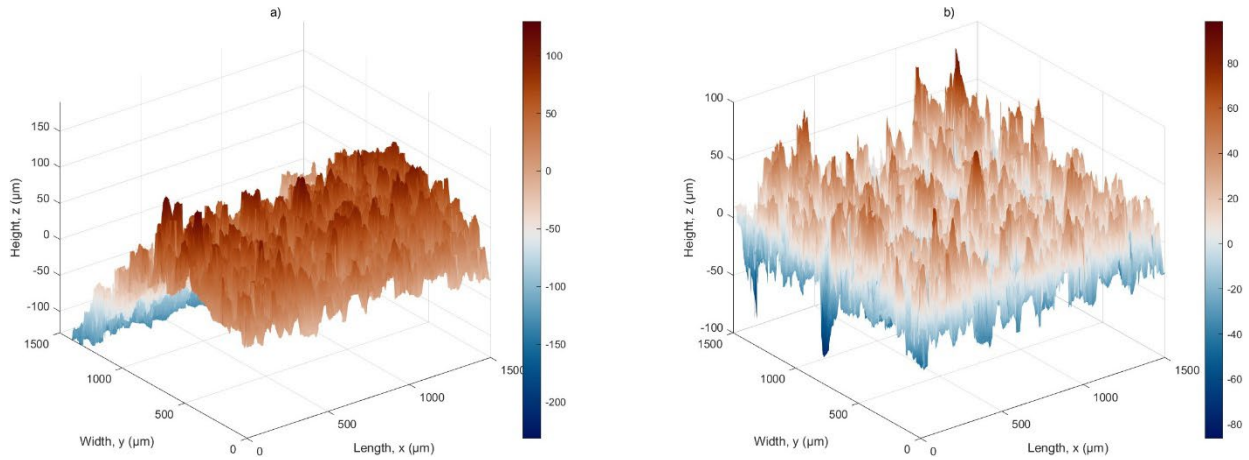

Figure U3: a) Unflattened SWLI raw surface scan sample 033 b) line by line flattened SWLI surface 033

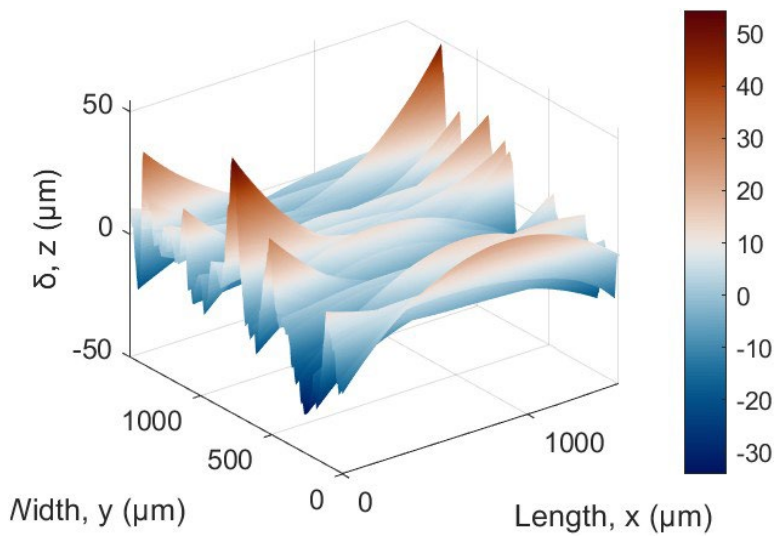

Figure U4: Difference between plane flattening and line by line flattening for sample 033

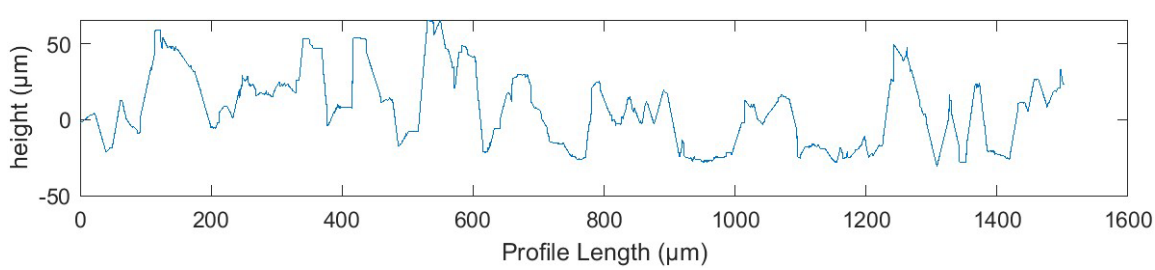

Figure U5: Flattened selected line profile from SWLI dataset sample 033

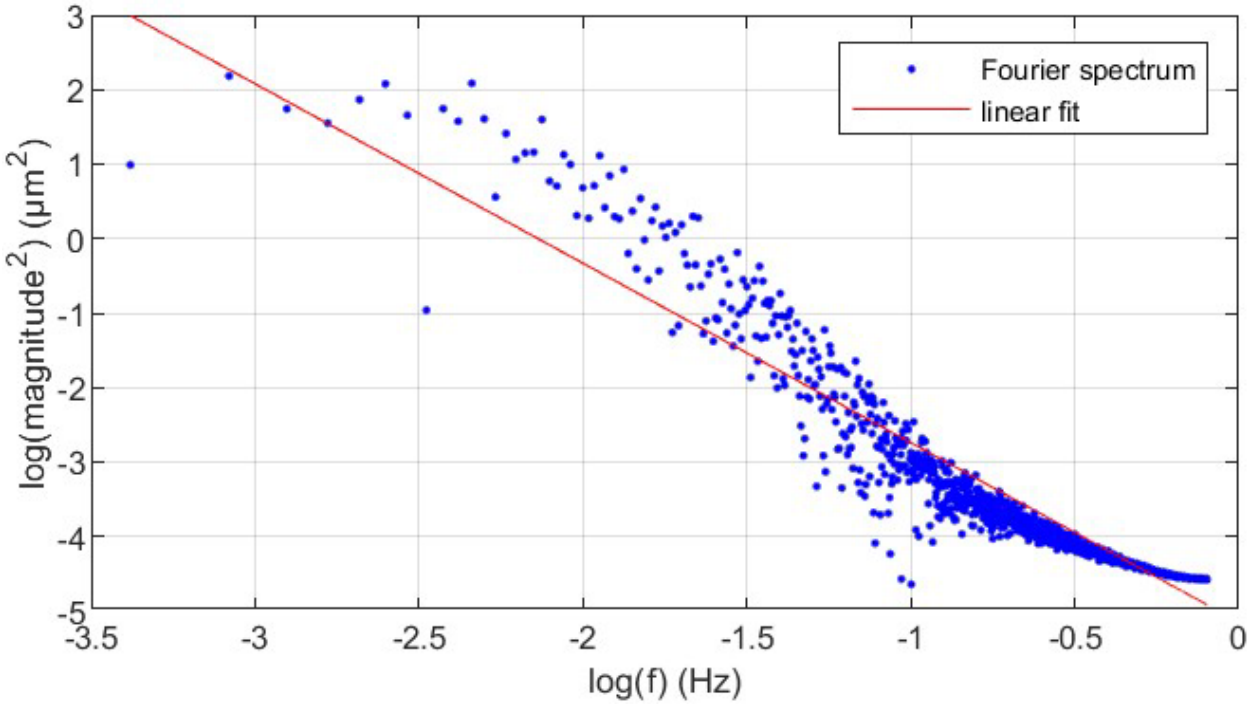

Figure U6: Fourier analysis and linear fit from Dektak line profile data sample 033

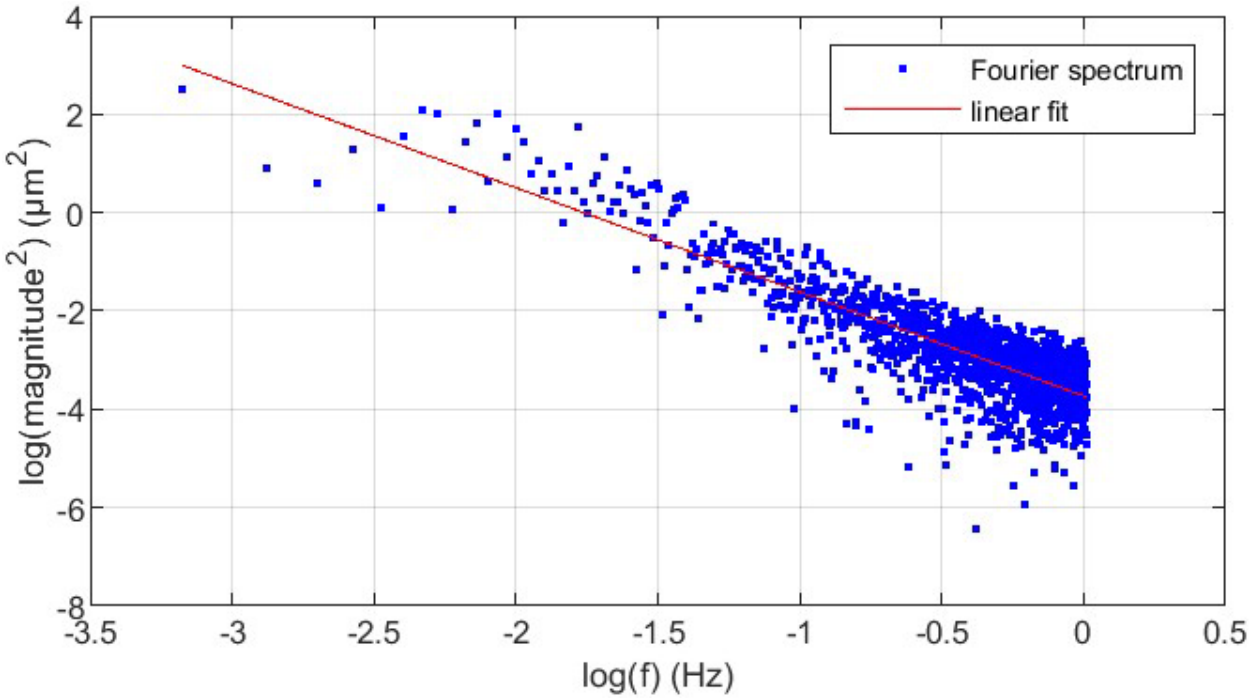

Figure U7: Fourier analysis and linear fit from SWLI line profile data sample 033

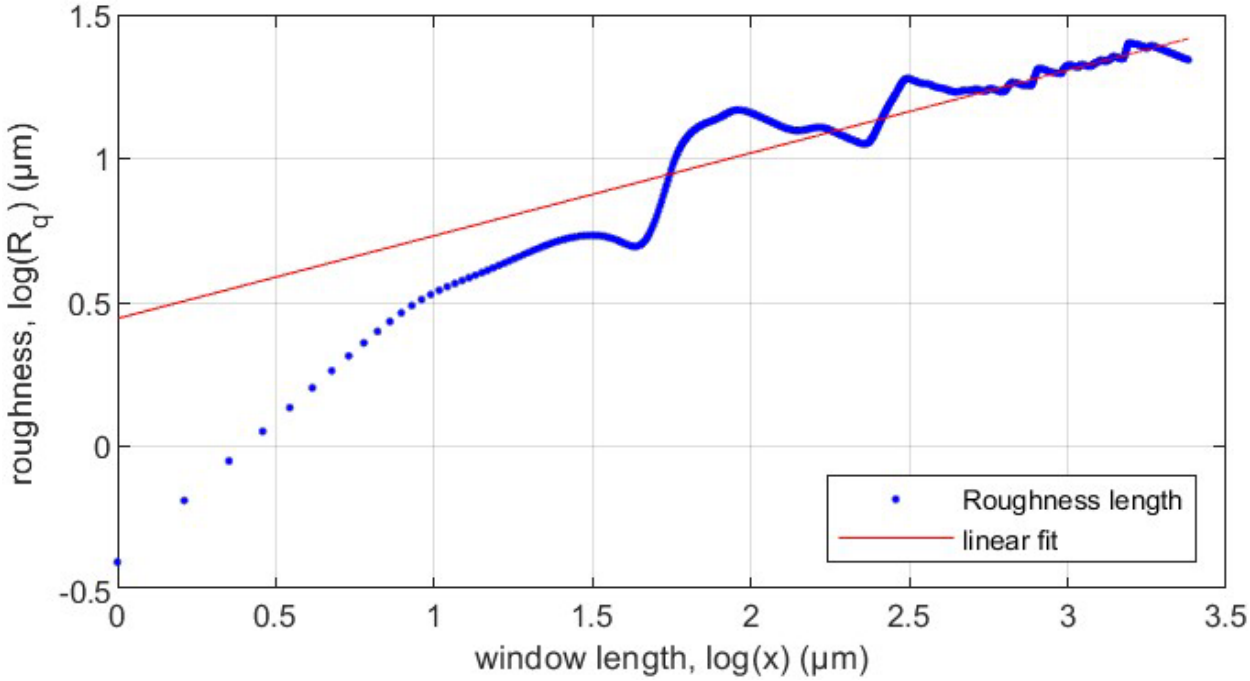

Figure U8: Roughness length method for calculating *H* from Dektak profile sample 033

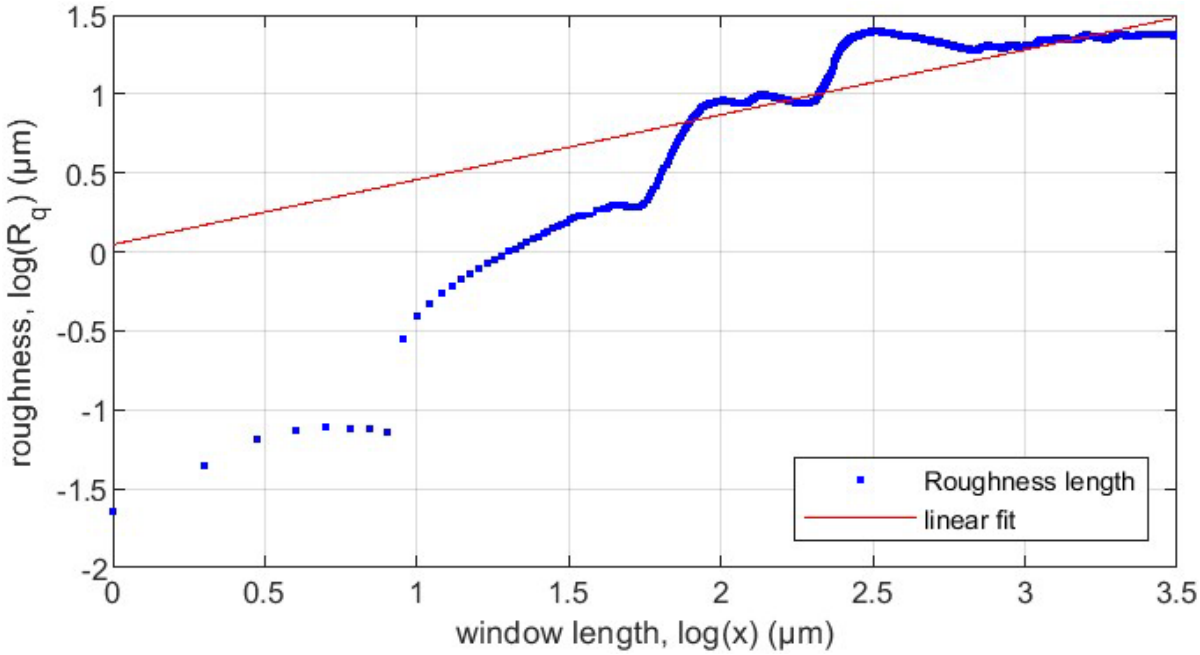

Figure U9: Roughness length method for calculating *H* from SWLI profile sample 033

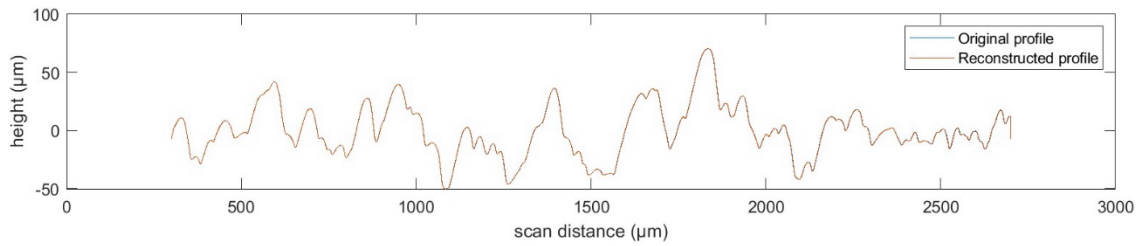

Figure U10: Original Dektak line scan profile and Fourier recreated line profile for sample 033

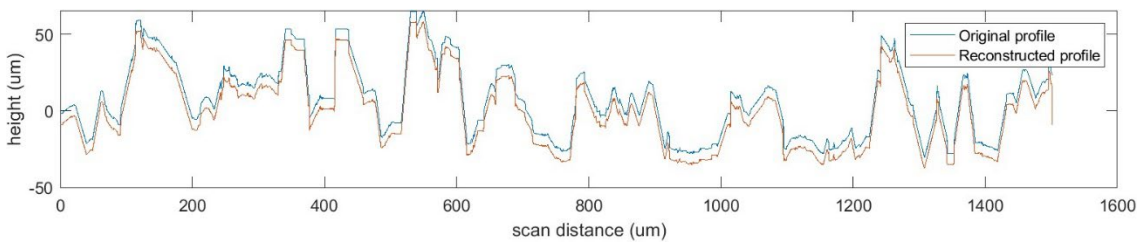

Figure U11: Original SWLI line profile and Fourier recreated line profile for sample 033

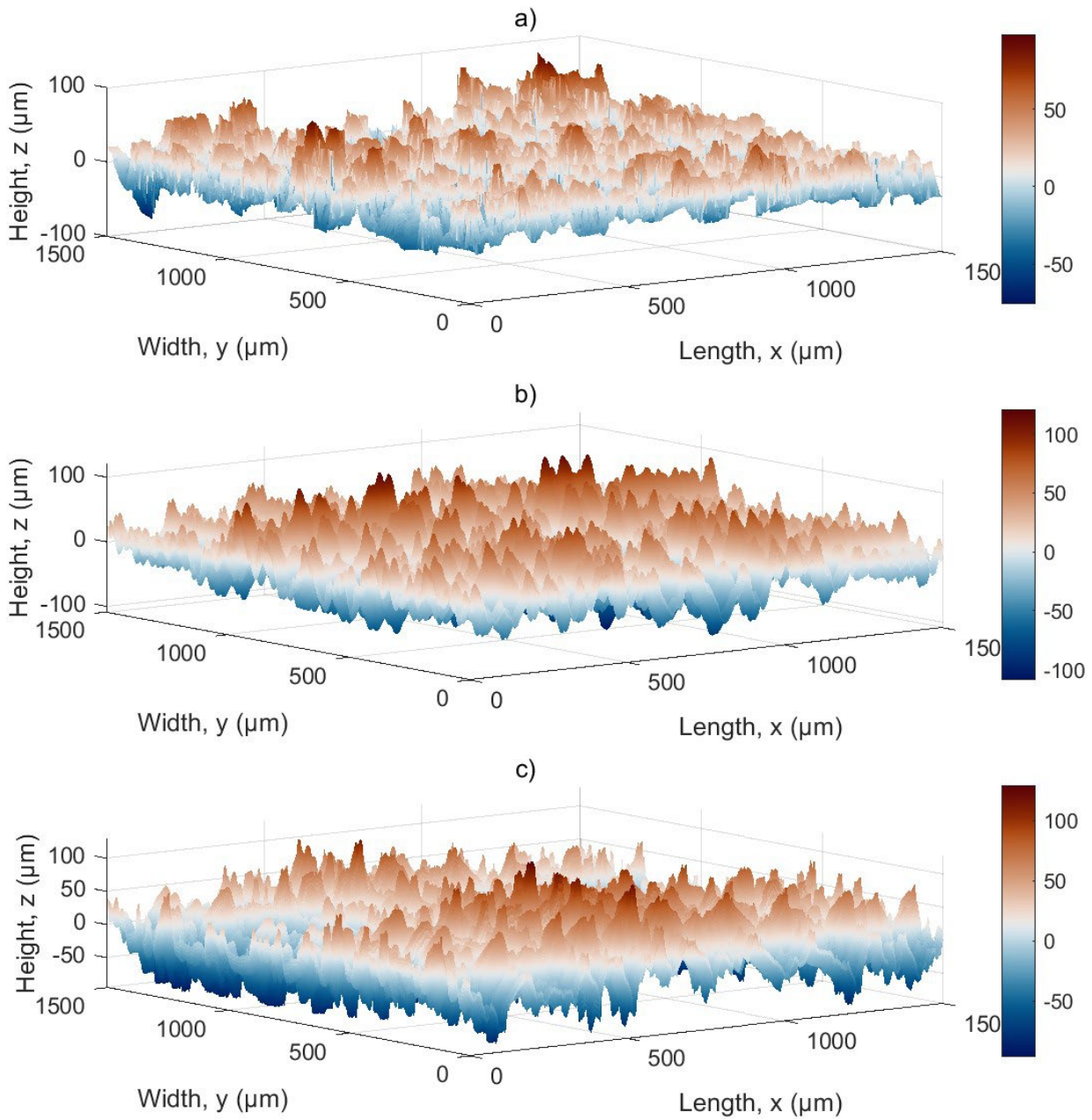

Figure U12: SWLI scan of sample 033 b) orthogonally extrapolated surface using spectrum data from Dektak line scan for sample 033 c) orthogonally extrapolated surface using spectrum data from SWLI line profile for sample 033

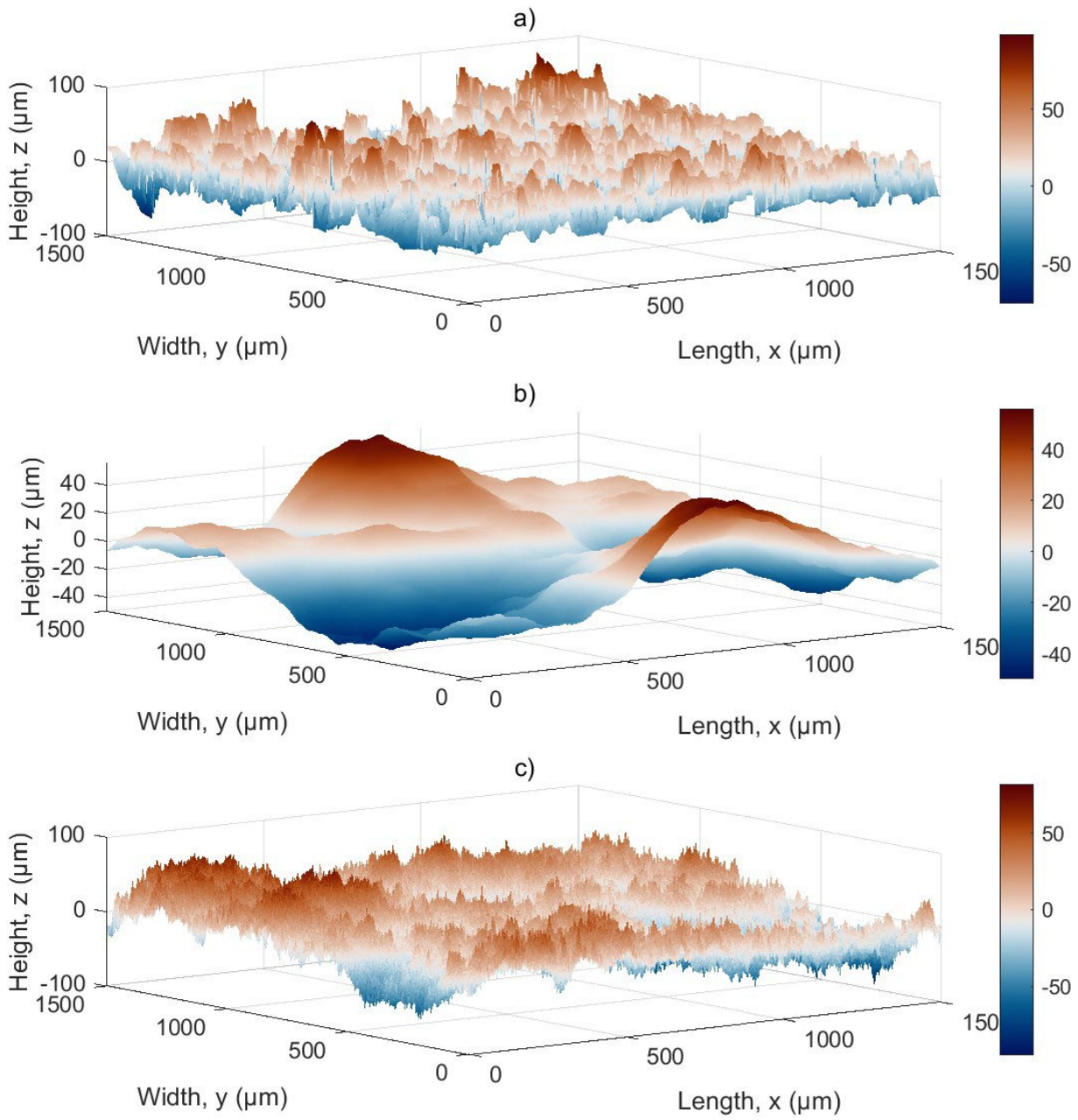

Figure U13: SWLI scan of sample 033 b) artificial_surf generated surface using Fourier method from Dektak line scan for sample 033 c) artificial_surf generated surface using roughness length method from Dektak line scan for sample 033

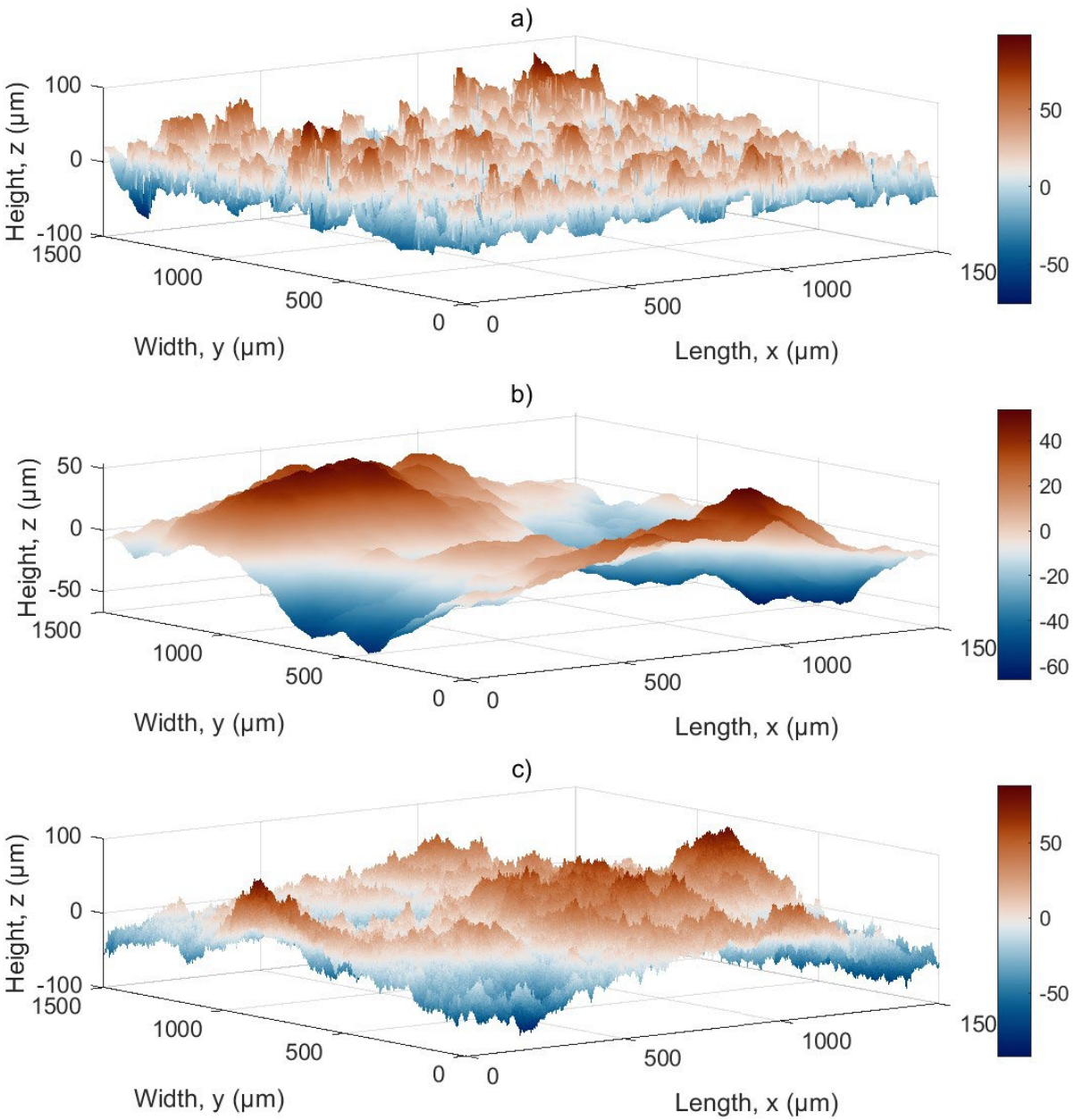

Figure U14: SWLI scan of sample 033 b) artificial_surf generated surface using Fourier method from SWLI line profile for sample 033 c) artificial_surf generated surface using roughness length method from SWLI line profile for sample 033

Appendix V: Parameter variation between measured and generated surfaces

The errors between parameters extracted from surfaces generated/extrapolated and those extracted from the measured SWLI surfaces are compared in the following tables. The measured surface's parameters are treated as the ground truth value (despite the optical scan not perfectly representing surface topography). The generated surface's parameters are treated as experimental or measured values for the purpose of the error calculations. The error values calculated are shown in the subsequent tables.

Table V1: Average roughness error results for surfaces vs measured SWLI surface

CSP	Max error (%)	Average error (%)
orthogonal	42.20	19.84
Artificial surface-Fourier spectrum	32.82	15.59
Artificial surface-roughness length	31.96	15.83

SWLI	Max error (%)	Average error (%)
orthogonal	57.71	31.32
Artificial surface-Fourier spectrum	42.14	11.30
Artificial surface-roughness length	39.08	11.75

Table V2: Maximum valley depth error results for surfaces vs measured SWLI surface

CSP	Max error (%)	Average error (%)
orthogonal	100.4	33.13
Artificial surface-Fourier spectrum	61.51	39.47
Artificial surface-roughness length	45.01	20.32

SWLI	Max error (%)	Average error (%)
orthogonal	103.38	63.46
Artificial surface-Fourier spectrum	60.26	18.73
Artificial surface-roughness length	75.33	29.53

Table V3: Total valley depth error results for surfaces vs measured SWLI surface

CSP	Max error (%)	Average error (%)
orthogonal	1114.97	214.68
Artificial surface-Fourier spectrum	443.43	105.57
Artificial surface-roughness length	1274.76	242.46

SWLI	Max error (%)	Average error (%)
orthogonal	1376.28	306.39
Artificial surface-Fourier spectrum	613.49	103.99
Artificial surface-roughness length	1752.88	264.52