STRUCTURAL FUNCTIONAL SURFACE DESIGN AND MANUFACTURE

JAMES TREVOR WHARTON

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

September 2017

Abstract

The main purpose of this investigation was to explore the potential benefits of structural functional surfaces using facilities available within the University. The potential benefits were demonstrated by applying functional surfaces to a set of particular engineering applications. The thesis mainly concentrated on improving the frictional performance of a surface structure for hydrodynamic bearing application. This thesis has also included some preliminary investigation into drag-reducing riblet structures but this chapter mainly discusses the development of a novel experimental apparatus which is needed for precise boundary layer profile measurements and also to obtain the actual surface drag for each sample.

To be able to assess these surfaces experimentally, they first, have to be manufactured. So, an extensive literature review of current manufacturing technologies was carried out. Each manufacturing method was ranked in its ability to cost-effectively produce surfaces with accuracy and repeatability also being considered. It was concluded that rolling, currently, has the best ability to structure large surface areas with the lowest costs associated. Other manufacturing methods, such as laser surface texturing, provide excellent repeatability and accuracy as well as the ability to create complex surface structures but is incredibly time-consuming for large surface areas. It was suggested that a hybrid of multiple manufacturing technologies would be incredibly useful for structuring surfaces. By combining rolling with more elaborate surface texturing methods (i.e. use a method such as LST to texture the roller surface), it is possible to amplify the productivity of less efficient methods, substantially.

Before any journal components were textured, it was decided to test a batch of ground components. These components were finished with an abrasive tape process. The process parameters were varied

for each sample and by doing this, a set of components with different roughness characteristics should have been obtained. The components were measured for 2D roughness parameters, 3D roughness parameters and surface energy. The components were tested on a tribometer apparatus in order to obtain a coefficient of friction (COF) for each sample. Correlation coefficients were then generated for the different surface measurements against COF, so that any strong correlations or trends could be identified. The idea was to try and obtain a reliable performance indicator (PI) so that frictional losses could be identified. It was found that the roughness parameters Sc (core void voume), Ssc (mean summit curvature) and Rku/Sku (profile/surface kurtosis) showed promise in the ability to predict the performance of a surface.

The next stage was to texture the surface of the journal component. This would done by the application of the type III texturing grinding process, described by Stepien (Surface Engineering, 24: 219-225), to the cylindrical grinding process. Some initial components were manufactured and the textures generated were found to be of an ellipsoidal shape. In order to guarantee the benefits of such surfaces, the configuration of the surface pattern has to be optimised. A python script was developed during this investigation in order to automate a full modelling process. The computational fluid dynamics (CFD) modelling used a full 3D Navier-Stokes approximation. This script was used in conjunction with the Taguchi optimisation technique and a best surface configuration was found, resulting in a maximum surface drag reduction of 16.6% at a 3µm clearance.

Further grinding trials were performed and the input parameters of the process were designed so that surface patterns were close to the recommendations of the optimisation process. The performance of the textured samples was impressive, with a maximum reduction in COF of 18.4% seen against a non-textured component with similar average roughness (Sa) value. Again, all components were measured for the aforementioned roughness parameters and surface energy. Sku continued to predict the best-performing component, showing promise as PI for both non-textured and textured samples.

Acknowledgements

I would like to take this opportunity to thank the following people for their support, contributions and guidance. This thesis would not have materialised without their involvement and the least I can do is express my deepest gratitude to all of them.

Firstly, I would like to thank my director of studies, Professor Xun Chen. Without you, I would not have had this opportunity to advance my academic career in the fields that I have a passion for. You have introduced me to incredible ideas and have always challenged me with problems that I never would've envisaged when I undertook the PhD initially. By doing this, you have made me a much more rounded, multidisciplinary engineer, enabling me to approach future problems with a broader spectrum of skills and knowledge. I would also like to thank you for the opportunity of organising the visit to Jilin University. This trip allowed me to explore a different culture, create connections in the field of biomimetics and the rewards of such collaborations continue to generate interesting research between our institutions. Finally, I am also thankful that you allowed me the freedom to explore my own ideas, allowing me to contribute and influence the research direction.

Next, I would like to take this opportunity to send a special thanks to Dr David Allanson. You have been my mentor and supervisor for over 6 years now and it truly astonishes me how much have I learnt from you. No matter how busy you are, you always make time for me, as well as others and I feel sometimes that you are undervalued for the continual support and friendly advice you give to everybody. I am using this moment to recognise this and I am sure, as well as everybody else, is sincerely grateful for such acts of kindness. I came to Liverpool John Moores University to further my engineering career and I never would've imagined myself developing such a strong passion for computational physics, fluid dynamics and thermodynamics. I can quite honestly say, that your

enthusiasm and passion for the subject, initially observed in lectures, was probably what *sparked* my interest in these subject areas. Your constant encouragement to explore new ideas and thought-provoking concepts have, on many occasions, led me to breakthroughs in my research. I thoroughly enjoyed teaching with you in Advanced CFD and when the students created the emblem and Latin motto for the class, referring to it as a "pressure cooker" and using the phrase "the weak may leave now", I think it may have indicated how much we enjoyed teaching the module and it was great to see other students develop a passion for the subject, just like I did. Both as a friend and as a student, I would like to say thanks again for all that you have done for me and I hope that one day, we can finish the quest of finding all Trappist beers.

I would also like to thank Mr Paul Wright (Engineering Technical Manager) for all the technical support and knowledge you have given me over the years. I thoroughly enjoyed building the high-speed open-loop wind tunnel with you and I can definitely say that the experience has enabled me to be more confident with electrical engineering. Many late nights were required in order to materialise that experimental rig and I think it would be appropriate to express just as much gratitude to his wife, Anne, for *putting up with* us working late on so many occasions. It really is appreciated (P.S. I wish you good luck in all your culinary adventures). Paul, it is without question that the faculty would cease to function properly without your vast technical advice and dedication to the job. I will miss our "putting the world to right" discussions.

I also need to express my deepest gratitude for all the technical support given to me by the engineering workshop team, especially to Peter Moran, whose hard work, dedication and experience has enabled cylindrical textured grinding to become a reality. I'd also like to thank the IT support team (Neale Murray, Michael Earner and Neil Hughes) because without you, these simulations would never have been possible. There are many others who have shown me so many acts of kindness within the Faculty of Engineering and Technology over the years and it goes without

question that my experience of this institution has been a happy one, both as an undergraduate and post-graduate.

Now, I (Grantham, Grandad and Nigel Thornberry) have to give a special mention to the “llamas” (Graham, Ross, Nat, Rachel, Doz and Jamie). Over the years, we have had some great times and I have so many good memories of all the antics we have got up to. It is why my thanks need to be expressed to you all because you have done one of the most important jobs, keeping me sane. Thanks for being so “smashing”.

I dedicate this thesis to:

Mum and Dad

Thank you for the encouragement, motivation and wisdom you have given me throughout my life. It is without question that I would have not got this far without your continual support and I will always be deeply indebted to you both.

Table of Contents

List of Figures

List of Tables

Nomenclature

Roman:

a_d	Dressing depth of cut (μm)
a_e	Real depth of cut (μm)
a_p	Depth of cut (μm)
a_s	Wheel wear (μm)
b_d	Dressing tool engagement width (μm)
c	Texture land width (μm)
d	Pipe diameter (m)
f_d	Dressing feed rate ($\mu m.rev^{-1}$,$mm.min^{-1}$)
h	Grid spacing (μm)
k	Height of roughness (μm), turbulent kinetic energy
l_g	Square root of riblet groove cross-sectional area
$\dot{m}$	Mass flow rate ($kg.s^{-1}$)
n	Rotational speed ($rev.min^{-1}$), outward normal direction
p	Pressure (Pa), order of accuracy
p_A	Atmospheric pressure (Pa)

$\bar{q}$	Dynamic pressure (Pa)
r	Radius of the journal (mm), Pearson's correlation coefficient, grid refinement ratio
r_d	Dressing tool tip radius (μm)
s	Riblet spacing (μm)
t	Time (s)
$\bar{u}$	Mean velocity in streamwise directions ($m.s^{-1}$)
u'	Fluctuating velocity in streamwise direction ($m.s^{-1}$)
u_τ	Friction velocity ($m.s^{-1}$)
v	Surface speed ($m.s^{-1}$)
$\bar{v}$	Mean velocity in wall-normal direction ($m.s^{-1}$)
v'	Fluctuating velocity in wall-normal direction ($m.s^{-1}$)
v*	Workpiece to wheel surface speed ratio
v_f	Feed rate ($mm.min^{-1}$)
$\bar{w}$	Mean velocity in spanwise direction ($m.s^{-1}$)
w'	Fluctuating velocity in spanwise direction ($m.s^{-1}$)
x	Distance in streamwise/x-direction (m)
y	Distance in wall-normal/y-direction (m)
z	Distance in spanwise/z-direction (m)
A	Area (m^2)
A_g	Riblet groove cross-sectional area

C	Arbitrary constant
C_{fr}	Coefficient of friction
D	Diameter (m)
E	Error
F	Force/load applied (N)
F^*	Non-dimensional friction force
F_f	Friction force (N)
F_r	Radial force (N)
F_s	Factor of safety
L	Characteristic length scale (m)
N	Number of cells
R	Pipe radius (m), convergence ratio
R_s	Grinding wheel radius (mm)
Re	Reynolds number
Re_{crit}	Critical Reynolds number
S_ϕ	Source/sink of scalar quantity
T	Torque (Nm)
Ta	Taylor number
Tu	Free-stream turbulence intensity
U	Velocity ($m.s^{-1}$)

U_∞	Free-stream velocity ($m.s^{-1}$)
U_d	Overlap ratio
V	Volume (m^3)
X	Deflection (mm)
X_{exp}	Thermal expansion (mm)
$\bar{X}$	Mean value for dataset
X_i	Trial/sample value
$\bar{Y}$	Mean value for dataset
Y_i	Trial/sample value

Greek:

α	Angle (radial)
γ	Interfacial energy ($mN.m^{-1}$)
δ	Boundary layer thickness (m)
ε	Turbulence dissipation rate ($m^2.s^{-3}$), relative error
θ	Static contact angle
κ	von Kármán constant
λ	Resistance factor
μ	Dynamic viscosity (Pa.s)
ν	Kinematic viscosity ($m^2.s$), surface speed ($m.s^{-1}$)

ρ	Density ($kg.m^{-3}$)
ρ_w	Grinding wheel profile radius (mm)
τ	Shear stress (Pa)
ϕ	Concentration of the scalar quantity
ω	Rotational speed ($rad.s^{-1}$)
Γ	Coefficient of diffusion
Δu^+	Roughness function

Subscripts:

rms	Root mean square
s	Sand grain, solid, grinding wheel
l	liquid
w	Workpiece
x	In streamwise/x-direction
y	In wall normal/y-direction
z	In spanwise/z-direction
LG	Liquid-gas interface
SG	Solid-gas interface
SL	Solid-liquid interface

Superscripts:

-	base component
+	Inner scaling, acid component
d	Dispersive component
p	Polar component

Abbreviations:

2D	Two-dimensional
3D	Three-dimensional
AE	Acoustic emission
AMTReL	Advanced Manufacturing Technology Research Laboratory
APG	Adverse pressure gradient
BC	Boundary condition
CAD	Computer aided design
CBN	Cubic boron nitride
CFD	Computational fluid dynamics
CNC	Computer numerical control
COF	Coefficient of friction
CPU	Central processing unit
CSV	Comma separated value

DNS	Direct numerical simulation
FEA	Finite element analysis
FST	Free stream turbulence
GAMG	Geometric-algebraic multi-grid
GCI	Grid convergence index
GUI	Graphical user interface
LES	Large eddy simulation
LST	Laser surface texturing
PDE	Partial differential equation
RANS	Reynolds-averaged Navier-Stokes
RMS	Root mean square
RNG	Re-normalisation group
SE	Surface energy
SEM	Scanning electron microscope
SiC	Silicon carbide
SIMPLE	Semi-implicit method for pressure linked equations
STL	Stereolithography
T-S	Tollmein-Schlichting
UV	Ultraviolet
ZPG	Zero pressure gradient

1. Introduction

1.1 Background

The research that is presented in this thesis has been undertaken within the Advanced Manufacturing Technology Research Laboratory (AMTRel) at Liverpool John Moores University. Due to the recent technological developments in the fields of manufacturing, metrology and modelling, the world is on the brink of a surface engineering revolution. Many scientists are now looking to functional surfaces in order to make further efficiencies in systems which all consume energy, a large of proportion of which is still generated by fossil fuels. The focus of this research is to carry out an evaluation of current manufacturing technologies surrounding the creation of these functional surfaces and critically analyse them in relation to their ease of use, costs related to producing parts, applicability and accuracy. After this evaluation took place, the selected manufacturing processes would be adapted in order to create micro-structured surfaces relating to two specific areas of interest: hydrodynamic bearing technology and drag reducing surfaces.

As this thesis will discuss later on, there are many difficulties and gaps in knowledge surrounding the design of these surfaces and how they relate to their functionalities. In order to answer some of these questions, significant effort has gone into developing computational fluid dynamics (CFD) models. The modelling of these surfaces will not only lead to a better understanding of the surface's mechanism that gives rise to the beneficial properties but will also allow for optimisation of the shape, orientation and dimensions of these micro-structures for each particular scenario.

Unfortunately, modelling isn't the 'real world' and sometimes it is possible that the physics can be under-represented, leading to the generation of false conclusions. Usually, the reason for this is

because the most important stage of the modelling process, validation, is often overlooked or done poorly. So, to make sure that the simulation work carried out in this project does in fact represent the physics, experimental studies have been undertaken. In order to facilitate the experimental studies, considerable engineering has led to the design and creation of a high-speed, open loop wind tunnel with sophisticated, precise instrumentation that enables for the investigation of boundary layers over these textured surfaces. A hydraulically fed, hydrodynamic bearing testing rig was designed and manufactured before the project took place but some modifications were also made to the rig during the course of this research in order to alleviate some of the reliability issues. This bespoke tribometer allows for the frictional performance of different bearing surfaces to be evaluated.

The main goal of this research is to contribute to the surface engineering field by demonstrating a range of application of functional surfaces which can be manufactured, cost-effectively and feasibly, using current technologies. In the process of doing so, the work sheds some light on the complex fluid mechanics related to the mechanisms that lead to the beneficial properties of these surfaces through the use of numerical modelling and experimentation.

1.2 Aims and Objectives

The aim of this investigation is:

"To investigate how a micro-scale surface geometry can be manipulated in order to improve the properties related to textured-induced lift, frictional reductions and surface drag reductions."

The specific objectives of the research presented in this thesis are:

i. To investigate and optimise the beneficial behaviour of micro-scale textured surfaces interacting with moving fluids by using a computational fluid dynamics software (CFD) package.
ii. To validate the CFD predictions by performing experimentation using facilities available within the University.
iii. To investigate the manufacturability of structural surfaces and develop a feasible manufacturing strategy.
iv. To demonstrate the benefits of structural functional surfaces for engineering applications.
v. To perform a literature review of functional surface studies including those which have taken inspiration from nature.

1.3 Thesis Outline

This thesis contains seven chapters. The second chapter introduces the reader to surface flow interactions, drag-reducing surfaces, a tribology-orientated viewpoint on structured surfaces for bearings and also a summary on existing manufacturing methods for surface texturing. The third chapter experimentally attains whether surface roughness parameter or surface energy can be used as performance indicator (PI) for predicting the frictional losses in a hydrodynamic bearing application. The fourth chapter presents some initial grinding trials which have been used to grind the surface of the journal component. This chapter then discusses the development of a python script for automating a computational fluid dynamics (CFD) modelling process. This script is then used to optimise surface pattern configurations. Chapter five uses the recommendations of the optimisation study with further grinding trials. The textured components generated from this grinding study are then used with the tribometer apparatus to validate the scale/magnitude of

surface drag reductions observed in the optimisation study. Chapter six presents some preliminary work into riblet structured surfaces and their potential for surface drag reductions. This chapter mainly concentrates on the development of a novel workholding solution which is required for the precise measurements needed in boundary layer profile data generation. Chapter seven, which is the final chapter, presents the conclusive remarks of the investigations and raises some potential areas for future investigation.

2. Literature Review

So, what is a functional surface and how important is surface engineering to the global challenges of the 21st century? A surface can be described as "functional" when there is a relationship between some measured value that represents the actual geometric surface (i.e. surface roughness parameter like Ra) and a performance indicator which reflects the surface's ability to achieve the designed, beneficial property (function) such as self-cleaning (super-hydrophobicity), drag/frictional reductions and improved heat transfer capabilities. Such surfaces offer the potential to help with one of the biggest problems facing the human race at this moment in time: energy consumption and the use of fossil fuels to provide this ever-growing demand. Fossil fuels are a cheap source of energy and developing countries who are on the verge of becoming economic superpowers turn to coal and gas sources to power their industries, making sure they can compete globally. In doing this, large amounts of greenhouse gases are being emitted. As the scientific community becomes more aligned with the view that global warming does exist, in the face of mounting evidence, attempts are being made to improve efficiencies in systems whilst alternatives to fossil fuel technologies are being developed. One of the biggest consumers of fossil fuels is the transport industry. Holmberg et al. (2014) performed a study into the global usage of petroleum in heavy-duty vehicles and found that 33.5% of the total energy losses in this industry was down to frictional losses, this equates to 180,000 million litres of fuel in 2012. So even a small contribution from surface engineering fields to reduce the friction between mechanical components would be beneficial. Zhang et al. (2011a) showed evidence that it is becoming increasingly difficult to find savings in the design of pipelines but using experimental and modelling techniques, however an optimised surface structure led to a 10% reduction in the

surface drag forces. A saving of this magnitude would massively decrease the global pumping capacity requirements and hence, lead to significant reductions in energy consumption.

Over the last decade, the amount of research relating to functional surfaces has grown significantly but why is this? Bruzzone and Costa (2013) recently stated: “recent technological developments now permit us to texture surfaces in a flexible way and to assess the tribological efficiency of different microtopologies”. Developments in the fields of manufacturing, metrology and modelling have all contributed to this major breakthrough in surface engineering. Manufacturing processes have been transformed using computer numerical control (CNC) and sophisticated feedback control systems with precise sensor instrumentation. Acoustic emission (AE), force/power monitoring, accelerometers and optical-based sensors (e.g. interferometers) are examples of instruments which have proved to be significant in improving accuracy and repeatability in various machining methods but these were once only available to research (Byrne et al. (1995)) due to the associated upfront costs of purchasing and implementation within existing production systems. Since then, industry has more widely adopted these sensing technologies as related costs have reduced, off-the-shelf machinery are fitted with such sensors and knowledge surrounding such components (and the use of them) have been transferred to workforces within manufacturing businesses. With these developments, today’s manufacturing systems have shown that they are quite able to produce micro-structured surfaces on components, with varying degrees of success (Wharton et al. (2014)).

In order to assess how well these surfaces have been manufactured, the component needs to be inspected and recent advances in the field of metrology allow for them to be more readily assessed. Traditionally, a surface would only be assessed using a two-dimensional, statistical surface roughness value, such as Ra (average deviation from mean), using a surface profiling instrument. Depending on the magnitude of this value, various different assessments could be made about the manufacturing process, such as:

i. The longevity of the tool could be evaluated (e.g. Does the grinding wheel need re-dressing?);

ii. The appropriateness of the machine tool for the component in question (e.g. Are the machine tool damping and stiffness characteristics appropriate for the workpiece material hardness?);

iii. Input parameter selection. (e.g. Inappropriate feeds/speeds and tool overhang causing undesirable vibrations (chatter));

The surfaces being generated are a result of the manufacturing process and the actual geometric structure of these surfaces are not being intentionally created. The size of these roughness features/surface irregularities depends on how well refined the manufacturing process is. Another problem with using Ra as an assessment of a surface, is that the same value can represent surfaces that are significantly different in geometry (Arnell et al. (1991)); Figure 2.1 gives a useful visual representation of this.

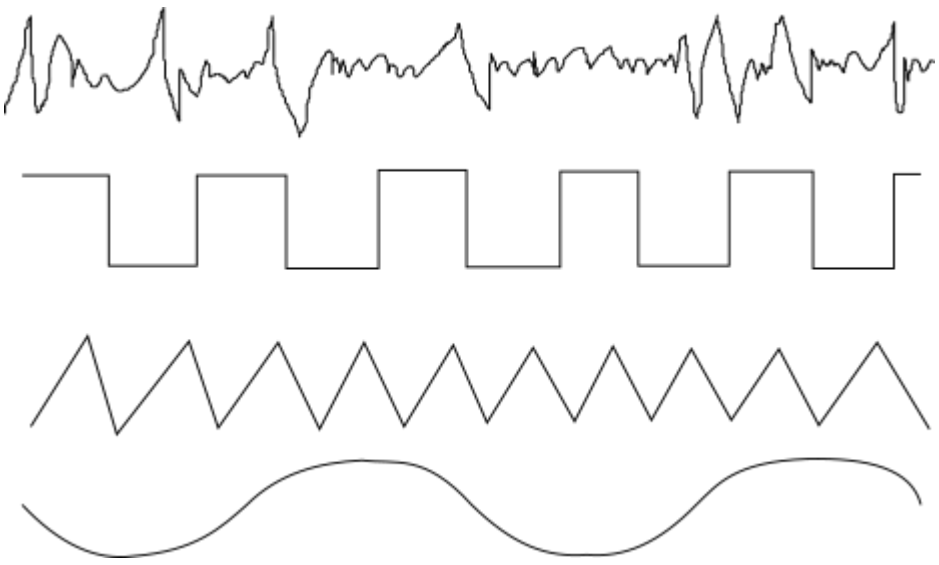


Figure 2.1. Different surface profiles with the same Ra.

Large peaks in the surface structure, which are masked by the use of this assessment, are poor for bearing applications (Stachowiak and Batchelor (1993)). Increased wear rates would be seen if such a surface was used and due to the precise tolerances related to high-speed bearings, this could significantly shorten the life of the component and consequently lead to catastrophic failure of this system. Therefore, to get a better description of the surface, other roughness parameters are commonly used in combination with Ra, such as Rp (maximum peak height). From this 2D approach, further developments in metrology have led to the use of optical techniques (i.e. white light interferometry) which allow for a surface to be inspected qualitatively, using three-dimensional surface plots and the quantitatively, using 3D statistical roughness parameters such as Sa, an extension of Ra. These improvements allow the user to assess a surface more accurately, which means engineers can control the geometry of the surface with a higher level of precision, enabling for a more reliable description of a surface's ability to perform some function.

The functionality being investigated for this research is solely related to the flow past stationary surfaces (boundary layers) or surface driven flows (Couette flow). Therefore, the remaining parts of this chapter will firstly, discuss the existing knowledge surrounding flow over surfaces. Secondly, some previous research relating to the effects of altering the surface structure will be discussed and some examples of naturally occurring surfaces (as well as engineered replicas) by looking to the field of biomimetics will be presented. Biomimetics is a field of engineering that attempts to design components or systems that mimic some of the mechanisms observed to operate in the natural world. The reason for including such content, is that nature has undergone billions of years of evolution and natural selection had led to species being optimised for their surroundings (i.e. animals with the greatest ability to perform within their surroundings survive and pass their genes onto their offspring). Therefore, there are some species that already present some interesting and "naturally optimised" mechanisms that relate to the surface drag reductions being explored for this research. Some of these engineered replicas will be discussed as they

inspired some of the designs presented in this thesis. The final part of the literature review will present a critical analysis of current surface manufacturing methods.

2.1 Flow over surfaces

The interaction of moving fluids with any object is complex and because of this, the earliest fluid dynamics tended to analyse objects using perfect fluids. A perfect fluid is one that is inviscid (has no viscosity), incompressible, has zero surface tension, adiabatic and is single phase. Unfortunately, neglecting the effects of viscosity results in d'Alembert's paradox. Jean le Rond d'Alembert proved in 1752 that if a body is moving through a perfect fluid in an equilibrium state of constant velocity, there is zero drag force. Obviously, this is not true in reality and therefore shows a flaw in the theoretical assumption of a perfect fluid. In reality, interacting layers of fluid have both tangential (shear) and normal (pressure) forces acting on each other (Schlichting (1979), p.5). Only by including the viscosity of a fluid can the shear stresses be accounted for. To help explain the effects of viscosity to the reader, a schematic of a simple surface driven flow (Couette flow) is provided (Figure 2.2). The flow being investigated is between two infinitely long plates. One surface is at rest (zero velocity) and the fluid at the surface of the plate is assigned a no-slip boundary condition. Due to molecular adhesive forces that exist at the solid-fluid interface being greater than the cohesive forces which act at the fluid-fluid interface, fluid that is right next to the surface is brought to rest. This is called the no-slip boundary condition and this means the fluid has zero velocity at the surface. The second upper surface has a velocity in one direction only (Ux) and again, due to the no slip conditions, the fluid attains the same velocity as the moving plate. In summary, the following assumptions are made for Couette flow:

i. No flow in either the y or z-direction;
ii. Fully developed flow;

iii. Steady state (no changing conditions, temporally)

iv. At y=0, no-slip boundary condition exists (**u** = (0,0,0))

v. At y=H, **u** = (Ux,0,0)

vi. No pressure drop in the x direction. dp/dx = 0.

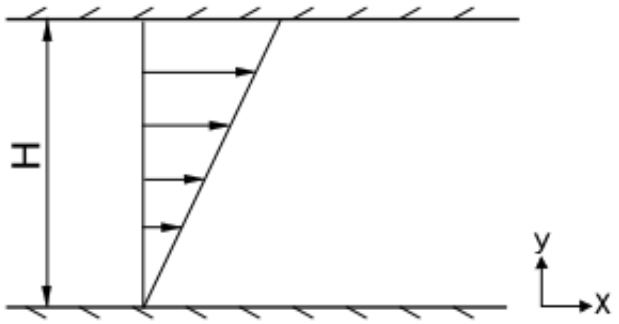


Figure 2.2 Couette flow schematic (Wharton et al. (2016)).

Using these assumptions, the Navier-Stokes equation can be simplified to express the linear velocity gradient between the two surfaces (eqn. (2.1)).

$$u(y) = \frac{u \cdot y}{H} \tag{2.1}$$

The only forces that are being exerted from the fluid onto the plate surface and retrospectively, the forces being applied from the plate onto the fluid are tangential (forces resulting from friction). Per unit area of the plate surface, the magnitude of these applied frictional forces generates shear stresses within the fluid. The shear stresses acting on the plate surface are proportional to the linear velocity gradient (du/dy) across the height (H) of the flow for a Newtonian fluid. The coefficient of proportionality being known as the viscosity of the fluid. The higher the viscosity of the fluid, for the same velocity gradient, the higher the shear stress that acts upon the plate surface. Shear stress (τ) is a result of both the velocity gradient and the dynamic viscosity property (μ); see eqn. 2.2.

$$\tau = \mu \frac{du}{dy} \tag{2.2}$$

When the effects of viscosity are now included, how does the flow field around a moving body differ from that of one moving through a perfect fluid? There are in fact two answers. In the case when a body that is streamlined, like an aerofoil, the drag will be mainly as a result of frictional forces. In these situations, the drag effects are relatively small unless the body is moving at high speeds or is moving through fluids which has large viscosity (such as glycerol). A more common situation is one that is encountered by so-called “bluff bodies”. In this scenario, drag arises from the combination of pressure fields that are being exerted on a thin layer of fluid near the wall of the body called the boundary layer. Adverse pressure gradients may give rise to boundary layer separation and this leads to the formation of a relatively large wake behind the object and this results in form drag (pressure-based drag). This produces a flow field that looks completely different to an ideal (inviscid) fluid scenario. To better understand the mechanisms behind separation and the resulting form drag, one must explore boundary layer theory.

The term “boundary layer” was first coined by Prandtl (1904) and the concept describes the thin region of fluid adjacent to the wall of a body. This region is a result of the viscous effects discussed previously where a no-slip condition occurs at the wall. The easiest way to visualise this phenomena is to observe boundary layer growth across a flat plate that is aligned parallel with the flow direction (Figure 2.3). Before the fluid encounters the plate, the velocity of the fluid is uniform. As soon as the flow travels across the plate, the effects of viscosity cause fluid to be at rest at the surface. The adjacent layers of fluid are also slowed down due to the cohesive forces that exist between fluid molecules. Traversing away from the wall, in the wall normal direction, the effects of friction weaken and the velocity of the fluid particles increase towards the free stream velocity. Eventually, at some distance away from the wall, the fluid particles are unaffected by friction. The velocity that these particles have is called the free-stream velocity (U_∞). The distance between the zero velocity encountered at the wall and where the free-stream velocity begins is called the boundary layer thickness (δ). As the flow continues down the plate, in

the streamwise direction, the boundary layer thickness increases due to the action of shear stresses that further retard the fluid in the boundary layer.

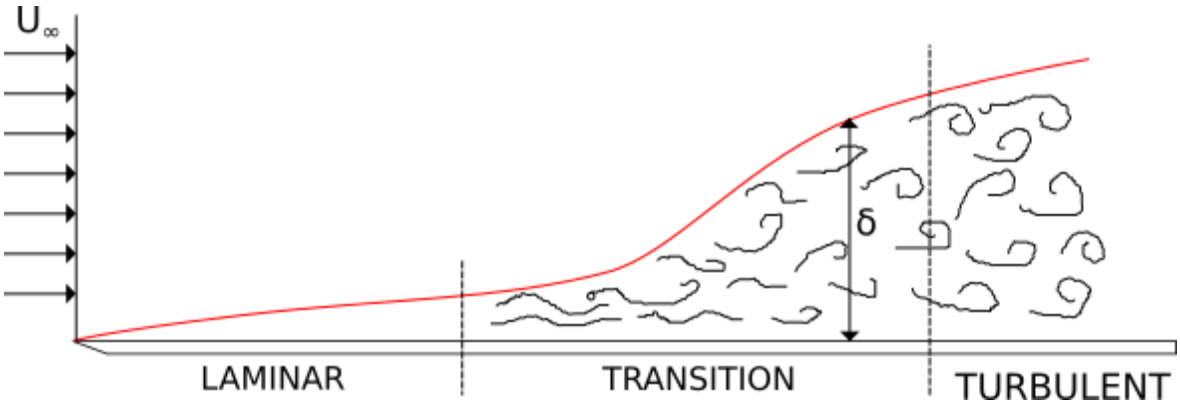


Figure 2.3. Diagram showing an exaggeration of boundary layer growth on a flat plate.

In a bluff body scenario, not only do to the fluid particles slow down due to the effects of friction (boundary layer) but their motion is also being opposed by adverse pressure gradients. The momentum of the particles reduces until a point at which the boundary layer no longer has enough energy to overcome the opposing pressure gradient. At this point, the velocity gradient at the wall becomes zero and the boundary layer flow separates from the surface, flowing over the stationary fluid. This location is called the separation point. Downstream from the separation point, fluid travels against the direction of the bulk flow and vortices are formed, giving rise to a complex wake. The presence of a wake is responsible for the large changes in pressure distribution around a bluff body, resulting in form drag. It is therefore important to make the link that boundary layers, under the influence of adverse pressure gradients and wall friction dictate the location of separation and hence, the size of the wake.

Not all scenarios experience form drag problems though. Flow through a long length of straight pipe experiences only surface drag effects, but substantial losses can still occur. Reynolds (1883) performed an investigation into the changing flow conditions in pipe flow. By injecting dye into the pipe flow, as a flow visualisation technique, Reynolds was able to make some interesting

observations. One of the conditions showed dye move along the pipe with very little deviation and the streak remained clear amongst the water throughout the entire length of pipe. This condition can be referred to as laminar flow. In a laminar flow regime, streamlines of the flow remain parallel to each other and provided the pipe conditions are steady state, the flow field should remain reasonably consistent with time. Two more complex flow conditions were observed: transitional flow and fully turbulent flow. In the transitional state, the dye initially appeared to have the same characteristics as described in the laminar state, but as it progressed further through the length of the pipe, the streak diluted partially and some unsteady deviation occurred. In the fully turbulent state, the dye rapidly disappeared from the point of injection and any remaining dye was observed to fluctuate chaotically. The work of Reynolds demonstrated that the previously discussed examples (pipe flow and the boundary layer growth over a flat plate) are not only affected by frictional forces but also inertial forces The ratio of these forces can be described by the dimensionless property called the Reynolds number (Re) (eqn. 2.3).

$$\mathrm{Re} = \frac{\text{Inertial Force}}{\text{Viscous Force}} = \frac{\rho \mathrm{U}^2/L}{\mu \mathrm{U}/L^2} = \frac{\rho \mathrm{U} L}{\mu} = \frac{\mathrm{U} L}{\nu} \qquad (2.3)$$

Where, L, is the characteristic length scale. In the case of pipe flow, this would be the pipe diameter. U is some appropriate velocity scale, for a pipe it is the mean bulk velocity and v is the kinematic viscosity.

Flow perturbations exist in all real flows but the generation of them and their magnitude varies considerably. If the viscous forces are dominant within a flow, the perturbations present are damped by viscous forces and diminish in size. As the inertial forces in the fluid become greater, the ability to dampen perturbations reduces and once the flow reaches a critical value of Reynolds number (Re_{crit}), the perturbations, instead of being damped are amplified, resulting in turbulent flow. The characteristics of turbulent flow are best described by Stewart (1968) as “turbulence syndrome” (i.e. a set of symptoms that all must exist for the flow to be considered turbulent). Stewart describes the three symptoms as:

"…**disorder**, not reproducible in detail…"

"…**efficient mixing**…"

"…**vorticity**, irregularly distributed in three dimensions…"

Even though Stewarts' description seems simple, it is extremely useful in describing turbulent flow. Revisiting the flat plate, a growing boundary layer that is in the laminar regime does in fact have two dimensional characteristics. There are changing velocities traversing in the wall normal direction and changing velocities in the downstream direction. But no change is observable in the spanwise direction. Therefore the flow cannot be turbulent and any analysis of such flow can be simplified. Soon as a boundary layer becomes turbulent, these simplifying assumptions cannot be made and the flow field changes continually without any repeatability (disorder). Another of the symptoms that is particularly interesting, "efficient mixing", can be interpreted as increased rates of diffusion. When turbulent flow occurs, diffusion of heat, momentum and matter significantly increases. This transport behaviour can be extremely beneficial, such as for the dispersion of pollutants, or problematic, significantly reducing the momentum of fluid particles within a boundary layer and hence, causing large losses within a length of pipe.

It is often stated that pipe flow becomes turbulent at $Re_{crit} \approx 2300$ (Schlichting (1979), p. 39) but it is probably more accurate to specify some range of critical Reynolds number (2000-100,000 (Tritton (1988), p. 284)). The reason for this is that the different levels of perturbations give rise to delayed or earlier than expected points of transition. This results in a certain amount of arbitrariness in defining what the critical Reynolds number is for each type of flow. The level of perturbation that exists within the free-steam can be defined by turbulence intensity (Tu). The description of this parameter is seen in eqn. 2.4.

$$Tu = \frac{u'_{rms}}{\overline{u}} \tag{2.4}$$

Where, u'_{rms} is the root mean square of the velocity fluctuation at any given point and $\overline{u}$ is the mean velocity at that same point (the RMS and mean values being interpreted either as time or ensemble averages). Although the definition of Tu in eqn. 2.4 is correct, the discussion is in reference to the onset of transition in a boundary layer and the critical Reynolds number, so the definition should be based on free-stream behaviour (i.e. $\overline{u}$ can be replaced with U_∞, the free stream velocity). The ratio of these values is often expressed as a percentage. If a value of 0% occurred, which is unrealistic, no transition would take place. Depending on the level of turbulence intensity, two different transitions can take place within a boundary layer: natural (sometimes referred to as "orderly") transition and bypass transition. As well as the perturbations that exist in the free-stream, other sources of perturbations exist. For example, roughness which will be discussed later in this chapter, can also delay or enhance the transition. Just as described with perturbations that exist in the free-stream, perturbations caused by the roughness can also change the transition's route.

The method in which transition takes place within a flow is complex but incredibly important to the engineering community. If the mechanisms that lead to transition are better understood, then systems can be better designed in order to prolong or accelerate the effects of turbulence. Schlatter and Örlü (2012) studied spatially developing zero-pressure-gradient (ZPG) boundary layers in detail using direct numerical simulation (DNS). In the investigation of Schlatter et al., different transitions were observed by varying the inflow conditions and also making use of various numerically-based tripping techniques. In one of the cases, H-type natural transition is observed (named after the work of Herbert (1988)). At the inlet of the computational domain, two-dimensional Tollmein-Schlichting (T-S) waves, are generated by some harmonic forcing method. As these T-S waves move further downstream, secondary instabilities occur. These secondary instabilities take the form of Λ-shaped vortices. These vortices then break down further into "turbulent spots" through the generation of hairpin vortices. These vortices then spread and

develop into more groups of vortices until, eventually, a fully turbulent state is reached where turbulent structures cover the whole span. Using isosurfaces of Q-criterion, the second invariant of the velocity gradient tensor, a parameter devised by Hunt et al. (1988), Λ-vortices can be identified. Λ-vortices are given this description because they appear in arrowhead-like shapes composed of two legs which have counter-rotating streamwise vorticity. Sayadi et al. (2013) performed a direct numerical simulation and showed that when the T-S waves break down to form these vortices , they can give rise to aligned (K-type, named after Klebanoff et al. (1962)) or staggered (H-type) grids of Λ-vortices. Natural transition develops extremely slowly due to the weakly unstable nature of T-S waves. Consequently, the often quoted critical Reynolds number for a spatially developing boundary, $5x10^5$ (characteristic length being distance from the leading edge of the plate), can be conservative and in fact has shown to go into the millions. Arnal and Juillen (1978) performed wind tunnel experiments and witnessed the different transition that occurred when the free stream turbulence (FST) was varied between 0.12% and 1.1%. The higher levels of turbulence intensity (0.5 to 1%) showed that T-S waves were no longer the dominant disturbance in leading to transition. In fact, a much more rapid breakdown to turbulence was observed, a description of what is referred to as bypass transition. This term “bypass” was first devised by Morkovin (1968). Morkovin stated: “Apparently, with the affinity of the scales assured, we can bypass the TS mechanism altogether if we can replace it with another strongly amplifying mechanism”. Although bypass transition is not itself a single route but actually depending on the amplitude and conditions of the perturbations, various routes can be taken (Zaki (2013)).

So, when FST of ≈1% or more is present in the flow, the boundary layer undergoes bypass transition. This type of transition is not understood as well as one by an orderly process but a brief explanation of the current theory/knowledge shall be made. Bypass transition can be split into three separate stages: buffeted laminar boundary layer, intermittent turbulent spot formulation and fully turbulent boundary layer (Jacobs and Durbin (2001)). In the first stage, the laminar

boundary layer is highly perturbed by elongated streamwise perturbations (i.e. u-component fluctuations). Kendall (1991) showed that there is a remarkable similarity to these “streaks” to that of Klebanoff modes and for that reason, this how they are commonly referred to. Hunt and Durbin (1999) showed that the initial flow within the laminar boundary layer only inherits the low frequency section of the FST. This is down to the so called “shear sheltering” effect which describes the mechanism in which higher frequencies of FST are filtered out. The perturbations that are able to be inherited by the boundary layer are then amplified and elongated into the streaks mentioned earlier by the effects of shear and the streaks continue further to grow in perturbation strength downstream. This amplification can be explained by rapid distortion theory (Philips (1969)). In the second stage of bypass transition, localised perturbations cause shear layer instabilities. The DNS studies by Wu and Moin (2010) show these instabilities lead to the formation of intermittent turbulent spots amongst the streak flow. The role of Klebanoff modes in the creation of these secondary instabilities has not been proven but the role of further DNS studies to generate masses of interesting post-processed data will be the main tool in trying to determine this. These turbulent spots increase in size, both span- and stream-wise, until the entire span is covered in turbulent structures (third stage).

The turbulent boundary layer profile can be split up into two different regions which are commonly referred to as the inner and outer layers. The inner layer, which represents approximately a 1/5 of the thickness (up to $y \approx 0.2\delta$), can be in the most part be described by the law of the wall (eqn. 2.6) function apart from a thin region called the viscous sublayer as well as the buffer layer (Bradshaw and Huang (1995)). It should be stated that the proportion of inner layer stated is for ZPG (zero pressure gradient) or weakly APG (adverse pressure gradient) scenarios, whereas strong gradients result in a smaller fraction of the thickness. The velocity profile is commonly represented in dimensionless form, where the mean velocity is represented by u^+ and the wall normal distance is represented by y^+. u^+ is the velocity divided by the friction

velocity, u_τ (see eqn. 2.13), a parameter which expresses the local shear stress at the wall in units of velocity. y^+ is the y-distance divided by the viscous length scale (ν/u_τ). The viscous sublayer occurs at $y^+ \leq 5$, where the turbulent stresses are negligible and in this region, the velocity gradient is linear, described in eqn. 2.5. Above the viscous sublayer and buffer region, the log-law layer exists and can be defined by eqn. 2.6.

$$\frac{u}{u_\tau} = \frac{y u_\tau}{\nu} \tag{2.5}$$

$$\frac{u}{u_\tau} = \frac{1}{\kappa} \ln \frac{y u_\tau}{\nu} + C \tag{2.6}$$

Where κ is the von Kármán constant and C is an arbitrary constant. From experimental data, eqn. 2.6 applies from a y^+ value of approximately 30 to 50, using κ and C values of 0.41 and 5.0, respectively (Bradshaw and Huang (1995)). It should be noted that these regions and constants are just proposed by Bradshaw and Huang. They are not certain values and other publications have suggested alternative values (Zanoun and Durst (2003) and Castro et al. (2013)).

2.2 Effects of geometric surface alteration

Now that the effects of viscosity and the resulting boundary layers that cause surface drag related problems have been discussed over a “smooth” surface, this literature review shall now concentrate on the effects of roughness, alterations of the surface structure and some of the interesting beneficial functions of these surfaces.

One of the earliest documented investigations into the effects of roughness was the one carried out by Hagen (1854), who looked at pressure losses in relation to the roughness of water channels. Then, nearly fifty years later, Nikuradse (1933) performed the more well-known flow through rough pipes study. This comprehensive study created artificial roughness in pipes by applying closely compacted sand to the pipe wall with a thin lacquer and through the use of a

pitot-static tube, the pressure drop was measured. With the use of various pipe diameters (25-100mm) and sand grain sizes, different reciprocal relative roughness (R/k_s) values (15-507) were tested. Where R is the pipe radius and k_s is the sand grain roughness size. Nikuradse was able to control the sand grain size accurately by passing ordinary building sand through a series of fine sieves. The pipe flow conditions were varied from a Reynolds number of 600 to 10^6 and for each roughness, the resistance factor (λ) was calculated (see eqn. 2.7).

$$\lambda = \frac{dp}{dx}\frac{d}{\overline{q}} \tag{2.7}$$

Where dp/dx is the pressure change per unit length of pipe, d is the diameter of the pipe and $\overline{q}$ is the dynamic pressure, taken from Bernoulli's equation, calculated by eqn. 2.8.

$$\overline{q} = \frac{1}{2}\rho\overline{u}^2 \tag{2.8}$$

Where ρ is the density of the fluid and $\overline{u}$ is the mean velocity. From the experimental data, three separate regions are identified and are referred to as hydrodynamically smooth, transitional and fully rough. In the first range, the roughness is so small that it exists purely within the laminar sublayer and for that reason, the surface acts like a "smooth" one and no additional losses are created by the roughness. This means that the resistance factor in this regime is only a function of the Reynolds number and not the relative roughness, which led to the expression given in eqn. 2.9.

$$\lambda = \frac{64}{Re} \tag{2.9}$$

In the transition region, some of the relative roughness can behave like a "smooth" surface initially and Blasius' Resistance Law applies (eqn. 2.10) to predict the resistance factor but eventually, the resistance is completely dependent on the relative roughness. For some of the highest relative roughness, the resistance at no point can be approximated by eqn. 2.10. In the transition region additional losses are generated by some of the roughness structures (peaks)

penetrating past the height of the laminar sublayer and for that reason an additional form drag is generated.

$$\lambda = \frac{0.316}{Re^{1/4}} \tag{2.10}$$

In the final region, completely rough, all the roughness peaks are generating form drag and this is the main contribution to pressure losses (resulting in a quadratic law of resistance). In this region, the resistance factor is completely dependent on the relative roughness (eqn. 2.11), which means that further increases in the Reynolds number do not alter associated resistances.

$$\lambda = \frac{1}{\left(1.74+2\log\frac{R}{k}\right)^2} \tag{2.11}$$

In summary:

i. Hydrodynamically smooth: $0 \leq k_s^+ \leq 5$

ii. Transition: $5 \leq k_s^+ \leq 70$

iii. Completely rough: $70 \leq k_s^+$

k_s^+ is referred to as the roughness Reynolds number (eqn. 2.12), where u_τ is the friction velocity calculated by eqn. 2.13.

$$k_s^+ = \frac{k_s u_\tau}{v} \tag{2.12}$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{2.13}$$

Even though the study by Nikuradse was one of the first to make a link between surface roughness and frictional based losses, it should be pointed out, this roughness, which is described by k_s^+, only applies to sand-based roughness. It was not until the work of Moody (1944), who used experimental data generated by Colebrook (1939), were links were made between commercially available metallic piping to the sand-based roughness seen in the study by Nikuradse. For this reason, this is why k_s is usually referred to as the effective roughness because other roughness is

being equated to having the same effects as a certain magnitude of sand roughness size. The study by Colebrook showed that manufacturing processes, such as casting, which are used to produce pipes, can result in an exceptionally rough pipe which will result in significant pressure losses.

To include the effects of the roughness on the boundary layer profile, another term, Δu^+, is added to the equation that describes the logarithmic region (eqn. 2.6). Δu^+ is often referred to as the roughness function. Scholz (1955) showed that as k_s^+ increases beyond hydraulically smooth for rough surfaces, the value of the roughness function increases, leading to a downward shift in the log-law profile.

Generally though, these earlier studies showed that as the roughness height increases for a particular diameter of pipe, the resistance to flow increased. However, this is not always the case, not all roughness adheres to the trend seen in the aforementioned experimentation and it is being realised that some roughness profiles actually result in a reduction in the surface drag, in comparison to that generated over a smooth surface.

2.2.1 Surface drag reduction

The most commonly referred to roughness when considering drag reduction is that observed on the skin of sharks. It is well known that the skin of sharks gives rise to exceptionally low drag properties and for that reason, plenty of research has been related in trying to replicate or mimic the effects through engineered, micro-structured surfaces. The shark skin surface is made up of complex, micro-sized, organic geometries referred to as dermal denticles (they are referred to as this because they appear tooth-like on the skin) (Reif (1985)). These scales are complex in shape but viewing them simplistically, they have grooves aligned with the flow direction and have sharp tips which stick out of the viscous sublayer (Zhang et al. (2011b)). Due to the complexity of the

geometry and the resulting 3D flow field, the drag-reducing mechanism of shark skin is not well understood. Zhang et. al. attempted to try and resolve this by performing a computational fluid dynamics (CFD) study on shark skin. This study scanned biological samples from a shark to create a CAD model of a single scale. Fully-developed channel flow was modelled by using a smooth wall and a rough wall, whose surface was formed by an assembly of the single CAD model. The digitally realised scales were angled at 15^{o} from the wall, this being the average angle determined from the scanned biological sample. The channel was setup to be 20 times the scale height ($h_{channel}=20k_{scale}$, $k_{scale}\approx11.7\mu m$), a height sufficient enough to reduce interaction of the surface flows and the average bulk flow velocity was varied from 5.1 to $6ms^{-1}$, where the highest velocity was stated to be above the critical Reynolds number and to be fully turbulent. Drag reduction was observed across the entirety of the flow conditions, ranging from a minimum of 7.09% at $5.1ms^{-1}$ to 13.63% at $6ms^{-1}$. Using the data provided by Zhang et. al., at the highest velocity of $6ms^{-1}$, k^{+}, which is the roughness height divided by the viscous length scale (ν/u_{τ}), was calculated to be 5.6. The distance (in viscous length units) between the two grooves on each scale and the spanwise distance between the patterns (wavelength) was found to be 28.6 and 54.8, respectively. The value of k^{+} means the roughness protrudes above the laminar sublayer, making it transitionally rough. According to the earlier specified definition this should increase the frictional resistance over one that is hydraulically smooth. This is obviously not the case. Luo et al. (2014), who summarises on published numerical analyses, concludes the drag reducing mechanism is a result of two key flow phenomena, which shall be explained with the aid of Figure 2.4. The first flow phenomena is a result of the tips extending beyond the laminar sublayer. The tips are shown to inhibit turbulence but have minimal frontal area in this region, in order to keep the associated form drag small as possible. Secondly, the attack angle of the shark skin scales results in a decrease in turbulence intensity. Luo et. al. explains that this is a result of “the occurrence of tiny back flow on the valley of the scale”. The paper makes links with the scale’s ability to generate rotating regions of fluid, due to the groove and tip geometry and lower shear stresses acting on

the wall but further investigation is needed to link the effects with the inhibiting effects on turbulence. The CFD study selected the k-ε RNG (Re-normalisation group) model with enhanced wall treatment for turbulence modelling. The RANS (Reynolds averaged Navier Stokes) based choice was selected because of the computational requirements relating to the high mesh densities required to resolve the flow over such organic surfaces. From the findings of Luo et al., it may be concluded that further work could be undertaken using the large eddy simulation (LES) and direct numerical simulation (DNS) turbulence simulation techniques. These modelling techniques will allow for a more detailed evaluation by resolving the flow structures generated by the shark skin surface.

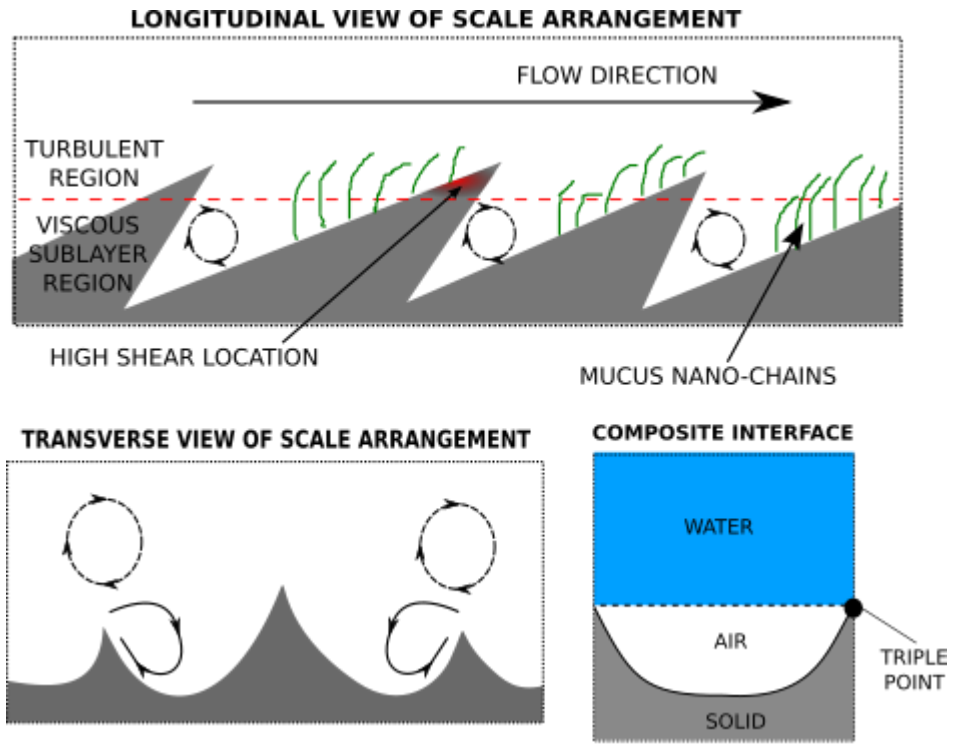


Figure 2.4. A schematic of the flow over shark skin and also an illustration of features that lead to the superhydrophobic characteristics.

Luo et. al. also mentions a third phenomena, which is not represented in the numerical model but does contribute to the drag reduction properties of shark skin is the superhydrophobicity of the surface. Daniel (1981) and Ball (1999) showed that as a shark accelerates through water, glands

are forced to secrete nano-chains of mucus. A so-called hierarchical structure is then present, made up of a microstructure, the denticle form, combined with the nanostructure of the mucus. This promotes a low surface energy fluid-solid interface, encouraging “slipping” at the wall (Jung and Bhushan (2010)). Jung and Bhushan showed that the presumption of a non-slip wall starts to break down once an interface shows hydrophobicity. This is because the adhesive forces are weaker and fluid molecules no longer have the same velocity as the surface speed, hence reducing the velocity gradient near the surface. Another feature that contributes to the exceptionally low surface energy is the composite interface (see Figure 2.4). When the shark comes up to the surface of the water, air is introduced into the pockets of the denticles. When the skin is then exposed to water again, air remains trapped in these pockets and results in a liquid-gas-solid interface (composite interface). This interface, which can be described by the Cassie-Baxter equation (Nosonovsky and Bhushan (2008)) is not particularly stable, so eventually this interface does decompose and will need the reintroduction of air at some point but for the time it does exist, Jung and Bhushan (2006) have shown that an already hydrophobic “rough” surface shall have further hydrophobic effects and will result in smaller frictional resistances.

Another assumption in the modelling by Zhang et al. (2011b) is that the scales are rigid. Lang et al. (2011) came to the conclusion that the dermal denticles are in fact pliable and move under the influence of the flow, passively. In fact, Oeffner and Lauder (2012) determined the magnitude of the error related to this assumption, experimentally, measuring a 12.3% difference between rigid and pliable sharkskin foils.

With the complexity of this pliable three-dimensional surface geometry, it is easy to see why engineers have not been able to produce accurate, sharkskin-inspired surfaces for industrial applications. Instead, these surfaces are often simplified to capture the essential structures linked with drag reduction such as the often investigated, two-dimensional riblets. These riblets are usually designed to represent the streamwise patterns seen on shark skin. Some of the earlier

riblet investigations (Walsh (1982) and Walsh (1983)), showed up to 8% drag reduction using a range of different sized V-shaped grooves and spacing. Walsh demonstrated that provided the spacing and the height of the grooves measured less than 25 viscous length units, drag reductions were observed.

Some effort has been made to summarise a large collection of published research surrounding turbulent flow over patterned surfaces (see Table 2.1) and for each of the investigations described, the best performing surface has been noted in Table 2.2 with the resulting drag reduction it achieved. It should be noted that not all of the investigations mentioned are listed, only those which provide all the required information to calculate the salient parameters. The surface patterns have been described by the dimensionless roughness height, k^+ and the spacing, s^+, between each repeating pattern (made dimensionless by dividing the length by the viscous length scale). Revisiting the earlier statement made by Walsh, that all of the streamwise-orientated riblets, which have dimensions smaller than 25 wall units, all show drag reduction. The variation in drag reduction for this riblet orientation is quite large though (3-34%) and there seems to be little correlation between the spacing or height dimensions to the improvements in drag over a "smooth" surface. Orlandi et al. (2006) explored a variety of riblet sizes and tried to identify any correlation between the dimensions (or ratio of) to the roughness function. Even though the ratio, k/s, provided some degree of correlation it was shown to be very much dependent on the structure shape, so it could not be used universally for riblets. García-Mayoral and Jiménez (2011) also came to the same conclusion when data from various studies was used to find optimal spacing and roughness height parameters. This investigation then made use of an alternative length scale, ℓ_g^+ (eqn. 2.14), which is the square-root of the riblet groove cross-sectional area (A_g^+).

$$\ell_g^+ = \sqrt{A_g^+} \tag{2.14}$$

When optimal values of ℓ_g^+ were plotted against drag reduction, there was much improvement on the correlation. In fact, Garciá-Mayoral and Jiménez state that the optimum values for s^+ and k^+ have approximately 40% scatter when plotted on a histogram, as opposed to only 10% scatter for the ℓ_g^+ optimal value, which was found to be 10.7±1. Looking at Figure 2.5, where the dimensions of the riblets (from Table 2.2) have been plotted against drag reduction, the majority of values for ℓ_g^+ do in fact lie near the stated optimal range (when the effect of the surface is seen as beneficial).

Table 2.1. Summary of literature on turbulent flow over patterned surfaces.

No.	Researcher	Flow Type	Pattern	Exp./CFD
1	Walsh (1982)	Channel	2D V-Groove Stream.	Exp.
2	Choi et al. (1993)	Channel	2D Sawtooth Stream.	CFD DNS
3	Goldstein and Tuan (1998)	Channel	2D Sawtooth Stream.	CFD DNS
4	Bechert et al. (2000)	Channel	2D Blade Riblets Stream.	Exp.
5	Chu and Karniadakis (1993)	Channel	2D Sawtooth Stream.	CFD DNS
6	El-Samni et al. (2007)	Channel	2D Blade Riblets Stream.	CFD DNS
7	Bechert et al. (1997)	Channel	2D Blade Riblets Stream.	Exp.
8	Martin and Bhushan (2014)	Channel	2D Blade Riblets Stream.	CFD LES
9	Chen et al. (2013)	Channel	3D Herringbone Riblets	Exp.
10	Stenzel et al. (2011)	Airfoil Surface	2D Trapezoidal Groove Stream.	Exp.
11	Frohnapfel et al. (2007)	Channel	2D Square Groove Stream.	Exp.
12	Choi (1989)	Flat Plate TBL	2D Sawtooth Stream.	Exp.
13	Baron and Quadrio (1993)	Flat Plate TBL	2D Sawtooth Stream.	Exp.
14	Park and Wallace (1994)	Flat Plate TBL	2D V-Groove Stream.	Exp.
15	Walsh (1983)	Flat Plate TBL	2D V-Groove Stream.	Exp.
16	Chatzikyriakou et al. (2015)	Channel	3D Hemispherical Elements	DNS + LES
17	Sutardi and Ching (2003)	Flat Plate TBL	2D Square Groove Trans.	Exp.
18	Gruneberger and Hage (2011)	Channel	2D Trapezoidal Groove Stream.	Exp.
19	Gruneberger and Hage (2011)	Channel	2D Trapezoidal Groove Trans.	Exp.
20	Bixler and Bhushan (2013)	Channel	2D Rectangular Groove Stream.	Exp.
21	Bixler and Bhushan (2013)	Channel	2D Sawtooth Stream.	Exp.
22	Zhang et al. (2011b)	Channel	3D Shark Skin	RANS CFD
23	Zhang et al. (2011a)	Channel	2D Sawtooth Stream.	RANS CFD
24	Saravi et al. (2014)	Flat Plate TBL	2D Serrate-Semi-Circ. Stream	Exp.

Notes for Table 2.1. Stream. (streamwise) = aligned with flow direction. Trans. (transverse) = spanwise pattern. (See Table 2.2 for details on best performing pattern and resulting drag reduction for each reference.)

Table 2.2. Dimensions of best performing pattern for each reference (Table 2.1) and achieved drag reduction.

No.	k^+	s^+	k/s	l_g^+	Drag Red. (%)	No.	k^+	s^+	k/s	l_g^+	Drag Red. (%)
1	20.0	17.0	1.2	-	4	13	12.0	12.0	1.0	8.5	6
2	20.0	17.0	1.2	13	6	14	14.0	28.0	0.5	-	4
3	8.3	23.0	0.4	9.8	3	15	10.0	15.0	0.7	-	8
4	9.6	16.0	0.6	12.3	9	16	10.0	20.0	0.5	-	(-7)
5	17.1	17.1	1.0	12.1	6	17	128.0	128.0	1.0	128.0	(-4)
6	8.9	17.7	0.5	12.4	11	18	8.5	17.0	0.5	10.6	8
7	8.2	16.4	0.5	11.5	10	19	0.8	1.3	0.6	0.8	(-0.3)
8	10.0	25.0	0.4	15.7	12	20	2.4	15.0	0.2	5.0	19^u,34^c
9	14.0	23.0	0.6	12.7	16	21	15.0	15.0	1.0	10.6	26
10	9.0	18.0	0.5	11.3	6	22^i	5.6	28.6	0.2	-	14
11	0.8	2.4	0.3	0.8	25	23	13.7	18.6	0.7	11.3	9
12	13.0	20.0	0.7	11.4	3	24	11.0	19.0	0.6	-	7

Notes for Table 2.2. Brackets designate drag increase. Superscript notations: ^{u} uncoated, ^{c} hydrophobic coating, ^{l} dimensions stated related to streamwise grooves on scale. -, not all data is available to calculate l_g^+.

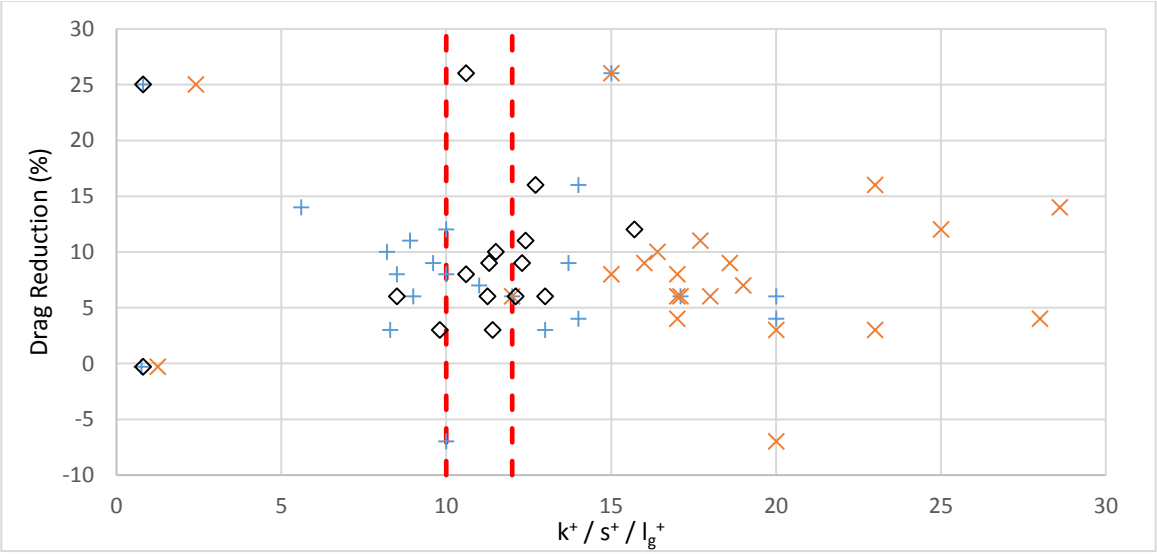


Figure 2.5. Comparison of riblet dimensions against drag reduction using data in Table 2.2.

Note: Orange x, s^+ spacing; blue +, k^+ roughness height; black ◊, ℓ_g^+; red dotted line, optimal ℓ_g^+ range determined by García-Mayoral and Jiménez (2011). Pattern number 17 has not been included, considered far outlier. Pattern 19 was omitted, due to surface structure not being optimal, only included in data for hydrophobic coating effect comparison.

One study, that by Frohnapfel et al. (2007), does not seem to follow the trend of drag reduction near the optimal ℓ_g^+ value. In this study, Frohnapfel et. al. measures the pressure drop through a channel and compares the differences between the smooth and grooved surface. To guarantee fully turbulent conditions, the turbulent intensity of the free stream at the inlet was measured (4%) and a trip was installed (blockage ratio of 12%) with a development length of 0.7m before the flow encountered the grooved surface. The riblet surface is made up of square, longitudinal grooves which have been milled and polished. What is interesting about the results, unlike some of the other studies mentioned, is there seems to be two ranges of k^+ where drag reduction peaks. The first range of k^+, between approximately 8 to 11 viscous length units, shows up to ≈4% reduction in drag which is comparative with the other studies. The second range of optimal conditions occurs within a much smaller range of roughness Reynolds number (0.8 to 1) and has a much higher reduction of surface drag (up to ≈25%). What is actually quite interesting as well, is the experimental work actually under-predicts the reduction in comparison to the DNS-based numerical simulations (Frohnapfel (2007)), where an actual maximum reduction of 33% is predicted. The height of this roughness is way below the viscous sublayer thickness for this particular condition and therefore can be concluded to be hydraulically smooth, where it should have no effect on the pressure drop but this data says otherwise.

The best performing surface which does adhere to the optimal ℓ_g^+, is from the study of Bixler and Bhushan (2013). Two surfaces from this study have been included in Table 2.2 because each surface achieves high drag reduction for different reasons, one optimal for the hydrophobicity effect and the other for optimal structure-only related effect. As discussed earlier, the effect of hydrophobicity is seen on the naturally occurring shark skin denticles through the use of a hierarchical surface. The combination of a segmented blade-type/rectangular-groove riblet for the microstructure and a nanostructure, which was created by dipping the surface in a solution of resin and 50nm silica particles, allowed the surface to achieve a 34% reduction on drag. Bixler

and Bhushan claim that drag reduction related to the structure is down to the lifting and pinning of turbulent vortices, balanced with the wetted perimeter of the riblet. In a drag reducing case, riblets force vortices to separate which are usually equal in size to the spacing of the riblet pattern. As well as the spacing, the riblet must also have an optimum height where the structure is large enough to lift the vortex away from the bottom surface but also not too deep. The deeper the groove is, the higher the resulting wetted perimeter which subsequently, leads to greater frictional forces. Therefore if the riblet is large enough in height, the drag reducing effect of the vortex pinning and lifting mechanism is overruled by the frictional forces related to the wetted perimeter. On a normal flat surface, vortices make contact with each other and cause entanglement, an effect which Bixler and Bhushan relate to an increase of momentum losses. If the riblet geometry has a tip or blade thickness, this should also be optimised. In theory, this should be made as small as possible but a limiting factor for realistic purposes, is down to the wear properties and ability to be manufactured. On top of these riblet blades, smaller vortices with a much higher rotational speed are detected and thicker blades allow for the interaction between the two rotating regions, again, an unwanted characteristic if the surface is to be beneficial. These findings are also backed up by the work of El-Samni et al. (2007) where drag increasing cases showed no lifting of vortices and clear rotational regions remained within the riblet grooves.

The discussion about riblets so far has been in relation to longitudinal, streamwise-orientated grooves. What about transverse/spanwise orientation? Well, for comparison, two trapezoidal-based riblets have been included in Table 2.2 from the experimental work of Gruneberger and Hage (2011). Each pattern described was found to be the best performing in that particular orientation. In the streamwise pattern, an 8% surface drag reduction was observed around the optimal range of ℓ_g^+. For the transverse case, all patterns actually increased frictional losses and the data showed that the smaller the riblet dimensions, the lower the losses. Sutardi and Ching

(2003) assessed similar grooves with larger height values and the data also showed that by introducing transverse grooves, the drag was only increased. Gruneberger and Hage actually fitted the data from the transverse riblet study to the earlier discussed work of Nikuradse (1933). Unlike longitudinal riblets, transverse orientated were found to have a remarkable similarity to the sand-based roughness up to a k^+ of 15. After this point, the riblets deviate and were found to be much worse for frictional losses than sand-based roughness. In the transition to fully turbulent roughness regime, the sand-based roughness explored by Nikuradse benefits from being closely packed together and grains which experience form related drag, due to protruding past the thickness of the viscous sublayer, are sheltered in the wakes of nearby roughness elements. Due to the spacing of riblets this is not the case and hence, results in higher losses.

Even though two-dimensional based riblets have shown to be quite effective when they are optimised, they are still a simplification of biological generated surfaces. The streamwise patterns seen on shark skin are not continuous and the ribs actually appear in a staggered-like pattern. Bechert et al. (2000) came to similar conclusions and performed experimentation on both 2D and 3D staggered fin arrays. To test different configurations of riblets, an ingenious test plate design was created, made up of brass rods. The spacing between the rods allowed for spring steel sheets to be inserted, which were shaped to be either the continuous rib shape or fins with trapezoidal grooves. The test plate is attached to a force balance, allowing for the true drag to be measured, rather than approximating by use of boundary layer profile measurements. Even though one could argue that the three-dimensionality of the surface should be closer in representing the shark skin case, the experimentation performed by Bechert et. al. only reached a maximum of 7.3% drag reduction for the many different configurations tested, whereas the 2D riblet configuration managed to achieve a drag reduction of 9%. An example of a 3D surface that has been proven to be more effective but is not based on shark skin, is the herringbone pattern (Chen et al. (2013)). This pattern is based on the layout of barbs that are present on the feathers of

birds. Looking at the pattern through a streamwise orientated plane, it looks remarkably like thin-blade based 2D riblets. The three-dimensionality of the structure arises from the chevron-like arrangement of the riblets across the span. Even though Chen et. al. report on a 16% drag reduction, the similarity between the herringbone riblet and the bio-inspired surface that was tested and simulated only lies with the chevron-like arrangement and not with the actual 2D shape. The engineered surface is more of a sawtooth geometry than a thin blade-type, chosen for its ease-of-manufacture.

2.2.2 Frictional reductions in hydrodynamic bearings

The other application of functional surfaces that this research will concentrate on is hydrodynamic bearings; specifically, journal bearings. The section will not only concentrate on the effects of surface structures on the fluid film lubrication effects but also make references to other tribological benefits such as wear related improvements.

So what is a journal bearing and how does it work? Well, making reference to the diagram seen in Figure 2.6, it can be seen that the bearing operates by allowing relative motion to occur between a journal and a bush, in this case the bearing bush and a rotating shaft called the journal. A film of fluid which is usually some synthetic oil or grease separates the surfaces in relative motion. When the bearing is not loaded, the clearance or separation of these two surfaces is conformal and the only fluid phenomena is one described as Couette flow (described earlier in section 2.1), where a linear velocity profile is created between the stationary and moving surfaces. When some load, F, is applied to the bearing, the shaft does not run concentrically to the bush and displaces away from the centre point by some distance, referred to as the eccentricity. Due to the eccentricity, the clearance between these two surfaces converges at some point, resulting in the formation of a “hydrodynamic wedge”. The combination of the surface driven flow and the reducing clearance

leads to the generation of a positive gauge pressure which enables the bearing to support the load and ensure the separation of the surfaces.

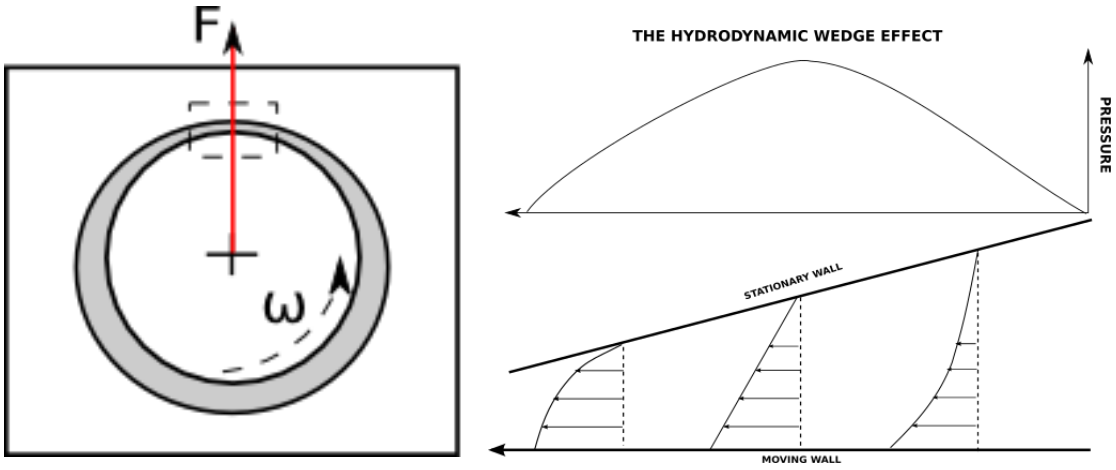


Figure 2.6. Schematic of a hydrodynamic journal bearing and the hydrodynamic wedge effect. Left image is reproduced from Wharton et al. (2016) and right image is based upon the work of Reynolds (1886).

To further improve the characteristics of the hydrodynamic lubrication, much research has been published on the introduction of structured surfaces. From the hydrodynamic film lubrication interaction aspect, the textures add additional load carrying capacity and due to the increased pressure generated, the film thickness increases and therefore reduces frictional resistances, a conclusion made by Ramesh et al. (2013). This effect is described as texture-induced lift. Another reason for structuring the journal surface is improving the performance of a bearing surface when it is starved of lubrication fluid. A non-textured surfaces would otherwise have actual contact when starved of lubricant and promote wear, creating debris in the interface. The introduction of surface structures on the other hand, are able to retain fluid better and provide some lubrication (Andersson et al. (2007)). As well as this, if contact were to occur with such a surface, debris that is created can be directed away from the interface into the pockets of the structures. Ramesh et al. carried out both experimental and analytical studies relating to the use of micro-dimples (square and circular) on the journal surface. Depth (1-100µm), pattern alignment (i.e. staggered

or aligned), width/diameter (20 to 1000μm) and texture density (4-63%) were the surface variables used in the study. Texture density is the percentage of the surface area covered by the texture area. The associated CFD simulation made use of the steady state, iterative based solver, SIMPLE (semi-implicit method for pressure linked equations), with periodic boundary conditions applied. One of the assumptions made in the model setup, was that it could be performed two-dimensionally, thus reducing calculation time substantially. The authors claim that this can be used to predict the hydrodynamic load performance of 3D-based textured surfaces. Unfortunately, this claim seems a little bold even though they back this argument up with referenced sources. Ignoring the effects of secondary flows, which would be expected in the dimples, should contribute to significant errors. The experimental study made use of a unidirectional, pin-on-disk configuration fitted with a six-axis strain gauge load cell. The textured pin and disk face was submerged in 85W-140 gear oil. This experimental setup, provided that the surfaces are parallel with each other, should be a reliable method for studying the contribution from the textures only. Any surface inclination will result in additional hydrodynamic load which cannot be attributed to the effects of the structured surface. Keeping the surface speed of $0.36ms^{-1}$ constant, the non-dimensional friction force (see eqn. 2.15) was calculated for each of the cases.

$$F^* = \frac{F}{p_A \cdot A} \tag{2.15}$$

Where F, is the friction force, p_A, is the atmospheric pressure and A is the area of contact. Experimental and analytical predictions did show similar correlations but for the majority of F^* values, an over-prediction of performance was shown by the CFD method described. If one were to ascertain the main factor in this consistently occurring error, it could be, with some certainty, be predicted to be a result of the 2D simplification. What is interesting from the experimental work, is that for all the cases except one, where similar magnitudes were observed, the introduction of these structured surfaces reduced the frictional resistances. As well as the

reduction of frictional losses, textured components showed much improved wear characteristics. The research did not make any effort to quantify wear after the samples were tested but SEM (scanning electron microscope) apparatus was used to inspect samples, qualitatively. The SEM images revealed scratch-like marks only on surfaces which had not been geometrically structured, no wear was visible on surfaces that were. The study concluded that the dimpled surfaces were most effective when there was a texture density of 20-30% and the depth was approximately equal to the designed film thickness.

Another approach to assessing the performance of these surfaces, was that by Andersson et al. (2007). This investigation tested the samples in a more hostile manner by exposing them to limited amounts (starved) of lubrication. The tribometer used for the experimentation was a ball and disk setup. A polished ball was oscillated back-and-forth over a laser-processed, surface patterned, high-speed steel disk. Before each of the trials, a finite amount of oil (≈ 50µl) would be applied with a rubber scraper to imitate the starvation of lubricant. Each of the trials would run for a maximum of 1000 sliding cycles whilst monitoring the friction force during operation and if the test reached a coefficient of friction value of 0.2, the trial was terminated early. The results of the study are impressive, noting a maximum of 11 cycles for the polished surface, whereas the best performing patterned surface actually remained below the frictional resistance target for the entire trial of 1000 cycles. Even the worst performing pattern trial still outdid the polished surface by a factor of four times. Although, patterning a surface is not always found to be beneficial. Andersson et al. came to the conclusion that textured surfaces would not outperform ordinary polished surfaces if the texture density is too small; the size of the structure is either excessively large or small; the orientation of the shape is not optimal, unless it is circular, or finally, if the frictional losses between the surfaces (without the introduction of surface patterning) is already relatively small in magnitude then, again, it may be ineffective to engineer the surface in such a manner. In other words, to assess whether the proposed surface structure will be beneficial or

not, some initial optimisation should be carried out for that particular set of conditions. This conclusion is further reiterated in the publication by Syed and Sarangi (2014) who performed a comprehensive study on how different structure shapes and the aforementioned parameters relate to the hydrodynamic performance of the surface. Many different shapes (square, circular, elliptical, ellipsoidal, triangular and hexagonal) were examined, analytically, using a modified version of the Reynolds equation which actually takes the inertial effects into account. This partial differential equation which was derived in the publication by Reynolds (1886) is a simplified version of the Navier-Stokes equations. The equation, which makes the assumptions seen in the list below, describes pressure variations in thin film lubricant flows.

i. Lubricant flow is assumed to be laminar as viscous forces dominate the flow;
ii. Assume pressure is constant across film thickness;
iii. Gravity body forces are neglected;
iv. Assume lubricant behaves in a Newtonian manner;
v. Non-slip at wall, so fluid adjacent to wall inherits the same velocity;
vi. Inertial forces are neglected.

For most hydrodynamic bearing cases, solving discretised Reynolds equations over a mesh representing the geometry provides an adequate approximation of the bearing performance. Unfortunately, the assumption to ignore inertial forces starts to break down when the bearing approaches higher surface speeds ($Re_{Cou} > 8$, eqn. 2.16). In fact, as the inertial forces become more dominant, the validity of the assumption that the flow is laminar is also questionable. Venkateswarlu et al. (1990) states that when the Reynolds number is smaller than five times the critical Reynolds number (see eqn. 2.18), the assumption that the flow is laminar is perfectly valid. If the value is larger than this criteria, then the pressure field (and the resulting load carrying capacity of the bearing) will not be stable and accurate modelling of these conditions will then require turbulent modelling techniques. The critical Reynolds number that Venkateswarlu et al. is

referring to here, is based on the critical Taylor number and is calculated using eqn. 2.18. Two types of destabilising forces arise in lubrication flow within hydrodynamic bearings: inertial forces for parallel shear flows (Couette flow) and centrifugal forces for flow between concentric cylinders. The instability in the second type comes in the form of Taylor-Gortler vortices and the Taylor number, seen in eqn. 2.17, gives an approximation as to when this instability would lead to an unstable flow. The only problem with the presented critical value for the Taylor number is that it is based on the bush and journal being concentric. If large eccentricity is expected, then other critical Taylor number values must be found from existing published data. So, to determine whether the lubricant flow is turbulent, one must assess the conditions using both the Taylor number and the Reynolds number (eqn. 2.16) (Szeri (1980)).

$$Re_{Cou} = \frac{\Omega \, \mathrm{r} \, \mathrm{c}}{\upsilon} = \frac{U \, c}{\upsilon} \quad \text{(Critical } Re_{Cou} \approx 2000\text{)} \tag{2.16}$$

$$Ta = \left(\frac{c}{r}\right) Re_{Cou}^2 \quad \text{(Critical Ta}\approx 1708\text{)} \tag{2.17}$$

$$Re_{CritTa} = 41.3 \left(\frac{r}{c}\right)^{0.5} \tag{2.18}$$

(Where, r, is the journal radius, Ω, is the rotational speed of the journal (in rad/s), U, is the linear surface speed and c, is the radial clearance). Syed and Sarangi (2014) give evidence that the range of flow conditions analysed are in fact laminar but one of the assumptions that still arises with the use of the Reynolds equation, even with the inclusion of the inertial terms, is that there is no pressure variation across the film thickness. Unfortunately, with the introduction of textures onto the surface, representing all three-dimensional geometries with two-dimensional pressure fields will incur large errors. Syed and Sarangi admitted the limitations of the Reynolds equation by stating: “texture height ratio for positive texture should be kept within limits… otherwise, calculation of 2-D pressure may not be accurate”. However the research did not investigate textures that protruded beyond the minimum film thickness. Still, for the ranges of surface texture investigated, some interesting observations were made. Out of all the shapes

investigated, the elliptical structured surface provided the best load support and smallest frictional resistance but was shown to be highly sensitive to orientation. The angle of 90^{o} (i.e. perpendicular to driven surface direction) provided the best performance for this particular shape, whereas, 0-40^{o} actually was outperformed by the triangular pocket in any orientation. It was also shown that increasing the number of texture elements, for any shape in the transverse direction, per unit length, reduced the surface's effectiveness. The texture height ratio, which is the depth or height of the cross sectional shape divided by the clearance size, seemed to be optimal around the 0.5 to 0.6 range. As discussed previously, Ramesh et al. (2013) came to the conclusion that circular pockets were most effective in the region of 20-30% for the texture density parameter, similarly, Syed and Sarangi's data show 20-40% for all the different shapes investigated. Regrettably, it cannot be assumed that the elliptical shape is truly the optimal shape, as the process in which they used to come to this deduction is flawed. All of the shapes were at a set orientation for the preliminary work and then the best performing structure shapes, assumed to be triangular and elliptical *were then* rotated through a range of angles. One of the conclusions made by this work, is that some shapes are highly sensitive to orientation, so there is the question, what is stopping other shapes being more effective in other alignments?

2.3 Surface manufacturing methods

Now that the background behind the functionality of these surfaces has been discussed, the next logical step, which is often overlooked, is to consider if the surface can be actually made? It may well be that, theoretically, a particular structure will provide the minimal amount of surface drag but it could be impossible to make or could be costly to mass manufacture. This section will discuss the manufacturing methods (machining and forming types) that could be implemented using facilities that already exist. Their applicability in structuring surfaces and also their cost effectiveness for industrial application will be critically evaluated.

2.3.1 Machining

One of the more traditional and conventional machining methods used to form surfaces is milling, or sometimes referred to as micro-milling, due to the application of this method to produce sub-millimetre features. Brandner et al. (2006) used both milling and chemical etching (which will be discussed later) to produce a microstructure for use in crossflow heat exchangers. Using an end mill tool, rectangular channels of 100µm width and 70µm depth were machined into thin stainless steel sheet which was layered upon each other and bonded together. Brandner et al. stated that the surface roughness of the features always remained below 0.1% of the channel height (or 70nm). This remarkably low roughness is not replicated in the work of Frohnapfel (2007) though. In this study, considerable burrs were created on top of the thin, blade riblets. Consequently, this resulted in an additional polishing process which altered the shape of the channels from a square to a trapezoidal feature, a result of the post-processing method which removed more material from the top of the channels than at the trough. Another inherent problem with milling, is down to the very nature of the mechanical process itself. The reciprocating motion of the tool limits the complexity of the geometries that can be created and trying to replicate some of the three-dimensional, bio-inspired surfaces is pretty much near impossible with such a process but even for the manufacture of 2D riblets, there are many flaws in the process. Dean and Bhushan (2012) detailed quite a few disadvantages when micro-milling was used to produce their blade riblets. The first being related to how time consuming it is structure a surface with this method. Bechert et al. (1997) highlighted this with a reported 1-2 weeks manufacturing time for each of the 0.4m x 0.5m milled plates. Dean and Bhushan selected a smaller computer numerical controlled (CNC) milling machine to improve the accuracy needed for the micro-scale machining (as the build tolerances will be tighter). The only problem with this is that the spatial operating limits of each axis are drastically reduced in comparison to a conventional milling machine. In fact, the limits

were so small that in order to actually machine the entire surface, the workpiece had to be unclamped and moved several times. Repeated setup of the workpiece is incredibly time consuming and also reduces the accuracy of the machined part. To programme the required tool path into the machine, a G-code program has to be generated. Usually, a commercially available computer aided manufacture (CAM) software would be used to generate the G-code program but Dean and Bhushan concluded that this was not possible as the software is not designed to output toolpath coordinates to the required tolerances. To overcome this, the G-code was manually created using a spreadsheet. The last problem that was faced in the machining process was the effects of backlash. A milling machine will always have backlash errors due to the allowable tolerances that are placed on the manufacture and assembly of machine parts. If the backlash is not taken into account, even on the small CNC setup used in this study, the accumulative error would produce out of tolerance parts. So, to overcome this, the effects of backlash was taken into account when manually generating the G-code program.

Another conventional machining method that has been proven to be effective in developing structured surfaces is grinding. This can be mainly achieved by texturing the grinding wheel surface and cutting that profile into the workpiece surface. A recent review on textured grinding wheels by Li and Axinte (2016) showed that textured wheels have been used since the 1920s but their purpose has mainly been to improve the characteristics of the grinding process. Some of the early designs used slots in the wheel surface to break up the cutting interface, in order to lower cutting temperatures. Later on in the 20th century, wheels were being structured to enhance the cooling performance. An example of this was using different slots shapes on the wheel surface to improve the transport of cutting fluid (intermittently) to the interface. As concluded by Li and Axinte, it has only been in the last decade that grinding research has been able to texture wheels down to a micro-scale level with the intention of structuring surfaces. An example of this, is the study by Denkena et al. (2008) which aims to use the cylindrical grinding process to add

streamwise, trapezoidal-based riblet features of 40µm spacing and 20µm height to the surface of compressor blades in turbine engines. Prior to this investigation, it had been shown experimentally, using a wind tunnel facility, that these ground structures on a flat plate reduced skin friction losses by 3.4%. Unfortunately, one of the main obstacles that make this machining method commercially unviable is the costly form wheel dressing operation. To produce the X20Cr13 flat plate (material similar to that of a compressor blade) for the initial investigation, a diamond form roller was used to form dress the inverse profile into the vitrified bond, SiC-based grinding wheel surface (grit size of 17µm). Using the profiled wheel, a surface grinding operation is then used to manufacture the plate surface. To dress each of the channel profiles into the grinding wheel, one-by-one, is incredibly time consuming (approximately 15 minutes for 20 microprofiles) and so to improve on this Denkena et al. experimented with the profile dressing method. Not only was the profile dressing method successful, it drastically reduced the dressing operation down to 1 minute. The dressing trials looked at the influence of the dressing speed ratio (peripheral linear velocity of the dressing roller over the peripheral linear velocity for the grinding wheel), number of roll-out rotations and the radial dressing feed. Increasing the infeed of the profile roller results in a larger force being applied to the grits which in turn results in larger breakouts of the profile tips. A dressing feed ratio of -0.7 was chosen (the negative value indicating up dressing). Up dressing, as proven by the experimental results, produces a smaller grinding wheel roughness than down dressing. The number of roll-out rotations for the sparking out process was varied from 0 to 100 and it would be expected that the longer the sparking out time, the better the profile accuracy. Actually, this is not the case. If too shorter time is allowed for spark out, then the grinding wheel will be out-of-roundness but if the spark out process is too long, then the frictional forces at the cutting interface build up and actually start to degrade the surface profile. After the dressing trials were performed in order to select the appropriate dressing parameters, the investigation looked at modifying the channel sides. As the trapezoidal channel sides were made steeper, the stresses acting on the grinding wheel profile were

increased, particularly at the tips of the wheel surface profile (proven in a 2D finite element, static analysis of a single, homogeneous grinding wheel surface micropattern). This was reflected in the grinding trials by the increased tool wear rates and larger dimensional errors associated with the surface profile. This grinding study has shown that this way of manufacturing surfaces can be applied to large surface areas quickly and cost effectively, unlike the milling method. Although some care may need to be taken in selecting parameters, to ensure stresses at the profile tips are minimised for the designed tool life. Otherwise, frequent re-dressing will be occur, incurring large per part costs and profile errors will be unacceptable. Another alternative solution to this is to improve the rigidity of the cutting profile. Denkena et al. (2010) achieved this by using a CBN based grinding wheel to produce the same patterns. Not only did the study show that a CBN wheel reduced tool wear rates, in comparison to the SiC material but it also allowed for steeper profile geometries. Denkena et al. made the following conclusions from the study:

i. Riblets of 60µm spacing and aspect ratio of 0.5 were manufactured successfully;
ii. Spiral side burrs, which occur at the profile tips, can be eliminated with the use of a higher workpiece speed in the sparking out process;
iii. A surface removal rate of $3cm^2min^{-1}$ can be achieved;
iv. A riblet tip of 1µm radius was manufactured;
v. The resulting riblet geometry achieved a 4% reduction in surface drag at a s^+ of 18;
vi. Tip profile angles of 60^o and above can be produced.

Another approach to structuring a surface is through the use of chemical etching (Wang et al. (2003), Brandner et al. (2006)). As previously stated, Brandner et al. used both milling and etching to create microchannel exchangers. The study made use of photolithography based chemical etching to create three-dimensional textured surfaces (aligned and staggered square column arrays). The first part of this manufacturing method uses a negative resist process where the

areas that are wanting to be kept un-etched are exposed to UV light. Areas which are not to be exposed to UV light are physically masked by some method. Areas where the photoresistive layer has been exposed to UV light become polymerised and this results in the polymer layer being harder to dissolve when exposed to the chemical. The entire surface is then exposed to the corrosive chemical and any non-exposed areas are essentially dissolved away by the etchant. Brandner et al. concluded that chemical etching not only allows for a surface to be structured in a more cost effective manner than milling but it actually enables for more complex, three-dimensional surfaces to be created. However, there are some disadvantages with this process. Surfaces that are created with this process tend to be pretty rough. The foils etched in this study have mean roughness values of micrometres or a few percent of the channel height. Though it should be noted that this particular study was interested in heat transfer functionality, so the inherent roughness of the surface actually promoted turbulent flow, enabling for a better dissipation of heat. In regard to the objective of this research though, such surfaces would not be particular useful in trying to minimise surface drag. Another drawback to this method, is down to the underlying chemical process which results in features being made up of full radii. Surfaces that are designed to have sharp corners and flat edges cannot be produced by this manufacturing method as the sharp edges are subject to more rapid dissolution than planar surfaces.

The last machining method to be discussed is laser-based techniques. Mishra and Polycarpou (2011) used the laser surface texturing (LST) technique to create circular dimples on the surface of pin samples for pin-on-disk based tribological experiments (pin and disks were made from grey cast iron). A range of pocket diameters, 40-60µm and depths, 4-10µm, were successfully manufactured. After the manufacturing process had been performed, the surfaces were examined and material “pile-up” or bulges were found around each of the circular pockets. These so-called bulges are common when micro-machining metals and are a result of the LST method which causes material to melt and solidify around the edges of the pockets. The height of the

bulges were significant in proportion to the depth of the dimple features, ranging from 0.8 to 1 for the bulge height to pocket depth ratio. Zum Gahr et al. (2009) , Andersson et al. (2007) and Braun et al. (2014) showed that once the surface has been laser textured, an additional polishing process can be effective in removing such defects and debris with little effect upon the internal surfaces of the pocket geometries. On the other hand, Mishra and Polycarpou did not employ any post-processing method because an initial attempt to polish the pin surface led to an out-of-tolerance flatness value. Unfortunately, retaining these defects in a hydrodynamic bearing application would lead to excessive wear rates during the run-in period and even though these unwanted features would eventually be worn away, the large amount of debris generated in the interface would almost-certainly shorten the operational life of the component. Mishra and Polycarpou performed wear and durability tests for each of the samples and found that even with these defects contributing to increased wear rates during the initial stages, the friction coefficient was still measured as being lower than the non-textured surface (that surface was the only one to not complete the full three hour testing period). Wear rates were also calculated for the trials and concluded that, even with the removal of the bulge features, they were similar in magnitude to that of a protective tribological surface coating. It should be noted that not all materials display this pile-up defect, for example the work carried out by Wakuda et al. (2003) and Wang et al. (2001) looked at LST ceramic-based components and this feature was not present. Although, it should be said, some initial trials may be needed in order to minimise the amount of thermal energy being introduced to the surface. All of the component surfaces processed in the work by Wang et al. have been subjected to too much thermal energy, indicated by the small cracks that surround each of the pockets. Such cracks, which appear consistently across the component surface, are detrimental in the acceleration of fatigue related failures. Very little details are given to the reader about the laser specification and parameter selection, so it is hard to make any comparisons between this article and other mentioned publications. So how economically viable is the LST method? Schreck and Zum Gahr (2005) laser texture 100Cr6 bearing steel plates (6mm x

15mm) for a cylinder-on-plate tribology test. Two different types of surface patterns are manufactured: dimples and channels. Using a Nd:YAG laser, set at 5kHz pulse mode, it is reported that the dimples require 12 pulses to attain the required depth (10µm) and diameter (65µm) at a laser power of 5W. So, for the largest texture area density of 55% (best performing), it would take approximately 35.8s. For the channel based geometry, three different orientations were experimentally investigated: parallel, perpendicular and crossed. Again, the higher texture area density performed best (crossed) and was found to be even better than the dimple structure. According to the report, the scanning speed was 50mm·s^{-1}. Each of the channels required 10 passes in order to achieve the required channel width (100µm and spacing between channels is also 100µm). So, for the plate to be textured with this pattern, it would take approximately 180s. (It should also be noted that these manufacturing times do not take into account the polishing process which is performed after the LST process). For such a small sample area, it can be indicated from these figures that LST is not particularly cost effective, especially when one includes the additional costs required to remove debris and pile-up defects from the surface. On the other hand, laser texturing has shown to be adequate in producing a large array of different three-dimensional structures with good repeatability, in comparison to the other machining methods described. One can also assume that as laser manufacturing technology improves, the time taken to texture the same surface areas will decrease and maybe become more economically viable in the future.

2.3.2 Forming

One process that has proven to be particularly economical at surface texturing is rolling (Pawelski et al. (1994), Hilgenburg and Steinhoff (2015)). As the sheet passes through a set of rollers, the textured roller imprints the inverse pattern onto the formed sheet and as demonstrated by Zhiqing et al. (2014), the produced surface structure can be produced with little distortion. Micro-

scale riblets can be introduced to metal sheet surfaces by the rolling process and even some of the less ductile materials (e.g. Ti-6Al-4V) have shown to be formed using a cold-working process without damaging the roller structure (Klocke and Feldhaus (2007)). What is particularly interesting about rolling, is that some of the aforementioned machining methods that would be deemed economically unviable for manufacturing a surface on its own could be used in conjunction with rolling, primarily to create the roller component surface. In the study of Klock and Feldhaus, turning was used to create the roller, a process which is ideal for manufacturing rotationally symmetric parts, with little cost or complication. The cutting edge was a modified, carbide-based indexable insert, which was ground to an almost sharp-edged point with very little corner radius. The high-speed steel (HS 6-5-2) roller was profiled before encountering a hardening process, which raised the hardness to 65HRC. The formed trapezoidal riblet structure, with ideal, thin walls, was of 340µm spacing and 162µm height. One advantageous property, proven by a finite element analysis study in this report, is that the formed, cold worked parts benefit from localised, strain hardened areas, particularly around the top of the blade-like walls of the riblet. This particular riblet geometry is setup to minimise the high shear contact areas, like the denticles of shark skin but unfortunately, these areas then suffer from increased wear rates. Coincidently, the materials from this cold worked process are strain hardened in such areas, so it may allow blade riblets to have a longer operational life. One of the limiting factors in this study, is the ability of the turning process to produce such small features. As the radius of the cutting edge is minimised, in order to make smaller riblet features, the concentration of stress becomes more local to the tool tip. This results in a cutting edge which is more prone to breakage (chipping) or certainly an increased wear rate. To try and reduce the size of the riblet features, Hirt and Thome (2008) investigated a novel process where the roller was wrapped in round wire. A structure is machined into the roller to help guide the wire for the winding process but also to resist the axial loading which would otherwise dislocate the roller profile. The smallest, scalloped-based riblets manufactured using this process was 33µm depth and 100µm spacing. This rolling method

produced excellent, sharp riblets with both good form and dimensional accuracy. Although, there is one main disadvantage to this manufacturing process, wire breakages. Due to the wire not being bonded to the roller, any wire breakages will encourage the whole winding to unravel. An additional bonding process, such as brazing, could be used so that if any wire breakages were to occur, the damage would only be local to the failure.

Another method is to actually mould the surface during the curing of paint or resin (Stenzel et al. (2011), (Zhang et al. (2011a) and Luo and Zhang (2012)). Luo and Zhang constructed a device which is an assembly of elastic, textured rollers for feeding down lengths of pipe. The idea is to apply a resin to the pipe walls and before the material has set, the device imprints a riblet surface. Luo and Zhang (2013) took this idea one step further in a process referred to as the bio-replication rolling method. Essentially, the idea behind this is to mould a silicon rubber roller against an actual biological sample of the shark skin. The roller is then used in the same way, to deform the uncured resin material. The process does not perfectly replicate the biological sample though, as shrinkage occurs in moulding the roller (≈10% reduction in final pattern height). Some difficulties also arise in trying to capture the sharp wedge angle on the back of the scales, an essential feature for the composite interface. The ability to scale the shark skin denticles for different flow conditions is also important, so Pan et al. (2013) detail an intermediate solvent-swelling-based amplification of the microstructure, which the silicon rubber roller is then cast off. This allows for denticles of any size, within reason, to be reproduced.

Table 2.3. Comparison of the machining methods reviewed.

Ranking	Manufacturing Process	Advantages	Disadvantages
1st	Rolling	- Can be applied to large surface areas; - Rollers are simple to manufacture; - Wire wound rollers can produce small, sharp features; - Beneficial strain hardening for improved wear properties	- Wire wound roller has little strength without brazing process; - Harder materials may damage roller profile;
2nd	Moulding	- Complex geometries, include the replication of biological surfaces, can be applied to large surface areas; - Surfaces of vehicle (i.e. aircraft fuselage) can be painted and pattern can be applied during curing process; - Sample features can be scaled using the solvent swelling process	- Resins and paints are susceptible to wear; - Bio-replication method requires actual biological sample; - Curing process can lead to shrinkage; - Sharp back angles cannot be produced
3rd	LST	- A whole range of three dimensional structures can be manufactured with good repeatability and accuracy	- Requires polishing to remove debris and pile-up defects; - Time consuming for large surface areas;
4th	Etching	- Three-dimensional structures can be created; - Setup costs are low (dependant on masking method)	- Not cost effective for mass manufacturing (time taken to apply mask and exposure to etchant for large areas); - Rough surface finish; - Channel geometry is limited to curved and round edges;
5th	Grinding	- Can be implemented on existing CNC machine; - Profile dressing allows for efficient texturing of grinding wheel; - Machining of components is relatively quick compared to other methods	- Processes can produce a haphazard surface with little definition; - Sharp angled features leads to large tool wear and large form errors;
6th	Micro-milling	- Channel surfaces have low roughness;	- G-code produced by CAM software is not to tolerance required (requires manual generation); - Geometries that can be manufactured are limited; - G-code needs to account for backlash; - Very time consuming. - Polishing may be needed to remove burrs;

2.3.3 Comparison and Ranking of Manufacturing Methods

To summarise, the main points relating to the applicability and cost effectiveness of each manufacturing method have been detailed in *Table 2.3*. The methods have also been ranked in terms of their ability to mass manufacture textured surfaces and it was judged from the information attained that the rolling method is currently the most effective process. The lower ranked methods reflect that these process types may need further research and development in order to be applicable for industrial application. It should be noted these processes are ranked on their sole ability to structure a surface. It is not to say that some processes, like LST, could be combined with a process like rolling to produce a surface.

The rolling method is a very effective process that make use of other elaborate texture generating processes and effectively amplify their productivity.

3. Influence of Surface Roughness on the Frictional Performance of a Hydrodynamic Bearing

3.1 Introduction

The first part of the investigation is to deduce, experimentally, whether there is a relationship between surface roughness, being described by some standardised statistical roughness parameter and the frictional forces experienced between a moving surface and a lubricant. If such a correlation exists, a manufacturing process can be optimised and assessed quantitatively by a parameter, knowing that there is a direct correlation between that and the frictional performance of the bearing surface. In other words, it would then become truly what is known as a "functional surface". A surface's effectiveness in achieving a function (in this case to minimise frictional forces) is described by some descriptive parameter (e.g. Sa). To ascertain whether this is a reliable method in assessing a surface for this application, thirteen components, which were manufactured using an abrasive tape finishing process, have been tested using a bespoke tribometer. All of the components have been subjected to roundness measurements, 2D as well as 3D roughness measurements and a wettability study; to determine the surface free energy of each sample (discussed later in the chapter). All of these measurements are then correlated, using Pearson's correlation coefficient, against the measured coefficient of friction for each sample.

3.2 Journal Components and Tribometer Apparatus

Before this research project was undertaken, the General Engineering Research Institute was in contact with a company who specialised in the finishing of crankshaft components for the automotive industry. The company was kindly asked to produce a set of crankshaft journals, varying the manufacturing process parameters for each sample in order to produce a range of stainless steel components with different roughness characteristics. An abrasive tape process was used to finish the components and each parameter selection is displayed in Table 3.1. The dimensions of the component were based on an existing crankshaft assembly manufactured within the external company. A sample connecting rod was also supplied, which formed the basis of the tribometer used in this study (see Figure 3.1).

Table 3.1. Abrasive finishing process parameter selection for each sample.

Sample	Grit Size (μm)	Index Length (mm)	Revolutions	Oscillation Speed (RPM)	Oscillation Length (mm)	Spindle Speed (RPM)
1B	9	20	15	150	1	100
2B	9	25	12	190	1	100
3B	9	20	15	100	1	100
4B	9	15	15	190	1	100
5B	9	15	15	100	1	100
6B	50	25	12	190	1	100
7B	30	25	15	190	1	100
8B	30	25	15	150	1	100
9B	9	25	15	150	1	100
C1	9	20	9	None	0	100
C2	9	25	9	400	1	100
C3	30	20	12	400	1	100
C4	30	25	12	None	0	100

3. Influence of Surface Roughness on the Frictional Performance of a Hydrodynamic Bearing

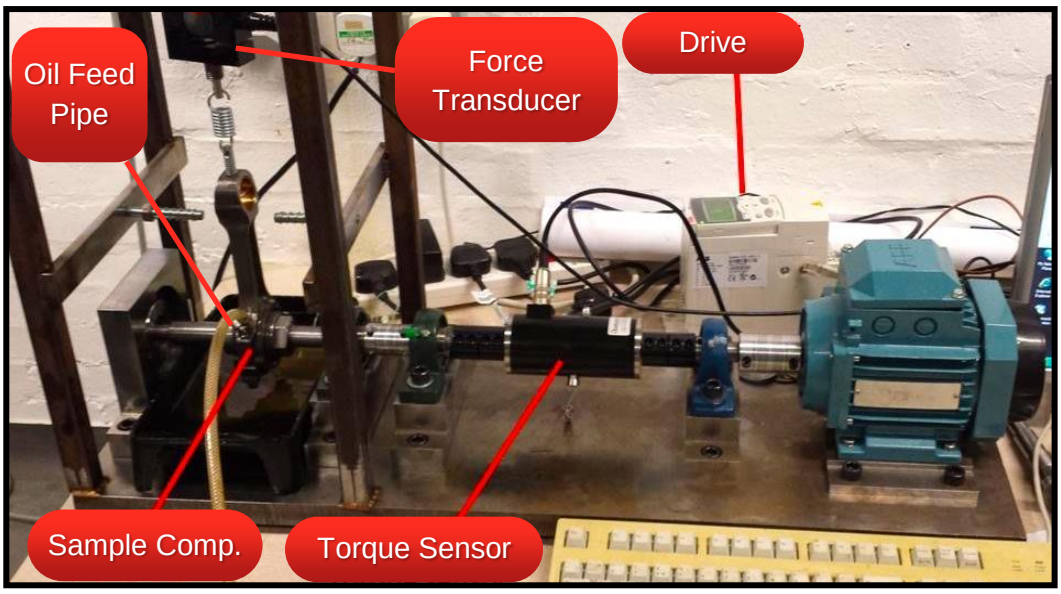


Figure 3.1. Tribometer setup.

Again, the tribometer was designed and manufactured before the project began. The idea behind this rig is to try and replicate the interacting behaviour of a crankshaft journal and connecting rod. The internal diameter of each journal sample is assembled concentrically to the shaft and fixed up against a shoulder with a large nut. On both sides of the component, a set of male and female spherical washers are used. It is important that while running the experiment that the shaft is running true so that the connecting rod doesn't drift off to the side (known as run-out). If the run-out is excessive, large vibrations will give rise to poor results, due to additional frictional contributions and may damage the rig components. Before each of the trials, the shaft is turned by hand in order to determine the amount of run-out. The nut and washer assembly is adjusted until the run-out is minimised. The shaft is driven by an electrical motor which is controlled by an ABB ACS310 general purpose drive with a pre-programmed ramp/acceleration. The target speed is manually adjusted using the on-board control panel (open-loop control process). This does mean that if a higher load is applied to the shaft, human intervention is required to carefully regulate the speed of the shaft to the correct value. A torque sensor is fitted to the shaft which sends rotational speed and torque data to the connected computer. The Datum Electronics M420

torque sensor (serial number 86871) is rated up to 10Nm torque and 2,500rev.min^{-1} rotational speed. The calibration certificate shows 1.676mV/V for a 0-10V output voltage range. To apply a load on the bearing, a threaded bar is connected to a threaded through hand knob. Adjusting this pulls the connected force transducer up and transfers the load to the connecting rod through a spring. Because this is a hydrodynamic bearing, lubricant needs to be fed into the cavity. This is done by a displacement pump with an attached reservoir full of Trident 15W40 (k at 40^{o}C is 100cSt, k at 100^{o}C is 13.9cSt and has a specific gravity of 0.883 at 15^{o}C). Oil flows from the pump into the cavity through a small hole in the side of the connecting rod. The pump flow rate is set to the minimal flow rate required to feed the cavity. The idea of the hydraulic system is not to add additional pressure contribution to the bearing, like a hydrostatic bearing setup. The load capacity of the bearing is to be only supported by the pressure generated from the hydrodynamic wedge effect of the eccentric journal.

3.3 Inspection of Samples

3.3.1 Roundness

The first two-dimensional surface measurement was performed in order to verify the overall quality of the sample, mainly for roundness. To do this, a Taylor and Hobson Talyrond instrument was used; an off-line roundness inspection tool. This tool is equipped with a turntable and a three jaw chuck centrally fitted to the top face. This particular workholding solution enables the component to be located concentrically to the turntable without any time-consuming alignment whilst gripping the sample on either the internal or external diameter, fixing it into position. A spring loaded probe arm is placed against the surface of the component and as the turntable rotates, radial deviations are measured and captured by the connected computer. Large

deviations on the journal surface are attributed to increased wear rates, higher frictional forces and problematic vibrations which all lead to a significant reduction in the designed operational life of the bearing. Also, the aim of this investigation is to link the surface structure to the performance of the bearing and having such out-of-roundness features (and also big changes in roundness values between samples) will make it near impossible to distinguish between the effects of the surface structure and the increased frictional resistances that are a product of such defects.

For this study, it would be acceptable for the roundness to be within a few microns but on the whole, all the specimens presented much higher values and were deemed low quality. The worst out-of-roundness sample was 9B (trial 2) at 15.2µm. Looking at the actual profile for this particular component (see Figure 3.2), a large peak feature is present. As discussed previously, this type of feature is not only poor for such bearing applications but will also make it hard to distinguish between potentially non-beneficial surfaces and increases in frictional forces that are contributed by out-of-roundness defects.

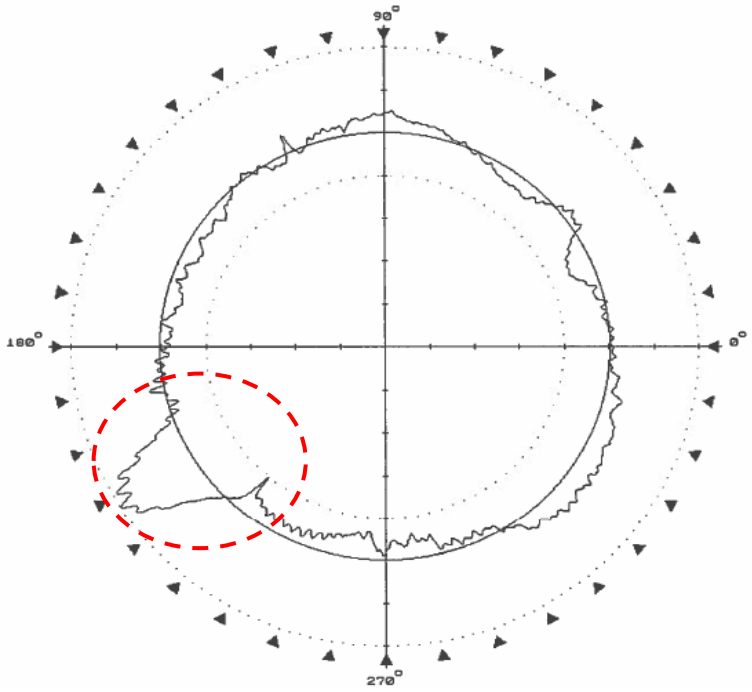


Figure 3.2. Component 9B highlighted defect. Each division represents 2µm.

In conclusion, most components presented large peak-like defects and this seems to be the main reason for the large roundness values observed across the entire batch. Due to the components being manufactured off-site at the external company, it is hard to evaluate and more importantly, control the contributing factors in the abrasive process that could result in these poor parts. In order to make results in future experimentation more reliable, it was decided that samples being used for remaining studies were in future to be made on-site within the University.

3.3.2 2D Surface Roughness

The next measurement that was performed was a 2D surface roughness study. This was performed using a Taylor Hobson Talysurf (see Figure 3.3) which is a piece of equipment that moves a probe along the surface of the component in the x-axis and stylus deviations (in the z-axis) are detected. The deviation data is sent to the connected computer and the Taylor Hobson µltra software then processes the profiles in order to generate two-dimensional statistical roughness parameters. An example profile from a measurement can be seen in Figure 3.4.

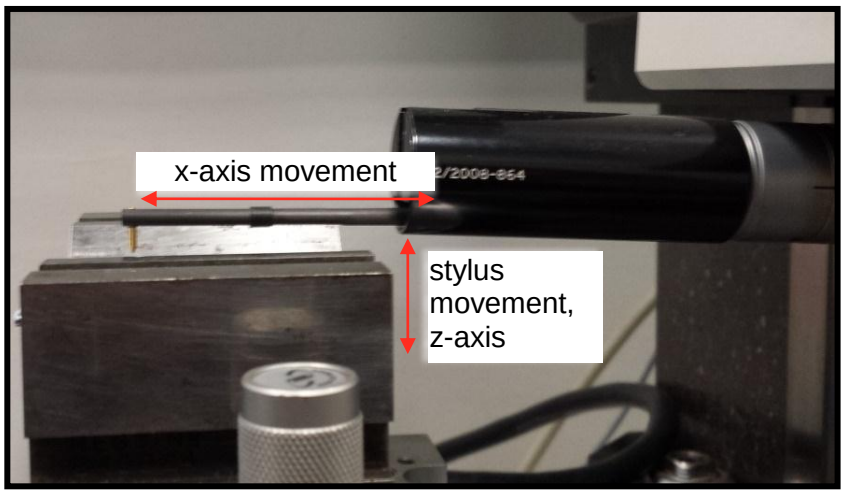


Figure 3.3 Talysurf apparatus.

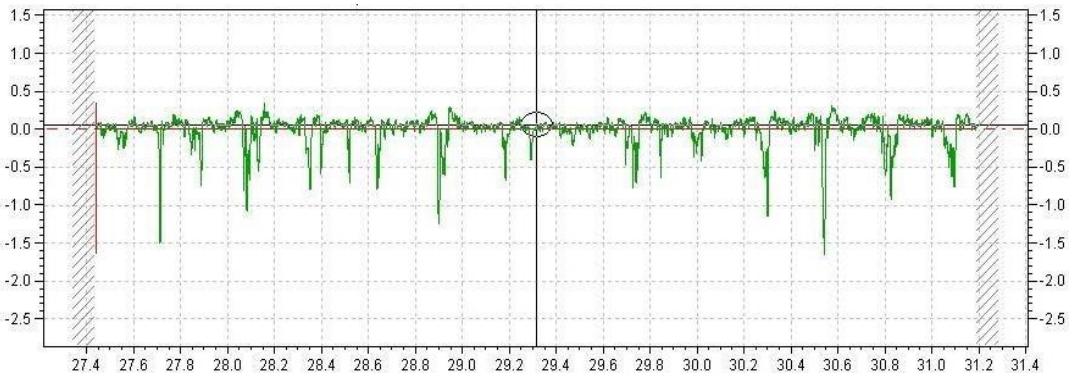


Figure 3.4. Example 2D profile. X-axis is in millimetres and y-axis is in micrometres.

Each of the components were measured three times and for each of the two-dimensional roughness parameters considered, a mean value was calculated (see Appendix A). This data will be presented with the experimental results of the bearing test rig later in this chapter, in order to plot some correlations and draw some appropriate conclusions as to the effect of roughness.

3.3.3 3D Surface Measurement

Three-dimensional surface measurements are superior over two-dimensional measurements for two reasons. The first reason being related to accuracy; the ability to capture a section of surface in three dimensions allows for a much better description of the surface roughness. Whereas previously, few profiles are taken and therefore limit the “field of view”. Another contribution to the improved accuracy is the argument of physical probing against optical probing. In the previously mentioned methods, a physical probe is moved along the surface. The radius of the probe tip will differ the surface definition that is being measured (although the magnification level from an optical method will have the same effect) and therefore influences the quality of the result. The second benefit is in relation to the surface structure. As well as having statistical roughness values to describe the surface, it is also useful to have a three-dimensional surface scan available to the user. This then enables a surface structure to be analysed both qualitatively and

3. Influence of Surface Roughness on the Frictional Performance of a Hydrodynamic Bearing

quantitatively. Roughness values may not be effective in describing a surface and so therefore, a qualitative analysis may be more useful. To perform three-dimensional surface measurements, a Bruker Contour GT-K white light interferometer (lateral resolution of 0.26µm and RMS repeatability of 0.02nm) was used.

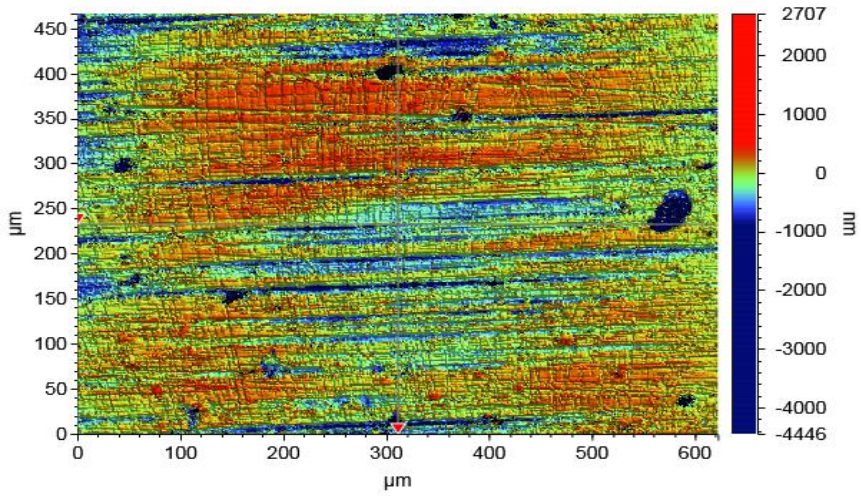


Figure 3.5. 3D surface plot.

The white light interferometry method is an optical based method which basically consists of an initial beam which is split into a reference beam and an offset beam. These are shone at the surface of interest and depending on how they interact with each other (i.e. they cancel each other out or summate to increase in intensity), which is detected by a camera, the change in height can be calculated. Variations in reflected light are calibrated to a change in height before measurements take place. The variations, in three dimensions, are post-processed on the connected computer using the Vision64 software where three-dimensional roughness parameters can then be calculated. Again, for each of the samples, three measurements are taken and the calculated 3D roughness values are presented in Appendix B, C and D.

3.3.4 Wettability and Surface Energy

The next measurement that was performed on all of the samples was an analysis of the surface wettability. Wettability can be explained by the diagram seen in Figure 3.6. In this diagram, a drop of liquid is *sat* on the solid surface and the angle being referred to as θ is called the static contact angle. This angle being the tangential angle (in relation to the solid interface) of the drop surface (gas-liquid interface) at the triple point. The equilibrium of the three coexisting interfaces (gas, liquid and solid phases) is a thermodynamic equilibrium which can be described by Young's equation (see eqn. 3.1), first described by Young (1805).

$$\gamma_{SG} - \gamma_{SL} - \gamma_{LG} \cos\theta = 0 \tag{3.1}$$

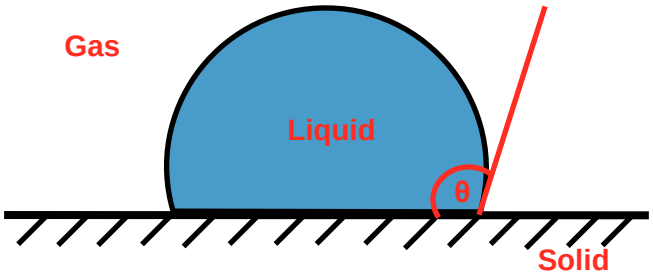


Figure 3.6. Contact angle schematic.

Eqn. 3.1 is split into three different interfacial energies which act at the solid-gas interface (γ_{SG}), solid-liquid interface (γ_{SL}) and the liquid-gas interface (γ_{LG}) (also known as surface tension). The interaction of these interfaces results in the contact angle shown. The resulting contact angle can be split into four different ranges: superhydrophilic ($\theta < 10^{o}$), hydrophilic ($\theta < 90^{o}$), hydrophobic ($\theta > 90^{o}$) and superhydrophobic ($\theta > 150^{o}$). (It should be noted that the prefix *hydro-* can refer to any liquid even though it does technically refer to water. The only exception to this is when the liquid is either oil or an organic based liquid which is when the prefix *oleo-* should be used). The smaller the contact angle, the higher the wettability of the surface. Bhushan and Jung (2011) stated

3. Influence of Surface Roughness on the Frictional Performance of a Hydrodynamic Bearing

"surfaces with high energy, formed by polar molecules, tend to be hydrophilic, whereas those with low energy and built of non-polar molecules tend to be hydrophobic". Hydrophobic surfaces are said to prevent the liquid from *wetting* the surface. So how does surface energy relate to surface friction for hydrodynamic bearings? Well, recent studies (Nosonovsky and Bhushan (2008), Nosonovsky and Bhushan (2009)) showed that the roughness of a surface influences the wettability of a surface which then affects the capillary adhesion force which in turn will change the frictional characteristics of the bearing surface.

So how can the contact angle and the surface energy be determined? Well the samples' surface energies were determined using the Owens-Wendt method (Owens and Wendt (1969)) which uses eqn. 3.2.

$$\left(\gamma_s^d \gamma_l^d\right)^{0.5} + \left(\gamma_s^p \gamma_l^p\right)^{0.5} = 0.5\gamma_l(1 + cos\theta) \tag{3.2}$$

Where γ_s^d is the dispersive surface energy, γ_l^d is the liquid surface tension property (dispersive), γ_s^p is the polar surface energy, γ_l^p is the polar component of the liquid surface tension property and γ_l is the total surface tension property for the liquid. The surface tension properties are predetermined from existing published data (taken from Lee (1996)). Looking at eqn. 3.2, it is clear that this equation on its own cannot determine the surface free energy because there are two unknowns, the dispersive and polar surface free energy properties. So, in order to determine these values, the equation must be solved simultaneously with each equation representing a different liquid. The two resulting linear simultaneous equations are presented in eqn. 3.3 and 3.4.

$$x + ay = b(1 + cos\theta_1) \tag{3.3}$$

$$x + cy = d(1 + cos\theta_2) \tag{3.4}$$

Where: - x = $\left(\gamma_s^d\right)^{0.5}$;

- y = $\left(\gamma_s^p\right)^{0.5}$;

- a, b, c and d represent the dispersive and polar surface tension values for the liquids;

- θ_1 and θ_1 are the measured contact angles for the two different liquids.

To determine the static contact angle for each of the liquids, a static sessile drop approach is used with an Attension Theta Lite contact angle goniometer. The goniometer is made up of a platform, where the specimen surface is placed; a light source and a camera, which faces in the direction of the light source. The monochrome filtered camera take several hundred images of the projected surface as a droplet is placed upon it. Before each of the samples are used in this experimental method, the component surface is cleaned with an alcohol solution. This is done so that the contact angles for each specimen are reproducible, as surface cleanliness has an effect on the resulting contact angle (Drelich (2013)). The following steps detail this experimental process:

i. Adjust the threaded plunger syringe until a droplet is formed at the end of the syringe. The size of the droplet remains constant over the experiment. Using the live camera feed through the Attension theta software on the connected computer, adjust the syringe further until the droplet meets the guide height.

ii. Make sure the software is set to slow and fast speed capture. This means that for the first time period 250 frames of 38ms are taken then five frames of 1s are then captured (usually). This setting is chosen to capture both the initial changing behaviour of the contact angle then capture the near steady state equilibrium which is required for the determination of the *static* contact angle. Figure 3.7 shows an example convergence of contact angle over time. If the target residual error of 0.5^o is not met then the aforementioned capture times will be extended in order to satisfy this criteria.

iii. The trigger point must be set at this step near the surface. This point allows the software to start capturing images when it detects a change in colour (which will be when the droplet is placed upon the surface). Once the focus and settings are all acceptable, the software is set to trigger and record.

iv. Using the spring loading arm, which holds the syringe, carefully place the droplet upon the surface and then release the arm.

v. Wait until all frames have been captured. The post processing feature should then automatically display. At this point, the user needs to tell the software where the top surface plane is and where to look for the droplet surface periphery. The software then takes each of the frames and fits a curve to the droplet surface. Where the curve intersects with the surface plane, a contact angle can be calculated (as per the diagram in Figure 3.6).

vi. Looking at the generated data, if the target residual error has been satisfied then the last measured contact angle can be recorded.

vii. This process is repeated three times for each of the samples per liquid, at different positions on the bearing surface and a mean contact angle is then calculated.

This process was performed for both distilled water and glycerol. The results of the static sessile drop experimental process can be found in Appendix E. Using the mean contact angles for each specimen (for each liquid), eqn. 3.3 and 3.4 were then used to calculate the dispersive and polar surface free energy values (this data is presented in Appendix F) using the liquid properties presented in Table 3.2.

Table 3.2. Surface tension properties for Owens-Wendt method (taken from Lee (1996)).

Liquid	Surface Tension Values (mN/m)		
	Dispersive	Polar	Total
Water	21.8	51.0	72.8
Glycerol	34.0	30.0	64.0

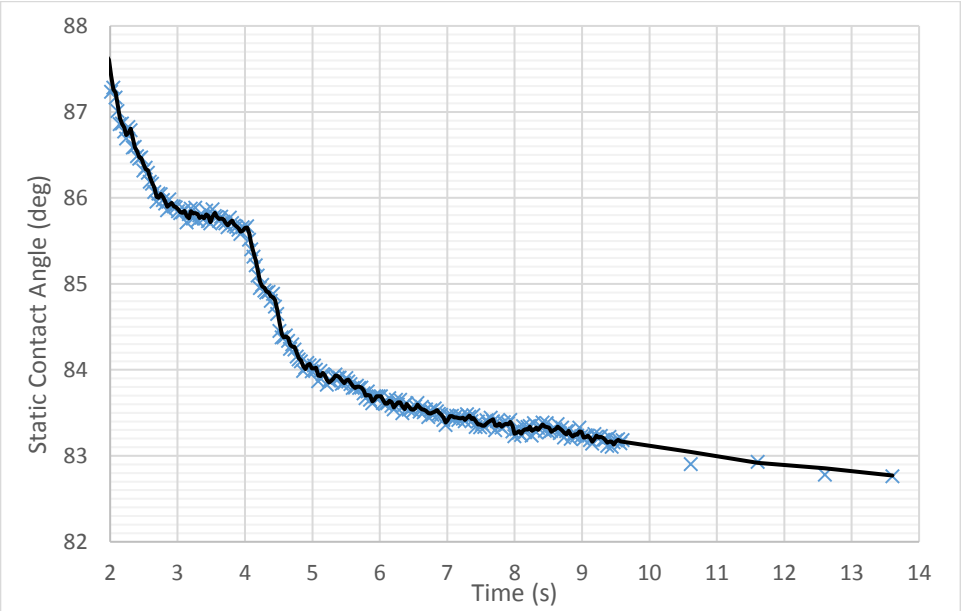


Figure 3.7. Static contact angle convergence over the experimental time period.

3.4 Bearing Test Rig Experimentation

Now that all the surface measurements have been discussed, the final measurement to be taken was to measure the coefficient of friction (COF) for each of the bearing samples. Using the tribometer discussed in section 3.2, the experimental method used for determining the COF was the following:

i. Place component onto shaft with a spherical washer set on each side and then tighten nut by hand.

ii. Insert shaft into flexible coupling and locating plate then tighten grub screws up.

iii. Rotate component inside journal and adjust spherical washer to minimize axial run-out. Once run-out is minimised, then fully tighten with spanners.

iv. Turn on displacement pump to start feeding cavity with oil. Before progressing to the next step, make sure oil is visibly flowing from the bearing cavity into the drip tray.

v. Set inverter to an appropriate speed for the test and press start. Motor shall then increase in speed by preset ramp.

vi. When ramp in angular speed has stopped and the correct speed has been achieved, apply a load of 100N by rotating the thread through knob connected to the load transducer and spring.

vii. The additional load may have reduced the speed, so make the necessary adjustments to correct speed then start timer and logging program (records torque and rotational speed).

viii. Run program for 4 minutes and whilst running, monitor temperature on multimeter at regular intervals and record.

ix. After 4 minutes, stop the logging program and press “stop” on inverter, shaft will then decelerate by the preset ramp.

Each of the samples were tested at 500rpm, 1000rpm, 1500rpm and 1850rpm, which was the maximum achievable shaft speed when loaded.

A few problems were encountered with the setup used. The first problem was related to the hydraulics which had been designed before this project. The displacement pump could not run for more than 5 minutes at a time, due to the component overheating and the flow rate being produced from the cavity was far too much. Only a small pressure and flow rate were needed to ensure that one, the bearing surface was lubricated and two, to ensure that the bearing was not of a hydrostatic nature. The bearing type being explored should be able to generate enough load capacity from the hydrodynamic effect. After the initial investigation, further testing actually

resulted in the pump failing. An engineering analysis of the hydraulics led to the conclusion that the hydraulics had been poorly designed. The first issue with the hydraulics was that there was no "slack" in the system to accommodate any back pressure. Due to the incompressible nature of the lubricating liquid, if any large back pressures are experienced by the fluid, that load is transferred back to the pump. To overcome this, the hydraulics needs to include two additional components: a non-return valve and an accumulator. The non-return valve would be installed after the pump and would ensure that the flow goes only one direction. The accumulator would be installed between the non-return valve and the bearing. The purpose of this component is to act as a reservoir for the incompressible fluid when back pressure exists. The accumulator, by some mechanical means (e.g. a spring system) also keeps the fluid in the reservoir under pressure so it is able to feed the flow to the bearing when needed.

The other problem was related to the thermocouple installed on the connecting rod near the bearing. The thermocouple temperature reading on the multimeter seemed to be varying chaotically as the test was operating, making it difficult to come to some conclusion about the varying rig conditions. Although, it should be noted that during the experiment, the oil was also pumped from a reservoir at room temperature. The oil from the cavity always remains in a separate reservoir. The other conclusion that can be made about the rig conditions is that the rig is always allowed to cool between runs to allow for the hydraulics system to cool down. So, inadvertently, the poorly designed hydraulics forces the experimental rig to have a cooling down period which enabled similar starting conditions.

The resistive torque, which acts in the opposite direction to the driving torque, is a product of the resulting frictional force and the radius from the centre of the journal to the surface where shear stresses act (eqn. 3.5). The coefficient of friction then relates the magnitude of the friction force to that of the load applied. Using eqn. 3.6, the coefficient of friction for each sample can be calculated.

$$T = F_f r = C_{fr} F_r r \tag{3.5}$$

$$C_{fr} = \frac{T}{r F_r} \tag{3.6}$$

Where T represents the measured torque, F_f is the friction force, r is the radius of the journal, C_{fr} is the coefficient of friction, F_r is the radial force (load applied during test).

The other process that must occur before the coefficient of friction is calculated is to adjust the torque to only represent the resistive torque of the bearing. Due to the fact that other components (i.e. other bearings) are needed to support the shaft, the resistive torque of these components must be measured separately and taken away from the measured torque. This torque was measured (Table 3.3) and subtracted from each of the sample's measured torque values. Unfortunately, therein lies another problem with the tribometer. Due to the design, the load cannot be applied to the shaft when no hydrodynamic bearing/sample is present. Therefore, the measured resistive torque of the other components do not truly reflect the torque that would occur during the experimentation because a radial load is applied.

Table 3.3. Rig friction torque values.

	Speed (rpm)			
	500	**1000**	**1500**	**1850**
Torque (Nm)	0.08	0.11	0.13	0.14

Regardless of these issues, the results of this experimentation are plotted in Figure 3.8. The first observation that should be expected, is that as the rotational speed is increased, the coefficient of friction value should increase. Provided the viscosity value remains fairly constant (although some changes may occur due to temperature variations), the one component of shear stress that will change will be the velocity gradient. All the components except 2B, 4B and 5B produce as predicted for this behaviour. In relation to the best performing component, no component is clearly indicated across the speeds as sufficiently lower in coefficient of friction (C_{fr}). The best performing component, for a particular speed, is seen with sample C3 which achieves a C_{fr} value

of 0.098. Compare this to the mean value for this speed (0.148) and this demonstrates the C3 sample as having 66% of the mean value. For 1000rpm, 2B performs best at 73% of the mean value. At 1500rpm, 6B performs best with a C_{fr} value that is 81% of the mean sample value and finally, at 1850rpm, 5B is seen to have coefficient of friction value that represents 86% of the mean value. So what conclusions can be made from this? Well, it isn't obvious from looking at these results that a particular ground component surface has resulted in a consistent outperformer and therefore, that may lead to some initial conclusion that each particular surface is better suited to a particular speed (or lubricant flow condition). To try and relate the roughness of a surface to the coefficient of friction, Pearson's correlation coefficient is used in the following section in order to detect and relationships from the data produced.

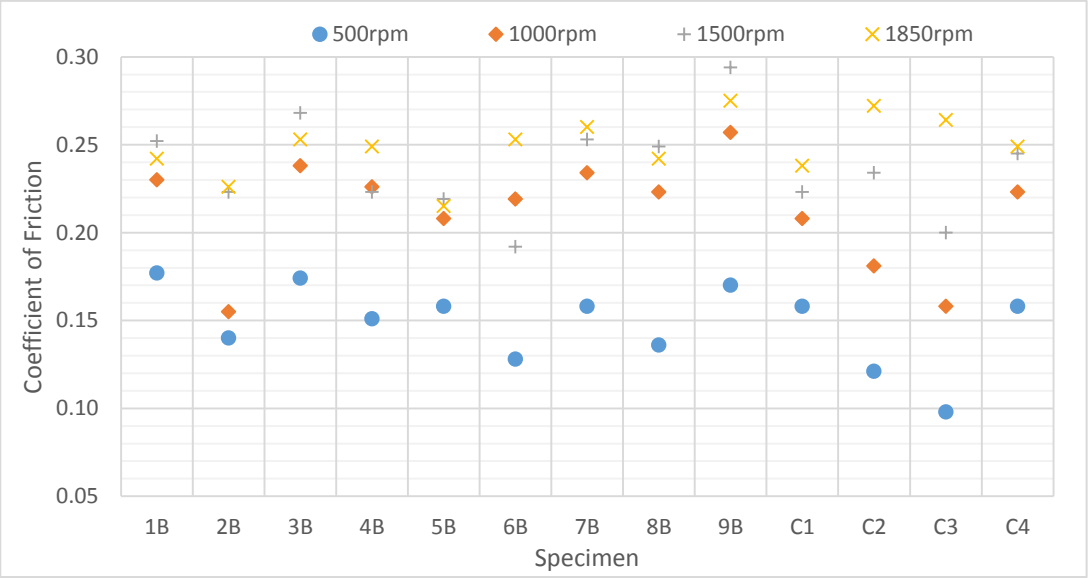


Figure 3.8. Coefficient of friction results for each sample (data used for plot can be found in Appendix G).

3.5 Correlation Results

Pearson's correlation coefficient, r (see eqn. 3.7), is used to distinguish whether there are any correlations between any of the surface measurements and the coefficient of friction. If the value

of r is equal to 1, then this represents a perfect positive correlation for the dataset. If the value of r is equal to -1 then it represents a perfect negative correlation. The further the value is away from zero, which indicates no relationship is present, the better the correlation. It should be noted that even though Pearson's correlation coefficient is powerful in detecting trends from data, it is only useful for identifying linear relationships.

$$r = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2 \sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2}} \tag{3.7}$$

In eqn. 3.7, X_i or Y_i represents one trial/sample value of each dataset and $\bar{X}$ or $\bar{Y}$ is the mean value for that dataset. All of the calculated values for each of the datasets are presented in Figure 3.9 and the actual data can be found in Appendix H.

Some interesting observations can be made from looking at Figure 3.9, starting with the largest observable correlations. RPc, which is referred to as the peak density per length of profile, is a parameter which describes the number of times the profile deviates past the set height from the mean roughness line for every centimetre of the profile. This parameter at 500rpm shows the largest correlation, with an r value of -0.700, which represents a fairly strong negative correlation. With this is mind, could this be possibly used to monitor the effectiveness of a surface in minimising friction? Not really based on just this evidence. Looking at the other range of speeds for this roughness parameter, the correlation gets smaller and even flips to a positive correlation at the highest speed. So the reliability of this particular parameter as a monitoring tool is questionable.

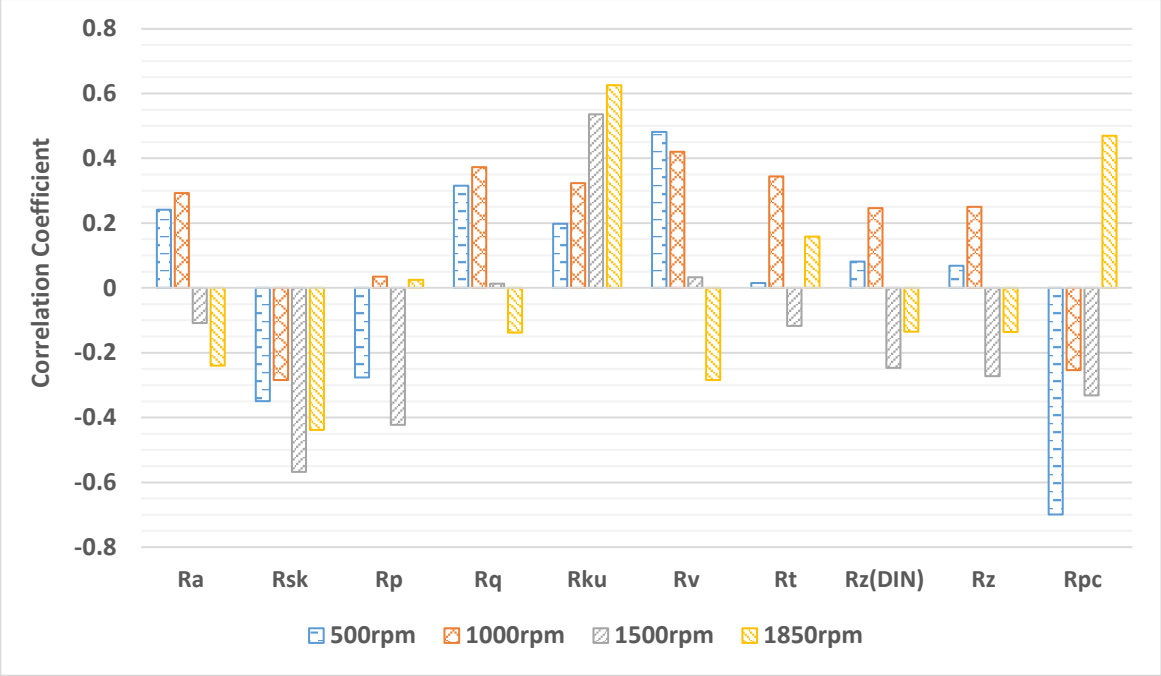


Figure 3.9. Correlation coefficient for coefficient of friction against 2D surface roughness parameters. Calculated data can be found in Appendix H.

The other conclusion that could be made is the increased reliability from using three-dimensional (3D) roughness parameters (Figure 3.10, 3.11 and 3.12) as opposed to two-dimensional (2D) based (Figure 3.9). Regardless of the magnitude they attain, looking at the values of Ra and Sa, which are the same parameters but different in dimensionality, more consistent correlations are seen with the 3D version across the speeds as a whole than were observed for 2D. Ra is commonly used to describe surfaces but as per the discussion in section 2, the use of this as a surface descriptor is misleading, if used on its own. This is backed up by the correlation values calculated for Ra in Figure 3.9. Little evidence is shown here that this parameter can be effective in assessing a surface for resulting surface friction losses.

The second largest correlation (0.626) for two-dimensional based is Rku, which is referred to as kurtosis of the profile. More simply put, the higher the value of Rku, the sharper the peaks and increased gradients on the “hill sides” (sometimes referred to as increase in the *peakedness* of the surface). Having a higher kurtosis profile would indicate large sharp peaks, which are usually a

source of wear. Having such minimal contact area on the *peak-tops* with the loads applied will usually attribute to increased wear of the peaks. Sometimes, bearings can usually exhibit a *breaking-in* period. This is where the initial run-time of the bearing will have increased wear rates, removing these peaks. Over time, the kurtosis and *peakedness* of the profile will reduce, lowering the frictional losses. Sku (Figure 3.10), the three-dimensional kurtosis parameter, does show similar magnitudes of correlation to its 2D counterpart.

Looking at the correlations for Sds (Figure 3.12), which is the three-dimensional version of peak density, no clear, strong correlations exist throughout the tested speeds, indicating that peak density may not be that important.

Looking at the 3D roughness parameters (Figure 3.10, 3.11 and 3.12), the largest correlation coefficient seen (-0.755) was for Ssc which is the mean summit curvature. In other words, this parameter describes the mean peak radius for the surface. This strong correlation value describes that as this roughness value increases, the coefficient of friction decreases but since the resultant unit is one over the length scale (i.e. 1/μm), this actually reflects that as the peak radius decreases, so does the frictional losses. Again, this fits in with the conclusion made before in relation to minimising surface contact area.

The second best correlation that exists for the 3D parameters lies with Sc which is the core void volume that supports 10-80% of the bearing ratio Abbott curve. This parameter is related to core fluid retention and increasing the retention fluid volume according to this data would indicate that it is beneficial for minimising frictional losses. What is also interesting, is another roughness descriptor that is related to Sc is Sci (core fluid retention index). Sci is calculated in a similar way to Sc but describes 5-80% of the bearing ratio curve and is non-dimensionlised by dividing the result by Sq. Looking at Figure 3.11 though, Sci is no way near as consistent as Sc and nor does it show any similar magnitudes of correlation.

3. Influence of Surface Roughness on the Frictional Performance of a Hydrodynamic Bearing

One of the other parameters that needs to be discussed is the surface energy correlations. One would conclude from looking at Figure 3.11 that surface energy is not a reliable way of characterising a surface for this function but there are some issues with the calculated data. The first issue is in relation to the chosen liquids, water and glycerol. Looking at the values for each liquid in Table 3.2, it can be seen that the dispersive component of the surface tension is close in size. This becomes a problem because when trying to solve for the components of the surface free energy using the simultaneous equations (eqn. 3.3 and 3.4) because it results in larger errors. It is better to use two liquids which are much more dissimilar, reducing the error significantly. A much more appropriate choice would be diiodomethane (γ_l^d=50.8mN/m and γ_l^p=0mN/m) and water (γ_l^d=21.8mN/m and γ_l^p=51mN/m). Another possible improvement to the calculation of surface free energy would be to look at alternative formulations/methods. As well as the Owens-Wendt method, other methods such as the van Oss method (van Oss et al. (1988)) could be used. This method splits the polar component of the Owens-Wendt equation into its acid and base component values. Because of the higher number of components (and subsequently the higher number of simultaneous equations to be solved) the robustness of the surface energy determination is improved but this method does suffer from a higher sensitivity to changes in the input values (Zenkiewicz (2007)). Another assumption with using these methods, is that they assume no composite interface exists (see section 2.2.1) because it is based on the Young's equation (eqn. 3.1). This method assumes a perfectly flat with no roughness surface. Yan et al. (2011) reviews and suggests some alternative approaches which take into account the roughness of a surface for the calculation of surface energy. One of the alternative approaches is to use the Wenzel equation (eqn. 3.8) which does take into account roughness, assuming a homogeneous wettability scenario. This means that the liquid totally penetrates the roughness with no gas encapsulated within the roughness, meaning that a composite-type interface cannot exist under this description.

$$r(\gamma_{SG} - \gamma_{SL}) = \gamma_{LG} \cos\theta_W^* \quad (3.8)$$

Where, r, is the roughness factor. This represents the ratio of roughness surface area to the flat no-roughness geometric surface area. This roughness factor is then used with eqn. 3.9 to convert the static contact angle measured, to the parameter, θ_W^*, Wenzel contact angle.

$$\cos\theta_W^* = r\cos\theta \quad (3.9)$$

Another alternative method that includes the effect of roughness is the Cassie-Baxter equation (see 3.10) which produces a modified contact angle, θ^*. This method assumes a heterogeneous wetting state and therefore allows for gas to be *trapped* within the roughness; this means that the effects of the composite interface could be included. Methods which make the assumption of no composite interface or heterogeneous state may break down further in the following section when substantial structures are introduced onto the surface which could hold/trap air when liquid is present on the surface. However, due to the interface being unstable, the assumption of no composite interface being present will eventually become valid some time after surface wetting.

$$\cos\theta^* = -1 + \varphi_S(\cos\theta + 1) = \varphi_S\cos\theta + \varphi_s - 1 \quad (3.10)$$

Where, φ_s, is the ratio of solid-liquid interface contact area to the total area (both gas-liquid and solid-liquid interface) in a plane parallel to that surface.

To conclude, this initial investigation has shown that there is some evidence that parameters Ssc and Sc could be effective in describing the functionality of ground, polished surfaces in order to minimise frictional surface losses/drag related forces. The next question to answer will be in their effectiveness at describing surfaces that are made up of microstructure elements.

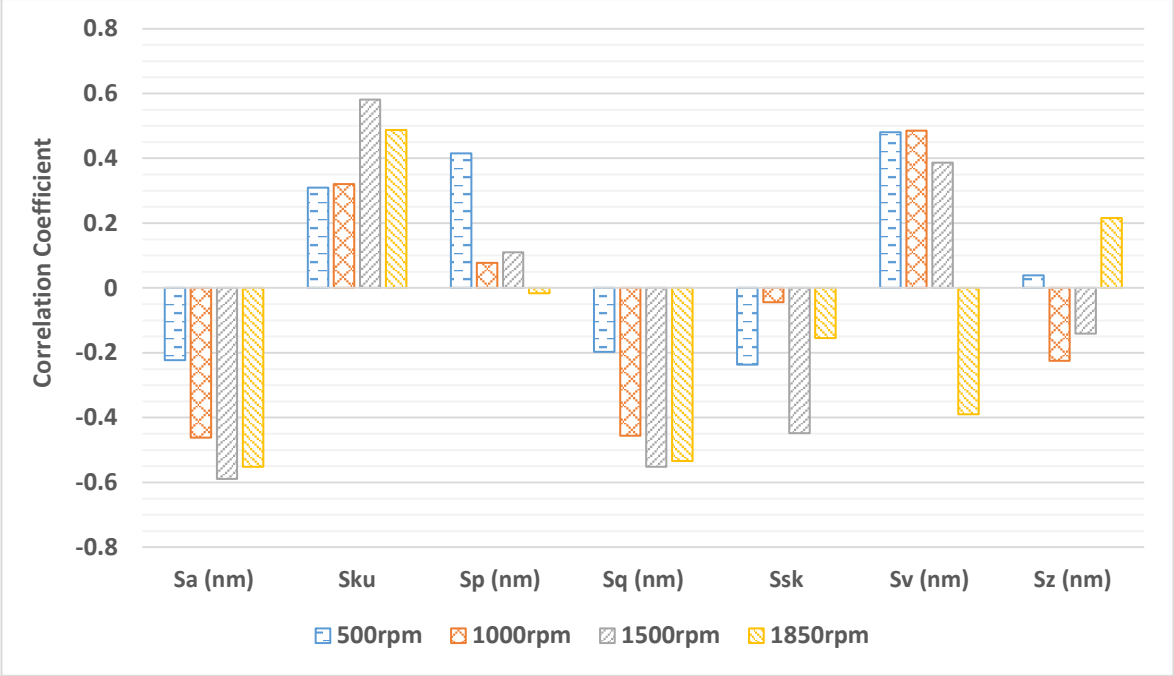


Figure 3.10. Correlation coefficient for coefficient of friction against height based surface roughness parameters. Calculated data can be found in Appendix H.

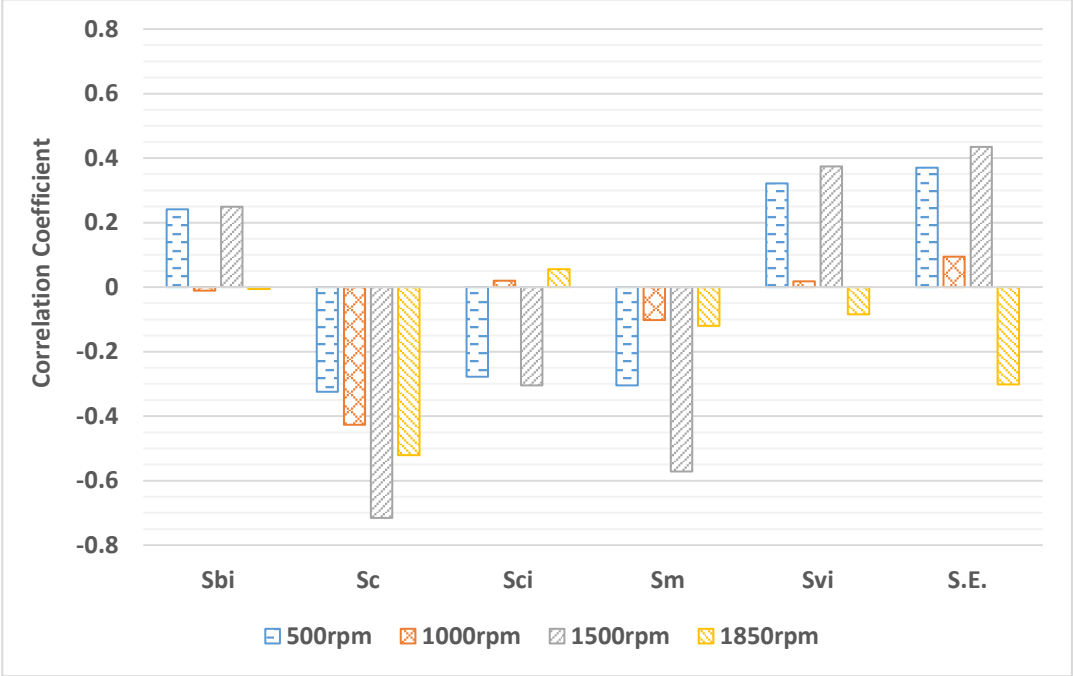


Figure 3.11. Correlation coefficient for coefficient of friction against functional surface roughness parameters and surface energy. Note: S.E. stands for surface energy and actual calculated data for this figure can be found at Appendix H.

3. Influence of Surface Roughness on the Frictional Performance of a Hydrodynamic Bearing

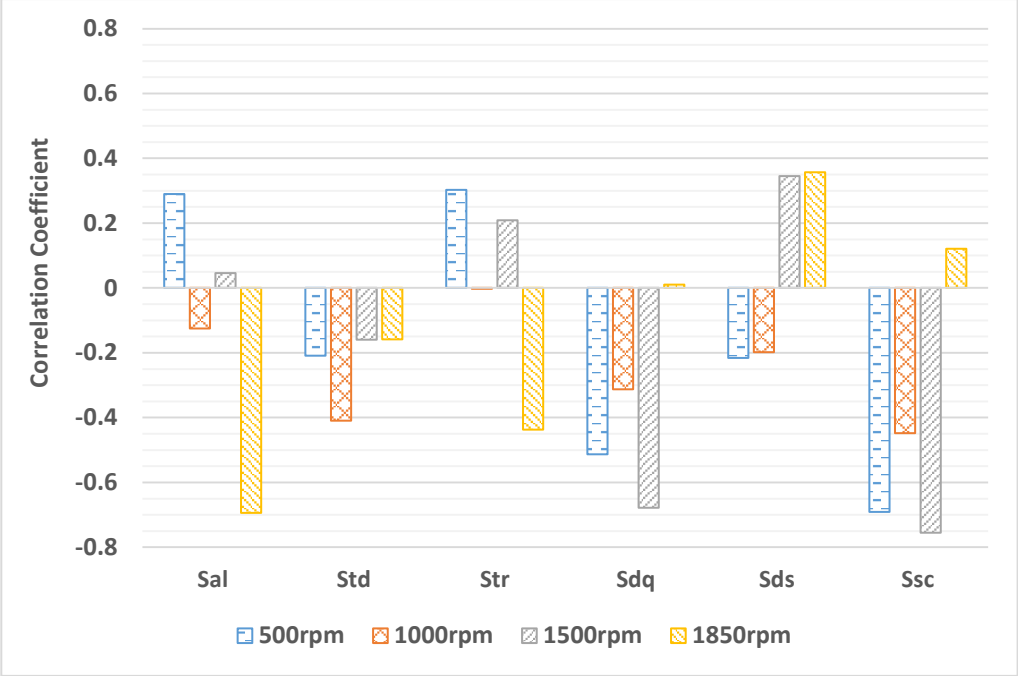


Figure 3.12. Correlation coefficient for coefficient of friction against both spatial and hybrid based roughness parameters. Calculated data can be found in Appendix H.

3.6 Conclusive Remarks of the Initial Bearing Investigation

To conclude the chapter, the following points summarise the initial investigation:

- Some roughness parameters (Sc, Ssc and Rku/Sku) show promise in the ability to describe a ground surface for its effectiveness in minimising friction but further investigation needs to verify its use when describing microtextured surfaces;
- The correlations presented by surface free energy determination are weak and are inconsistent but beneficial changes to the methodology have been highlighted. This includes changing the chemicals used for the Owens-Wendt method (water and diiodomethane);
- Components that need to be included within the hydraulics for the tribometer to run reliably have been disclosed (non-return valve and an accumulator). Also, a new

3. Influence of Surface Roughness on the Frictional Performance of a Hydrodynamic Bearing

displacement pump should include lower flow rate delivery and should contain a valve for minor adjustments;

- The actual bearing journal samples that were manufactured for this investigation have been shown to exhibit poor roundness defects with large peak-like structures identified for the majority of the batch. All further samples will be made on-site using a cylindrical grinding machine, allowing for better control of process parameters.

4. Numerical Investigation into Ground Textured Surfaces

4.1 Introduction

This part of the thesis will concentrate on the modelling of viscous flows over surfaces that have had micro-textures introduced onto them. Prior to this modelling taking place, some grinding investigations were performed in order to evaluate whether a cylindrical grinding technique could be used to texture the surface of a journal bearing effectively.

This investigation led to a novel grinding operation which uses a single-point dressing operation. It is recognised that the grinding wheel surface structure is well-linked to the path in which the dressing tool moves during the dressing operation (Chen and Rowe (1996)) and of course this, to some extent, then influences the resulting workpiece surface. The shape of the diamond tool tip, the tip radius and the cutting depth are also influential on the resulting wheel surface. According to Chen and Rowe, when dressing the grinding wheel, the dressing force can either lead to a fracture of the grain or within the bonding material. A larger dressing depth of cut, a_d, or decreased dressing feed, f_d, will lead to higher dressing forces and will increase the probability of a bond fracture, which effectively means the grains of the wheel will be *pulled out* rather than fracturing the grains (Pande and Lal (1979)). Grain fractures are either micro-, where small, fragile fractures appear on top of the grains or macro-, where large plateau-like areas are created across the grain. The ratio of the dressing tool engagement width, b_d, to the dressing feed is called the overlap ratio, U_d. This ratio describes how often each point of the grinding wheel surface interacts with the dressing tool tip. If the overlap ratio is less than one, it means that not all of the grinding

wheel surface has been in contact with the dressing diamond edge and due to the helical, kinematic motion between the two objects, a thread-like groove will be present in the wheel surface after dressing. Assuming that the diamond cutting edge is of a paraboloid shape, the resulting grinding wheel surface will be shaped similarly to the pattern seen in Figure 4.1 (the diagram is representing the stochastic surface as homogeneous for the sake of simplicity; in axial, cross-section view). If this dressing operation is then repeated a second time, cross grooves will be produced on the wheel surface (Figure 4.2). The resulting protrusions on the wheel surface are the cutting edges that will form the dimple textures on the surface. Looking at Figure 4.1, it can be seen how the dressing parameters will lead to the resulting profile and how the depth of cut, d, which will be smaller than the dressing depth of cut, will result in the workpiece not engaging with the entire profile, specifically, between C and D.

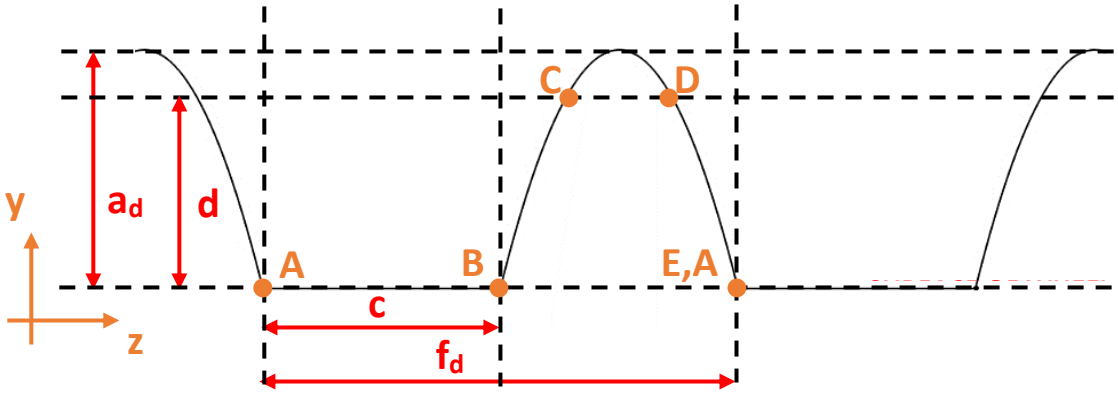


Figure 4.1. Schematic of textured grinding wheel surface (axial cross-section view).

Figure 4.2. Representing homogenous grinding wheel surface after first (top geometry) and second (bottom geometry) dressing operations.

So, how can the profile of the grinding wheel at any point along the z-direction (axis aligned with grinding wheel width) be described? Well, referencing to Figure 4.1 and 4.3, if y_w represents the y-distance from wheel surface to centre point of wheel, y_w is a function of z-distance (eqn. 4.1) (Stepien (2007b)).

$$y_w = f(z) \tag{4.1}$$

Also, Stepien expresses that if α is the angle, in radians, travelled from some origin then z can be calculated by using eqn. 4.2.

$$z = \frac{f_d}{2\pi}\alpha \tag{4.2}$$

From point A to B, this part of the wheel has had no material removed during the dressing operation and grinding wheel radius profile, ρ_w, is equal to the grinding wheel radius, R. From point B to E, ρ_w will be the wheel radius minus some height which is a function of z. This is summarised in eqn. 4.3 and 4.4 (Stepien (2007b)):

$$\rho_w(\alpha) = R \quad \text{(from point A (α=0) to B (α=}\alpha_B\text{))} \tag{4.3}$$

$$\rho_w(\alpha) = R - f\left(\frac{f_d}{2\pi}\alpha\right) \text{ (from point B (α=}\alpha_B\text{) to E (α=2π))} \tag{4.4}$$

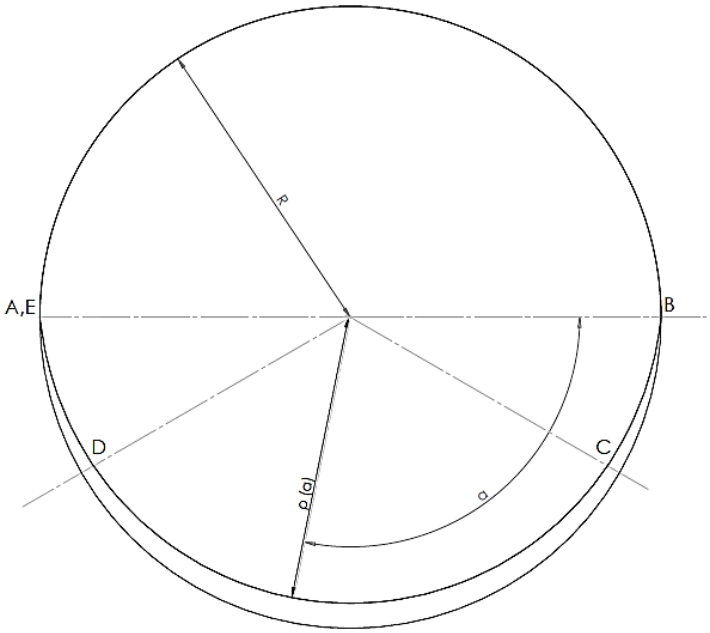


Figure 4.3. Schematic of grinding wheel in cross-section.

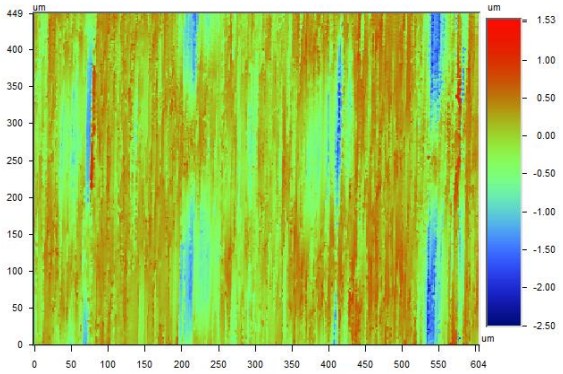
According to Stepien, if the diamond tool tip is assumed to be spherical with a radius of r_d, then eqn. 4.4 can be expressed as eqn. 4.5 (for the grinding wheel profile where the diamond tool has engaged with the surface):

$$\rho_w(\alpha) = R + r_d - a_d - \sqrt{r_d^2 - \left(f_d\frac{\alpha}{2\pi} - \frac{f_d+c}{2}\right)^2} \tag{4.5}$$

All of the pattern dimensions have been explained except the pocket land width, between points C and D. This land will be produced provided that the depth of cut is lower than the dressing depth of cut. Eqn. 4.6 calculates ∝ for points C or D at a profile radius of R minus the depth of cut.

$$\propto_{C,D} = 2\pi\left(1 - \frac{\sqrt{a_d(2r_d - a_d)}}{f_d} \pm \frac{\sqrt{(a_d - d)(2r_d - a_d + d)}}{f_d}\right) \tag{4.6}$$

(Where - is used to calculate point C and + is to calculate point D). As mentioned earlier if the wheel is dressed twice and cross-grooves are produced on the surface of the wheel, dimples can be introduced onto the journal. An example of a surface from this initial investigation is seen Figure 4.4 (with the grinding parameters detailed). Later on in this thesis, further, refined grinding trials will be reported on, which made use of optimal pocket sizing/orientation, predicted by the numerical investigation.



a_d	20	µm
f_d	300	µm·rev^{-1}
n_s	1665	rpm
n_w	10	rpm
v_s	35	ms^{-1}

Figure 4.4. Example of ground, textured surface (left) with grinding parameters used (table on right) from initial investigation into surface texturing by cylindrical grinding technique.

Using the scan from the white light interferometer, it was found that micro-dimples from this grinding process were of ellipsoidal shape (when viewing the surface from a smoothed, homogenous view). These ellipsoidal pockets were also produced in a similar method by Stepien (2008) on steel shafts but the features were millimetres in size. This then formed the basis of the simulation: to find out what size (length, width and depth), orientation and whether a staggered or aligned pattern is most suitable (if at all) for a specific operating condition. It is important to optimise the pattern for both texture-induced lift and surface drag reductions because as Syed and Sarangi (2014) concluded: if a textured surface pattern has not been optimised, then a smooth surface could well be superior. The purpose of modelling was to study the interaction of oil flow (hydrodynamics) with textured, ground surfaces and by doing this, over different pattern

setups, identify the potential drag reduction effects. The simulation was performed using a computational fluid dynamics (CFD) code, OpenFOAM® (version 3.0.1). Numerous codes were evaluated for this particular application and this code was chosen for a number of reasons. Firstly, the code is distributed through an open-source licence. This means the code is freely available to anyone and in return, the community further develops the core code and makes it available to the developers. Another well-known code, EDF's Code_Saturne®, was also evaluated but a limited English speaking community exists for the support of this CFD software. This lack of support makes it harder to engage with the software at first (i.e. few tutorials for getting started) but also the community is important in identifying bugs. Code_Saturne also uses FORTRAN language, which the author was not experienced in and as a user, would cause further delays to the project. Unfortunately, OpenFOAM® is only a solver with meshing functionality and unlike a commercial type package, it cannot provide computer aided design (CAD) solutions, a must when complex shapes are to be studied and neither can it be used to post-process the results. Other open-source software solutions were used to overcome these downfalls. EDF's Salome-Meca® has good CAD capabilities and version 2015.2 was used to generate the stereolithography (STL) files used to describe the geometry and subsequently needed for the meshing process. ParaView was used to post-process and visualise the large datasets generated by the modelling process.

So what is CFD modelling and what use does it have? Within the fluid mechanics field, most problems are three-dimensional but some flows can be simplified down to one-dimensional approximations using, for example, Bernoulli's equation but this approach is highly limited because it is based on ideal fluid and only provides properties along a one-dimensional streamline. In order to perform three-dimensional analysis, CFD is used. There are three conservation laws used within fluid mechanics: conservation of mass (continuity equation), conservation of momentum (Navier-Stokes equation) and conservation of energy. The

incompressible form of the continuity (eqn. 4.7) and Navier-Stokes equations are shown below (X, Y and Z direction forms are eqn. 4.8, 4.9 and 4.10, respectively).

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{4.7}$$

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=X-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right) \tag{4.8}$$

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=Y-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right) \tag{4.9}$$

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=Z-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right) \tag{4.10}$$

Taking eqn. 4.8 (X-momentum) as an example, each of the terms shall be explained. $\frac{\partial u}{\partial t}$ is the temporal term. $u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}$ is the advection term which represents acceleration due to positional change in the flow field. X is the body force term and can represent additional forces, for example, gravity. $\frac{\partial p}{\partial x}$ is the pressure gradient term. $\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)$ are the forces attributed to friction (i.e. viscous forces). Unfortunately, due to the highly non-linearity nature of these equations, these partial differential equations (PDEs) cannot be solved as they are, so they are transformed by integration into a set of scalar transport equations (explanation will follow). These equations are then discretised across a grid (so that velocities and pressures can be calculated at a number of discrete locations). The discretisation method used in OpenFOAM® is the finite volume (variable storage is cell-centred and pressure is co-located). This means that the grid is effectively made up of cells (hence, finite volumes) which are in effect control volumes. The discretisation process then results in a set of algebraic equations which can then be solved for each cell in the computational domain.

The scalar transport equation (eqn. 4.11) describes the transport of a scalar quantity by the overall movement of the fluid (advection) and also by the mechanism of diffusion. ϕ represents

the concentration of the scalar quantity. The term $\frac{d}{dt}(\rho V\phi)$ describes the changing concentration within the control volume. $C\phi$ is the advection term where $C = \dot{m}_{fluid} = \rho \boldsymbol{u}\boldsymbol{A}_{\text{face}}$. The diffusion term, $-\Gamma\frac{\partial\Phi}{\partial \boldsymbol{n}}A_{\text{face}}$, comes from Fick's law of diffusion. Where $\frac{\partial\Phi}{\partial \boldsymbol{n}}$ is the concentration gradient (along n, the outward normal direction), Γ is the coefficient of diffusion and A_{face} is the area of the face. Diffusion, which is a product of the random movements occurring within the fluid, always acts to reduce the concentration gradient, hence the negative sign. S_ϕ is the source or sink of the scalar quantity being considered. So, how can eqn. 4.11 describe velocity as a scalar quantity when it is a vector quantity? Well, a scalar transport equation can be used to describe each component of the velocity vector individually. Also, velocity can be described as a concentration of momentum. The resulting scalar transport equation for X-momentum is seen in eqn. 4.12.

$$\frac{d}{dt}(\rho V\phi) + \sum_{\text{faces}}\left(C\phi - \Gamma\frac{\partial\Phi}{\partial \boldsymbol{n}}A_{\text{face}}\right) = S_\phi \tag{4.11}$$

$$\frac{d}{dt}(\rho Vu) + \sum_{\text{faces}}\left(Cu - \mu\frac{\partial u}{\partial \boldsymbol{n}}A_{\text{face}}\right) = X - \frac{\partial p}{\partial x} \tag{4.12}$$

Each of the scalar transport equations are then transformed into a set of algebraic equations (example seen in Figure 4.5) for each of the control volumes, where the value, a, represents a collection of similar terms.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}\begin{bmatrix} \Phi_{east} \\ \Phi_{west} \\ \Phi_{south} \\ \Phi_{north} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix}$$

Figure 4.5. Example of algebraic equations.

These equations can then by solved iteratively by some appropriate numerical method (i.e. Gauss-Siedel). Each time an iteration is performed, a complete set of equations are solved for the

complete computational domain. After an iteration has been completed, each of the equations will have a residual error, which is the imbalance in the equations when the results are back substituted. Before the solver is run, the user is required to specify a target residual error and the solver will continue to perform successive iterations until this criteria has been met (the residual error is below the target).

It should be noted that computational fluid dynamics is only a prediction of the flow, it is not reality and therefore there will always be some error in the results. The amount of error is an accumulation of many different factors, such as:

i. The appropriateness of the boundary conditions;
ii. Discretisation errors (mesh refinement, order of schemes used…);
iii. Imbalance in equations being solved (residual error will always exist);
iv. Limitations of the machine's precision;
v. The representation of turbulence (i.e. by use of appropriate/inappropriate turbulence model);
vi. The actual governing equations themselves (assumption of the flow as continuum and other assumptions made in their derivation);

There are many more…

Therefore validation of numerical modelling results (CFD/FEA etc.) should *always* be carried out. It is the most important part of a robust modelling process and should be used to judge not only the accuracy of the modelling against reality but also make sure that human error hasn't been involved. The often used phrase, *"the decimal point was in the wrong place"* comes to mind.

Now that the reader has been informed of the modelling objectives and the CFD procedure, the remaining part of this chapter will cover the modelling process used to determine an optimal surface pattern configuration.

4.2 Model Setup

This section informs the reader of the modelling setup and process automation, implemented by use of python scripting. The scripting was used to automate the modelling process (for optimisation sake) but also to link the SALOME-MECA software to the OpenFOAM® solver.

To model the textured bearing efficiently, some simplifications were made to the model. The decision was made to only model a small section of the bearing surface (see Figure 4.6 for diagram) as if it were a Couette flow (see section 2.1 for more detail on this). A similar model setup to that seen in the publication by Ramesh et al. (2013). Although, one of the main drawbacks in the modelling by Ramesh et. al., is that two-dimensional domains were used to represent three-dimensional flow and as per the discussion in section 2.2.2, this led to an over-prediction in performance for the CFD results when mapped against experimental results. The computational domain used in this study is three-dimensional, representing a section of surface with four pockets (2 x 2 array) which results in a large number of cells (millions). In section 2.2.2, the work of Syed and Sarangi (2014) is also discussed. Their research showed a comprehensive study of different pocket geometries but unfortunately, a modified version of the Reynolds equation was used and this limited the investigation to certain texture height ratios (to ensure the inaccuracies relating to the pressure field predications were kept to a minimum). The modelling described in this will make use of the Navier-Stokes equations, minus the temporal term as the problem is steady state, to overcome such restrictions (the governing equations are shown in eqn. 4.7, 4.8, 4.9 and 4.10).

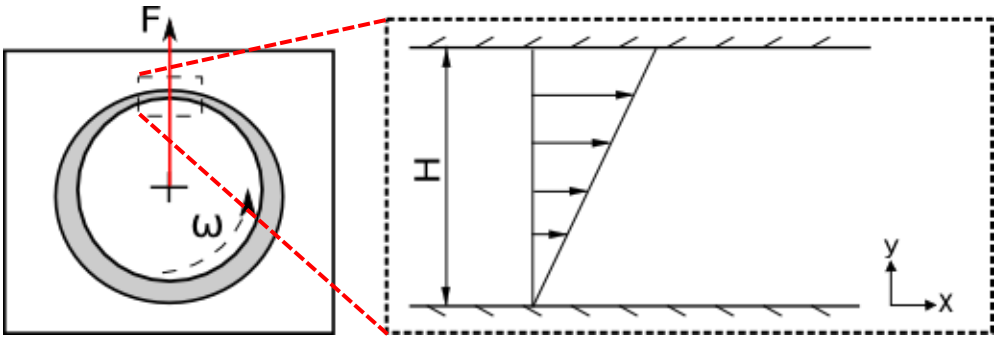


Figure 4.6. Schematic of finite section to be modelled Wharton et al. (2016).

To represent the fully developed Couette flow, a set of boundary conditions were applied to the computational domain (Figure 4.7). The inlet and outlet faces of the domain were selected as being translational cyclic. This was due to the fact that Couette flow is supposed to be a fully developed flow (with a surface of infinitely repeating geometry) in this direction. During the simulation, the cyclic boundary condition (BC) will take the values of the pressure and velocity fields at the outlet of the computational domain and extrapolate them back to the domain inlet. Another assumption made is that the surface is infinitely long in the z-direction. To model this, cyclic boundary conditions are applied on the front and back faces. The top face (the sliding wall BC) is represented by a uniform velocity of $5ms^{-1}$ in the x-direction and zero velocity in all other directions. This velocity of $5ms^{-1}$ is the maximum operational linear speed of the tribometer used in the experiments. The bottom face will represent the textured surface and the fluid-surface interaction is represented by a non-slip BC, so all velocities in all directions will be zero. All faces which are not explicitly defined as being cyclic, have zero gradient (Neumann) pressure boundary conditions applied and the pressure field is initialised with zero gauge pressure. The kinematic viscosity of the fluid being used is 95cSt, based on the engine oil used (Trident 15W40 at 40^{o}C) in the experimentation. Another assumption was that the flow conditions lie within the laminar flow regime. Based on the criteria stated in eqn. 2.16 and 2.17, the flow conditions result in a Reynolds ($Re_{Cou} \approx 0.4$, based on max clearance investigated (7µm)) and Taylor ($Ta \approx 18 \times 10^{-6}$) number well

below the suggested critical values. Therefore, based on this calculation, this assumption made should be valid.

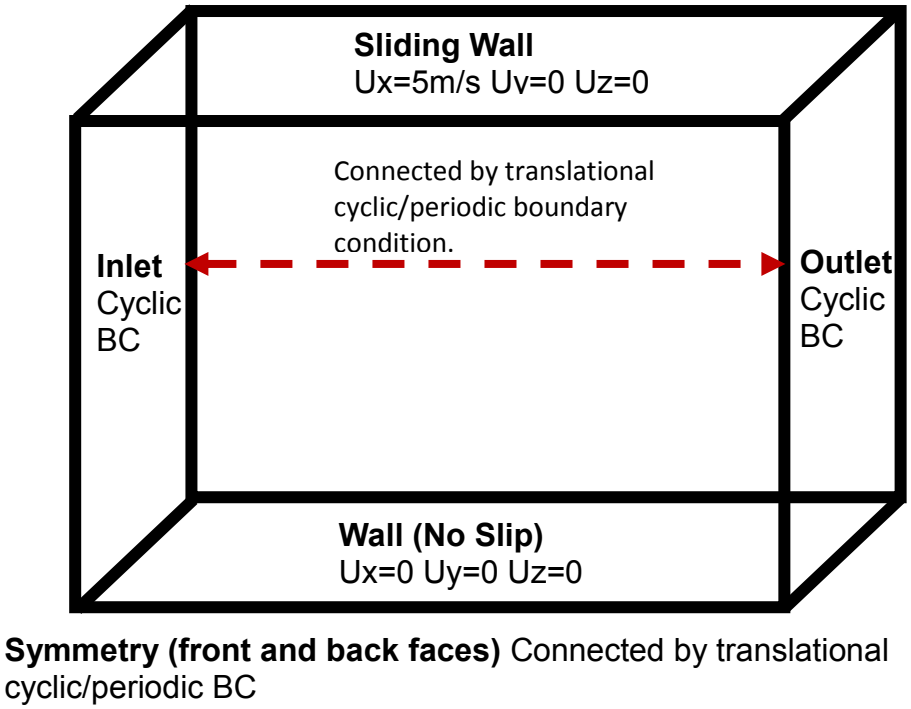


Figure 4.7. Diagram of applied boundary conditions.

The Couette flow is assumed to be steady state and for that reason, temporal details were not required, so the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) iterative-based solver by Patankar and Spalding (1972) was used to solve for pressure and velocity fields. A time marched solution is an alternative method that could be used to reach a steady state solution and is noted to be a more robust approach than SIMPLE but at the cost of CPU time. In this investigation SIMPLE was chosen for efficiency purposes. The SIMPLE solver works by using the following method:

i. An initial pressure and velocity field is applied (i.e. a guess);

ii. The momentum equations (Navier-Stokes) are then solved in order to find a better estimation of the velocity field;

iii. The continuity equation is then used to generate a pressure correction equation;

iv. The resulting pressure correction is then multiplied by an under-relaxation factor;

v. The weighted pressure correction is then applied to the original pressure field;

vi. Likewise, a velocity correction (again weighted by some under-relaxation factor) is then calculated and applied to the velocity field.

vii. An iteration has been completed. An imbalance in the solved equations (residual error) will be present. Starting with the better guessed fields, step ii to vi will be repeated for each successive iteration until the residual error is below the target value (specified at 1e-8 for this modelling which is slightly lower than the value used in the study by Ramesh et al. (2013)).

A sensible choice of under-relaxation factor of 0.3 for the pressure solver and 0.7 for the momentum solver was initially applied. Regrettably, one of the downsides to using the SIMPLE solver is that some guesswork is needed in applying these settings and differs from model-to-model. Initial runs of the simpleFoam solver with some different dimple texture configurations presented some divergent behaviour and this resulted in a lowering of the under-relaxation factors to 0.2 and 0.5 for the pressure and momentum solvers, respectively. The maximum number of iterations was also increased to 5000. The lowering of the under-relaxation factor values will often increase the number of iterations required but will help guarantee the stability required for the automated parametric study.

Both of the pressure and velocity equations were solved, iteratively, using the Gauss-Seidel method. This means that equations are solved using *last known values* from the same iteration (or soon as they become available), unlike the Jacobi method, where they can only be taken from the previous iteration. This simple modification significantly reduces the number of iterations required for the target residual error to be met. The pressure solver was also setup to usa a geometric-algebraic multi-grid (GAMG) preconditioner with faceAreaPair (geometric based)

agglomeration selected. The use of GAMG is beneficial for two reasons. The first is to use the preconditioner in order to reach a better guess for the initial field by performing iterations on a coarser grid level (which will require far fewer iterations to solve). The field values are then interpolated from the coarse grid onto the finest (original) mesh whereby solving can take place. Using multi-grid techniques with the Gauss-Seidel solver has shown to vastly reduce the number of iterations required to reach convergence against using the solver without such a technique (Versteeg and Malalasekera (2007), p. 232-239). The second benefit of the GAMG preconditioner is to smoothen out high frequency errors. By solving on a coarser grid, long-wavelength errors on the finer grid become short-wavelength which are easily dealt with on that grid and reduce quickly. The faceAreaPair agglomerate, which is better suited to orthogonal based meshes, basically works out whether a geometrically neighbouring cell should merge with the cell in question in order to achieve the coarser level needed for GAMG. For each cell in each cluster, the agglomerate checks each of the neighbouring cells faces for the largest face weight and merges if appropriate. The discretisation scheme used for the convective terms was Gauss linear (central-differencing and Gauss referring to Gaussian integration method for the control volume) which is second order accurate but unbounded. The limitedLinear choice is also an option for discretisation where the linear scheme includes a limiting coefficient (usually a value of 1 is chosen), which will ensure boundedness for stability purposes. This will ensure that the gradient is limited so that when the cell-centred value is extrapolated to the cell faces, it does not exceed neighbouring value bounds. Provided the mesh generated is of high quality in terms of skewness, non-orthogonality, aspect ratio and smoothness (definitions shall be explained later), then the limitedLinear choice may not be required. Again, a linear interpolation scheme was selected for the Laplacian (diffusive) terms with full correction applied for the surface normal gradients. To calculate the Laplacian terms using a Gaussian method, surface normal gradients need to be calculated. Looking at Figure 4.8, when an orthogonal mesh exists, the normal for the face in question (highlighted red) aligns with gradient direction (between the two cell centres). So in this

case, it is a simple calculation of one cell centre value minus the other divided by the distance. In the non-orthogonal approach, the surface normal no longer aligns with the gradient direction, so a correction is required. The correction weights the surface normal gradient in accordance to the angle between the two vectors (face normal and between cell centres). Even though the background mesh used is orthogonal, the additional meshing procedure with the snappyHexMesh utility for the introduction of the complex textured surface will introduce non-orthogonality to the mesh, so this correction was required.

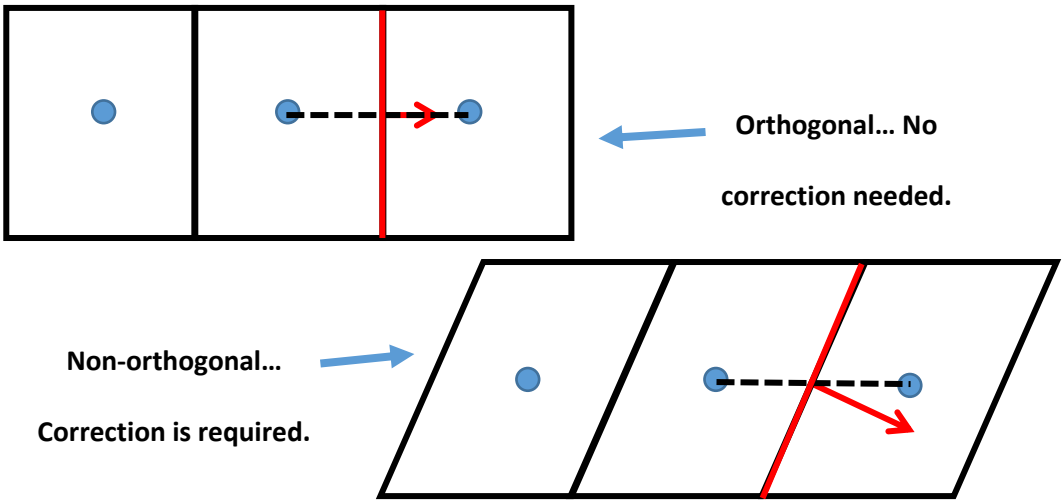


Figure 4.8. Orthogonal against non-orthogonal surface normal gradient correction.

Before moving onto the geometry creation and mesh refinement section, a little more detail will be given as to the operation of OpenFOAM® version 3.0.1 on the chosen Linux distribution, Ubuntu 14.04LTS. The code itself is a large collection of c++ libraries which are mainly solvers. To run a CFD job, first a collection of dictionaries are setup within the working directory. The working directory must be setup as shown in Figure 4.9. The three folders within the working directory are fairly explanatory. The folder “0” contains the starting conditions and boundary conditions. So, for this particular modelling exercise only two files are required “U” (velocity) and “p” pressure. The “constant” folder contains the “polyMesh” folder, which contains files describing the mesh

geometry, all of which is automatically generated by the meshing utilities. The other files contained within the “constant” folder are the so called “transportProperties” which details the fluid properties and “turbulenceProperties” which designates whether the modelling requires turbulence modelling or not and if so, which one should be used. The last folder “system” contains a large list of dictionary files which dictates settings applied to the solver or utility being run. The most important (and compulsory) files that are stored in this folder are “controlDict” which describes what solver is to be used and settings relating to time steps, output/writing frequency, surface or point history outputs, writing precision and writing format (binary or asci) are elucidated. “fvSchemes” details the discretisation schemes used for time, gradient, advection, Laplacian and surface normal gradients. “fvSolution” is used to select which solver to use for solving each of the flow variables (i.e. pressure and velocity) and it is here where the SIMPLE algorithm, target residual errors and under-relaxation factors are set.

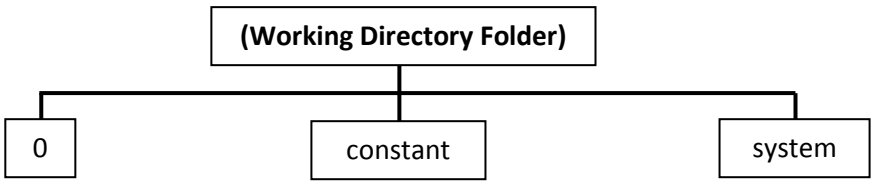


Figure 4.9. Required folder structure for case setup.

Within the Ubuntu operating system, the line “source /opt/openfoam30/etc/bashrc” (assuming “/opt” is where the code installation lies) is added to the file located at “~/.bashrc”. This means that all the library names can be called up without full directory routes. Assuming the terminal is pointing towards the working directory already, the SIMPLE solver command is “simpleFoam”. If a log file needs to be generated then “simpleFoam > nameOfLogFile” can be used. This command will run the solver in serial mode. The solver can also be run in parallel using Open MPI (Message Passing Interface). Provided that the mesh (and its initial field values) have been split into a number of directories within the working directory, the command “mpirun –np N simpleFoam –

parallel > nameOfLogFile" can be used (where N represents the number of subdomains or physical cores to be run on). Once the solver has finished, the domain needs to be reconstructed from the individual sub-domain results ("reconstructPar") then a modified ParaView ("paraFoam") can be used to view the results.

4.2.1 Geometry Creation within Salome-Meca

Now that the operation of the code has been explained, briefly, the creation of the geometry within the CAD facility of the Salome-Meca software shall be explained. To build the surface geometry every time, for each of the changing dimensions in the parametric study, would be time consuming. For that reason a python (version 2.7) script was created using the GEOM module. This removed the need to load the software graphical user interface (GUI) and instead the software ran in the *background*. By doing this, some of the performance requirements of the CAD modelling procedure will be reduced in comparison to those required for the user interacting with the GUI. The python script named "bearingStagger2.py" is explained in reference to the flow diagram (Figure 4.10) with code snippets taken from full script (which is provided in full in Appendix K).

The first part of the script, lines 1-7, imports the required modules (Snippet 4.1). The modules required are GEOM from Salome as well as the basic "math" module from the python library for some of the basic calculations needed within the script. All the commands for Salome are executed in the script by "geompy.*someFunction*".

```
00000001 import salome
00000002 salome.salome_init()
00000003 import GEOM
00000004 from salome.geom import geomBuilder
00000005 geompy = geomBuilder.New(salome.myStudy)
00000006 gg = salome.ImportComponentGUI("GEOM")
00000007 import math
```

Snippet 4.1. Imported modules for geometry creation script.

After the modules have been imported, all the variables needed for the script are defined, as shown in lines 9-31 (Snippet 4.2). All the variables which have been assigned a value, are needed as a starting value for loops executed later on in the script, except for the variables “even” and “odd” which is used to name objects in even and odd iterations within loops for referencing purposes.

```
00000009 xincrementNo = 0
00000010 yincrementNo = 0
00000011 totalIncrementNo = 0
00000012 xQuantity = changeNo1
00000013 yQuantity = changeNo2
00000014 xincrementInitial = 0
00000015 xincrementDist = changeNo3
00000016 yincrementInitial = 0
00000017 staggeredGrid = changeNo9
00000018 staggerDist = changeNo10
00000019 yincrementDist = changeNo4
00000020 xSize = changeNo5
00000021 ySize = changeNo6
00000022 zSize = changeNo7
00000023 radialClearance = changeNo8
00000024 d={}
00000025 even = 0
00000026 odd = 1
00000027 anglePattern = changeNo11
00000028 angle = changeNo12
00000029
00000030 xAngleCorrect = (xSize/2)*(math.cos(angle))
00000031 yAngleCorrect = (xSize/2)*(math.sin(angle))
```

Snippet 4.2. Defining variables used for the geometry creation script.

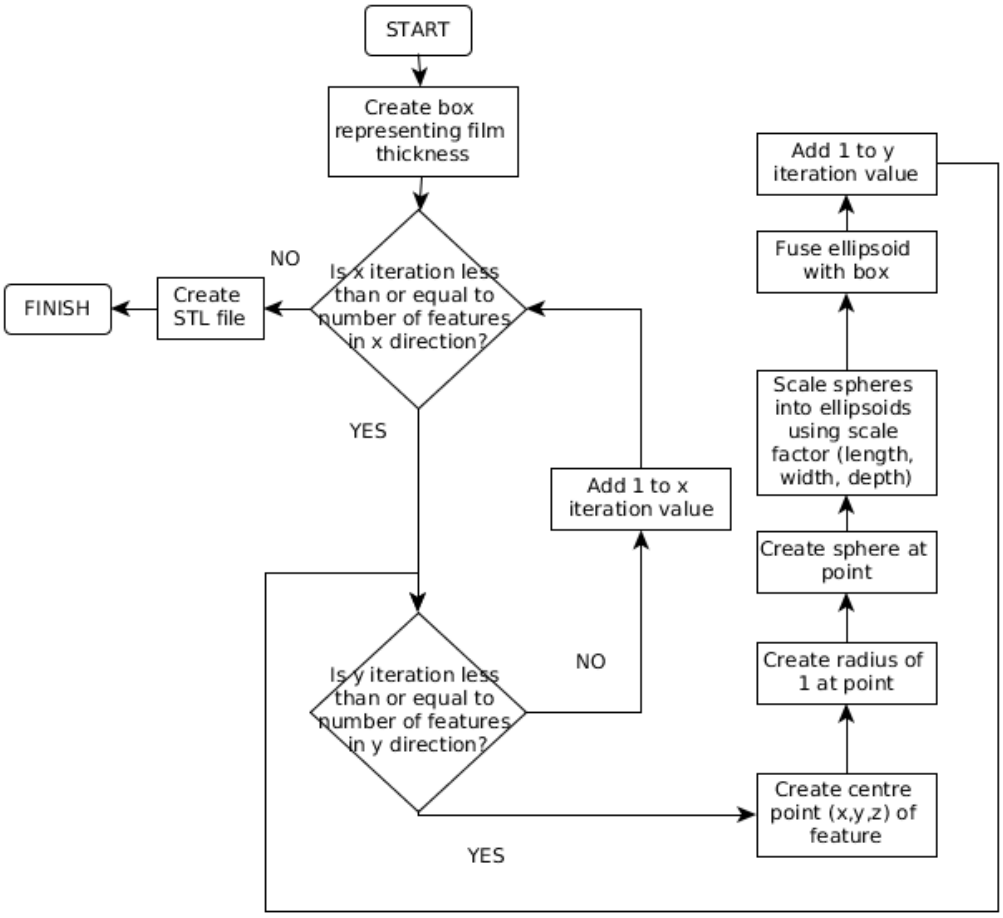


Figure 4.10. Flow diagram of SALOME GEOM python script for STL generation (Wharton et al. (2016)).

All the variables defined with the placeholder “changeNo#” are quantities that needed to be defined by the user. “xQuantity” and “yQuantity” are variables that need to be defined with an integer value that represents the number of pockets being modelled in the x and y-direction. As mentioned earlier, this will stay constant at 2 x 2 number of pockets. “xIncrementDist” and “yIncrementDist” need to be defined with a value (all physical dimensions are in units of µm) that represents the distance between each repeating pocket in the x and y-direction. The variable “staggeredGrid” needs to be answered with a *True* or *False* answer. If the pockets are to be aligned with each other then a *False* answer is required, else, *True* will result in every other row of pockets being shifted by some y-distance (defined by the “staggerDist” variable). “xSize”, “ySize” and “zSize” characterise the length, width and depth of the pockets, respectively.

"radialClearance" is the clearance height dimension. "d=" is a dictionary variable. "anglePattern" is answered with a *True* or *False* statement which enables the script to rotate the pockets by an arbitrary angle specified in radians ("angle"). If no rotation is required the lengthwise pocket feature is aligned with the x-direction. "xAngleCorrect" and "yAngleCorrect" are simply to correct the sizing of the pocket dimensions in relation to the x and y axis, when some degree of rotation is included.

```
00000035 if anglePattern == True:
00000036
00000037     pointMin = geompy.MakeVertex((-1 - (ySize/2)), (-1 -
yAngleCorrect), 0)
00000038     xDomainDist = (xQuantity * xincrementDist) - (ySize/2)
00000039     xincrementInitial = xAngleCorrect - (xSize/2)
00000040     if staggeredGrid == True:
00000041         yDomainDist = (yincrementInitial + (yincrementDist *
(yQuantity)) + (staggerDist)) + (yAngleCorrect) #even command 2nd 4th
etc
00000042     else:
00000043         yDomainDist = (yincrementInitial + (yincrementDist *
(yQuantity - 1))) + (yAngleCorrect) #odd command 1st 3rd etc
00000044 else:
00000045     pointMin = geompy.MakeVertex(-1, (-1 - (ySize/2)), 0)
00000046     xDomainDist = (xQuantity * xincrementDist)
00000047     if staggeredGrid == True:
00000048         yDomainDist = (yincrementInitial + (yincrementDist *
(yQuantity - 1)) + (staggerDist)) + (ySize/2) #even command 2nd 4th
etc
00000049     else:
00000050         yDomainDist = (yincrementInitial + (yincrementDist *
yQuantity)) - (ySize/2) #odd command 1st 3rd etc
00000051
00000052 print yDomainDist
00000053 print 'yDomainDist'
00000054
00000055 pointMax = geompy.MakeVertex((xDomainDist+1),
(yDomainDist+1), (radialClearance+1))
00000056 domainBox = geompy.MakeBoxTwoPnt(pointMin, pointMax)
```

Snippet 4.3. Calculation of dimensions and generation of clearance fluid geometry.

The first command, shown in the flow diagram (Figure 4.10) is to create a box which represents the film thickness, lines 35-56 (Snippet 4.3).The actual geometry is generated using the command on line 56 using the GEOM function "MakeBoxTwoPnt" which creates a box based on two points which are diagonally opposite each other. These two points are defined as "pointMin" and

"pointMax". (It should be noted that the origin is always located at the centre of the pocket first generated (x,y) and the z origin is located at the bearing surface or the bottom surface of the film thickness box). The calculation of these points depends on whether the pockets are defined with rotation and whether they are staggered or not, hence the reason for the IF statement on line 35. The reader may have also identified that a value of -1 is introduced on dimensions for "pointMin" and 1 for the "pointMax". This makes the film thickness slightly bigger than it needs to be, the reasoning for this shall be explained in the meshing section.

Apart from the last command, the remaining script is a single main WHILE loop (lines 59-107) with another internal WHILE loop nested within the main loop. The idea behind this is that the surface is split up into rows of pockets and the generation of each row is one iteration within the overall WHILE loop. The second WHILE loop within the main loop is to traverse along the row and generate each of the pocket geometries. The variable "xIncrementNo" serves as a counter of the number of rows completed. At the end of each loop, the command "xIncrementNo += 1" increases the variable by one and when this number exceeds the "xQuantity" value (number of pockets in x-direction) the loop exits. A similar procedure occurs for each of the pockets created along the transverse direction, which is counted by "yincrementNo".

```
00000066    while yincrementNo < yQuantity :
00000067        xDist = xincrementInitial + (xincrementDist *
xincrementNo) + (xSize/2)
00000068        yDist = yincrementInitial + (yincrementDist *
yincrementNo) + (Stagger)
00000069        # create a vertex
00000070        d["p{0}".format(yincrementNo)] =
geompy.MakeVertex(xDist, yDist, 0)
00000071        # create radius
00000072        d["radius{0}".format(yincrementNo)] = 1
00000073        # create sphere
00000074        d["sphere{0}".format(yincrementNo)] =
geompy.MakeSpherePntR(d["p{0}".format(yincrementNo)],
d["radius{0}".format(yincrementNo)])
00000075        # scale sphere into ellipsoid
00000076        d["scale{0}".format(yincrementNo)] =
geompy.MakeScaleAlongAxes(d["sphere{0}".format(yincrementNo)],
d["p{0}".format(yincrementNo)], (xSize/2), (ySize/2), zSize)
```

Snippet 4.4. Creation of each ellipsoidal geometry.

To make the geometry representing the fluid region within Salome, the film thickness geometry is merged with each of the ellipsoidal shapes. Resulting in half of the ellipsoidal shape being removed and the other half (representing the fluid within the pocket) protruding from the bottom surface of the film thickness. To explain how this is implemented, it is beneficial to start with the explanation of lines 66-76 (Snippet 4.4). Lines 67 and 68 calculate the current positions in x and y axes based on the number of rows and pockets that have been completed so far. The first step in geometry creation is to create a reference point/vertex at the current position (line 70). The name of this object (and the remaining objects are allocated similarly) is given by the dictionary command "d["p{0}".format(yIncrementNo)]". Where "p{0}" is the string (identifier) and {0} is replaced, in this case, by the current value of "yIncrementNo". The next stage is to create a sphere (line 74) based on a unit radius. This sphere is then scaled by the length, width and depth values in their relative directions (line 76). This scaling process *stretches* the sphere into an ellipsoidal shape.

```
00000077         if anglePattern == True:
00000078             d["pVector{0}".format(yincrementNo)] =
geompy.MakeVertex(xDist, yDist,  1)
00000079             d["vector{0}".format(yincrementNo)] =
geompy.MakeVector(d["p{0}".format(yincrementNo)],
d["pVector{0}".format(yincrementNo)])
00000080             d["angle{0}".format(yincrementNo)] =
geompy.MakeRotation(d["scale{0}".format(yincrementNo)],
d["vector{0}".format(yincrementNo)], angle)
00000081             if xincrementNo == 0 and yincrementNo == 0:
00000082                 d["fuse{0}".format(even)]=
geompy.MakeFuseList([d["angle{0}".format(yincrementNo)],domainBox])
```

Snippet 4.5. Rotation of pocket and merge commands.

The last remaining stage of geometry creation is to rotate the pocket (if needed) and merge the pocket with the film thickness geometry (Snippet 4.5). To use the rotate GEOM command "MakeRotation", the object, the axis of rotation and the angle need to be specified. The axis of rotation is specified on line 79 using the "MakeVector" command which requires two points

spaced apart in the direction of the axis required. Hence, “pVector{0}” created on line 78 is the same as point “p{0}” but spaced apart in the z-axis. The command on line 82 then fuses the pocket geometry (for this y-iteration) with the film thickness geometry. Once all the loops in both x and y direction have been completed the final stage is to export the geometry to STL file using the lines seen in Snippet 4.6. This STL file describing the complete model geometry is now ready to be used with the meshing process described in the next section.

```
00000110    geompy.addToStudy(d["fuse{0}".format(odd)],"domainBox")
00000111    geompy.Export(d["fuse{0}".format(odd)],
"bearingSurface.stl", "STL")
```

Snippet 4.6. Export to stereolithography (STL) file.

4.2.2 Grid Setup and Mesh Independence Study

The ideal mesh is hexahedral type, where the primary flow direction is known and the cells faces are aligned with that direction. Having cell faces aligned with the primary flow direction minimises the effects of numerical diffusion. An example of numerical diffusion is seen in Figure 4.11, where the flow direction is out-of-alignment with the cell faces. The model has two regions of fluid with the same flow direction but are at two different temperatures. The fluid has zero thermal conductivity, meaning the flow field should result as the schematic diagram shows, however it does not. This is due to the discretisation process, requiring fluid to only pass through each distinct cell face. Because the flow is aligned with the diagonal, it should go through the node but it can’t according to the mathematical description. Hence a *false* direction results from this, adding additional, artificial diffusion to the transport of physical quantities. If the number of cells are doubled in the example below, the physical distance of the *spreading* is reduced but the same number of cells are exposed to the diffusive problem.

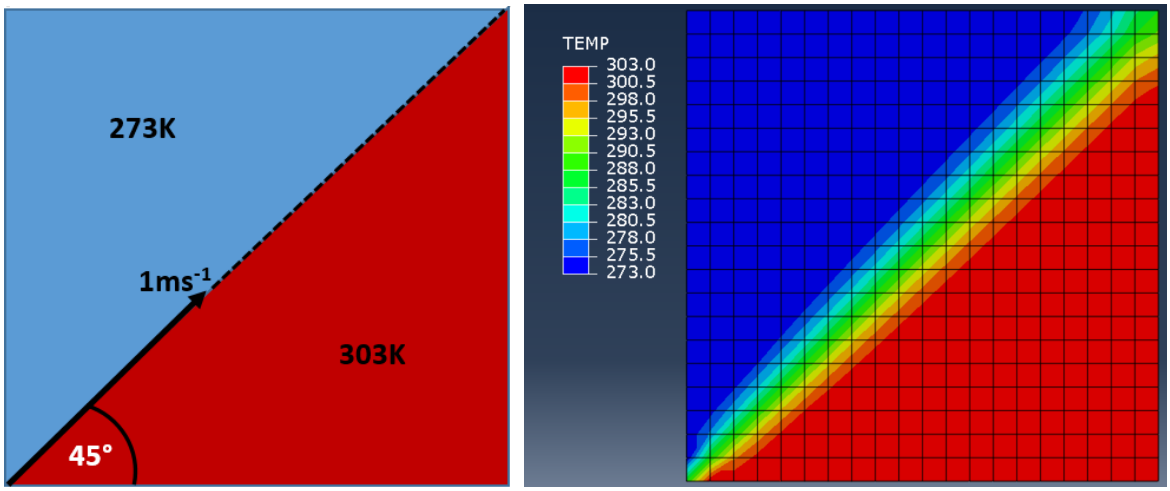


Figure 4.11. Example of numerical diffusion (Wharton and Allanson (2015)).

Unfortunately, a lot of real world CFD problems have swirling, three-dimensional flow movements and this is where other unstructured element types (tetrahedral, polyhedral, prisms…) become more attractive. Another problem with hexahedral meshes is that they are expensive to build. Even experienced users still require large amounts of time to build meshes around complex and organic surfaces. Fortuitously, for this Couette flow problem, the majority of the cells are aligned with the flow direction, except for the deviation of flow direction to be expected near the dimple structures.

There are two stages to the meshing procedure used this CFD investigation: the first stage is to create the background mesh using the blockMesh utility. The second stage is to adjust the background mesh against the surfaces of the STL file, which was generated by the bearingStagger2.py script, using the snappyHexMesh utility. The background mesh is initially completely *structured*, orthogonal and hexahedral-based but the 2nd stage of the meshing procedure will result in the mesh being *unstructured* in parts.

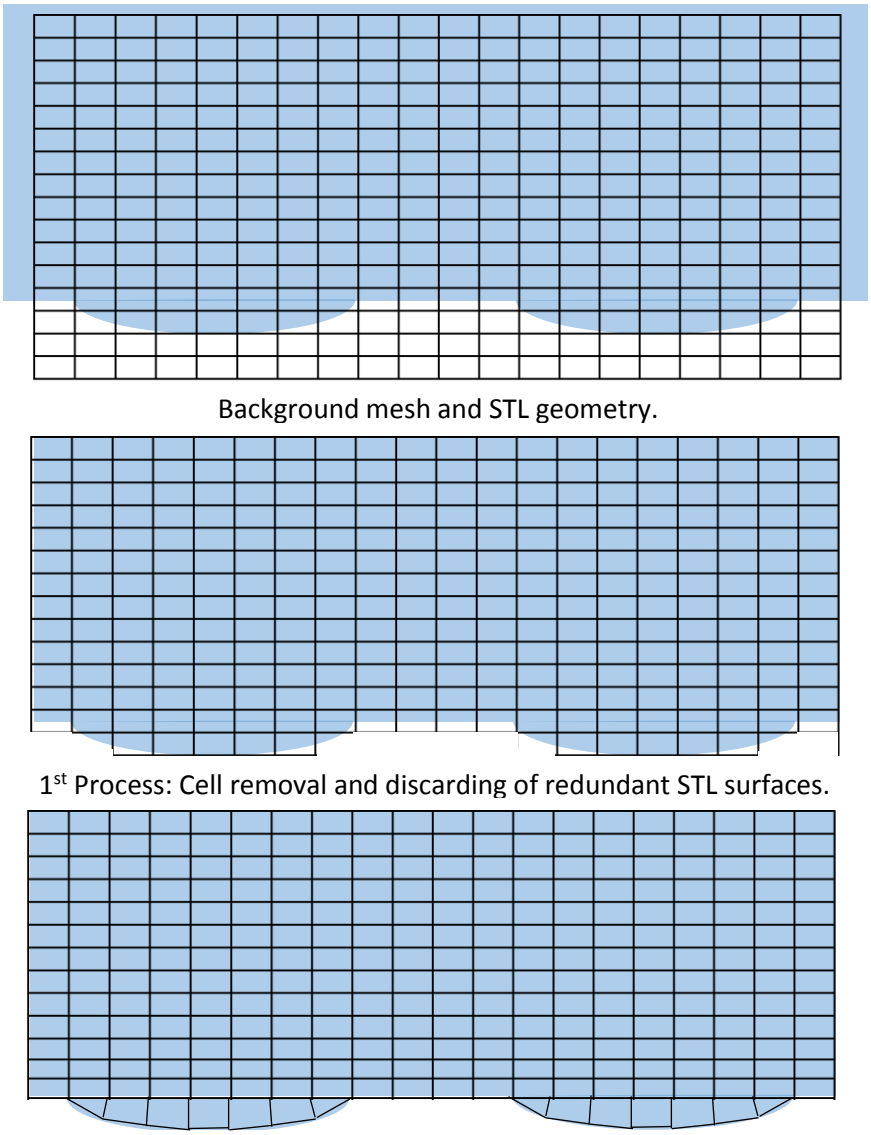


Figure 4.12. Simplistic view of meshing procedure.

An idealistic and simplistic representation of the meshing process when using the snappyHexMesh utility is displayed in Figure 4.12. The black grid represents the background mesh and the blue shaded object represents the STL geometry. The main rules in generating the background mesh are that it should be made of only hexahedral cells and the cells near the surfaces of the geometry should have an aspect ratio of approximately one. Then, before the first process of cell removal occurs, some surface and edge refinement is performed. This means that

the user can define the number of levels of refinement for cells which are intersected by a surface or edge feature of the STL geometry (each level splits the cell in every direction so that one cell becomes 2^3). This refinement may be needed in order to get the cell size, near the surface, more appropriate in order to resolve the features. It is important to note that cell subdivision preserves the aspect ratio of the parent cells. After refinement, the first process, cell removal, is performed which removes any cells that have less than, approximately, 50% of the volume filled by the bounding surface. Also, remember that the film thickness volume was made larger in some directions (see Snippet 4.3)? The reason for this, is that the faces that are closest to the "locationInMesh" user-defined co-ordinate are kept. Looking at Figure 4.12, you can see how the only surface kept from the STL file is the textured surface (the other surfaces of the STL file are not required and are discarded), which is automatically imported as a *wall* (non-slip BC). The other five faces remaining are from the background mesh with their allocated BCs. After cell removal has finished, the faces of the remaining cells do not align with the surface and a *jagged* representation is seen. To remove this low quality depiction of the geometry, the second process of *snapping* is used. The first part of the process is to move the nearest vertices to that surface, this distorts those cells that the vertices belong to. The algorithm then loops iteratively across the grid adjusting the near-surface cells and surrounding cells in order to meet the set mesh quality targets: this includes non-orthogonality, skewness, aspect ratio and smoothness. The last remaining, optional process is to create mesh layers near the surface in order to satisfy boundary layer modelling requirements.

To improve the accuracy of the final CFD solution, further refinement of the mesh (with a mesh independence study) will be required but on the other hand, as mentioned already, mesh quality metrics are also important. Non-orthogonality is a measure of the angle of deviation between the surface vector and the vector between cell centres (see Figure 4.13). Skewness is related to the magnitude of the deviation vector between two points on a cell face. The first point is the face

centroid and the second is the where the centre-to-centre vector intersects with the face. The smoothness mesh quality statistic is related to the cell expansion ratio (Δ1/Δ2). The steeper the transition, the more diffusive it becomes for the flow field. Figure 4.13 shows a poor, steep transition. The last important mesh quality statistic is the cell aspect ratio, which is the ratio between the largest and shortest dimension and it should be aimed to reduce this to as close to one as possible. Although, provided the cell is aligned with the primary flow direction, the size of each dimension can be reduced or increased according to the gradient. So, the size of Δx would be appropriate, provided that a small gradient exists along this direction.

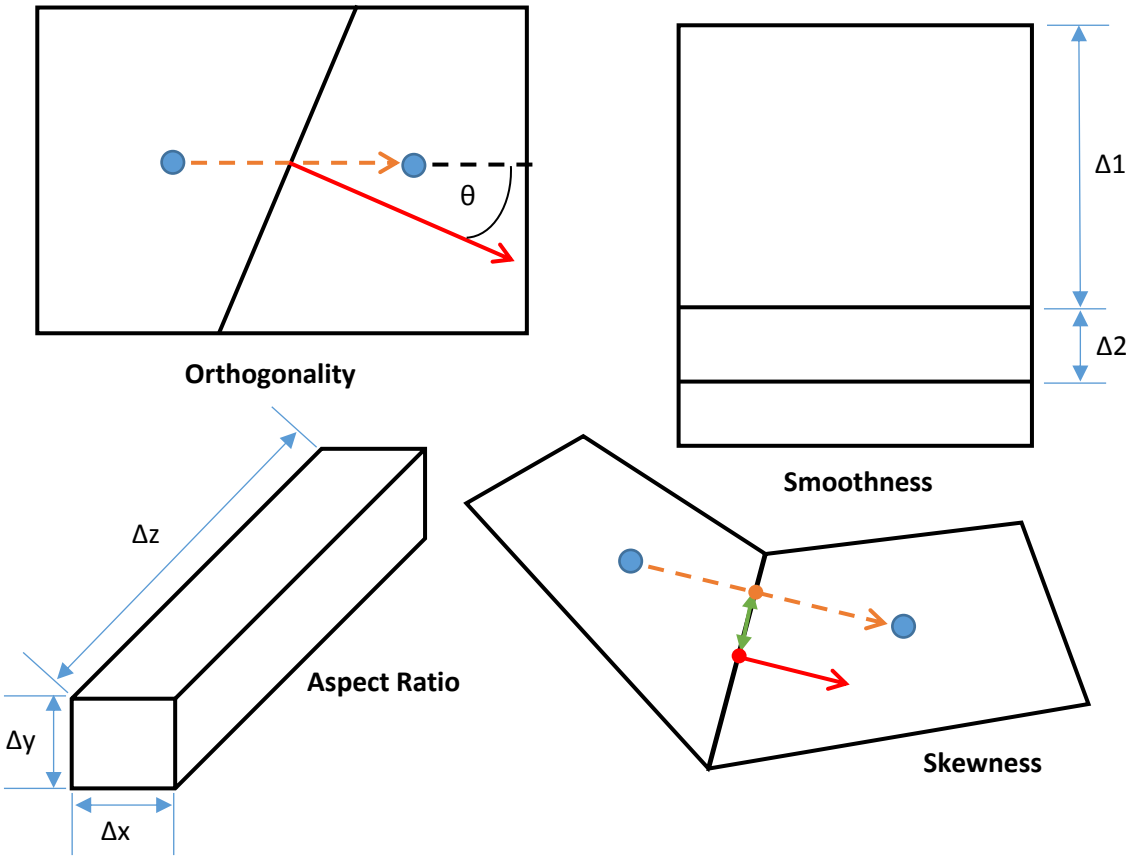


Figure 4.13. Mesh quality definitions. (Note: red arrow is surface vector, blue dot is cell centre point, green arrow is the deviation vector and orange arrow is the vector between cell centres.)

As discussed previously, the background mesh is created first in the meshing process. The grading of the background mesh is based upon the final z-direction sizing of the cells within the pocket.

For the entire mesh at first, an aspect ratio of 8:8:4 is created based on this size. A further two refinements occur using the topoSet (which selects cells to be refined) and the refineHexMesh utility. The first refinement splits all cells, in every direction, below half of the clearance height. The second refines all cells below the journal surface, where the final sizing will be 2:2:1 ratio. The refinement process results in a grading of cells, finer cells in the pockets and larger cells as z-distance increases away from the textured surface. After the background mesh has been created, no further refinement is carried out by the snappyHexMesh utility. The purpose of it is only to apply the cell removal and *snapping* process.

In order to minimise the errors associated with spatial discretisation, it is imperative that a mesh independence study is carried out in order to determine the actual grid sizing required. For this study, the independence of the mesh was quantified using the grid convergence index (GCI). Roache (1994) devised the grid convergence index (GCI) which is based upon the work of Richardson and Gaunt (1927). The theory of Richardson's extrapolation by Richardson and Gaunt is to determine an exact solution for a variable based on a series of lower-order numerical approximations. The discrete solutions of a variable of interest, f, which converges monotonically in the simulation as the grid spacing, h, reduces towards a zero value is described by the following series:

$$f = f_0 + g_1 h + g_2 h^2 + g_3 h^3 + \cdots \tag{4.13}$$

Where, f_0 represents the variable at zero grid spacing (i.e. the exact solution) and g are functions which are independent of the changing grid spacing. If the method is of the order p, then eqn. 4.13 becomes as follows:

$$f = f_0 + g_p h^p + \mathrm{HOT} \tag{4.14}$$

Where HOT stands for higher order terms. Lower order g functions are dropped as they are equal to zero. If two discrete solutions for eqn. 4.14, f_1 and f_2, are provided from two different

computational grids, which have different levels of uniform grading, then these equations can be combined to eliminate g_p and solve for f_0:

$$f_0 = \frac{(h_2^p f_1 - h_1^p f_2)}{(h_2^p - h_1^p)} + \text{HOT} \tag{4.15}$$

Where subscript 1 is the finer grid and 2 is the coarser grid. If the grid refinement ratio, r, which is equal to h_2/h_1 is used to replace the spacing terms in eqn. 4.15 and the HOT terms are dropped then it results in the following:

$$f_0 = f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{4.16}$$

Eqn. 4.16 is then used to form an error estimate for the fine grid solution:

$$E_1 = f_0 - f_1 = \frac{(f_1 - f_2)}{r^p - 1} = \frac{\varepsilon}{r^p - 1} \tag{4.17}$$

Where ε is equal to $f_1 - f_2$. It should be noted that eqn. 4.17 can be presented in fractional form and expressed as a percentage by using ε in relative form ($f_1 - f_2 / f_1$). A second error estimate for the coarse grid is presented in eqn. 4.18.

$$E_2 = f_0 - f_2 = \frac{(f_1 - f_2) r^p}{r^p - 1} = \frac{\varepsilon r^p}{r^p - 1} \tag{4.18}$$

Roache then devised the grid convergence index (GCI) which multiplies the error estimates by a factor of safety, F_s, (see eqn. 4.19 and 4.20) resulting in a more conservative approximation of the error. The reasoning behind this is to improve confidence in the result, statistically. The suggested value of F_s depends on the number of grids being used for the mesh independence study. Roache (2003) recommends for a triplet grid study, which is what will be used for this study, a factor of safety of 1.25 should be used.

$$GCI_{fine} = F_s E_1 \tag{4.19}$$

$$GCI_{coarse} = F_s E_2 \tag{4.20}$$

Further notation shall be expressed for a triplet grid study. For Richardson's extrapolation to be considered valid, the solutions of the reducing grid sizes should monotonically converge. To check whether this is the case or a diverging solution has occurred, the convergence ratio, R, should be calculated:

$$R = \frac{\varepsilon_{12}}{\varepsilon_{23}} \tag{4.21}$$

Subscript 12 refers to solutions between medium and fine grid, whereas, subscript 23 refers to the solutions between coarse and medium grid. The value of R can be highly influenced by the grid refinement ratio. A small selection of r, say below 1.1, could result in a value R near 1 (which means an oscillatory convergence is identified) due to the solutions being largely affected by noise (Roache (1994)). The order of accuracy, p, also needs to be determined for the extrapolation (eqn. 4.22). A central-differencing scheme was used for the discretisation in this study, so it should ideally lead to an order of 2 for the solution accuracy. Unfortunately, as reported by Roache, this is not the case. Other additional factors such as the influence of additional schemes (e.g. to calculate drag force for the variable), mesh related issues (i.e. strong grid stretching), errors in the coding, influence of non-linear flow mechanisms… Are a few to mention.

$$\frac{\varepsilon_{23}}{\varepsilon_{12}} = \frac{(r_{23}^{p}-1)r_{12}^{p}}{(r_{12}^{p}-1)} \xrightarrow{\text{if } r=r_{12}=r_{23}\text{ , rearrange for p}} p = \frac{\ln(\varepsilon_{23}/\varepsilon_{12})}{\ln(r)} \tag{4.22}$$

Once, the GCI has been calculated, the next stage is to compute the grid spacing required to achieve the required accuracy. The required grid convergence index, GCI*, is used with eqn. 4.23 to calculate the grid refinement ratio, r*, needed for the required accuracy. r* combined with the coarse grid spacing size will enable the determination of the actual grid spacing size.

$$r^{*} = \left(\frac{GCI^{*}}{GCI_{23}}\right)^{\frac{1}{p}} \tag{4.23}$$

The mesh independence study was carried out on a chosen set of conditions which are described in Table 4.1. Due to the whole grid being constructed and graded on the basis of the z-direction sizing of the cells in the pocket, this would be used to calculate the grid refinement ratio in the Richardson's extrapolation study. Three different grids were graded based upon the number of cells through the depth of the pocket: 10 cells (0.5µm), 20 cells (0.25µm) and 40 cells (0.125µm). This meant that a grid refinement ratio of 2 was selected. A grid refinement ratio of 1.5 was investigated before arriving at this value, however but this led to a convergence ratio that was too close to 1 and poor prediction of r* was observed.

Table 4.1. Description of surface used in mesh independence study.

Dimension	Value
Streamwise Spacing	500µm
Spanwise Spacing	150µm
Staggered?	Yes, ½ Pitch
Pocket Length	200µm
Pocket Width	75µm
Pocket Depth	5µm
Radial Clearance	7µm
Texture Alignment	0^o (Streamwise)

The shear/friction force (F_f) was selected to be the variable used for interrogating the mesh independence, as this is the quantity of interest for the parametric study. Table 4.2 shows that as the number of cells were increased, the force increased. A substantial increase of cells were observed across the differing grids resulting in solver times ranging significantly, from 53s on the coarse grid to 33069s on the finest grid. (Computation was performed on a workstation with dual Intel Xeon E5-2650 v4 (Broadwell) 2.9GHz, equating to 24 physical cores with 128GB of DDR4 RAM).

Table 4.2. Mesh independence study details.

Grading	No. of Cells	Cell Depth Size (um)	F_f (mN)
Coarse (3)	590,749	0.500	0.664699
Medium (2)	5,491,017	0.250	0.725600
Fine (1)	38,657,841	0.125	0.729640

Note: F_f – shear force, F_p – pressure force and F_T – total force.

The results of the Richardson's extrapolation study can be seen in Table 4.3. It can be seen that the convergence ratio is 0.066, showing that the solution across the grids does in fact monotonically converge, very well in fact. Using eqn. 4.16, the exact solution (at zero grid spacing) for the friction force was determined as 0.729353mN. A GCI* of 5% was used to calculate the grid spacing required for the parametric study. To some, this chosen error may seem fairly large but there is logical reasoning behind this. At 5%, the number of cells across the depth of the pocket would be approximately 12. For this particular surface, this results in a grid size of 2,224,688 cells. Now, depending on the length scale of the pocket depth to the spacing dimensions, the number of cells may increase/decrease. In fact, one of the conditions from the parametric study will result in a spacing to pocket ratio of 500 whereas the surface used for the GCI analysis will only have a ratio of 100. So substantially higher cell counts will be seen and for that reason, the 5% GCI* is chosen as a balance between accuracy and computational limitations. Images of the mesh generated for this chosen GCI* are presented in Figure 4.14.

Table 4.3. Details of the Richardson's extrapolation study.

Parameter	Resulting Value	
ε_{32} (Abs.)	0.060901	mN
ε_{21} (Abs.)	0.004040	mN
ε_{32} (Rel.)	8.39	%
ε_{21} (Rel.)	0.55	%
r	2.0	
p	3.914	
F_f Exact	0.729353	mN
R	0.066	
E_1 (Abs.)	0.000287	mN
E_2 (Abs.)	0.065228	mN
GCI_{21}	0.05	%
GCI_{32}	11.24	%
GCI^*	5.00	%
r*	1.230	
Cell Size Req.	0.407	um

Note: Abs. – Absolute and Rel. – Relative.

For a laminar Couette flow, some may come to the conclusion that the mesh requirements for this study is exceptionally large but there are reasons for this. The first being ease of use. SnappyHexMesh is an exceptionally powerful tool that allows complex surfaces to be meshed using hexahedral type cells but to achieve good quality meshes, it is best to try and reduce the background mesh to aspect ratios near one where surface intersections are expected and this in itself will result in high densities of cells in these areas. SnappyHexMesh is used instead of domain sectioning to achieve hex-based cells so that automation of the process is easily achieved. The second is in relation to the geometry of the surface structure. Ellipsoidal shapes are made up completely of curved surfaces and these are notoriously difficult to mesh and require large cell densities in order to resolve them. Because these are complex three-dimensional shapes, high cell counts are expected. As a result of this, some researchers have tried to justify the use of two-dimensional simplification of three-dimensional structures for full, Navier-Stokes CFD approximations.

Table 4.4. Mesh quality statistics for the chosen GCI* grid.

Quality Statistic	Value
Max aspect ratio	9.207
Non-orthogonality (max)	56.965
Non-orthogonality (avg.)	13.143
Max skewness	2.224

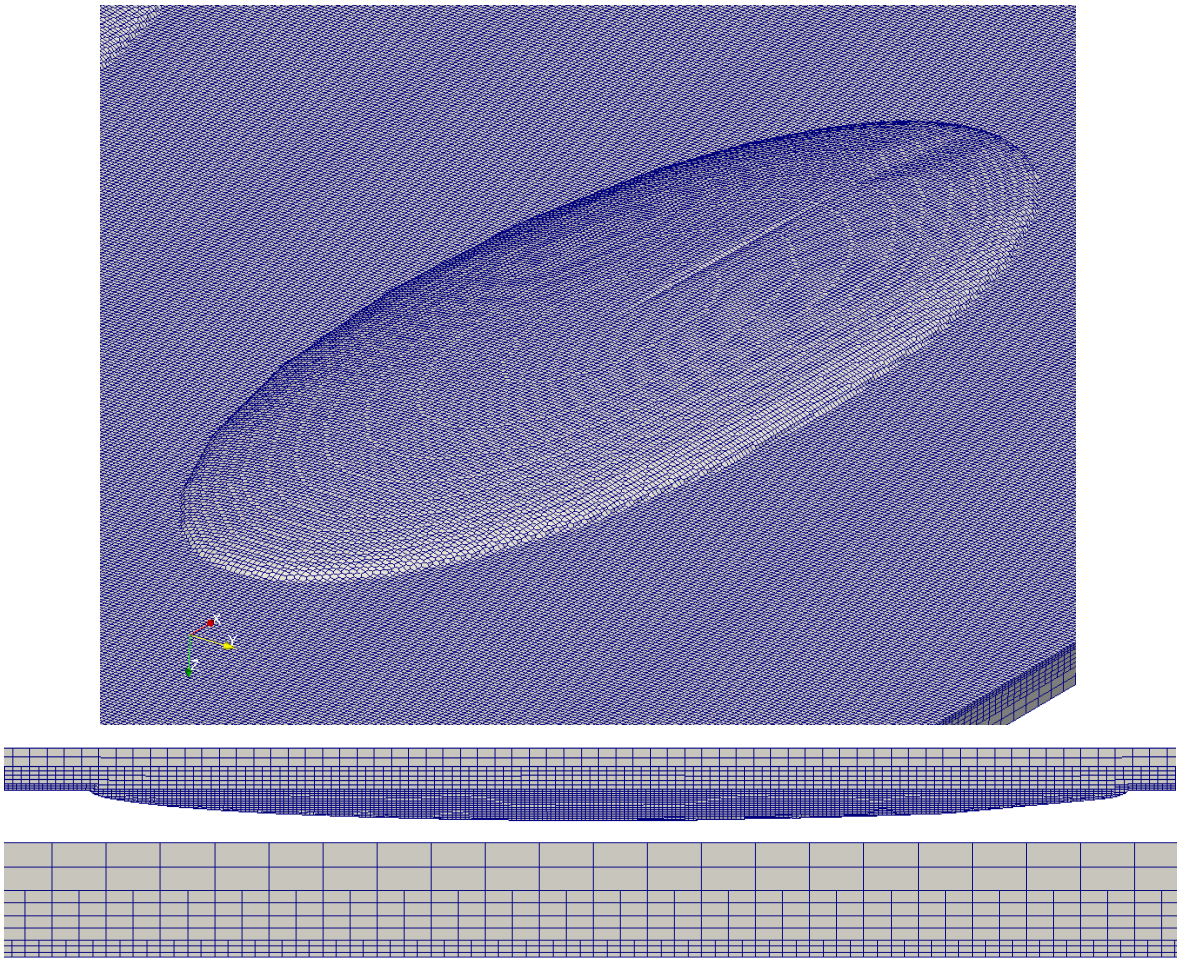


Figure 4.14. 3D and cross-sectional views of the grid at GCI*.

4.2.3 Parametric Study Script

Now that the grid setup and mesh independence study has been presented, the final part of the CFD setup is in the automation of the modelling for the parametric study. Again, like the geometry creation script shown in section 4.2.1, the modelling was automated by two python scripts. A flow diagram of the processes implemented by the two scripts are presented in Figure 4.15. The idea behind the compareStudy2.py script is to first read a comma separated value (CSV) file, containing all the different trials and conditions, which is then imported into an array. For each trial, the conditions are passed to the bearing_program.py which automates the whole modelling procedure. After one full modelling study has been completed, the compareStudy2.py script packages all the files and folders of the working directory into a folder named “study#” (where # is the trial number). This process repeats for each trial, as one iteration within the WHILE loop of the script, until all trials have been completed and the results stored in their appropriate folders.

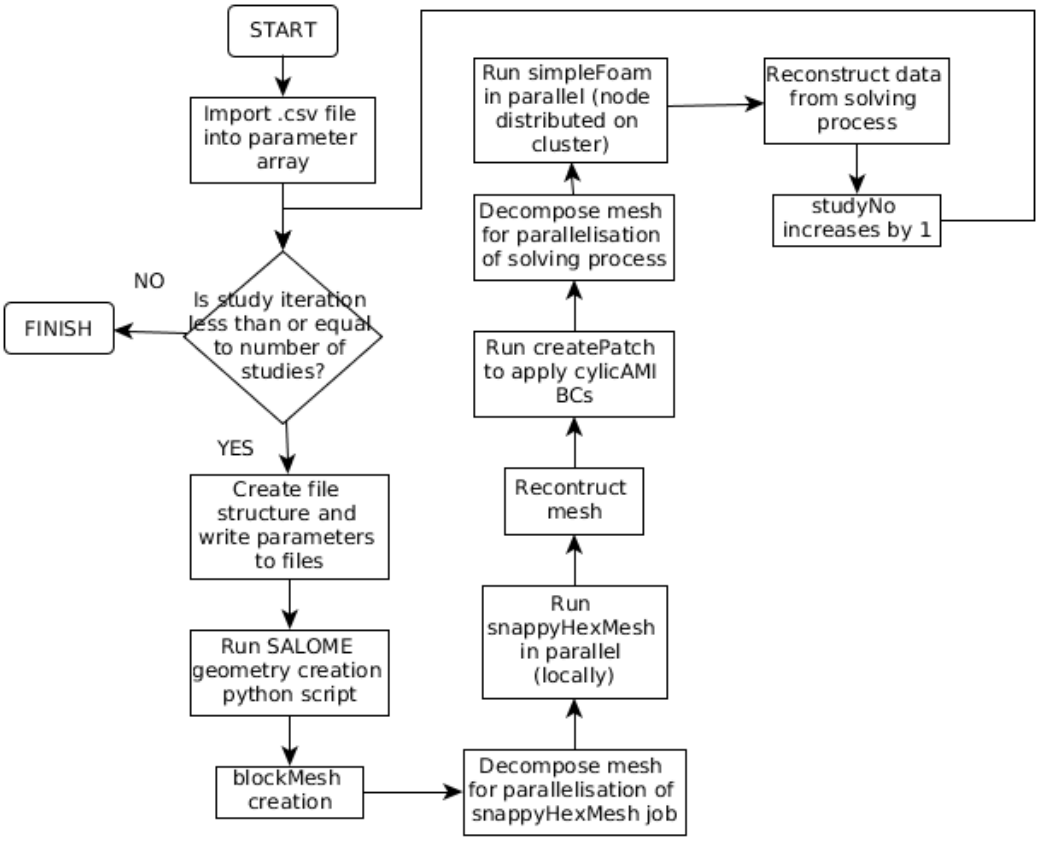


Figure 4.15. Flow diagram of *compareStudy2.py* and *bearing_program.py* (Wharton et al. (2016)).

Due to the simplicity of the script, snippets will not be *walked through* for the compareStudy2.py script but can be viewed in Appendix I for reference. The remaining part of this section shall concentrate on the functionality of the main script, bearing_program.py.

```
00000008 parser = argparse.ArgumentParser()
00000009 parser.add_argument("xSpacing", type=float)
00000010 parser.add_argument("ySpacing", type=float)
00000011 parser.add_argument("length", type=float)
00000012 parser.add_argument("width", type=float)
00000013 parser.add_argument("depth", type=int)
00000014 parser.add_argument("clearance", type=float)
00000015 parser.add_argument("stagDist", type=float)
00000016 parser.add_argument("angle", type=float)
00000017 args = parser.parse_args()
```

Snippet 4.7. argparse commands.

To call the modelling script, the command “python bearing_program.py %s %s %s %s %s %s %s %s” is executed. Each of the %s is an argument that is passed to the modelling script from the calling script and the order in which they appear, in the call, is the order they are interpreted by the “add_argument” functions (from the argparse module) seen in Snippet 4.7 (i.e. the first variable is “xSpacing” and the last variable is “angle”). The variable “args” is defined by “parser.parse_args()”. This means that each of arguments can be used within the script as “args.#”, where # represents the name of the variable given in the “add_argument” function. All of the arguments are immediately used to define the variables used in the script (see Snippet 4.8). Variables “xQuantity” and “yQuantity” specify the number of dimples in the x and y direction. The number of dimples was set to two in each direction. This is the minimum number of textures in each direction which can guarantee the periodicity of the repeating pattern for all different scenarios. If the periodicity of the surface could not be guaranteed for all scenarios, the validity of the periodic boundary condition would break down. “speed” is the linear speed of the moving wall. “procMesh” and “procSolve” are the number of processors that will be used for the parallel execution of the meshing and solver programs. “textRefine” is used to define the number of cells required across the pocket depth, which is used for the grading of the mesh and its value was determined by the mesh independence study. The “angle” and the “staggerDist” variables are used to determine the *True* or *False* values of “anglePattern” and “staggeredGrid” through the use of the IF statements seen on lines 38-46.

```
00000021 xQuantity = 2
00000022 yQuantity = 2
00000023 xincrementDist = args.xSpacing
00000024 yincrementDist = args.ySpacing
00000025 xSize = args.length
00000026 ySize = args.width
00000027 zSize = args.depth
00000028 radialClearance = args.clearance
00000029
00000030 staggerDist = args.stagDist
00000031 speed = 5
00000032
00000033 angle = args.angle
00000034 procMesh = 23
00000035 procSolve = 23
00000036 textRefine = 12
00000037
00000038 if angle == 0:
00000039    anglePattern = False
00000040 else:
00000041    anglePattern = True
00000042
00000043 if staggerDist == 0:
00000044    staggeredGrid = False
00000045 else:
00000046    staggeredGrid = True
```

Snippet 4.8. Variables defined in the modelling script.

The next part of the script uses the “cp –R” linux command to copy the “constant” and “system” folders as well as the bearingStagger2.py script from the source directory. The source directory contains a copy of the working directory needed to run a SimpleFoam job, except all the settings in the dictionary files are filled with placeholders. The script calculates all the settings based on the trial conditions and then overwrites the placeholders with the required values.

One of the first dictionary files to be written to, is for the blockMesh utility, which generates the background mesh. The computational domain is defined by eight vertices starting with vertex 0, which is where the axis origin is placed. The x-axis and its direction is defined by vertex 1 (going from the origin to this point). The y axis distance and direction is defined by going from vertex 1 to 2. Vertices 0, 1, 2 and 3 are all placed on the same plane for z position. The distance and direction of the z-axis is defined from vertex 0 to 4. For each plane in the z axis, the vertices are described in an anti-clockwise fashion (looking against the z-axis direction) from the origin vertex (0 on the

origin plane, 4 on the other). On line 120, the variable “vertex0Old” defines the string that is the placeholder within the file. Then on the next line, the variable “vertex0New” is the replacement string for that placeholder.

```
00000097 xAngleCorrect = (xSize/2)*(math.cos(angle))
00000098 yAngleCorrect = (xSize/2)*(math.sin(angle))
00000099
00000100
00000101 if anglePattern == True:
00000102     minVertexX = -(ySize/2)
00000103     if staggeredGrid == True:
00000104         maxVertexX = (xQuantity * xincrementDist) - (ySize/2)
00000105         maxVertexY = ((yQuantity -1) * yincrementDist) +
yAngleCorrect + staggerDist
00000106         minVertexY = -yAngleCorrect
00000107     else:
00000108         maxVertexX = ((xQuantity) * xincrementDist) -
(ySize/2)
00000109         maxVertexY = ((yQuantity - 1) * yincrementDist) +
yAngleCorrect
00000110         minVertexY = -yAngleCorrect
00000111 else:
00000112     minVertexX = 0
00000113     maxVertexX = (xQuantity * xincrementDist)
00000114     minVertexY = -(ySize/2)
00000115     if staggeredGrid == True:
00000116         maxVertexY = ((yQuantity - 1) * yincrementDist) +
(ySize/2) + staggerDist
00000117     else:
00000118         maxVertexY = (yQuantity * yincrementDist) - (ySize/2)
00000119
00000120 vertex0Old = '(x1 y1 z1)'
00000121 vertex0New = '(%s %s -%s)\n' % (minVertexX, minVertexY,
zSize)
00000122 vertex1Old = '(x2 y1 z1)'
00000123 vertex1New = '(%s %s -%s)\n' % (maxVertexX, minVertexY,
zSize)
00000124 vertex2Old = '(x2 y2 z1)'
00000125 vertex2New = '(%s %s -%s)\n' % (maxVertexX, maxVertexY,
zSize)
00000126 vertex3Old = '(x1 y2 z1)'
00000127 vertex3New = '(%s %s -%s)\n' % (minVertexX, maxVertexY,
zSize)
00000128 vertex4Old = '(x1 y1 z2)'
00000129 vertex4New = '(%s %s %s)\n' % (minVertexX, minVertexY,
radialClearance)
00000130 vertex5Old = '(x2 y1 z2)'
00000131 vertex5New = '(%s %s %s)\n' % (maxVertexX, minVertexY,
radialClearance)
00000132 vertex6Old = '(x2 y2 z2)'
00000133 vertex6New = '(%s %s %s)\n' % (maxVertexX, maxVertexY,
radialClearance)
00000134 vertex7Old = '(x1 y2 z2)'
00000135 vertex7New = '(%s %s %s)\n' % (minVertexX, maxVertexY,
radialClearance)
```

Snippet 4.9. Calculation of vertices for blockMesh dictionary file.

Lines 122- 135 describe the placeholder strings and replacement strings for the remaining vertices. Lines 101 to 118 calculate the maximum and minimum points for the x (streamwise) and y (spanwise) direction. These values are then used to define the %s values used in the replacement strings. To actually write these replacements to the dictionary files, an example of one replacement script is shown by Snippet 4.10, which writes the co-ordinate for vertex 0 in the blockMeshDict file. First the location of the dictionary file is specified in Line 69, then a temporary version of the file is given a name and location (line 70). Line 337 opens the original dictionary file in read-only state. It then opens the temporary copy of the file (line 338) with write permissions. On lines 337-342, the script searches the read-only file for the string and if the placeholder string exists within the dictionary file, the same line in the temporary file is replaced completely with the replacement string. The replacement string always ends with "\n" which will tell the write command to keep the current and next line separated. After this, the new version of the temporary file needs to replace the old dictionary file. So the old file is deleted (line 344) and the temporary file is renamed (line 346). Every time a placeholder is replaced, similar lines of code to that seen on lines 337 to 346 will be used.

```
00000069 blockMeshLoc = './system/blockMeshDict'
00000070 blockMeshLocTmp = './system/blockMeshDictTmp'
...
00000337 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000338     for line in input_file:
00000339         if line.strip() == vertex0Old :
00000340             output_file.write(vertex0New)
00000341         else:
00000342             output_file.write(line)
00000343
00000344 os.system('rm -R %s' % (blockMeshLoc))
00000345
00000346 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
```

Snippet 4.10. Script used for placeholder replacement.

Another file that is written to is the bearingStagger2.py script. Remember the placeholders "ChangeNo#" defining the variables within the script? These are replaced with the surface dimensions/descriptions needed to create the STL file. After each trial is completed, a copy of the script is kept within each folder so that the user can refer back to the configuration used for that particular trial.

The next dictionary file to be written to is responsible for the configuration of the createPatch function. This OpenFOAM utility enables for patches to be applied. A patch basically defines the type of boundary condition that will be applied to a particular surface and is usually specified within the blockMeshDict file. Unfortunately, there was a bug within the code that effected the application of periodic patches before the snappyHexMesh procedure, so the patches are applied after the meshing process has finished. So a temporary patch is applied to the inlet/outlet and back/front faces and then createPatch changes this to the cyclicAMI patch type after mesh generation has finished. To define each of the cyclicAMI patches, the linked face needs to be specified and the code needs to be told of the distance between them ("separationVector"). The calculation of these distances and the placeholder replacement strings can be seen in Snippet 4.11.

```
00000139 symSepDist = maxVertexY - minVertexY
00000140
00000141 xCyclicDist = maxVertexX - minVertexX
...
00000147 sepVect1Old = 'separationVector (xDist1 0 0);'
00000148 sepVect1New = '            separationVector (%se-6 0 0);\n'
% (xCyclicDist)
00000149 sepVect2Old = 'separationVector (xDist2 0 0);'
00000150 sepVect2New = '            separationVector (-%se-6 0 0);\n'
% (xCyclicDist)
00000151 sepVect3Old = 'separationVector (0 xDist3 0);'
00000152 sepVect3New = '            separationVector (0 %se-6 0);\n'
% (symSepDist)
00000153 sepVect4Old = 'separationVector (0 xDist4 0);'
00000154 sepVect4New = '            separationVector (0 -%se-6 0);\n'
% (symSepDist)
```

Snippet 4.11. Separation vector placeholder for createPatch function.

The next operation is to specify the mesh grading settings for the blockMesh file and searching criteria for the topoSet utility which enables refinement of the background mesh. Lines 427 and 428 converts the pocket depth and radial clearance values into units of metres from the input data in microns. Line 430 then calculates the sizing of the cells through the pocket depth. The pocket depth size is divided by the number of cells required (12) and then multiplies it by four. This is because the region undergoes two sets of cell splitting. The cell size for the x and y direction is made twice the size (line 432). Lines 431 and 433-434 then calculate, based on the computational domain sizing, the number of cells required for each direction (Nx, Ny, Nz) by dividing the dimension by the sizing. Because an integer value is required (there is no such thing as a fraction of a cell), the result of the division is then rounded up using the numpy command "np.ceil" and then "int" is used to convert the floating point result into an integer, removing any decimals. The placeholder (line 436) and its replacement (line 437) for the grading can be seen in the code snippet.

```
00000427 zSizeAbs = np.multiply(zSize,0.000001)
00000428 radialClearanceAbs = np.multiply(radialClearance,0.000001)
00000429
00000430 zCellSize = np.multiply(4,np.divide(zSizeAbs,textRefine))
00000431 Nz =
int(np.ceil(np.divide(np.add(radialClearanceAbs,zSizeAbs),zCellSize))
)
00000432 xyCellSize = 2 * zCellSize
00000433 Nx =
int(np.ceil(np.divide(np.multiply(xCyclicDist,0.000001),xyCellSize)))
00000434 Ny =
int(np.ceil(np.divide(np.multiply(symSepDist,0.000001),xyCellSize)))
00000435
00000436 meshGradOld = 'hex (0 1 2 3 4 5 6 7) (x y z) simpleGrading
(1 1 1)'
00000437 meshGradNew = '    hex (0 1 2 3 4 5 6 7) (%s %s %s)
simpleGrading (1 1 1)\n' % (Nx, Ny, Nz)
...
00000451 topoSet1z = np.divide(radialClearance,2)
00000452 topoSet2z = 0
...
00000454 topoSet1Old = 'box (x1 y1 z1) (x2 yz z2);'
00000455 topoSet1New = '            box (%se-6 %se-6 -%se-6) (%se-6
%se-6 %se-6);\n' % (minVertexX, minVertexY, zSize, maxVertexX,
maxVertexY, topoSet1z)
00000456 topoSet2Old = 'box (x1 y1 z1) (x2 yz z2);'
00000457 topoSet2New = '            box (%se-6 %se-6 -%se-6) (%se-6
%se-6 %se-6);\n' % (minVertexX, minVertexY, zSize, maxVertexX,
maxVertexY, topoSet2z)
```

Snippet 4.12. blockMesh grading and topoSet settings.

The topoSet box search setting is used, which basically marks all cells within the specified box and creates a set. When refineHexMesh is used, it uses this set to identify which cells to split. The search box dimensions are specified for the first refinement (applied to cells below half of the radial clearance) and second refinement (applied to cells below the journal surface) on lines 455 and 457, respectively.

For the snappyHexMesh dictionary file, the parameter which changes case-by-case is the “locationInMesh” coordinate. When the cell removal process is undertaken, the programme needs to know which section of the mesh to keep. Is it within the volume described by the STL file or outside of this volume (Figure 4.12)? This point is placed in the middle of the clearance geometry (line 499).

```
00000498 locInMeshOld = 'locationInMesh (x y z);'
00000499 locInMeshNew = '    locationInMesh (%se-6 %se-6 %se-6);\n'%
((maxVertexX/2), (maxVertexY/2), (radialClearance/2))
```

Snippet 4.13. Location in mesh point placeholder and replacement.

Before the meshing operation and subsequent solving takes place, the programme is told how many physical threads are to be employed (used) for parallel execution. To facilitate this, the mesh needs to be split into a number of subdomains which are allocated to each of the available cores. This is achieved using the decomposePar utility, with the scotch method. The simple method, an alternative to scotch, simply divides the mesh along the specified directions with equal spacing. As there is a variation in grid densities throughout the domain this can result in partitions which have higher cell counts and an increased number of source/target faces. Unbalanced partitions result in a slower solver time for two reasons. The first due to the larger number of points to be solved for some partitions over overs, leaving some cores *hanging* (waiting for other partitions to reach the end of that iteration). The second reason is due to communication between source and target faces on partitions. Faces on the edge of the partitions are linked and require data from other paritions. Some cores/nodes will lag behind others if larger amounts of data need to be received. The scotch algorithm analyses the mesh and performs a balancing calculation for sizing the partitions, ensuring that the drawbacks just mentioned are minimised. The “procMesh” variable seen earlier, which is manually adjusted by the user changes the placeholder seen in Snippet 4.14. Before solving, the same dictionary file will be altered again (line 631-643), replacing the exact same string but using the variable “procSolve”.

```
00000539 decompMeshLineOld = 'numberOfSubdomains procs;'
00000540 decompMeshLineNew = 'numberOfSubdomains %s;\n' % (procMesh)
```

Snippet 4.14. decomposePar dictionary setting.

Now that the placeholders in the required dictionary files have been edited, the next step is to execute the geometry creation script with Salome in shell mode (i.e. without the GUI). This is undertaken using the command on line 555: “subprocess.call('#/appli_V2015_2/salome -t python bearingStagger2.py --ns-port-log=salomePort.txt', shell=True)” where # represents the path to installation directory. Once this is completed, the STL file is moved to the required “triSurface” folder within “constant” of the working directory. Then the surfaceFeatureExtract command is run (line 562). This enables OpenFOAM to *fit/wrap* surfaces around the STL file, generating an .eMesh file. This file, which is recognised by snappyHexMesh, will use the extracted surfaces for detecting the intersecting cells, allowing for the *snapping* process.

The meshing process begins with the execution of blockMesh on line 565. This is the only procedure that has to be run in a serial manner. The remaining mesh manipulation processes can be run in parallel so as to reduce the time consumed. After the background mesh has been generated, the decomposePar (line 570) facility will partition the domain. Now partitioned, the background mesh is refined twice using the code seen in Snippet 4.15. A WHILE loop is used with the variables “meshRefInc” and “meshRefEnd” to designate how many iterations of refinement are needed. The combination of the “foamJob –p –s” with topoSet command allows the job to run in parallel. The argument –p tells the code to run topoSet in parallel and –s stands for screen, which means that the output is generated to screen (logged). refineHexMesh is then executed with the arguments, “c0”, which is the name of the set containing the marked cells for refinement and –overwrite means simply to overwrite the mesh file within the “constant/polyMesh” directory, rather than generating additional folders containing the mesh (only useful when debugging). The last stage of the meshing operation is then performed using the command on line 578: “foamJob –p –s snappyHexMesh”.

```
00000567 meshRefInc = 1
00000568 meshRefEnd = 3
...
00000572 while meshRefInc < meshRefEnd:
00000573     os.system("foamJob -p -s topoSet -dict
system/topoSetDict.%s" % (meshRefInc))
00000574     os.system("foamJob -p -s refineHexMesh c0 -overwrite")
00000575     meshRefInc += 1
```

Snippet 4.15. Refinement commands.

snappyHexMesh generates a separate folder for each stage of the mesh manipulation process, with the final mesh being output to the folder named “2”. OpenFOAM is instructed to reconstruct the mesh from this folder using the command “reconstructParMesh –time 2” on line 580. The mesh within “polyMesh” is then deleted and replaced with the newly reconstructed mesh. Now that the meshing process has completed, the “createPatch” command discussed earlier, can now apply the periodic “cyclicAMI” patch to the appropriate faces using the designated separation vectors.

```
00000602 speedOldLine = 'value           uniform (speed 0 0);'
00000603 speedNewLine = '        value           uniform (%s 0 0);\n'
% (speed)
00000604 intOldLine = 'internalField   uniform (speed 0 0);'
00000605 intNewLine = 'internalField   uniform (%s 0 0);\n' % (speed)
```

Snippet 4.16. Velocity boundary condition and initialisation of field.

Next, the boundary conditions in the “0” folder are copied from the “source” folder and the only file to be altered is the “U” file (velocity). The placeholders to be overridden in this file are the x-component uniform velocity for the sliding wall and the internal velocity field (see Snippet 4.16), dictated by the “speed” variable. Giving the internal field a prescribed velocity before solving is called initialising. By doing this, the solver is given a much better guess to start with and this should reduce the number of iterations required for the solving process. Again, the job is partitioned ready for solving (line 646). The last command before the solver is executed, is again, to improve the efficiency of the solving process: “mpirun -np %s renumberMesh -overwrite -

parallel" where %s is the number of processors to be used. The renumberMesh facility is a bandwidth minimisation facility. Using the method of Cuthill and McKee (1969), a sparse matrix of A (for the solving of A·x=B type problem) has the rows and columns permuted in a way that the non-zero values are aligned in a more diagonal form (it does this by reordering the cell list). Reducing the bandwidth is particularly useful for the solver because the new matrix will only contain the *diagonal* of the original matrix. Not having to store the many zero values outside this diagonal is better for memory requirements and also improves the efficiency of the solver because it is not having to perform pointless calculations on these values. The SIMPLE solver is now run using the command “mpirun -np %s simpleFoam -parallel > log” (line 648). When the solving process is finished, the domain partitions are reconstructed and all the files from the working directory are moved into a folder, ready for the next trial.

4.2.4 Trial Conditions for the Parametric Study

This section of the report details the trial conditions used for the parametric study. In order to perform some optimisation of the surface structure and study the effects of varying parameter values, some consideration went into the design of the trial conditions so that valid conclusions could be made from the results. It was decided that the orientation of the textures should be tested to two levels (streamwise and spanwise alignment) whilst all other parameters should be tested to three levels, in order to generate a more robust set of conclusions/findings based on this analysis results. The orientation was only tested to two levels because when some other angle was tested, the symmetry of model was violated due to the way in which the geometry and the cyclic BCs were setup. The values chosen for each of the parameter levels are described in Table 4.5. If a full factorial study were to be performed using these trial conditions, it would require 4374 trials. Based on the analyses that were performed, the average simulation takes approximately two hours to solve (not including other processes). Multiplying this by the number

of trials, a full factorial study would take roughly a whole year (364.5 days). Obviously, this was not a practical proposition. So, to solve this issue, Taguchi's factorial design of experiments method was used. Taguchi (1986) designed a set of orthogonal arrays which allow for an experimental design where all the parameter levels are equally weighted against each other, whilst substantially reducing the number of trials when compared against a full factorial study. By using this design method, each of the parameters and their effect on the measured variable, can be independently analysed. This fractional factorial approach enables for an accurate prediction of optimal parameters (and the effects on the measured variable) whilst being efficient in reaching those conclusions. The minimum mixed level orthogonal array needed to cover the trial conditions (Table 4.5) is a L18 ($2^1 3^7$) array, which was taken from the set of designed tables by Taguchi (see orthogonal array used in Table 4.6). In total, this experimental design only requires 18 trials which would, according to the average time to solve, take 36 hours to complete. In order to implement this investigation, a comma separated value (CSV) file, which was needed for the script input, was created based on the orthogonal array in Table 4.6, replacing each of the parameter level numbers for each trial with the values specified in Table 4.5. The recommended range of sizes for each of the dimensions, specified in Table 4.5, was chosen based on the limitations of the machining process and the machine setup used. The approximate limitations were found by the initial investigation which produced the surface shown in Figure 4.4. As mentioned earlier, the number of pockets in each direction will remain constant, two in each direction. This guarantees the periodicity of the surface ensuring the validity of the translational-type cyclic boundary condition. According to section 4.2.3 and Snippet 4.9, this also means that the sizing of the computational domain will based upon the spacing in each direction multiplied by the number of pockets, which in this case is two. For the trial conditions (presented in Table 4.5), this will create a maximum domain size of 1mm streamwise size and 0.45mm spanwise size. The minimum domain computational domain size will be 0.6mm in the streamwise direction and 0.1mm in the spanwise axis.

Table 4.5. Trial conditions.

Level	1	2	3
Angle (A)	0°	90°	
Length (B)	200µm	250µm	300µm
Width (C)	25µm	50µm	75µm
Depth (D)	1µm	3µm	5µm
X Pitch (E)	300µm	400µm	500µm
Y Pitch (F)	50µm	100µm	150µm
Stagger Dist. (G)	0	¼ pitch	½ pitch
Clearance (H)	3µm	5µm	7µm

Table 4.6. L18 ($2^1 3^7$) orthogonal array.

Trial No.	Angle	Length	Width	Depth	X Pitch	Y Pitch	Stagger Distance	Clearance
	A	B	C	D	E	F	G	H
0	1	1	1	1	1	1	1	1
1	1	1	2	2	2	2	2	2
2	1	1	3	3	3	3	3	3
3	1	2	1	1	2	2	3	3
4	1	2	2	2	3	3	1	1
5	1	2	3	3	1	1	2	2
6	1	3	1	2	1	3	2	3
7	1	3	2	3	2	1	3	1
8	1	3	3	1	3	2	1	2
9	2	1	1	3	3	2	2	1
10	2	1	2	1	1	3	3	2
11	2	1	3	2	2	1	1	3
12	2	2	1	2	3	1	3	2
13	2	2	2	3	1	2	1	3
14	2	2	3	1	2	3	2	1
15	2	3	1	3	2	3	1	2
16	2	3	2	1	3	1	2	3
17	2	3	3	2	1	2	3	1

4.3 Results of the Parametric Study

After the parametric study had completed, an evaluation of the results was performed using direct effect charts (Figure 4.16-4.19). These direct effect charts are created by taking the average of the results for all studies where that parameter level has occurred. The result, also known as a performance indicator was chosen to be shear stress, lift pressure, lift to drag ratio and percentage drag reduction. The direct effect charts are useful because they give a visual representation of how strongly each parameter influences the frictional resistance losses. These particular indicators were chosen because they reflect the aim of the study, which is to minimise surface drag effects. The shear stress and lift pressure indicators should really be referred to as average or mean values because they are calculated by dividing the shear or lift force by the area that they act on. All of the data produced from each of the trials in the parametric study as well as the data calculated for the direct effect charts can be seen in Appendix L. For the purpose of easily distinguishing effects on performance indicators, the clearance effects have been omitted. This is due to the fact that this, by far, has the strongest effect. This is not unexpected because shear stress itself is related to the velocity gradient; reducing the height will increase the gradient (eqn. 2.2). Shear stress at 3µm was on average 124.728kPa, whilst a value of 57.472kPa occurred at 7µm. To calculate the drag reduction percentage, the total drag (pressure drag plus viscous based drag) is divided by the shear force for a smooth surface without texture. The force for a non-textured surface is determined by first, calculating the shear stress, using eqn. 2.2, with a dynamic viscosity of 0.084075Pa.s (density of 855kgm^{-3}) and then multiplying it by the area it acts upon.

One of the first conclusions to be made from the study is that no trial, even for the large ranges used for the input parameters, resulted in a surface with increased drag. The lowest drag reduction was observed in trial 15 (0.28%). The maximum reduction seen in the study was for trial 5 at 14.87%. It can be seen that the scale of drag reductions observed in these trials is far greater than the ellipsoidal textures seen in the study by Syed and Sarangi (2014).

The parameter that appears to have the most consistent effect across all performance indictors was the pocket depth. It seems that increasing the depth of the pocket is beneficial in all aspects, with the greatest change seen between levels 2 and 3 (3µm to 5µm). This is advantageous in two respects. Firstly, this data shows that increasing the pocket depth will improve the frictional characteristics of the fluid-surface interaction at the journal bearing surface. Secondly, from a tribological view, increasing the depth will increase the surface reservoir size that retains lubricating fluid when starvation occurs. A property which was shown by Andersson et al. (2007) to significantly reduce wear and extend component life.

What about orientation of pockets? This is an important consideration because to manufacture bearing surface pockets that are orientated spanwise will be difficult using a cylindrical grinding machine tool. Looking at the direct effect charts, it can be seen that spanwise pockets increases viscous-based drag effects. Looking at Figure 4.19, where the performance indicator uses total drag (includes pressure-based drag), orientation has the greatest effect with streamwise orientation being the most effective. The difference in load support (or ratio to drag) between orientations, seems inconclusive as indicated by the small difference in performance indicator values.

Another parameter which makes the manufacturability of these surfaces more difficult is the stagger dimension. Looking at the drag reduction indicator for the clearance range studied, it is seen that having pockets aligned produces the worst performance, whereas the quarter pitch stagger configuration results in the best drag reduction. Unfortunately, this does mean that the optimal surface cannot be manufactured using this particular surface texturing grinding technique.

So are the parameters optimised? Using the drag reduction results, it can be seen that only parameters B (pocket length) and G (stagger proportion) indicate an optimal level has been achieved. Further increases to pocket width and depth, while decreasing pitch in both directions,

could potentially lead to further drag reductions. The next stage of the optimisation process is to confirm the best and worst controlling parameters through an additional study.

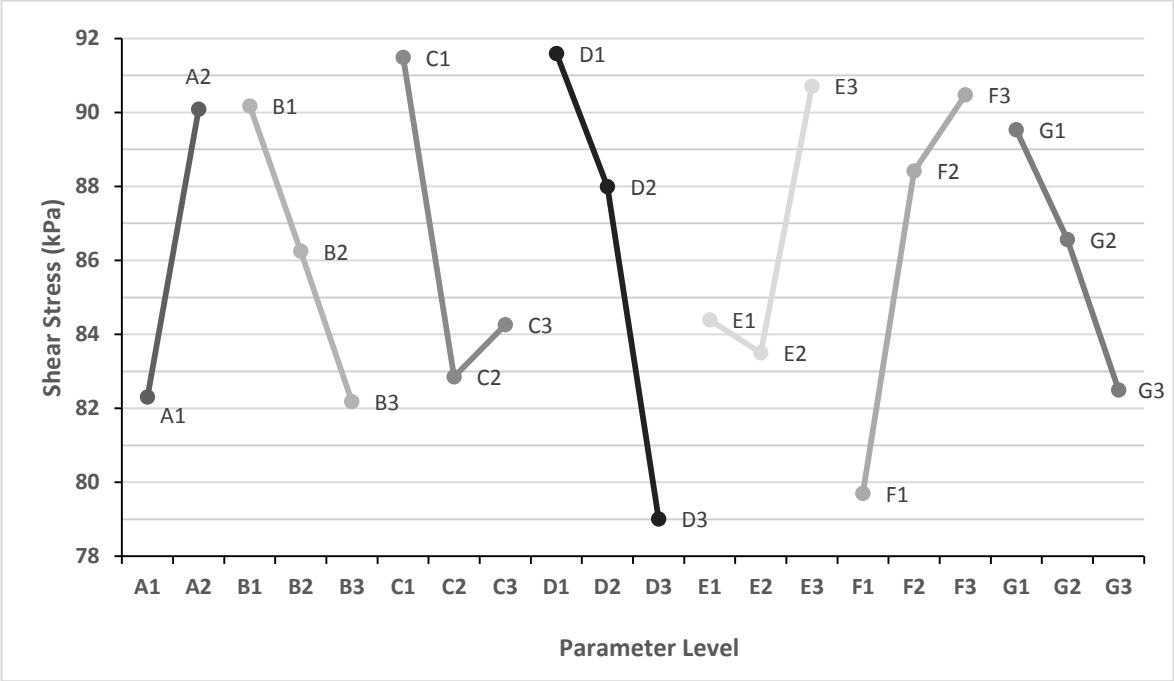


Figure 4.16. Direct effects of parameters on shear stress.

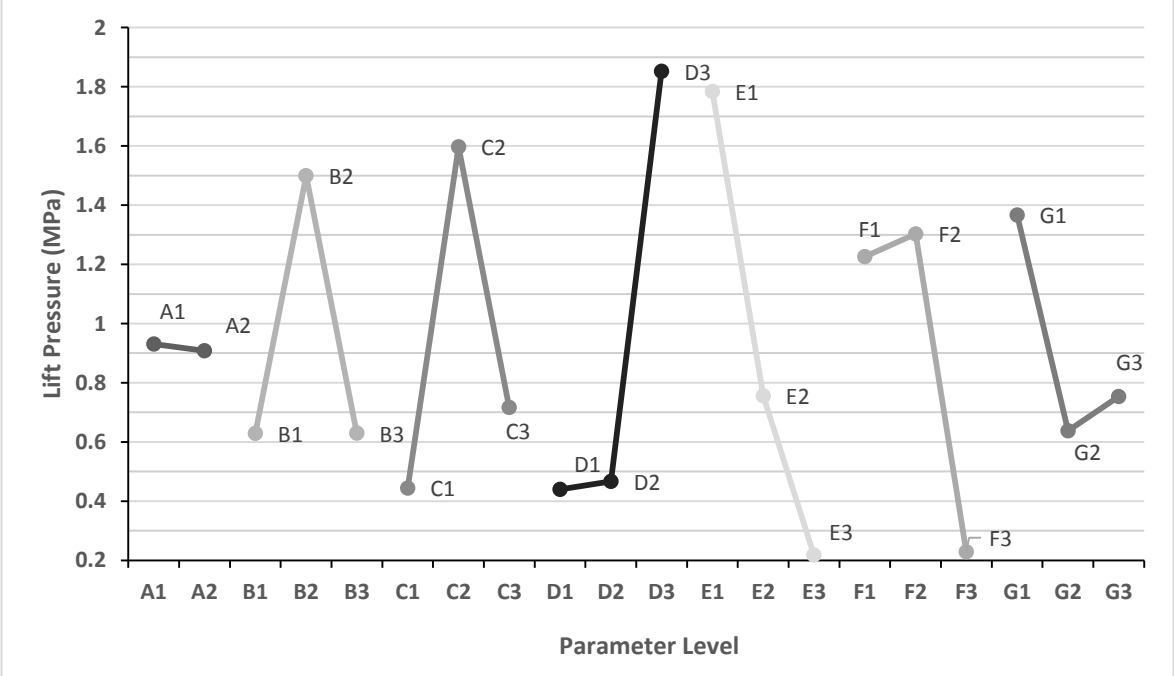


Figure 4.17. Direct effects of parameters on texture-induced lift.

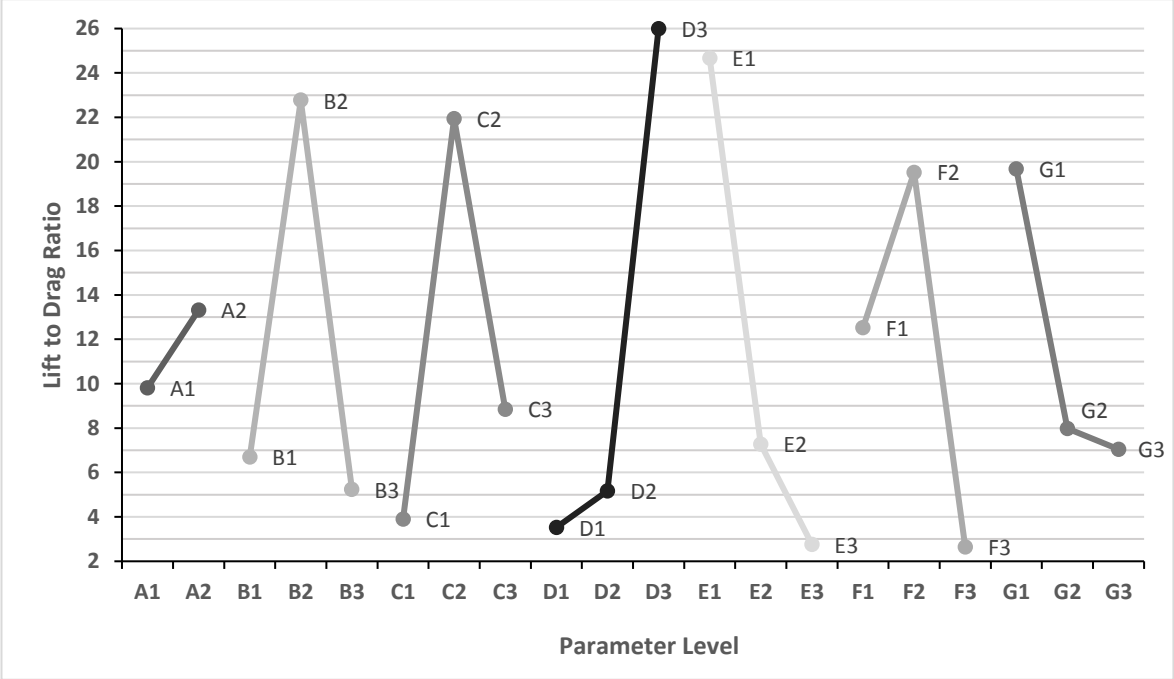


Figure 4.18. Direct effects of parameters on the lift to drag ratio.

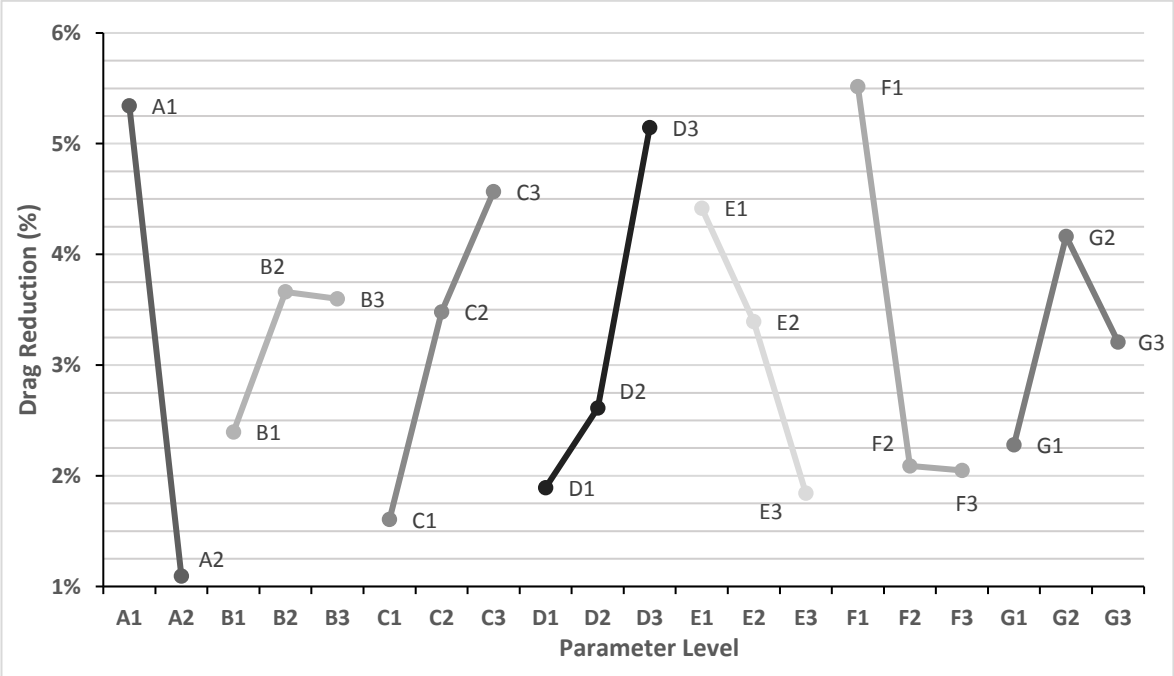


Figure 4.19. Direct effects of parameters on drag reduction.

4.4 Confirmation of Optimal Parameters

The idea behind this study is to confirm the effects of the lowest and highest performing level combinations. It was decided to judge parameter levels based on the drag reduction performance indicator (Figure 4.19). The parameter levels chosen for the confirmation trials are indicated in Table 4.7. As well as the clearance heights used in the previous study, additional trials were performed at clearances of 9µm and 11µm. The beneficial effects of these surfaces are expected to diminish further at increasing clearances but how considerably? Screenshots of the resulting worst and best performing surfaces can be seen in Figure 4.20 and 4.21. The better performing surface configuration is particularly interesting. From a qualitative viewpoint, there is an almost remarkable similarity between the inter-connecting staggered grooves present on shark skin with the geometry illustrated in Figure 4.20. Obviously, the functionality of shark skin is slightly different, but one of the key features identified by Luo et al. (2014) was the presence of recirculation within the grooves at the viscous sublayer region. This recirculation is a feature that is later identified to be present in the flow, when the surface is optimal for drag reduction.

Table 4.7. Best and worst parameters chosen for verification study.

Parameter		A	B	C	D	E	F	G
Level	**Best**	1	2	3	3	1	1	2
	Worst	2	1	1	1	3	3	1
Actual Value	**Best**	0°	250µm	75µm	5µm	300µm	50µm	¼ pitch
	Worst	90°	200µm	25µm	1µm	500µm	150µm	No Stagger

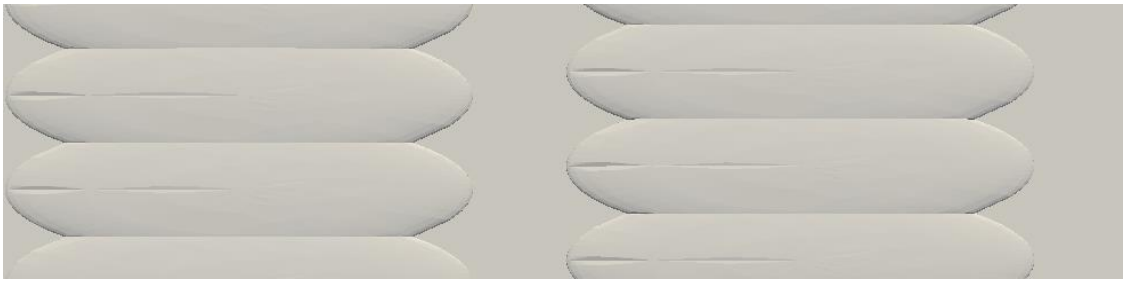


Figure 4.20. Best surface configuration.

The results of the confirmation trials are presented in Figure 4.22. The maximum drag reduction observed was 16.55% for the best surface with a k^+ of 0.605 and an s^+ of 6.050 (at 3µm clearance). The worst surface with a k^+ of 0.069 and an s^+ of 10.366 resulted in a drag reduction of 0.19% (at 11µm clearance). Very little change is seen in the worst surface performance across all the clearances (0.4%-0.2% reduction) but still no increase in drag is observed for the worst configuration. For future work, it would be interesting to investigate how large the clearance has to be for the effects of the best surface to become equal to the performance of the worst. Syed and Sarangi (2014) concluded that if a surface texture design has not been optimised by some method, then a "smooth" surface can be superior. Whereas, from the data presented here, even the worst surface still offers some benefit, even if it is relatively small.

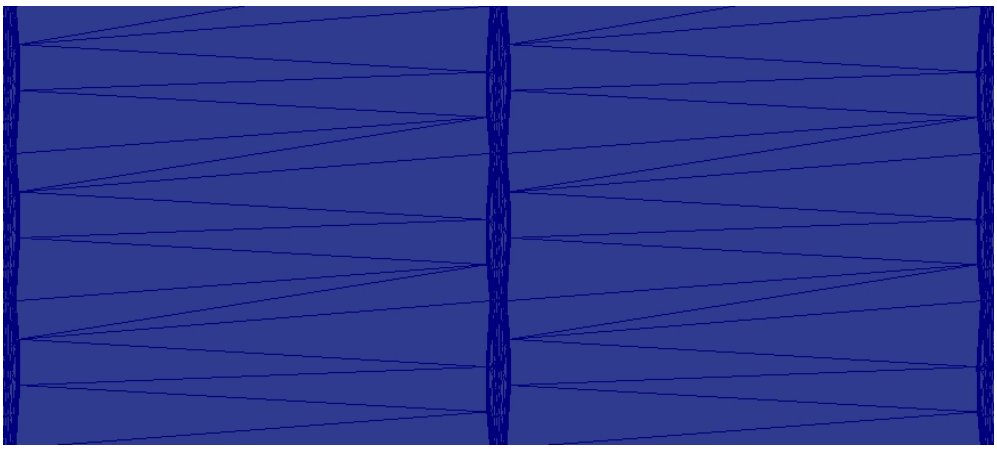


Figure 4.21. Worst surface configuration (triangulated surface has been displayed in order to identify the surface pattern with ease).

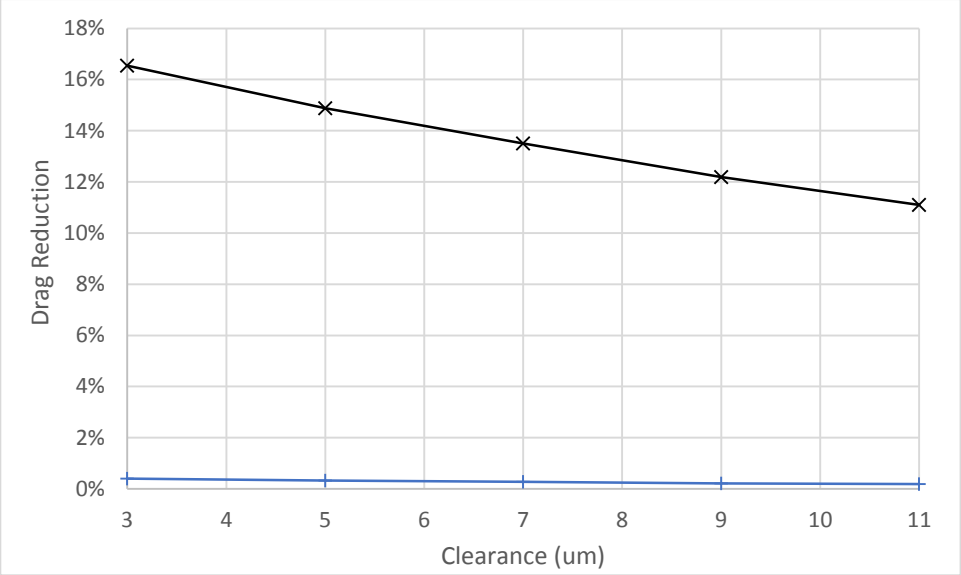


Figure 4.22. Drag reduction comparison (data used for this chart can be found in Appendix M). Best geometry – black X markers, Worst – blue + markers.

So, for the best and worst surfaces, what are the main differences in flow field characteristics observed? Well, looking at Figure 4.23, one of the main contrasts is the concentration of vorticity around the land areas and the related change in shear stress in this area (Figure 4.24). The vorticity in the lower performing field has a smaller limit and can be seen to be reasonably constant for much of the film. It seems from looking at these plots, that minimising land areas to provide the lift/contact pressure needed is key in achieving a higher performing surface. Another feature which is non-existent in the poorer performing surface is the presence of recirculation. Looking at Figure 4.25 (contour indicating negative streamwise velocity), a small and thin recirculation area can be identified in the bottom of the pocket geometry for the best performing surface. As mentioned earlier, this has a similarity with the “back flow” regions present in shark skin flow interaction.

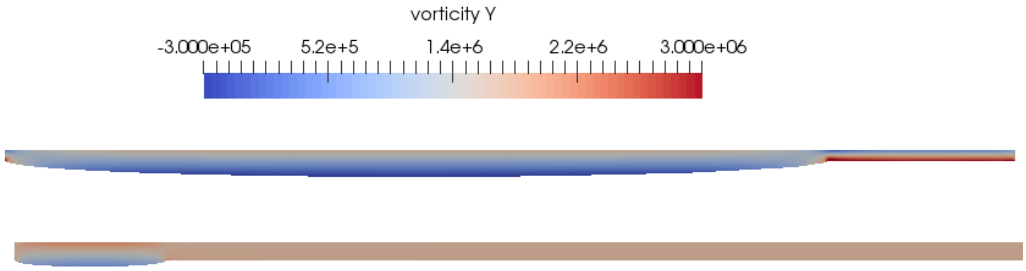


Figure 4.23. Vorticity (in 1/s) contour plot – cross-sectional view. Geometry - best (top) and worst (bottom).

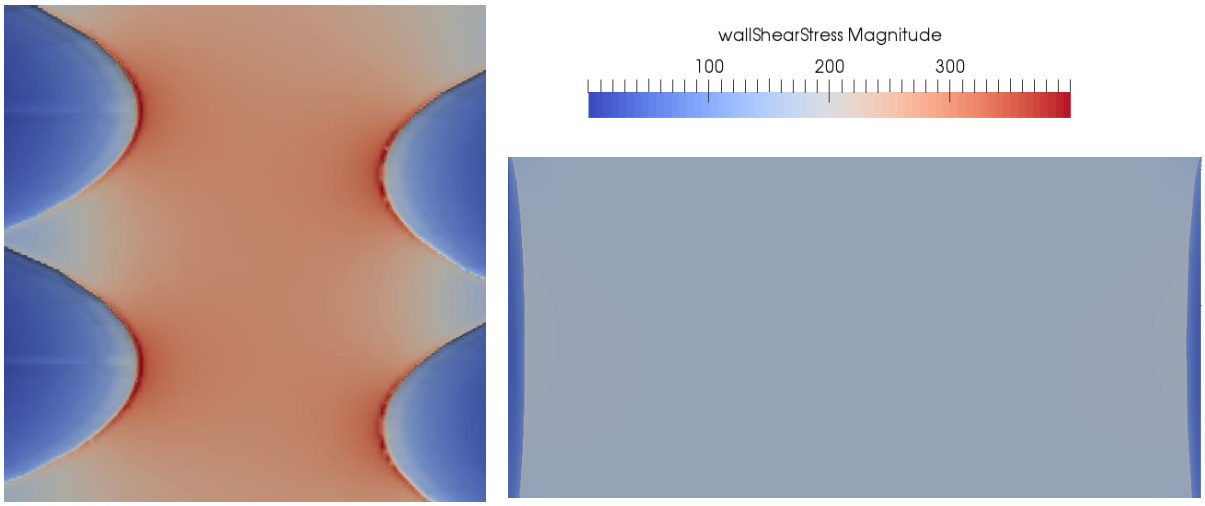


Figure 4.24. Wall shear stress on land between pockets – kinematic pressure units used (m^2s^{-2}). Geometry - best (left) and worst (right).

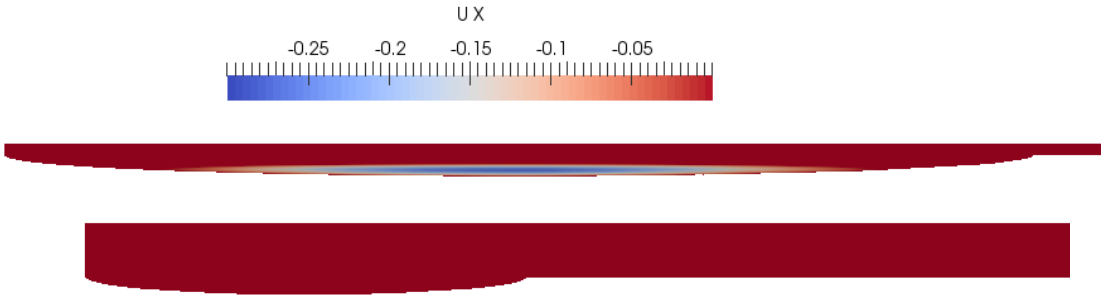


Figure 4.25. Negative streamwise velocity (in ms^{-1}) highlighting recirculation zones. Geometry - best (top) and worst (bottom).

Now that the numerical modelling has been completed, the final step (and the most important one) is to perform a systematic validation of the modelling process.

4.5 Validation using Infinite Rayleigh Step Bearing Theory

One of the problems with this particular modelling process, is that there is a lack of published data surrounding ellipsoidal based micro-textured bearing surfaces. Plus, the experimental setup used in this research is of a hydrodynamic setup and differences between the modelling assumptions and reality will not allow for comparison. So it was decided that in order to validate the modelling process and setup, the well-known infinite Rayleigh step bearing theory should be used in conjunction with rectangular pockets instead of ellipsoidal ones. This is easily implemented within the python code by replacing line 74 in the bearingStagger2.py script with the "MakeBoxTwoPnt" command. The script was then configured so that only one pocket appeared in the transverse direction and the width was equal to its spacing. The following derived equations presented are taken from Stachowiak and Batchelor (2006) (p. 129-131) whose works is based on the work of Rayleigh (1918).

The step dimensions used in the analysis were based on the following ratios:

$$\frac{h_1}{h_0} = 1.87 \qquad \frac{B_1}{B_2} = 2.588 \tag{4.24}$$

Where h_1 is the clearance height plus pocket depth; h_0 is the radial clearance; B_1 is the pocket length and B_2 is the land length. These are the optimum ratios for the Rayleigh step bearing, recommended by Cameron (1981). The actual dimensions used were: h_1=13.09µm, h_0=7µm, B_1=200µm and B_2=77.28µm. Looking at the linear gauge pressure profile produced by the CFD simulation (Figure 4.26), the maximum pressure was found to be 308.97kPa + 632.67kPa = 941.64kPa. Contours of streamwise velocity were produced during post-processing with an

additional contour identifying areas of recirculation (Figure 4.27 and 4.28). Recirculation is a feature which is similarly found in the ellipsoidal features. The maximum pressure can be predicted using the following equation, given by Stachowiak and Batchelor:

$$P_{max} = \frac{6U\mu(h_1 - h_0)}{\left(\frac{h_1^3}{B_1} + \frac{h_0^3}{B_2}\right)} \tag{4.25}$$

The velocity and dynamic viscosity were kept the same as in the previous CFD study and eqn. 4.25 results in a predicted maximum pressure of 981.30kPa, which yields a 4.04% difference in the results. This result means that, with some confidence, the modelling process used in this study is correct. Although, further work should be undertaken in order to make direct comparisons for the ellipsoidal dimples, whether that is through an experimental approach or using data that is generated by another CFD solver.

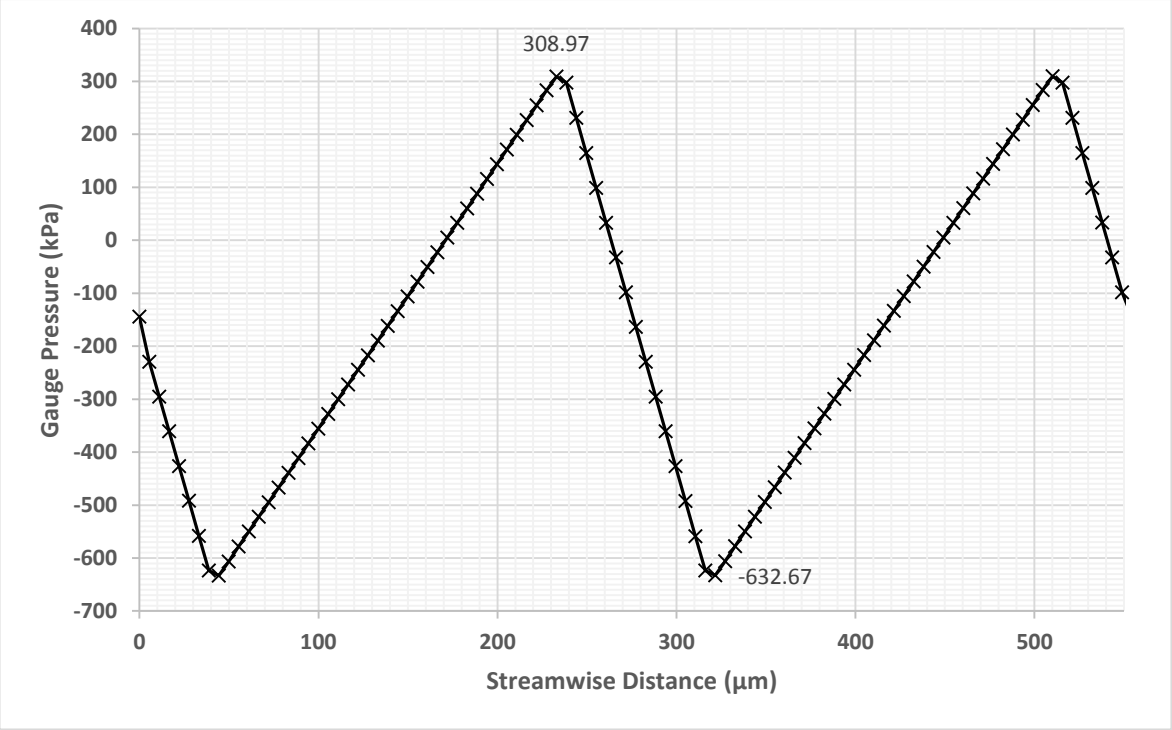


Figure 4.26. Gauge pressure profile along spanwise direction (maximum and minimum values indicated).

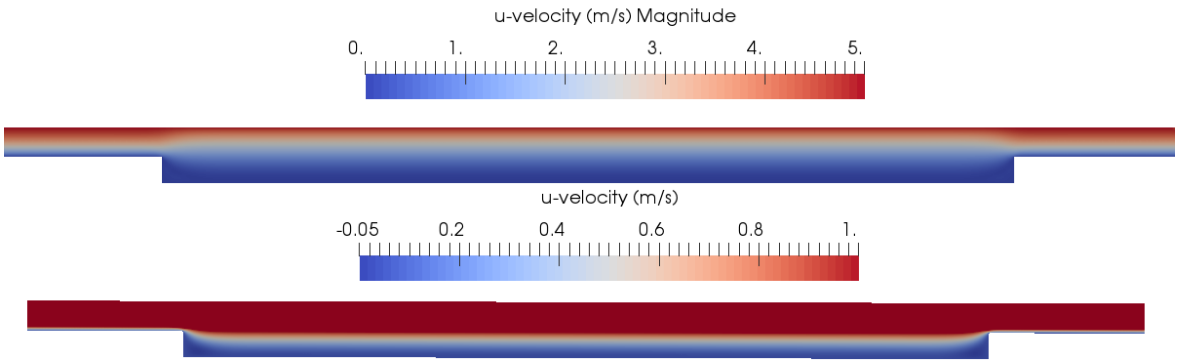


Figure 4.27. U-velocity for whole range and identification of low velocity areas.

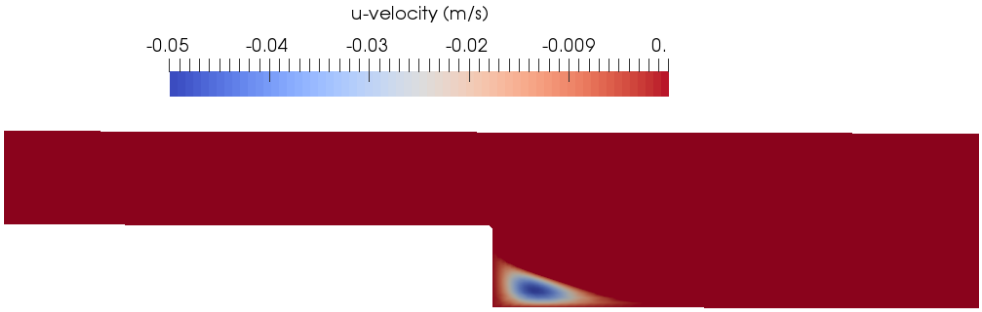


Figure 4.28. Identification of recirculating areas following a Rayleigh step.

4.6 Conclusive Remarks on the Numerical Investigation

The following conclusions were made from this investigation:

- It has been demonstrated using an array of open-source software tools, that full three-dimensional Navier-Stokes based approximations can be made for lubricating flow interactions with the proposed ground, surface textures. This approach differs from others mentioned, in the respect that these predictions give more accurate pressure fields than using the simplified Reynolds equation approach. The open-source approach

used in this study will allow for further investigation and research to be carried out without the implications of commercial licensing.

- The successful creation of the python interface script allows for the automation of: generating the CAD models required for the meshing procedure; creating a hexahedral-based mesh around the complex ellipsoidal dimples; preparing the working directory and dictionary files required to run the simulation and controlling the solving process. This allows a user with little knowledge or background in CFD to optimise surface configurations for a particular set of bearing conditions. Although, further improvements may be made to the script in order to automate the post-processing of results with the Taguchi-based optimisation technique.
- The script has been designed so that future users can, with a few simple edits, change the geometry of the texture being investigated (which was demonstrated in the validation exercise). The script is currently designed for internal flow regime but again, with a few simple changes to boundary conditions within the source files, an external flow study could be carried out (i.e. boundary layer growth over plates could be investigated). So, by design, what has been created is a script that allows the user, with a few simple edits, to study a whole range of flow scenarios without the time delay associated with learning open-source tools. This has overcome a commonly quoted disadvantage when comparing commercial against open-source CFD tools.
- Further validation needs to be carried out which will allow for direct comparison of simulated ellipsoidal dimples to reality. This could be undertaken using a pin-on-disk tribometer where additional hydrodynamic forces/interactions can be minimised. Some validation of the modelling process has been performed against infinite Rayleigh bearing theory.
- As discussed earlier, a truly optimal surface configuration has not been achieved according to the direct effect charts presented. Further investigation needs to study

textures with smaller spacing whilst increasing the pocket width and depth, in order to identify further possibility of drag reduction. It is key, from the analysis of the results of the numerical investigation, that minimising land area (which provides the contact pressure) shall reduce surface drag losses.

The results from the numerical modelling process have shown that the proposed ellipsoidal textures are useful in reducing surface drag. The study has also highlighted what configurations lead to a poorer/worse performance in drag reduction. The next section shall evaluate the manufacturability of the higher performing surface configurations and also further validate the magnitude of the surface drag reductions seen in the numerical modelling.

5. Experimental Investigation into the Texturing of Surfaces by the Cylindrical Grinding Method

5.1 Introduction

Previous to this investigation, a computational fluid dynamics study was used to optimize the size, spacing and orientation of grinding patterns to achieve the best possible texture-induced lift and surface drag reduction capabilities. The first row of *Table 5.1* shows details of the optimised features for a linear speed of $5ms^{-1}$. Unfortunately, due to the limitations of the machine being used to for this investigation and the dressing point radius, not all of the parameters could be achieved. Although, as this investigation is only to prove the effectiveness of these textures, an alternative set of parameters were selected (as close to the recommended values as possible). The previous simulation work concluded that minimising land area and maximising the texture area was key in allowing these textures to be useful in comparison to a nominally, smooth surface. This key point was kept in mind when selecting the achievable parameters that would be used for this grinding study.

Table 5.1. Optimised and achievable parameter details.

	Angle (deg.)	**Length (µm)**	**Width (µm)**	**Depth (µm)**	**X Pitch (µm)**	**Y Pitch (µm)**	**Stagger**
Simulation	0	250	75	5	300	50	¼ Pitch
Achievable	0	350	85	2	350	200	0

5. Experimental Investigation into the Texturing of Surfaces by the Cylindrical Grinding Method

The main purpose of this experimental investigation was to prove the effectiveness of ground surface textures in reducing the frictional forces between surfaces in a hydrodynamic bearing application. The optimised parameters seen in the simulation showed good surface drag reduction characteristics across a range of clearances. The following experimental investigation may not allow for direct comparison to the simulation results but will allow, at least, some conclusions to be made about the magnitude of the drag reductions observed in the numerical investigation.

5.2 Hydraulic Configuration for Tribometer Apparatus

Before any experimentation could take place, the first task was to resolve some problems relating to the hydraulics of the tribometer apparatus. The pump catastrophically failed soon after the initial bearing investigation was carried out. Engineering analysis of the system led to the conclusion that the pump failed early because of the poorly designed hydraulics (details explained in section 3.4). As a reference, Figure 5.1 shows a hydraulic diagram of the newly designed system. The variable displacement pump that was fitted had a maximum flow rate of 1ℓmin^{-1} and was rated at a maximum operating pressure of 170bar. A pressure relief valve was included within the hydraulic power pack and this could be adjusted to a set pressure. The relief valve was adjusted on the pump manually until the pressure was just enough to overcome the head losses of the system, providing oil to be fed into the cavity without contributing additional hydrostatic pressure to the bearing. After the pump, the fluid encounters a non-return valve (Parker 2301 check valve), this ensures that no back flow could enter the pump which would substantially increase the loading on the component. After the non-return valve, several pipe junctions were added. One of the junctions would lead to the actual bearing cavity and other junctions would lead to the accumulator or to another relief valve. The accumulator (Parker ELM series, diaphragm based with a 0.16 litre capacity) is required for two reasons: the first is to act as a

reservoir in case of a power outage to the system. As hydraulic fluid enters this particular type of accumulator, it is pressurised by the opposing diaphragm (which is pressurised by a volume of gas) and in the case of a power outage, the pressurised hydraulic fluid will continue to feed the bearing until the journal stops rotating. The second reason for the accumulator is to accommodate for any sudden back pressures: acting in both a relief and damping capacity. The additional relief valve (with a fitted gauge pressure indicator) provides the whole system with relief in case of complete blockage. Referring to Figure 5.1, it can be seen that the reservoirs connected to the pump and the bearing are kept separate. This ensures that hydraulic fluid being fed into the bearing cavity is at near-room temperature and will not be affected by any heat generation at the bearing surfaces.

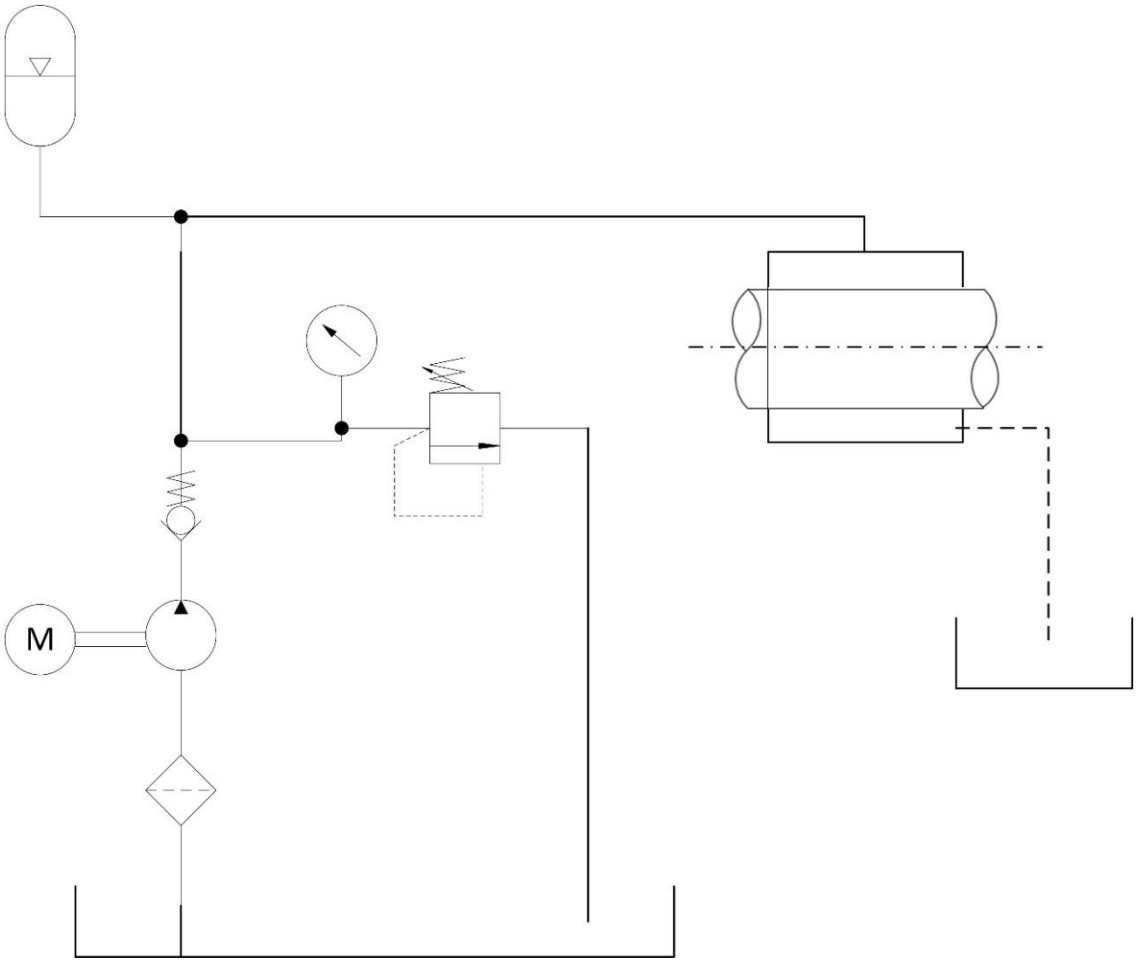


Figure 5.1. Hydraulic diagram of the tribometer apparatus.

It was found that the original hydraulic system could not run for more than five minutes without overheating. The new hydraulic system design was tested beyond this (to 15 minutes in a single run) and no overheating was observed. This redesign of the equipment allowed for the tribometer experimentation, detailed later in this section, to take place.

5.3 Experimental Method for Grinding Study

This section will now give the reader detail on the process used to grind the samples used in this study. Unlike the samples manufactured in section 3, these were made onsite using the University's Jones & Shipman Ultramat CNC cylindrical grinding machine. A 77A601J8V alumina (aluminium oxide) wheel with vitrified bond was used for this study (wheel has a width of 50mm and diameter of 0.385m). The grain size for this wheel is determined by the number 601 (F60 medium size macrogrit with an additional digit of 1 indicating some further information about the grit mixture by the manufacturer), which according to BS ISO 525:2013 (British Standards Institute (2013)) and BS ISO 8486-1:1996 (British Standards Institute (1996)) requires 100% of the grit to fall below a mesh aperture size of 425µm.

Before the study is carried out, each of the mild steel components were turned and then ground to a diameter of 53.01mm diameter. Also, prior to the study, the cylindrical grinding machine is switched on and the wheel is left to run for a minimum of 2 hours in order to reach steady-state conditions. All grinding operations were run with a sufficient flood delivery flow rate of cutting fluid.

The following describes the method used to manufacture the components:

i. Dress the wheel according to the polish grinding dressing parameters (*Table 5.2*).

ii. Following the plunge grind parameters in *Table 5.2,* polish the component. If it is a non-textured component, it is then finished to a size of 53mm and does not following the remaining steps.

iii. Dress the wheel using the parameters stated in *Table 5.3*. Again, using the putty, study the wheel surface.

iv. The surface of the bearing will now be textured using the plunge grinding cycle described in *Table 5.3*. Rapid feed the grinding wheel to a safe distance of 53.1mm and then fine feed (0.1mm/min) the grinding wheel to 53.05mm, initially. Then perform the specified cut (a_p) at a feed of 0.01mm/min. Monitor the acoustic emission (AE) output to monitor the cutting process. Repeat the process and move in radially by 0.01mm increments to a target diameter of 53mm, where actual cutting should be observed on the AE output.

v. Measure and record the three-dimensional surface roughness for the component using the Bruker white light interferometer. Then using the Talyrond profilometer, measure the roundness of the component.

Table 5.2. Polish grinding parameters.

Dress Twice	f_d=1mm.min^{-1}
	v_s=35ms^{-1}
	n_s=1665rpm
	a_d=1µm
Plunge Grind	f_1=0.1 mm.min^{-1}
	f_2=0.01 mm.min^{-1}
	Dwell for 10s with 100µm oscillation
	v_f=0.02 mm.min^{-1}
	n_w=125rpm
	10µm diameter removal

Table 5.3. Textured grinding parameters.

Comp. No.	Radial a_d (µm)	f_d (µm.rev^{-1})	v_s (ms^{-1})	n_s (rpm)	n_w (rpm)	Radial a_p (µm)	v_f (mm.min^{-1})	f_d (mm.min^{-1})
1	20	200	35	1736	4	2.5	0.01	347
2,4,5	10	170	38.3	1900	4	2.5	0.01	323

Note: Comp. - Component

5.4 Dressing Tool Measurement and Textured Grinding Parameters

As discussed in section 4.1, the dressing tool shape and size strongly influences the resulting dressed grinding wheel surface and more so in this case because of the overlap ratio, chosen to be less than one, resulting in a textured wheel (the theory of which is presented in section 4.1). In order to determine the achievable sizing of the surface patterns, it was decided to investigate the dressing tool edge shape. To do this, the same adhesive *putty* material used in the grinding trials was pushed past the dressing edge, leaving a *scratch-mark* behind. This valley was then examined using the GFM MikroCAD Premium 1 (Depth resolution is 0.1µm, other axes have a resolution 1.2µm) device (the surface can be seen in Figure 5.2). The device uses the structured light fringe projection profilometry method for performing surface measurements. The MikroCAD software is then used to process the data generated by the device.

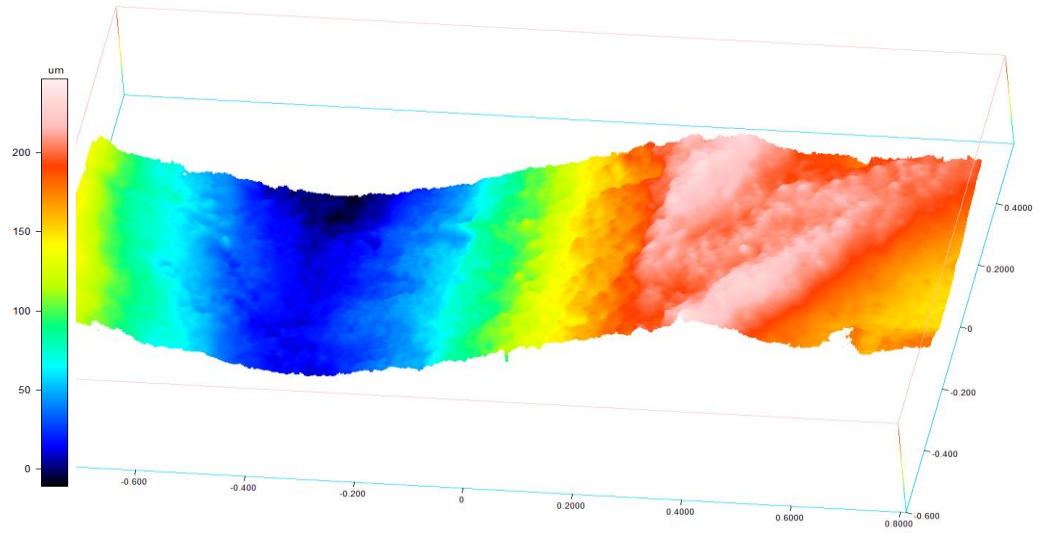


Figure 5.2. Surface plot of dressing tool indentation. Domain lengths are in millimetres.

The next step was to post-process the data in ParaView in order to ascertain the actual engagement width for the dressing tool (Figure 5.3 shows a visual demonstration of the data manipulation). The first step in post-processing was to remove the boundary noise at the very edge of the imported surface data by *cropping* the surface (any surface highlighted purple is kept). The second step in post-processing was to apply some smoothing in order to remove some of the noise from the data of interest. Then the final step was to export the co-ordinates of a line which resulted from intersecting an x-y plane with that surface (see Figure 5.4 for the profile of the valley). The profile displayed is for a 10µm radial dressing depth of cut (a_d) which is used to manufacture components 2, 4 and 5. From this profile, the dimensions of the texture can be calculated (based on a homogenous wheel surface assumption). The dressing tool contact width/engagement width (b_d) will be 154µm at this depth (10µm). This means that an overlap ratio (U_d) of 0.9 is observed for the dressing operation (calculated using eqn. 5.1).

$$U_d = \frac{b_d}{f_d} \tag{5.1}$$

The dressing feed of 170µm.rev^{-1} resulted in a y-pitch of 170µm (explanation of chosen feed is explained later). According to the kinematics of the texturing process described by Stepien

(2007b), in order to obtain the desired longitudinal pitch, L, for the textures, one revolution of the grinding wheel must equal the work material distance travelled of L. This means a surface speed ratio of workpiece to wheel, v^*, must be calculated in order to make sure the surfaces are synchronised (see eqn. 5.2).

$$v^* = \frac{v_w}{v_s} = \frac{L}{2\pi R_s} \tag{5.2}$$

Where, R_s, is the grinding wheel radius. For a longitudinal pitch of 350µm and a wheel radius of 192.5mm, the speed ratio was calculated as 0.289×10^{-3}. This means for the maximum wheel speed, dictated by the machine control panel, of 1900RPM or $38.3 ms^{-1}$, the workpiece surface speed must be set at $11.069 mm.s^{-1}$ (or ≈4RPM). The next step is to calculate the feed rate (v_f) required in order to obtain the correct depth of cut/in-feed per revolution (a_p). The time required for one revolution of the workpiece is given by eqn. 5.3. Multiply this by the feed rate, v_f, in order to calculate the depth of cut (eqn. 5.4).

$$\text{Time per workpiece revolution} = \frac{\pi D_w}{v_w} \tag{5.3}$$

$$a_p = \frac{\pi D_w v_f}{v_w} \rightarrow v_f = \frac{a_p v_w}{\pi D_w} \tag{5.4}$$

Where, D_w, is the workpiece diameter. According to the Stepien, the depth of cut will result in the pocket depth required. So, for a pocket depth of 2.5µm, the required feed rate will be $9.97 \mu m.min^{-1}$ ($\approx 0.01 mm.min^{-1}$). Even though this was not used for the grinding trials, an alternative, and possibly, a more useful parameter for predicting the depth of cut would be the parameter, a_e, real depth of cut (eqn.5.5). Rowe (2009) p.18-19, discusses that the real depth of cut will differ from theoretical a_p due to several factors:

- Deflection in the system when grinding forces are applied (X);
- Wheel wear (a_s) will decrease the depth of cut;

- And thermal expansion of workpiece (as well as machine components), X_{exp}, will usually cause an increase in the depth of cut.

$$a_e = a_p - X - a_s + X_{\exp} \tag{5.5}$$

The grinding technique used in this investigation does not incorporate any dwell time, which is needed to reduce errors relating to roundness and sizing. The dwell phase is required to overcome some of the issues relating to the differences between theoretical and real depth of cut. To quote Rowe: "Typically a_e is approximately a quarter of a_p depending on the workpiece hardness". So, for future trials, it may be of use to predict the real depth of cut, in order to make sure the pocket features are near to the designed sizes.

Based on the calculated value for a_p, the pocket and land width can now be predicted based on the dressing tool profile in Figure 5.4. At a radial depth of cut of 2.5µm (or 7.5µm profile depth), the workpiece would make contact with the profile between -0.425mm and -0.297mm (x-axis). This means that per 170µm y-pitch, 128µm would be the land width, resulting in a pocket width of 42µm. What is apparent from these calculations is how reliant the texture dimensions are on the dressing tool radius. In fact, a dressing tool edge that is more *worn,* which will result in a larger radius, which will result in pocket width that is more sensitive to depth of cut, giving rise to a much larger range of pocket dimensions for a restricted range of cut depths.

Now that the calculations for the textured grinding have been provided, the next section will discuss the grinding trials performed and give some explanation of the approach taken.

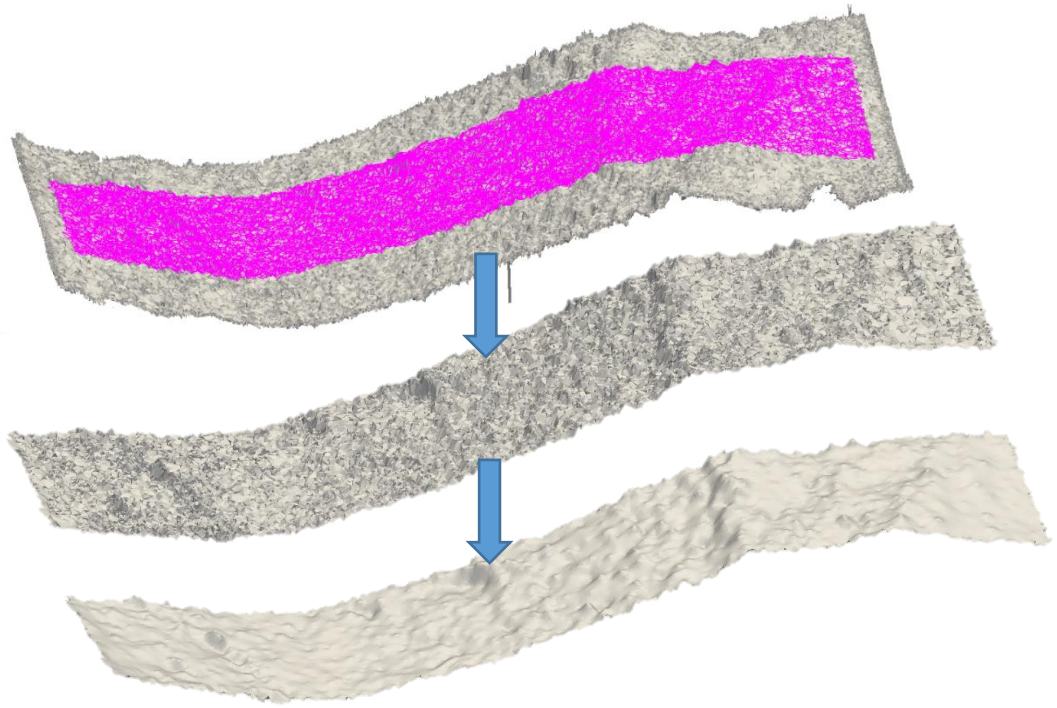


Figure 5.3. Visual demonstration of steps in post-processing surface data (cropping data then smoothing out noise from selection).

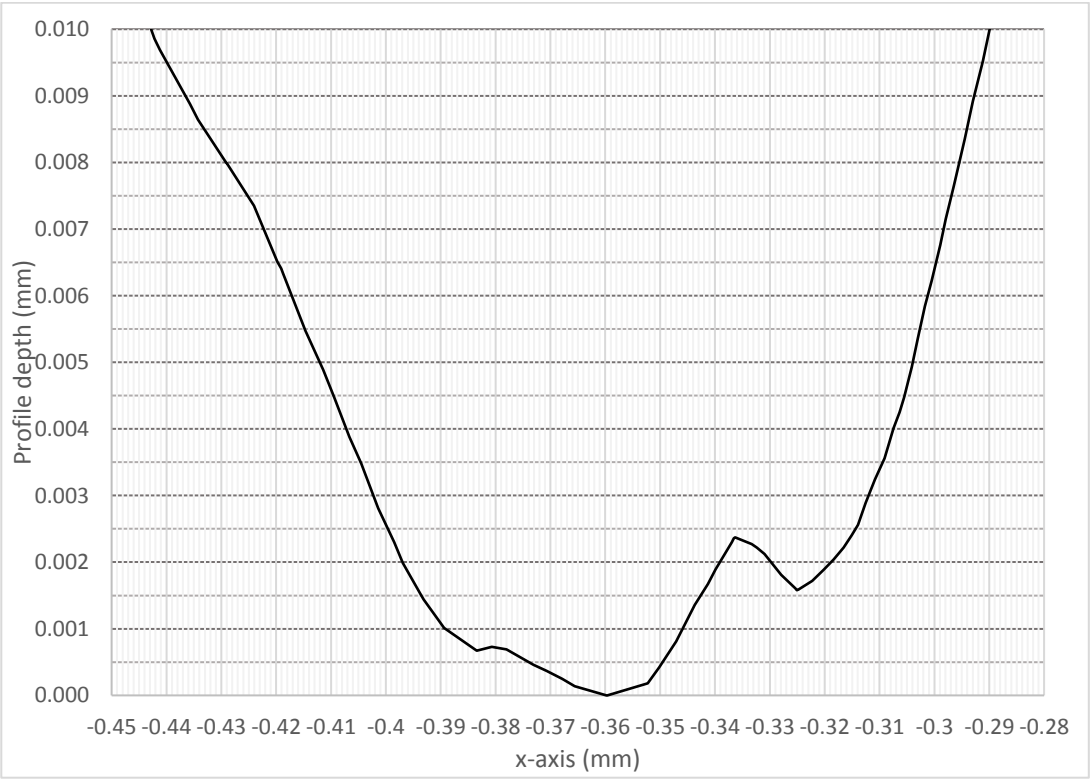


Figure 5.4. Cross section profile for dressing tool measurement.

5.5 Discussion of Grinding Trials

Using the parameters seen in Table 5.3, component 1 was ground for one revolution and was found to have a poor surface texture. Analysing this surface, qualitatively, using the 3D surface measurement presented in Figure 5.8, it can be seen that there is almost no consistent texturing observed. It was decided that the reason for this could be down to the aggressive dressing parameters. If the wheel is dressed aggressively enough, the dressing forces applied to the interacting grits will be so great, that instead of the desired fracturing process, grains will be pulled out of the bonding material. Pulling grains out of the wheel surface results in a reduction of *active grains* which causes two main problems for the surface texturing process. The first negative effect of reducing active grains, according to Stepien (2007a), is the increase of cutting forces during the grinding process. Decreasing the number of active grains (i.e. the grits that make physical contact with the workpiece during cutting) will increase the size of the undeformed chip thickness. This means that each active grit has to *work harder* to form each chip and therefore will increase the cutting force applied to each grain. This will substantially increase wheel wear to a point where a component's surface may not even be fully textured. The second effect is that there may not be enough active grains to form well-defined pockets across the surface itself, making the surface appear more stochastic.

To grind the second component, the wheel was dressed with a lower feed rate (reduced from 200µm.rev^{-1} to 170µm.rev^{-1}) and the dressing depth of cut was reduced from 20µm to 10µm. The calculation of the feature sizes discussed in the previous section were based on this wheel. Looking at Figure 5.9, it is without doubt that the texturing has improved on component 1. Unfortunately, due to operator error, the grinding that took place before the texturing process has resulted in the component being out-of-roundness. In reference to Table 5.4, it can be seen that component 2 has a roundness of 96.333µm and component 3, the non-textured, comparative sample had a similar roundness of 99.050µm. Looking at the profiles for these

components in Figure 5.5, it was concluded that the grinding process used to get the components to size, did not complete a full grind cycle. Due to the similar roundness error for these two components, it was decided to test them on the tribometer in order to ascertain the frictional performance for these surfaces. But, as expected, violent vibrations were experienced during testing, resulting in early termination of these trials. Due to the limited number of components available, these were re-ground later for components 5 and 6.

Using the same textured wheel, component 4 was ground. In this grinding trial it was decided to programme the machine so that the stock removed, would be equal to two revolutions worth of surface texturing. There were two reasons for this. The first being to improve the consistency of the pockets around the journal. Because dwell was not being used, the wheel was hardly engaging at the initial contact with the workpiece, resulting in very little texturing. The second reason was to see the effect on the roundness profile. Again, because of the no dwell time, a *camshaft-like* profile is seen on the textured components (an example of this can be seen in Figure 5.7). Removal or minimising this step on the profile will remove the directionality of the bearing's performance. Because the depth of cut is increasing at 2.5µm per revolution, the radial depth of cut will be 5µm. This means that the pocket sizing will change. Referencing to the cutting edge profile in Figure 5.4, at this depth of cut (or 5µm profile depth), the workpiece will engage between -0.412mm and -0.304mm. This allows for the prediction of the land width size at 108µm and a wider pocket of 62µm. Figure 5.10 and 5.11 show the surface resulting from this grinding trial. It is interesting to see that the consistency is excellent and the spacing/sizing of the textures adhere near to the designed parameters. So, could the phasing between revolutions be small enough to result in a surface pattern that is acceptable? Maybe the grinding machine control could be improved to include a closed loop control system. This control would improve the surface texturing process by using appropriate dwell times to achieve better roundness and sizing without causing large deviations in the designed pattern configuration.

Component 5 was created again, using the same wheel but this time, the machine was programmed to remove stock that was equal to one revolution. Analysing the 3D surface measurement for this component (Figure 5.12 and 5.13), qualitatively, it is clear that this trial has produced a poorer quality surface with little consistency in comparison to component 4.

Finally, a set of non-textured samples were produced using only the polishing parameters stated in Table 5.2. These samples were manufactured so that comparisons could be made after the tribometer experimentation had took place. Component 7 was compared with component 4 and component 6 against component 5. The comparative samples were produced so that the diameters were similar. Having similar diameters made sure that the eccentricity due to load would create the same hydrodynamic effect and any other effects would be down to the surface texture.

After all the components had been tested, it was decided to create another sample (component 8) by exposing component 7 to further polishing, using a vibratory finishing technique, for 15 minutes. The vibratory finishing was performed using a OTEC EF18 with a mixture of different sized granite-based media. Component 4 and its comparative, non-textured sample (component 7) had similar average roughness, Sa, values (0.237µm and 0.241µm, respectively). By performing this further finishing process, the Sa value was lowered to 0.187µm for component 8. This meant the performance of component 4 could be compared to a similar roughness value but also to one which was much lower. Usually, the side effect of performing this additional process is that the roundness will worsen but studying Table 5.4 shows that this is not the case.

5.6 Inspection of Samples

This section presents the measurements that took place in order to judge the overall quality of the manufactured components.

5.5.1 Roundness

Just as described in section 3.3.1, the Taylor Hobson Talyrond instrument was used to generate two dimensional profiles of each component surface. For each component, three profiles and three roundness values were generated. Some of the profiles are included within this section and a complete collection is presented in Appendix N. All of the roundness values are presented are in Table 5.4 as well as a calculated average value for each component.

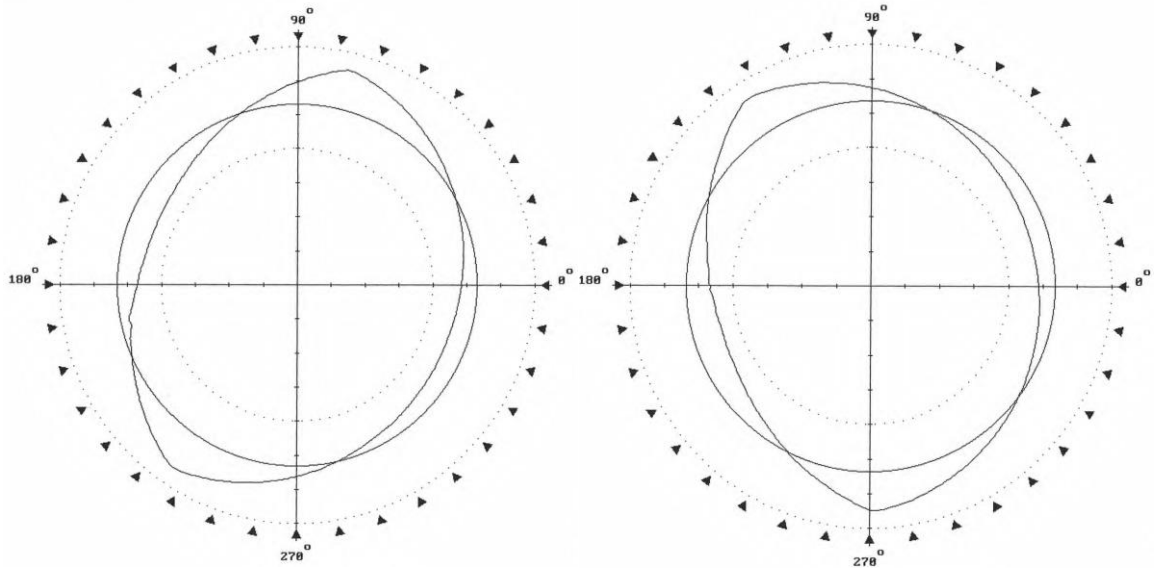


Figure 5.5. 2D profile for component 2 (left) and component 3 (right). Divisions are equal to 50µm.

Referring back to the conclusions made in section 3, one of the main reasons in deciding to manufacture the components on-site was to control the quality of the samples being used for this research. Large peak defects were observed on the components in section 3, resulting in roundness values of up to 15.2µm. Apart from components 2 and 3, which were a product of operator error, all samples were below this value.

5. Experimental Investigation into the Texturing of Surfaces by the Cylindrical Grinding Method

As mentioned already, all the components which were textured, using approximately one revolution of grinding, exhibit the same *camshaft-like* profile. A result of the exclusion of dwelling. Consequently, all of the textured components have a higher roundness than their non-textured counterparts. The highest roundness measured was exhibited by component 4, which was the only sample to be ground for approximately two revolutions. Even though the profile does show the removal of the step seen on other samples, it has now unfortunately been replaced with another large, *plateau-like* defect.

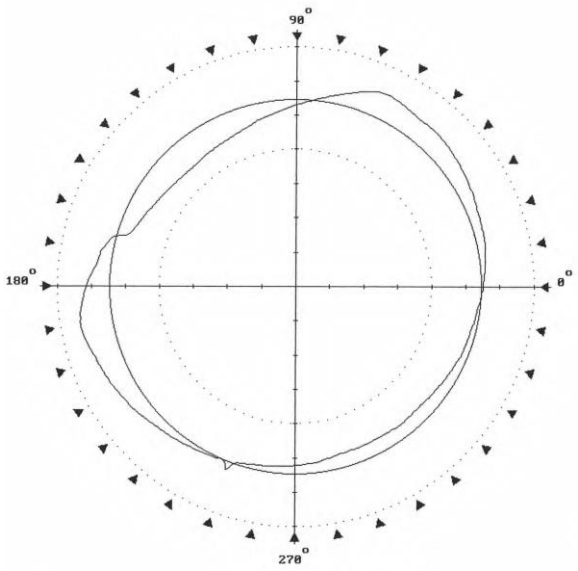


Figure 5.6. 2D profile for component 4. Divisions are equal to 5µm.

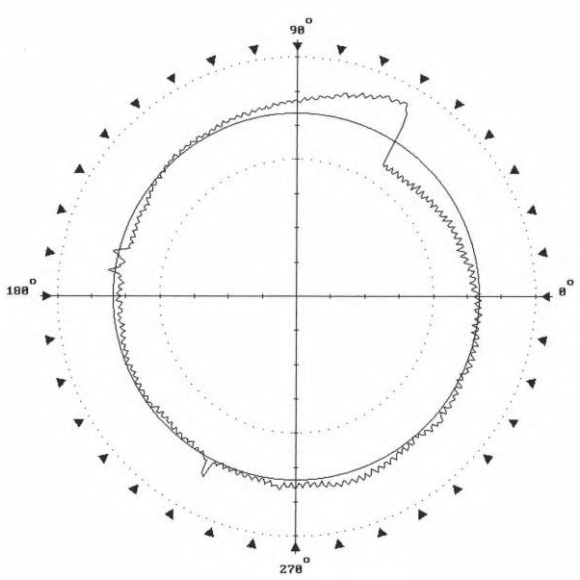


Figure 5.7. 2D profile for component 5. Divisions are equal to 1µm.

Table 5.4. Roundness measurement results.

Component No.	Roundness (µm)			
	Test 1	Test 2	Test 3	Average
1	3.400	2.200	2.250	2.617
2	97.150	93.200	98.650	96.333
3	99.350	101.650	96.150	99.050
4	9.450	8.800	9.050	9.100
5	1.950	1.900	3.300	2.383
6	0.650	5.150	2.000	2.600
7	6.700	2.800	1.950	3.817
8	1.350	0.550	2.250	1.383

5.5.2 3D Surface Measurement

To perform three-dimensional surface measurements, the Bruker Contour GT-K white light interferometer was used, the same equipment used to analyse the surfaces of components in chapter 3. Each component was measured three times. For every measurement that took place, a set of statistical roughness parameters were generated. All of this data can be found in the appendices: Appendix O for height-based, Appendix P for functional and Appendix Q for spatial

and hybrid roughness parameters. All images of the textured components have been included in this thesis, some within this section and the complete collection can be found in Appendix R.

In reference to the roughness data, one observation was the large deviation in some of the roughness parameters, particularly ones which were spatial and hybrid-based (Sdq, Sdr, Sds and Ssc). It was realised after the measurements had been taken that this was down to the changing magnification. By changing the magnification, additional detail is being resolved, causing discrepancies between the attained roughness values. So when further conclusions are made between this data, it shall be based on the same magnification. (For ease-of-use, the data in the appendices has been highlighted with the relative magnification level.)

At the end of chapter 3, it was concluded that three 3D statistical “roughness parameters showed promise in the ability to describe a ground surface for its effectiveness in minimising friction”. These parameters were Sc (core void volume), Ssc (mean summit curvature) and Sku (kurtosis of the surface). The next question is, are these parameters still useful when textures are introduced onto the surface? Well, based on the findings of chapter 3, a hypothesis was generated for the components used in the tribometer experimentation. For each set of comparative components, a best-performing sample was chosen based on the roughness data. See Table 5.5 for the required roughness data relating to the first set of comparative components and Table 5.6 for the second set. In relation to the data presented in chapter 3, it was concluded that increasing the values of Sc and Ssc would reduce the coefficient of friction, whereas, Sku needs to be minimised in order to reduce frictional losses. Using these rules, the best performing components have been selected (see Table 5.5 and 5.6). These hypotheses will be compared with the tribometer results later in the chapter, to see if they correctly predict the component performance.

Table 5.5. Roughness data for components 4, 7 and 8 (based on 10X magnification data).

Component No.	Average Roughness Value		
	Sc ($\mu m^3/\mu m^{-2}$)	Ssc (1/mm)	Sku
4	0.271	198.000	3.589
7	0.344	318.667	4.450
8	0.257	322.000	3.790
Best Performing	7	8	4

Table 5.6 Roughness data for components 5 and 6 (based on 2.5X magnification data).

Component No.	Average Roughness Value		
	Sc ($\mu m^3/\mu m^{-2}$)	Ssc (1/mm)	Sku
5	0.396	30.597	39.982
6	0.163	13.978	16.959
Best Performing	5	5	6

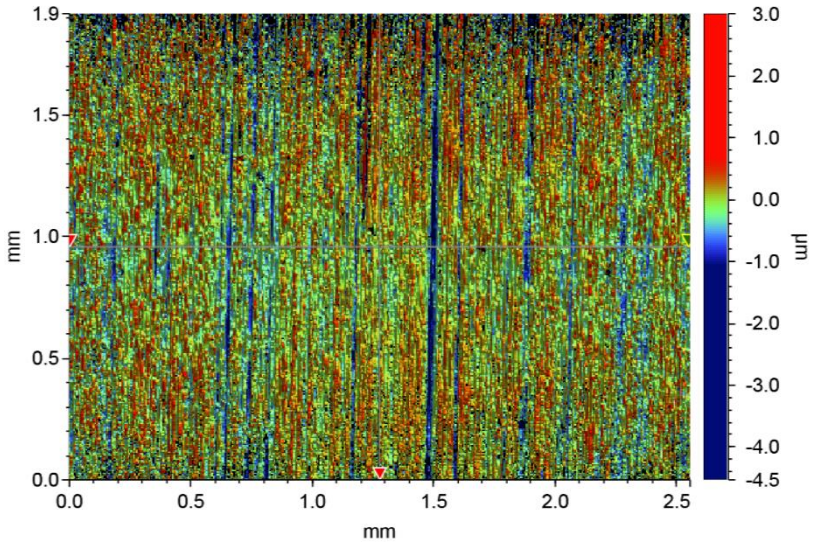


Figure 5.8. Surface pattern for component 1 (2.5X magnification).

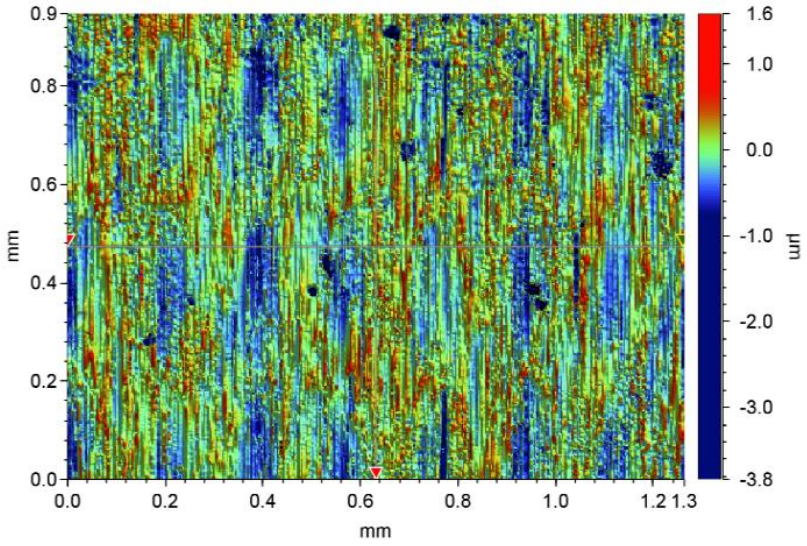


Figure 5.9. Surface pattern for component 2 (5X magnification).

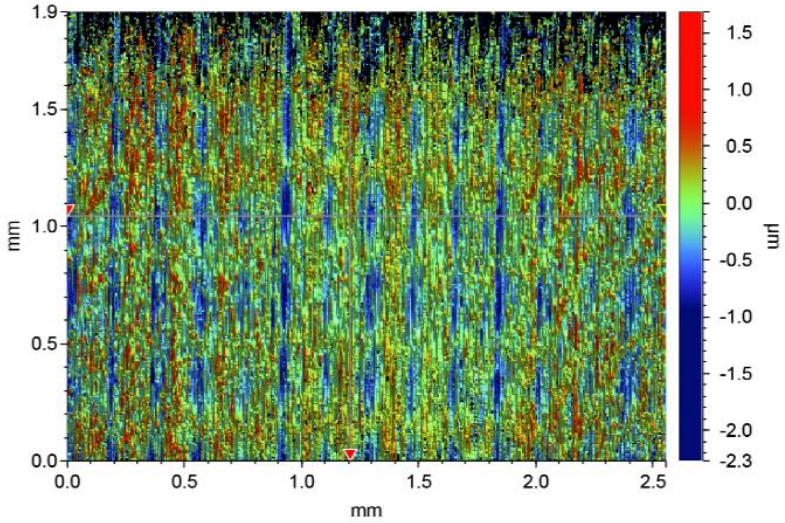


Figure 5.10. Surface pattern for component 4 (2.5X magnification).

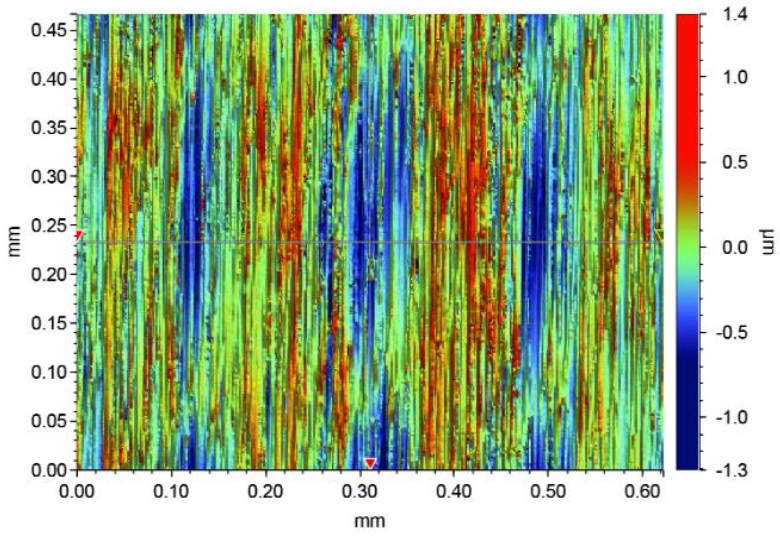


Figure 5.11. Surface pattern for component 4 (10X magnification).

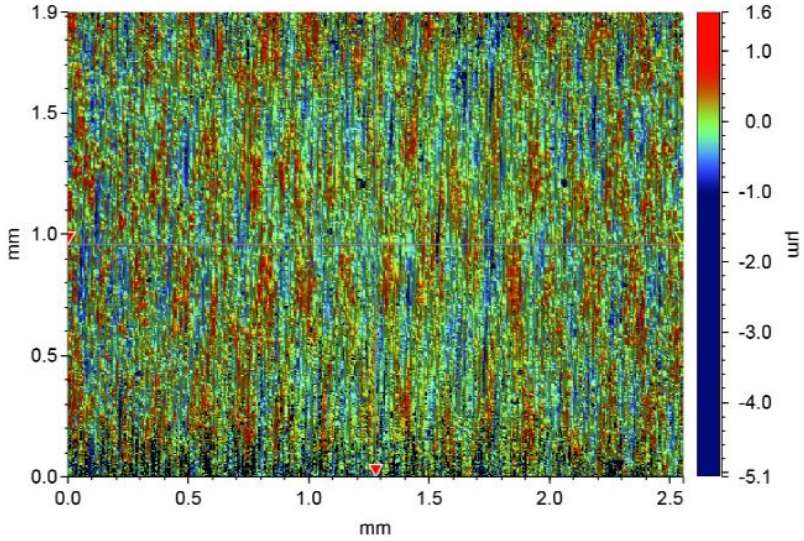


Figure 5.12. Surface pattern for component 5 (2.5X magnification).

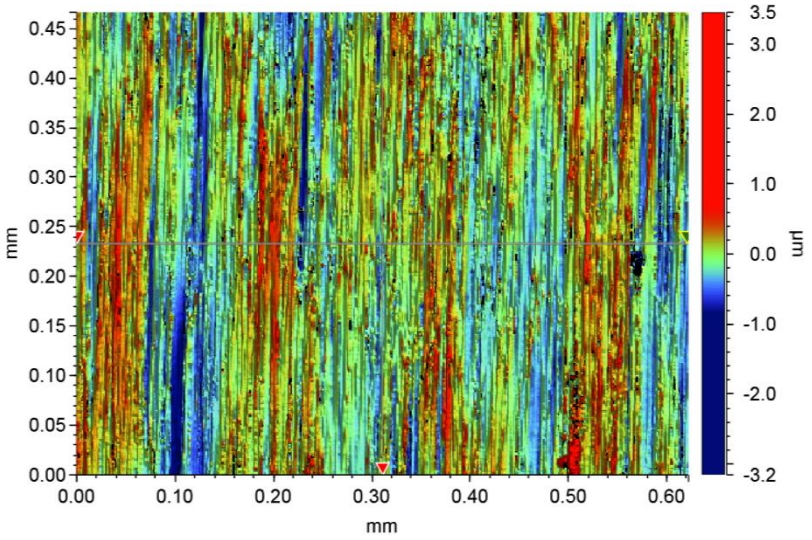


Figure 5.13. Surface pattern for component 5 (10X magnification).

5.6 Discussion of the Tribometer Experimentation

In order to attain the coefficient of friction for each component, the tribometer was used as described in section 3.4. The only difference in the experimental method was the force applied. The force applied previously was 100N. For the experiment described in this section, a 30N load was applied to all components during each trial. It was decided to lower the load applied in order to judge the performance under the worst conditions that could be measured. The load was chosen by incrementally increasing the force by 1N until the bearing was stable enough (so that it *ran true*) to be tested. By keeping the load as low as possible, the hydrodynamic wedge effect is minimised, reducing the lift needed to separate the surfaces and therefore will result in high friction conditions. Consequently, the coefficient of friction seen in this experimentation is much higher than any of the data produced using the higher load in chapter 3 (Appendix G).

Some problems were encountered during testing. One of the problems with testing 5 and 6 was the smaller diameter. As a result of the radial clearance being so large, the volumetric flow rate was increased and the closed reservoir filled up completely before the 4 minute testing period was met, so only 3 minutes of test data was generated. Another complication that was

experienced during testing was the temperature monitoring. The thermocouple being used had somehow become damaged before trials were run so no temperature data can be presented. Fortunately, due to the fact that the oil is not circulated and time between tests is enough to allow the components of the rig to reach room temperature again, the starting conditions are always the same. So any changes in temperature are a result of the actions applied during each trial and therefore the data from each trial can be compared against each other. The results of the friction testing can be seen in Table 5.7.

Table 5.7. Final coefficient of friction values.

Component	Diameter (mm)	Coefficient of Friction	Reduction (%)
4 – Textured	52.964	0.389	18.4
7 – Polished	52.970	0.477	
5 – Textured	52.570	0.288	(-83.4)
6 – Polished	52.574	0.157	
4 – Textured	52.964	0.389	17.1
8 – Polished & Vibratory Finish	52.961	0.469	

Note: brackets around value denote an increase percentage.

Textured component 4 showed consistently lower frictional forces than component 7 throughout the test (Figure 5.14). What is also interesting about this data is how the coefficient of friction decays over the test period. Component 7 shows a much steeper decline, initially, in comparison to its textured counterpart. Usually, this type of behaviour is a sign of some initial wear. One of the benefits of a textured surface, which was validated by Andersson et al. (2007), is that pockets retain oil which can feed a *starved* surface and maintain some texture-induced lift (load capacity) between the contact areas. This ability to provide surface separation under *starved* conditions

significantly reduces the chance of wear from occurring. If wear does occur, then the pockets also can act as a backup, collecting debris away from the contact interface. After 1:30 minutes, a constant decline with the same gradient as the other component is seen. This will be the temperature increase at the surface leading to a reduced viscosity. It is well known that hydrodynamic-type bearings create significant quantities of heat in comparison to other types (Marinescu et al. (2007), p.309-310).

The first thing to notice about the data from component 5 and 6 (Figure 5.15) is the smaller coefficient of friction than the previous data. This is down to the significant increase in radial clearance on components 4 and 7 (between 195µm to 200 µm). This data also shows no sign of wear on either of the test cases, which could be down to the fact that the clearance is considerably larger which does further minimize the chance of wear from occurring at the interface. But in this scenario it is clear that, the textured bearing does actually increase the coefficient of friction.

Figure 5.16 presents the data from component 4 against the trial for component 8. As mentioned previously, component 4 and 7 has similar values for average roughness, Sa. So, it was decided to lower the average roughness by further polishing this component (creating component 8) using a vibratory finishing technique. Well it is evident by looking at the drag reduction percentage in Table 5.7 that the vibratory finishing has had a positive effect, lowering the coefficient of friction (COF). Still, a drag reduction of 17.1% is observed against the better performing, textured sample (component 4). One thing that is noticeable for trial of component 4, is the small jump in COF at approximately 1 minute and 15 seconds. It is hard to actually determine what this actually is but it could maybe indicate some brief contact.

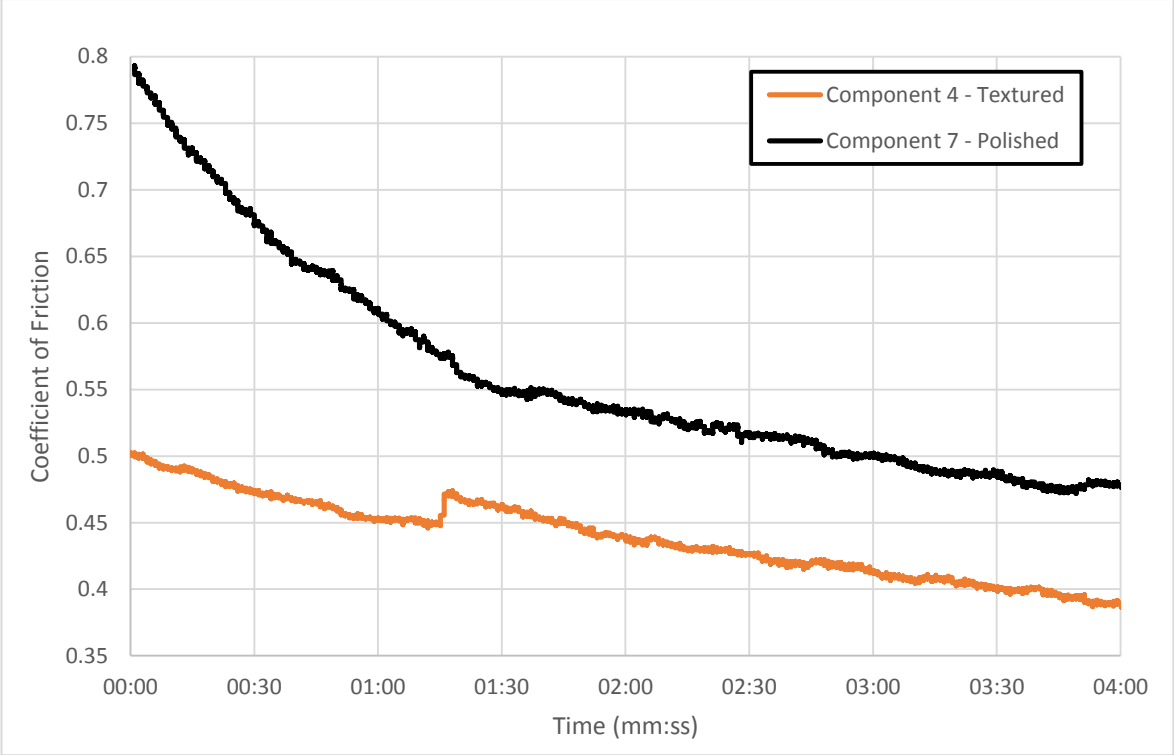


Figure 5.14. Comparison of coefficient of friction between component 4 and 7.

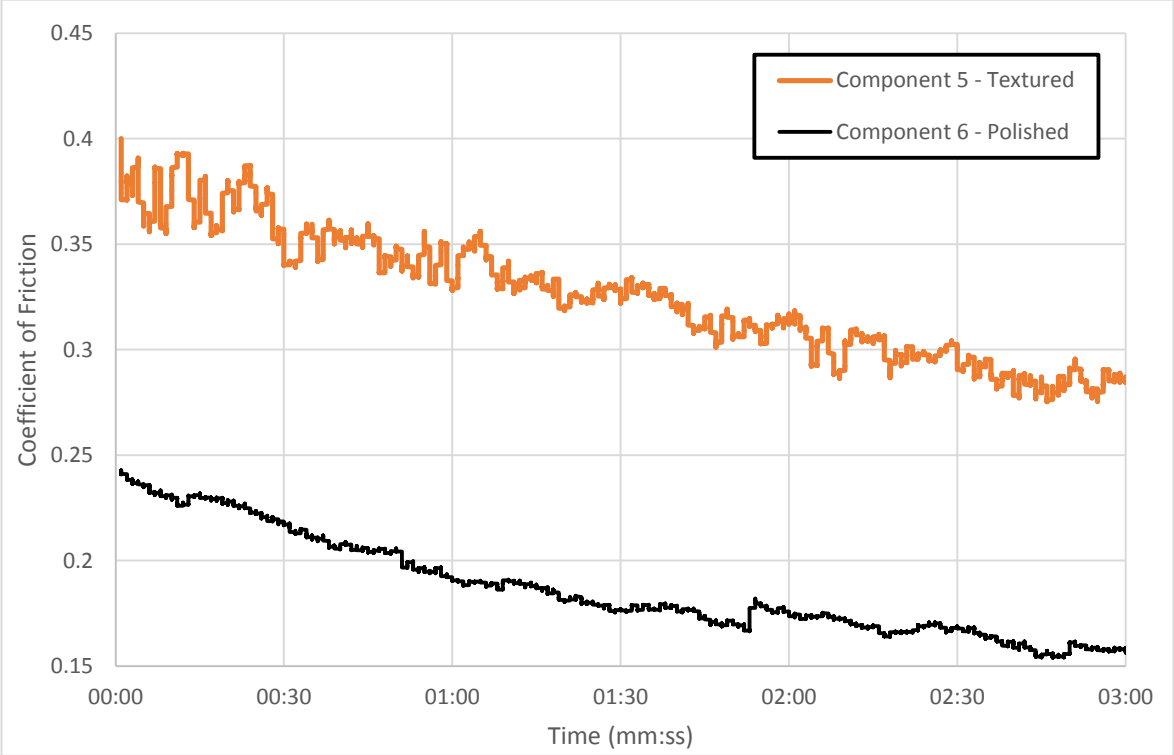


Figure 5.15. Comparison of coefficient of friction between component 5 and 6.

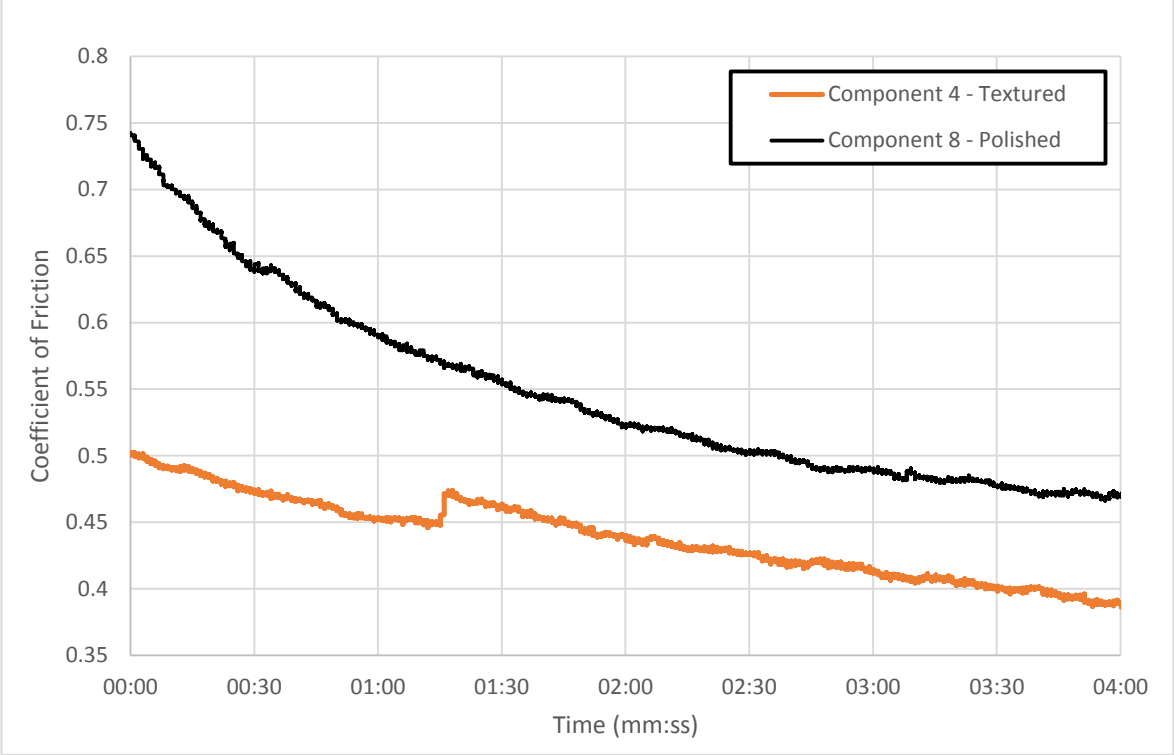


Figure 5.16. Comparison of coefficient of friction between component 4 and 8.

It is also apparent that when comparing the time history of the frictional performance of component 7 against 8, not only are the starting values of coefficient of friction lower but the initial period is not as abrupt. Giving indication that little wear or actual contact of the surfaces is occurring. A lower kurtosis value of 3.790 is observed for component 8 (Table 5.5) as opposed to the value of 4.450 for component 7. This means that through either, or a combination of, the first testing period and the additional vibratory finishing process, the *sharp peaks* of the surface structure have been removed. This initial testing of the bearing may have resulted in something that is referred to as the *breaking-in period*, where the bearing is run and initial wear removes the peaks of the surface, inadvertently producing a better performing surface. Although, according to Marinescu et al. (p. 309) wear is usually observed on start-up of a hydrodynamic bearing until an oil film is established.

5.7 Wettability? Surface Roughness Parameters? Do these predict frictional performance?

As discussed in chapter 2, a surface can only be described as *functional* provided that there is relationship between measured surface parameters and the performance indicators, which in this case is the coefficient of friction. This section will explore whether surface roughness parameters or surface energy can be used as a performance indicator.

In section 3.3.4, surface energy was discussed together with its relationship to the frictional performance of a bearing surface. For the components in chapter 3, the static sessile drop technique was used with the Attension Theta Lite goniometer in order to determine the contact angles for different liquids (glycerol and water). The measured contact angles were then used to solve for the surface energy of each component by the Owens-Wendt method (see eqn. 3.2). Unfortunately, it was concluded that from the correlations of surface energy against coefficient of friction (Figure 3.11) it was a poor performance indicator. It was determined that there were issues with the choice of the liquids used in the experiment. The dispersive component of water and glycerol (Table 5.8) were too similar in magnitude, so that when they are used to solve for the surface energy by a set of simultaneous equations, larger errors will result, as the resulting equations were ill-conditioned. As a result, it was decided to re-evaluate the set of components using water and diiodomethane for the Owens-Wendt method. The van Oss method (van Oss et al. (1986)) presented in eqn. 5.6, splits the polar component of the Owens-Wendt equation into its acid and base components. The introduction of additional unknowns for the surface means that three sets of liquids have to be used in order to determine the surface energy. As mentioned previously, this improves the robustness of the solution/calculation, which is particularly useful in

the case of higher surface energy surfaces, such as metals but as a trade-off, it is also subjected to a higher sensitivity to the values input (Zenkiewicz (2007)).

$$\left(\gamma_s^d \gamma_l^d\right)^{0.5} + (\gamma_s^+ \gamma_l^-)^{0.5} + (\gamma_s^- \gamma_l^+)^{0.5} = 0.5(1 + \cos\theta) \quad (5.6)$$

(Where the + superscript indicates acid and – superscript is the base values).

Table 5.8. Surface tension properties (taken from Lee (1996)).

Liquid	Surface Tension Values (mN/m)				
	Dispersive	**Polar**	**Acid**	**Base**	**Total**
Water	21.8	51.0	34.2	19	72.8
Glycerol	34.0	30.0	5.3	42.5	64.0
Diiodomethane	50.8	0	0	0	50.8

The contact angles determined from the static sessile drop experimentation can be found in Table 5.9. These values were then used with the Owens-Wendt method to predict the surface energy values for each of the components tested (Table 5.10). Comparing the components of similar diameter, 4 and 8, it can be seen that a lower surface energy result does correlate with a lower coefficient of friction. Now using the van-Oss method (results presented in Table 5.11), again the lower surface energy is predicted for the best performing component out of the comparative set. The surface energies predicted by the van-Oss method do seem to be lower in magnitude to those which were calculated using the Owens-Wendt but both predict the best and worst performing surface.

Table 5.9. Contact angles obtained from static sessile drop measurements.

Component	Trial No.	Contact Angle (deg.)		
		Diio.	Water	Glycerol
4	1	40.7	75.5	70.7
	2	41.5	74.8	69.5
	3	38.3	73.6	71.6
	Avg.	40.2	74.6	70.6
5	1	45.2	62.7	75.1
	2	45.6	64.0	73.2
	3	46.1	64.8	74.1
	Avg.	45.6	63.8	74.1
6	1	42.4	74.9	73.9
	2	41.0	68.3	74.5
	3	39.8	73.2	74.4
	Avg.	41.1	72.1	74.3
8	1	40.5	73.1	75.0
	2	46.7	62.6	78.4
	3	44.9	67.7	70.0
	Avg.	44.0	67.8	74.5

Table 5.10. Surface energy results determined using the Owens-Wendt method.

Component	Surface Energy (mN/m)		
	Dispersive	Polar	Total
4	39.5	5.5	45.0
5	36.7	11.5	48.2
6	39.1	6.6	45.7
8	37.5	9.1	46.6

Table 5.11. Surface energy results determined using the van-Oss Method.

Component	Surface Energy (mN/m)				
	Dispersive	Acid	Base	Polar	Total
4	39.5	8.7	0.0	0.7	40.3
5	36.7	22.7	0.7	8.0	44.7
6	39.0	13.1	0.4	4.5	43.5
8	37.5	18.1	0.6	6.5	44.0

What about surface roughness parameters as a performance indicator? Well, before the tribometer experimentation was performed, a set of hypotheses were generated in order to predict the best performing component based on the values of Sc, Ssc and Sku. The predicted best performing surfaces were presented in Table 5.5 and 5.6. So, how do these compare? Consistently, Sc and Ssc predict the incorrect components whereas, Sku actually predicts the better performing component for both comparative sets. What is also interesting is the magnitude of Sku as well. Looking in particular at the large increase in frictional losses (83.4% increase) for the textured component 5 against component 6 (Table 5.7), a substantial increase is also seen in Sku. Which, according to the conclusions presented in chapter 3, would indicate a much lower frictional performance. This then does agree with that relationship for both textured and ordinary, ground surfaces.

In conclusion, on the basis of the evidence presented within this thesis, surface roughness parameter, Sku, has shown to be a reliable performance indicator for frictional performance, whether that is for a surface with textures or for a surface without. In terms of the wettability indicator, surface energy, this also shows some potential as a performance indicator. On the other hand, only a small batch of components have been correctly tested with the appropriate liquid selection. A much larger sample size needs to be tested in order to make such a substantial claim. In order to overcome the limited sample size, the original components in chapter 3 were re-tested with the correct liquid combination but some components were displaying significantly different contact angles (for water and glycerol) when compared against the original data (see Table 5.12). Even though this could be attributed to contamination issues others have suggested that this phenomena is related to the change in surface chemistry over time (Ta et al. (2016) and Kietzig et al. (2009)). Ta et al. laser textured 304 stainless steel sheets and some measured an initial contact angle of 10^{o} on the day of manufacture. After 6 months, all surfaces presented a stable contact angle which had become superhydrophobic with little difference in the evolution of the contact

angle over time. Kietzig et al. concluded that the key factor in this change was a result of carbon dioxide in the air depositing carbon to the active magnetite in the surface. If the conclusions made from the wettability study are correct, then the coefficient of friction could reduce further with time.

Table 5.12. Change in contact angle over time for components from chapter 3.

Component	Water			Glycerol		
	Sep-14	Jan-16	Change	Sep-14	Jan-16	Change
1B	72.8	80.1	10.0%	73.2	76.5	4.6%
2B	77.2	77.6	0.6%	62.6	62.4	-0.4%
3B	67.7	76.4	12.8%	65.0	73.0	12.3%
4B	69.0	94.1	36.3%	65.2	72.1	10.5%
6B	78.5	82.0	4.4%	74.9	83.2	11.1%
7B	71.3	89.2	25.0%	73.8	74.7	1.3%
8B	62.8	85.9	36.8%	62.2	87.3	40.3%
9B	81.1	85.4	5.3%	65.9	75.8	15.0%
C1	70.3	82.3	17.1%	78.3	76.4	-2.4%
C2	68.4	83.8	22.5%	70.7	76.3	8.0%
C3	79.3	76.7	-3.2%	71.4	78.6	10.1%
C4	88.0	90.0	2.2%	70.5	88.6	25.6%

5.8 Conclusions of the Experimental Investigation and Suggestions for Further Work

This final section of this chapter presents conclusions relating to the grinding process used for this investigation and the performance of the components produced.

Textures introduced on the surface of the workpiece by the cylindrical grinding process was found to be relatively successful but some issues still remain to be resolved. The machine used presented limited capabilities when texturing the grinding wheel. The ability to control the movement of the dressing edge in relation to the wheel surface is critical and multiple passes (on this machine) required the dressing tool to move to the starting position before another pass was initiated. Being able to move the tool back and forth across the surface with synchronisation to the wheel rotation would be ideal. This capability would not only allow for control of staggered patterns but also enable wider tool engagement through the use of multiple passes, without causing grains to be removed from the bonding material. Another interesting suggestion would be to change the wheel grade to a smaller grain size, perhaps using a cubic boron nitride (CBN) based wheel. This would be beneficial for, firstly, improving on the number active grains. More active grains will lower the cutting forces applied on each grain and also improve the definition of the pocket structures. Secondly, it would improve the strength of the wheel for the aggressive cutting process. The textured wheel, which was used to produce component 4, was also used for component 5. Figure 5.13 shows that, qualitatively, little structural definition is observed. This may be down to the occurrence of excessive wheel wear. By improving the wheel strength, this effect could be reduced. Another issue with the process is the exclusion of a dwell period. The 2D profiles presented show a *camshaft-like* shape, which is a result of the feed movement into the rotating workpiece. Could it be possible to design the textures in a way so that they are approximately equally spaced on the component circumference? This means multiple revolutions of cutting could occur without destroying the surface structure. This would also require, a better

synchronisation of the workpiece rotation by the machine control system, making sure the calculated kinematics of the texturing process are more precisely adhered to. During the grinding trials, the acoustic emission (AE) signal was used to judge whether the wheel had actually engaged with the workpiece or not. Could the signal be used in a more useful manner by actually instructing the control system as to when the wheel has engaged, at a certain wheel position and moving to the desired position in relation the wanted depth of cut? Controlling the real depth of cut is important because a changing depth will result in a deviation in texture size. The different depths observed across the measured surfaces indicate considerable deviations from the expected dimensions for the existing grinding process.

In relation to the performance of the textured surfaces, some success has been shown in the ability to reduce frictional losses. Although, as explained throughout this thesis, control and definition of the surfaces is key. By not keeping the manufactured surface structure near to the optimised size and texture shape, higher frictional losses will be observed for the textured component against the non-textured counterpart. When it is adhered to, as shown by the performance of component 4, the beneficial properties become very apparent. Not only did this component produce reductions against a similar average roughness sample but it also outperformed one which was exposed to further finishing. As an indication of frictional performance, both the surface roughness parameter, Sku and surface energy show promise. Sku has shown to be beneficial in predicting better performing surfaces, both textured and non-textured. This gives the impression that reducing the sharp *peakedness* of a surface structure is of benefit. The wettability conclusions are less definitive. The surface energy performance indicator needs further investigation with larger sample sizes in order to make a more definite claim. But on the basis of the initial results presented in this thesis, using the correct liquid combinations, the results are encouraging.

The tribometer experimentation has shown to be relatively successful but issues remain with the ability to compare to the numerical study directly. Ideally, a pin-on-disk setup would be the correct configuration for comparison because it removes the additional hydrodynamic lubrication mechanism. Using a pin-on-disk, the clearance would be static and preset to the required distance. With the current experimental setup, the clearance is harder to control and also was difficult to predict. So for future work, it would be strongly suggested to surface grind the pin end to the required surface geometry (texture) and investigate the performance under similar and more repeatable test conditions.

6. Development of a Novel Experimental Apparatus for Analysing Drag-Reducing Benefits of Micro-Riblet Structured Sheets

6.1 Introduction

In section 2.3.3, current manufacturing technologies were ranked in their ability to create micro-textured surfaces, economically. The final and most important point raised was in relation to the combination of different methods. This hybrid method would make use of the rolling method with more elaborate texturing processes, such as grinding. By doing this, complex structures can be formed whilst amplifying their productivity. Complex, three-dimensional, drag-reducing surface structures are observed in nature (see section 2.2.1). Unfortunately, due to the limitations with present manufacturing technologies, these surfaces have been mostly simplified with two-dimensional riblet structures. Most of the 3D structures presented in the literature review actually increase drag. In this section of the thesis, some initial investigation into riblet technology will be presented, with the aim to apply such structured sheets to the surfaces of high speed trains. In order to test these surfaces at much higher velocities, new experimental apparatus had to be developed within the University. The development of this equipment shall be detailed for the reader. Even though the forming process presented in this chapter only produces 2D structures, it is understood that future investigation will make use of the grinding process detailed in this thesis for structuring the roller. The inverse of the structure seen on the bearing surfaces will produce some interesting, 3D, rib-like micro-structures. Hopefully, if the improvements

relating to the dressing procedure in the textured grinding process are implemented, then staggered patterns could be more reliably produced. Using this, in combination with the rolling process, should allow for the creation of staggered, micro-sized ribs. This would be a more accurate representation of the shark-skin surface structure.

6.2 Preliminary Investigation into the Manufacture and Testing of Riblet Sheets

The first step in this investigation was to actually produce some riblets. The initial riblet sheets were made out of aluminium and were formed using the rolling mill in Figure 6.1. The rolling mill was supplied and operated by members of staff at the Rolling Forging Institute, Jilin University, China. One of the rollers was profiled using the turning process. To make the aluminium sheet more malleable whilst also ensuring the longevity of the profiled roller, the material was heated to 600^{o}C in a furnace before being exposed to the forming process. The preliminary riblet sample had similar dimensions to the sample 3 component seen in Table 6.1. The two samples that were initially produced were: non-textured and rectangular groove based riblets. These samples were tested in the University's closed loop wind tunnel. The small, rectangular sheets were held by a rod with a slot in the end and a grub screw to fix the sheet in place. The other end of the rod was held in the force balance. The samples were placed flat (level with the wind tunnel wall) and were tested in 5ms^{-1} increments in the range of approximately 10-40ms^{-1}. The experimental procedure was repeated for streamwise, spanwise and 45^{o} oriented grooves.

The largest drag reduction observed was in the experimentation for the streamwise orientated grooves (results presented in Figure 6.2) at ≈40.8ms^{-1}, where a drag reduction of 7.2% was recorded. As the velocity was increased in this particular test, the drag reduction recorded also

increased. The results from these preliminary trials proved to be positive but obvious issues surrounded the workholding solution used.

Figure 6.1. Rolling mill used with textured roller.

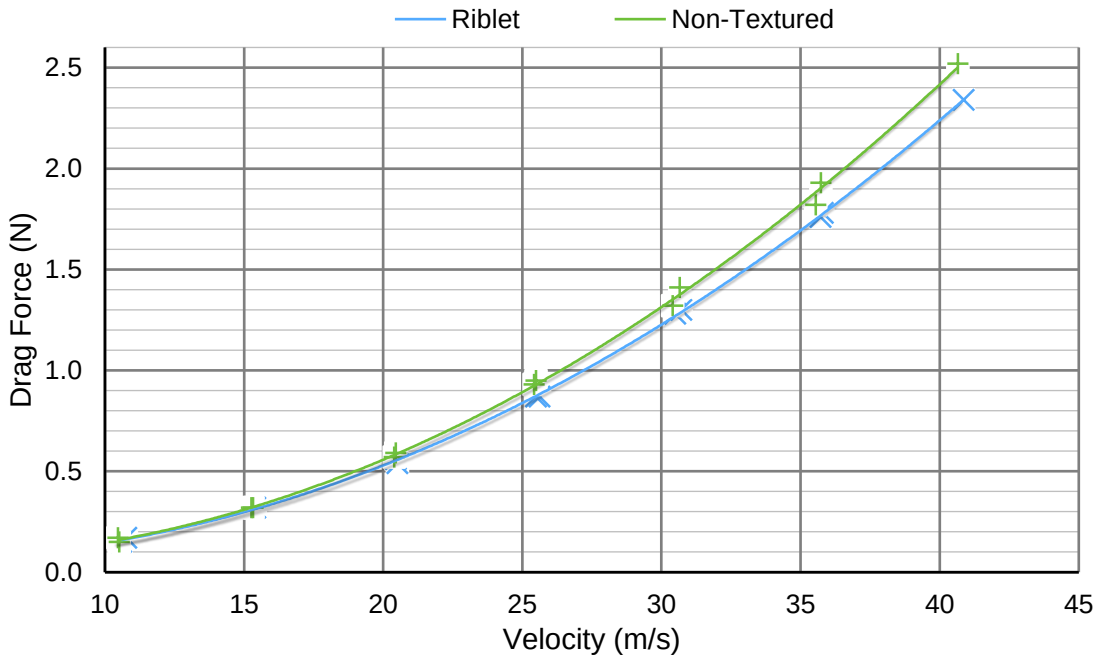


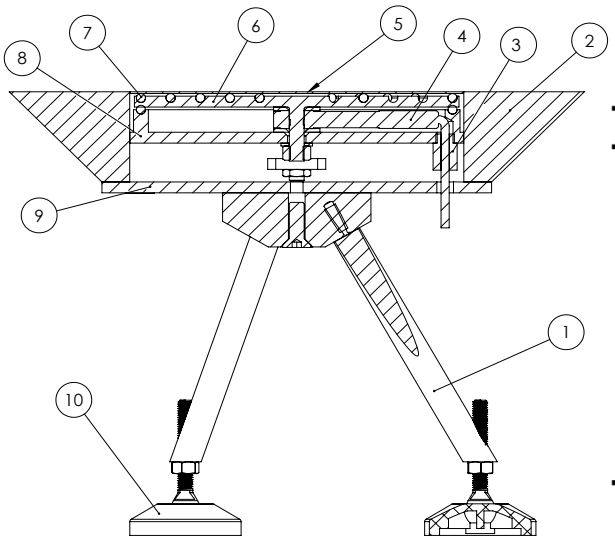
Figure 6.2. Drag force measurements for initial experimentation with streamwise orientated riblet sheet.

6.3 Development of the Novel Workholding Solution

The first problem with the workholding solution is damage to the surface. As different angles of orientation are investigated, the surface is damaged by the grub screw. The second problem is with the minimal support design. Because the sheet is only being held at one point, as the wind velocity is increased, the sheets tend to deflect and deform, reducing the repeatability of the experiment. Using Figure 6.3 as a visual aid, the development of the workholding solution shall be explained. In the first image, it can be seen where the sheet has been damaged by the contact with the screw end. It was then decided to try and glue the sheets to a circular plate. The plate would provide additional structural integrity, minimising the deformation observed in the preliminary trials, whilst maintaining the leading edge for any angle/orientation. Unfortunately, no matter what glue was used with the sample, the sheets would not adhere to the backing plate. The rolling process had introduced some bowing into the textured sheet and when tried to affix to the backing plate, the sheet could not be held completely flat for the curing of the glue. But the idea of having the circular sheet samples was carried forward into the next designs. The next idea is actually quite innovative; using a vacuum clamp. The fabricated vacuum clamp provides substantial clamping force whilst providing the ability to quickly change the sample sheet. The vacuum is provided by a Venturi vacuum pump (Norgren M/58112/09) connected to the airline in the laboratory. The idea behind the vacuum pump (schematic shown in Figure 6.5) is to provide high pressure air through a sudden expansion. Low, vacuum pressures are generated at this point which is where the vacuum clamp is connected to. In the technical specification supplied by the manufacturer, at a supply pressure of 6bar, the vacuum pump can provide a maximum gauge pressure of -0.83bar. So, for an area of 0.025m^2 (sheet diameter of 0.18m), a clamping force of ≈2kN can be applied to the sheet. Substantially more than what is required.

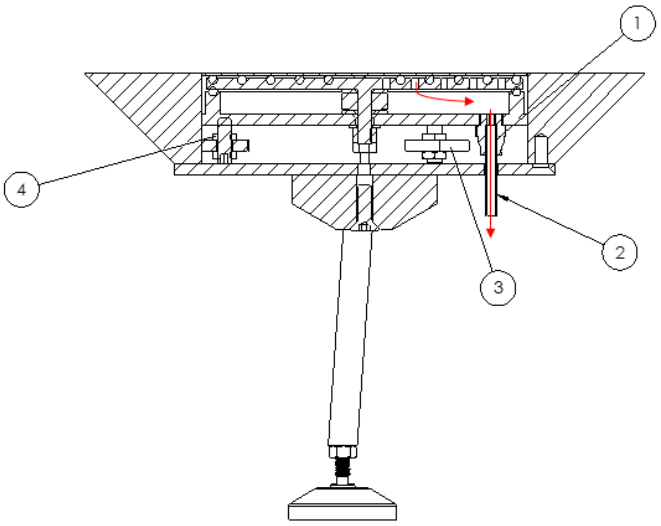
Figure 6.3. Development of workholding solution.

Even though there is no physical obstruction to the flow travelling over the plate, the flow is in fact still disturbed by the large thickness of the cylinder shape. This is because a large *pressure bubble* is formed at the leading edge and this delays the formation of the boundary layer, as it needs to reattach. This then leads to the next design of the workholding solution.



ID	Description
1	Leg support
2	Ring with angled sides
3	Cabled gland
4	Piezoelectric force sensor
5	Textured sheet sample
6	Top vacuum clamp plate
7	O-ring chord
8	Bottom vacuum plate
9	Bottom support plate
10	Adjustable levelling feet

Figure 6.4. Assembly drawing (section view) of workholding solution with part list (1 of 2).



ID	Description
1	Pneumatic straight adapter
2	Pneumatic exit pipe
3	Support lug
4	Adjustable height nut

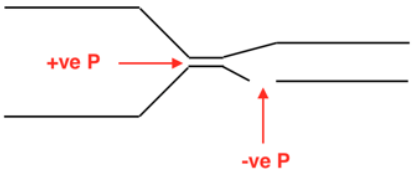


Figure 6.5. Assembly drawing (section view) of workholding solution with part list (2 of 2) and schematic of vacuum pump mechanism.

The first to notice with the final design (assembly drawings presented in Figure 6.4 and 6.5) is the angled sides. By having the leading edge coming to a sharp edge, this will minimise the *pressure bubble*. Another feature that was incorporated into the final design was a KISTLER 9602A-3201 piezoelectric force transducer. This will not only allow for drag forces being applied to the plate to be measured but also take into consideration time dependant forces (vibrations) which could be

damped by the structure of the surface. The riblet sheets are placed upon a set of O-rings spread equally across the diameter of the plate. Because of the large clamping forces, the previous design caused the sheet to bow in the centre. The O-rings will not only provide a tight seal but also help to support the plate. The cable connected to the force transducer is fed through a cable gland to ensure a tight seal for the vacuum chamber. The actual workholding solution is also supported by a tripod leg support arrangement. Because the walls of the wind tunnel are slightly angled, in order to minimise boundary layer growth, the levelling feet will provide the ability to level the top surface in relation to the moving air direction. Also, there are three support lugs inside the equipment. Three threads connected to the vacuum chamber are fed through these lugs and allow the user to level the top surface. The KISTLER force transducer was connected to a National Instruments NI-9219 24-bit C Series Universal Analog Input module through a National Instruments CompactDAQ (NI cDAQ-9174) chassis.

Even though this workholding solution, in principle, should have performed considerably better in the closed-loop wind tunnel facility, multiple problems were encountered. The first problem that was encountered was in relation to the force transducer. Because it is of piezoelectric type, the charge builds up over time and increases the measured voltage, leading to a false drag increase being detected during wind tunnel experimentation. In an attempt to eliminate this problem, two measurements were taken to measure the increase of voltage over time (phase shift measurement). Regrettably, the increase was not stable enough and the voltage deviation would account to approximately 10% of the measured voltage during experimentation (i.e. the increase of voltage due to load/drag force applied to the sensor). The second problem that was encountered was actually a result of the wind tunnel wall strength. At higher velocities, the walls would not provide the support needed for the rig and consequently, large vibrations were experienced. This makes it difficult to reliably measure the boundary layer profile. Also, these

large vibrations were being picked up by the force transducer. This then led to the development of the high-speed open-loop wind tunnel.

The open-loop wind tunnel needed to incorporate two main features. The first was to provide a stable working section assembly for the workholding solution, allowing for precise boundary layer measurements. The second feature required the equipment to provide air velocities in excess of $100ms^{-1}$ (speeds travelled by high-speed trains).

6.4 Development of the High-Speed Open-Loop Wind Tunnel

At the time of developing the open-loop wind tunnel, a redundant blower (1.2m rotor diameter, 10 backward facing blades) apparatus was found in the laboratories. It was decided to retro-fit the apparatus for the purpose of this investigation. Ducting (0.3m by 0.3m cross-sectional area) was constructed for the blower apparatus in order to direct the air to the working section. Air also passes through two flow straighteners to guarantee the maximum eddy size. Both flow straighteners have hexagonal structures with 5mm hydraulic diameter. The first flow straightener is encountered as soon as the air leaves the blower and the second is encountered just before the converging section. These eddies are then compressed through a converging section before the air enters the final section, where measurements are taken. The shaft connected to the rotor of the blower was driven by a 3.7kW motor (1:1 ratio) and at 130.2% power, the open-loop wind tunnel produces a maximum velocity of $88.749ms^{-1}$ ($0.346m^3s^{-1}$ volumetric flow rate). As this velocity was not high enough for the required specification, it was decided to calculate the required power and rotational speed of the wind tunnel. In order to perform this calculation, data was required for the current setup. Figure 6.6 and 6.7 show the power usage and rotational speed mapped against volumetric flow rate for the current setup. To perform this measurement, the unloaded motor was first ramped through a series of rotational speeds. The motor was then

connected to the rotor and ramped through the same rotational speeds. The power data shown in Figure 6.6 is only to drive the rotor. The power to drive the unloaded motor was taken away from the measured power, giving the power required to drive the rotor only. In order to achieve the required velocity, a volumetric flow rate of $0.4m^3s^{-1}$ was required. Using the trendline equations presented on the plots, it was predicted that a power of 5.561kW and a rotational speed of 3319.33RPM was needed to meet this target. Unfortunately, the maximum speed of a 2 pole motor is 3000RPM (50Hz supply). This meant that a gearing ratio was required from the pulleys (used to connect the motor to the shaft via a belt). A ratio of 1.188 was chosen, meaning that the power requirements would now be increased to 6.604kW. Including the 88% efficiency (electrical to mechanical power) stated by the manufacturer's specification (TECAB3-132-52), the minimum rated power for the motor should be 7.504kW.

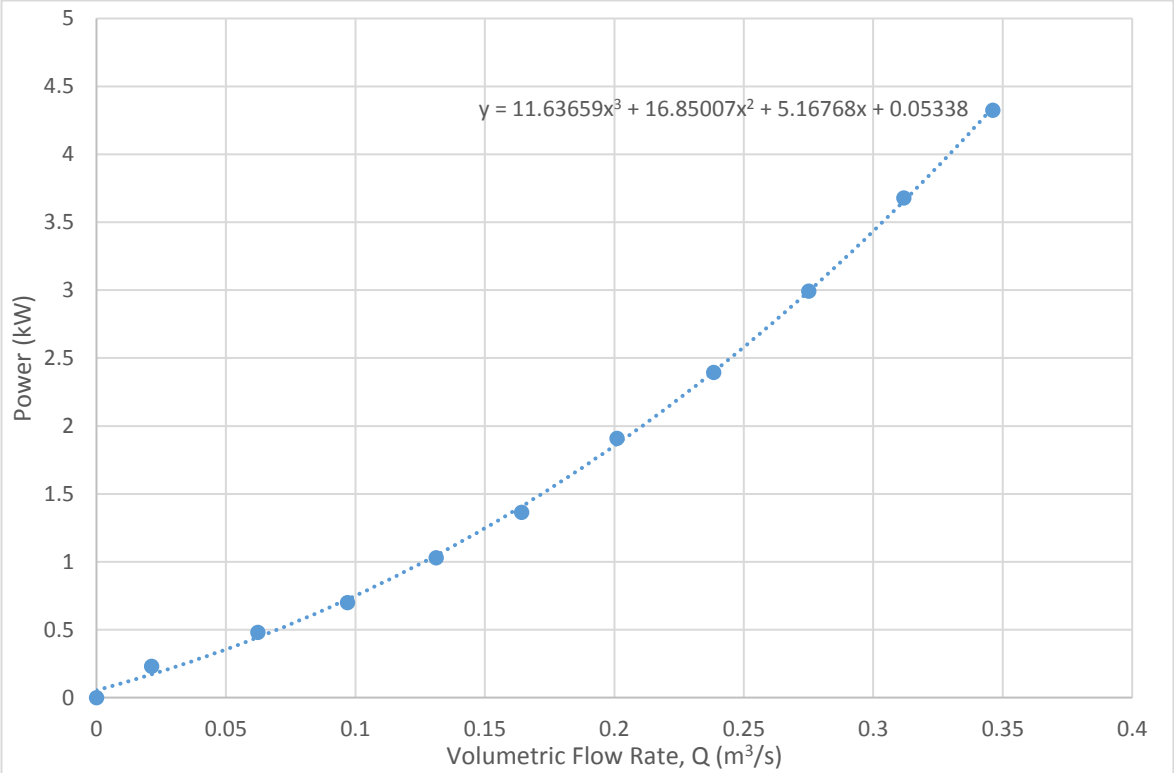


Figure 6.6. Mapping power requirements for open-loop wind tunnel.

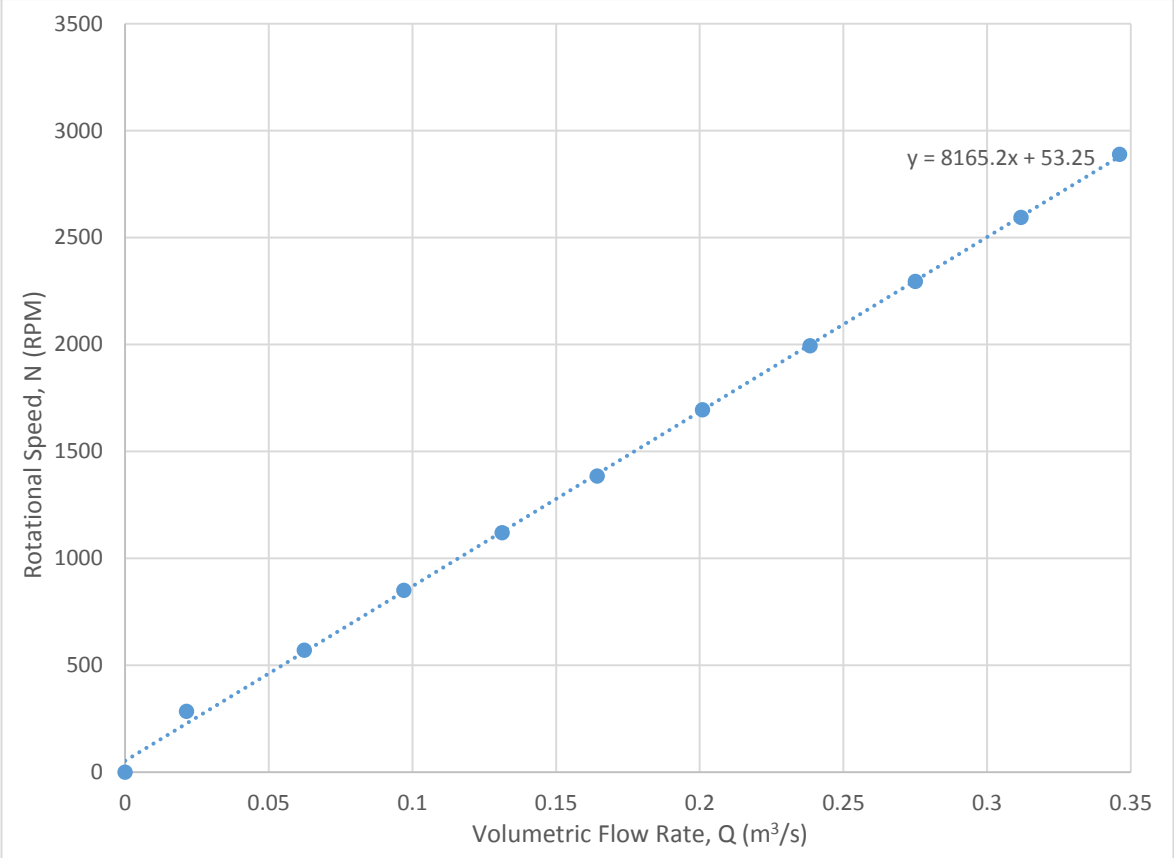


Figure 6.7. Rotational speed mapped against volumetric flow rate for the open-loop wind tunnel.

The next task was to design the working section for the open-loop wind tunnel (see Figure 6.8 for the assembly model). For stability and structural integrity, 2inch box section steel was used to support the slave plate of the working section. The slave plate itself was 25mm thick (1m long and 0.6m wide), flash-ground mild steel plate. To get the slave plate parallel with the ducting, the whole plate sits on three set screws. Once levelled, screws adjacent to the set screws can be tightened, locking the plate into position. Down each side of the plate are two stainless steel rails. For each of the carriages that support the overhanging beam, two linear bearings are connected to a rail. One of the carriages in this axis is connected to a Goodwin Technologies linear encoder (600mm travel). On the y-axis (across the beam), the whole carriage sits on two linear bearings which moves across a dovetail slideway; again this is connect to another linear encoder (350mm travel). On the z-axis (depth), the carriage sits on one linear bearing across another dovetail

slideway. In the x and y axis, the movement is free unless locked into place by a threaded handle. The z-axis is controlled by an attached height gauge. This allows for small incremental movements, which is particularly useful for profile measurements. The encoder connected to this axis is of a higher resolution (0.1µm). The encoder is connected to a National Instruments NI 9411 quadrature counting card. LabView software is used to automatically process the velocity measurements and record the z-distance that the probe is at. The main instrument used to measure the velocity (as well as the fluctuations of velocity) is a DANTEC 55PS16 single component, hot-wire anemometer (connected to a NI 9215 module).

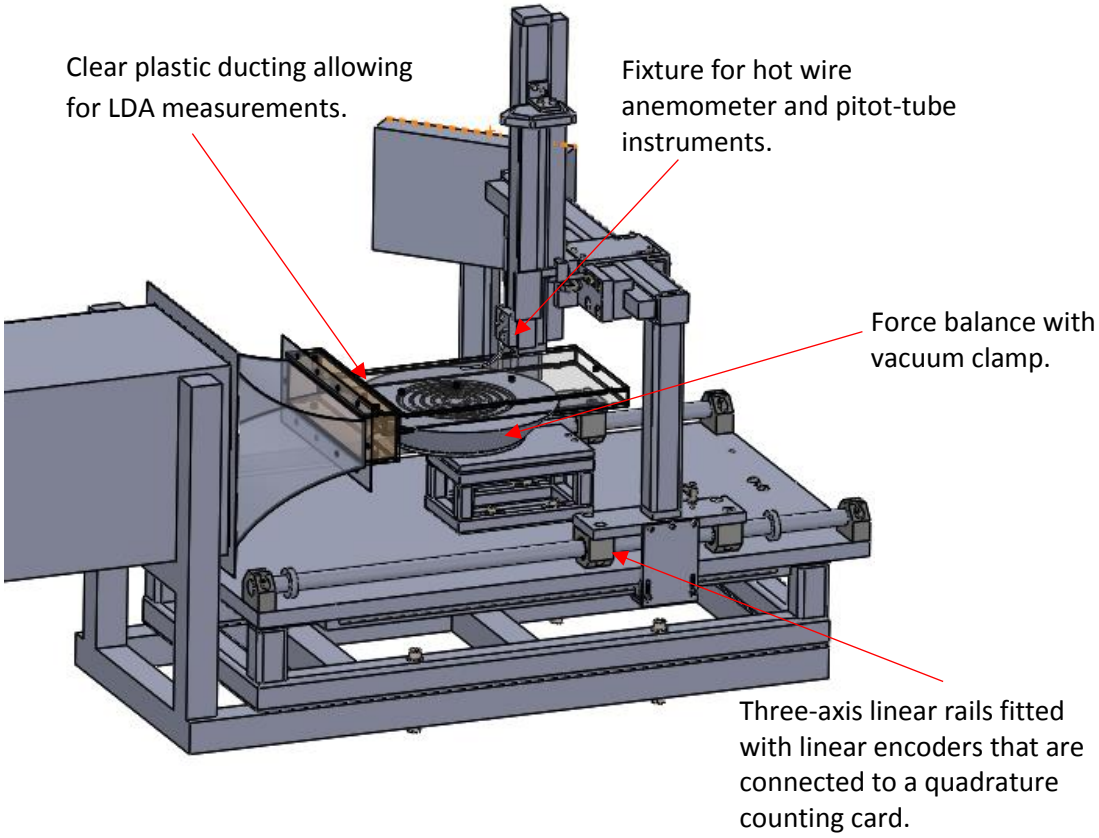


Figure 6.8. Working section assembly.

At the end of the converging duct section is another 3D printed converging section which then leads onto the clear plastic ducting. The ducting has been purposely made out of clear plastic acrylic so that laser Doppler anemometer (LDA) measurements can be taken in the future. Holes

in the top of the acrylic allow the probe to travel to the surface of the plate. Each of the holes have positioned with a point of interest (e.g. end of plate point). The internal height of the ducting has been designed so that it is at least 5 times the turbulent boundary thickness at the end of the ducting section.

In order to relate the voltage measured across the force transducer to the force/load applied, calibration of the workholding solution needs to be carried out. But one of the problems with the design was that it is particularly hard to fix weights to the assembled apparatus. So, some additional components were designed and manufactured so that the calibration can be easily carried out. Figure 6.9 shows how the additional components are setup in order to calibrate the sensor. The plate on top of the assembly is fixed rigidly to the top using the vacuum generated inside. The plate has a component attached to the top with a locating hole. A bearing sits between the two components so that the pulley locates straight to the centre point. The additional pulley component is detachable, as it is not needed during wind tunnel testing. The idea behind the calibration is not only to relate the voltage to the load applied but also to locate the primary directions/axes of the sensor. This is done by rotating the whole workholding solution until the maximum voltage is seen in one component and then fixed in place using a screw underneath.

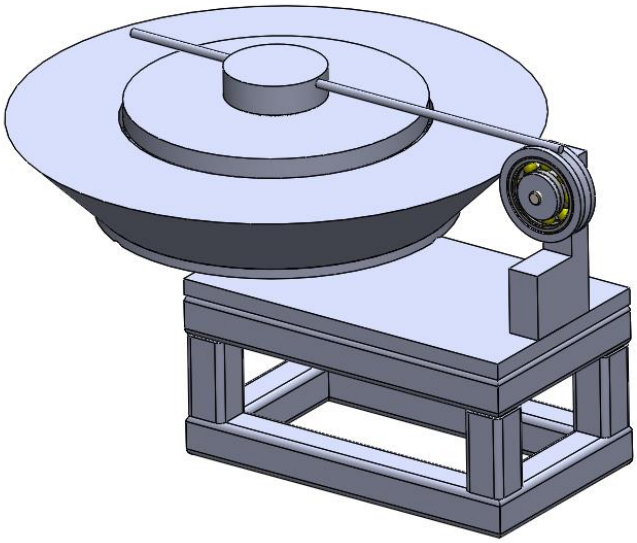


Figure 6.9. Setup for calibrating force sensor.

In order to overcome the problems associated with the previous force transducer, it was decided to design and manufacture a bespoke sensor to fit inside the vacuum chamber. Two Honeywell FSS1500NSB piezo-resistive based load cells were used as the basis of the sensor, to capture the forces orientated in the streamwise and spanwise directions. This particular sensor, has sensitivity of 0.14mV/g and has an input voltage of 5V DC. The load applied in the direction of the z-axis, wall normal, is not important. The sensors were embedded in the assembly as shown in Figure 6.10. The middle pin is loose and screws into the top plate. At one side of the pin, in each direction, the face is pre-loaded against the flat face. The pre-load is applied through the opposing screw, applying pressure to the opposite side of the pin. The actual pre-load is judged by measuring the voltage output of the sensor as the opposing screw is incrementally rotated and when a signal change is detected, the pre-load should be substantial enough.

Figure 6.10. Replacement force transducer setup.

The sensor is connected to the terminal box (shown in Figure 6.11) through a shielded, twisted pair cable with a 15 pin female D-sub connector. The terminal box provides a grounding point, two BNC connections (which provide the connection to the NI 9219 module) and two plugs for the power supply connection (positive and negative). Initial measurements of the voltage over time have shown the sensor to be incredibly stable, which is probably down to the piezo-resistive based load cell. The only flaw with this type of load cell, is that they are usually more sensitive to temperature changes but if this does become a problem, an alternative temperature compensated sensor is available from Honeywell, which has the same dimensions/envelope. This means that it could easy be implemented with the current assembly, if needed.

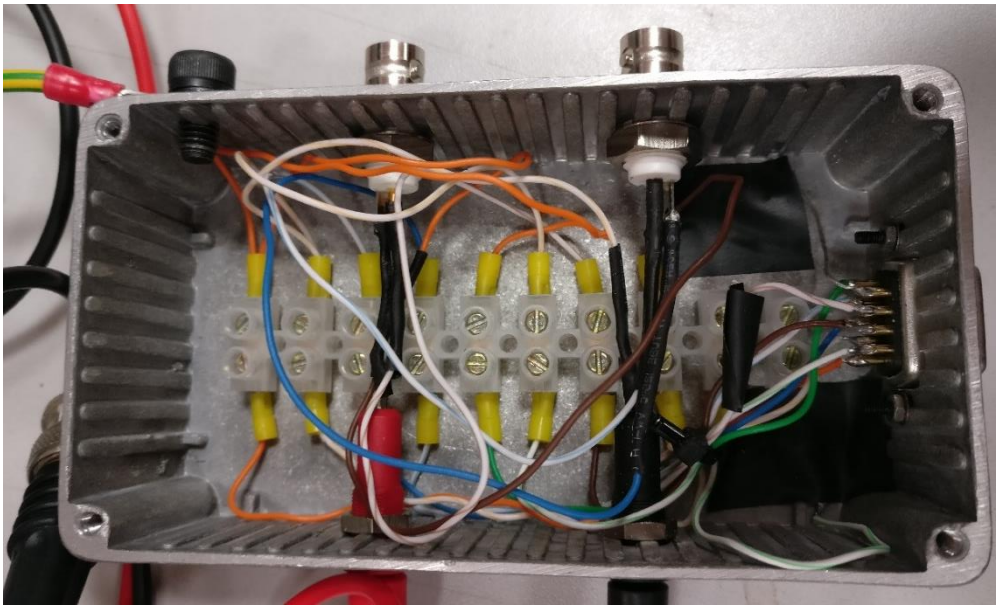


Figure 6.11. Wired terminal box for load sensor.

6.5 Micro-Riblet Structured Sheets

After the initial rolling process was carried out, another set of plates were made using the same rolling mill and textured roller. The clearance between the rollers is adjusted to get different riblet sizes. Three different plates were manufactured and the resulting dimensions are shown in Table 6.1. These dimensions were all measured using the Bruker Contour-GT white light interferometer (a surface measurement for sample 3 is presented in Figure 6.12).

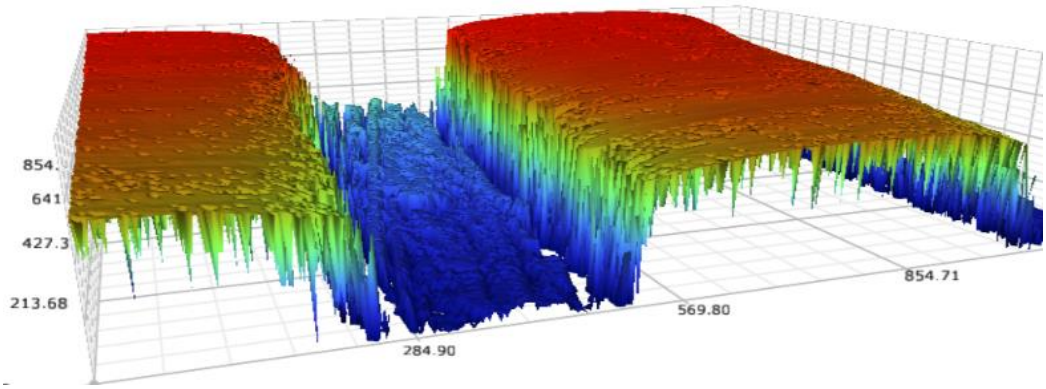


Figure 6.12. 3D surface measurement for riblet product with lowest force.

Table 6.1. Measured riblet dimensions.

Component	Measured Dimension (mm)		
	Groove Width	Spacing	Groove Depth
Sample 1	0.293	0.535	0.075
Sample 2	0.223	0.603	0.074
Sample 3	0.200	0.599	0.049

6.6 Conclusions

This section has given the reader some insight into the potential of using the rolling method for texturing large surface areas. It is envisaged that in the future, the cylindrical grinding process described in this thesis could be used to texture the roller components. This would provide three-dimensional micro-rib based structures which would be more comparable to shark-skin structures than what has been observed in literature, currently.

The preliminary wind tunnel testing has been beneficial in two ways. Firstly, it has shown that these rolled, textured sheets have the potential for reducing surface drag on high speed train vehicles. The second benefit of this experimentation, is that the flaws and difficulties in measuring the surface drag on these sheet samples have been revealed. These flaws have now been overcome with a novel workholding solution. This equipment allows samples to be clamped down without damaging the profile or causing the sheet to deform but has enough rigidity to enable precise and repeatable boundary layer measurements. The force sensor assembly inside the vacuum chamber allows for the determination of the surface drag. A lot of literature published uses the profile data to predict the drag forces being applied to the surface. Unfortunately, this may not reliably predict the form drag contribution reliably, if there is any.

The high-speed open-loop wind tunnel should provide the basis for some interesting research, particularly with the three-dimensional surfaces. As per the findings of the literature review, few

6. Development of a Novel Experimental Apparatus for Analysing Drag-Reducing Benefits of Micro-Riblet Structured Sheets

3D surfaces have shown to be beneficial in terms of surface drag reduction. Most of the better performing structures, have so far, been two-dimensional. This simplification is really a result of the manufacturing limitations but some recommendations have been suggested in order to overcome such issues.

7. Conclusions and Suggestions for Future Work

7.1 Conclusions

The main purpose of this thesis was to explore the potential benefits of structural functional surfaces using facilities available within the University. The potential benefits of such surfaces were demonstrated by applying them to a particular engineering application. The main application in this thesis was to reduce the frictional losses in hydrodynamic bearings. Using the tribometer apparatus, it has been demonstrated that a textured surface can out-perform a surface without any micro-structure introduced onto it (provided that the texture configuration has been optimised). Textured component 4 presented the best performance with an 18.4% reduction in the coefficient of friction (COF) value, when compared against a component with no texture and similar average roughness (Sa) value. When the textured component 4 was compared against a component with a much lower Sa roughness value, it still outperformed with a 17.1% reduction in the COF value. As well as this, a preliminary investigation into the performance of micro-riblet structures for the purpose of reducing surface drag for application to high speed trains has been undertaken. Some initial wind tunnel testing was performed with rolled aluminium sheets. One of the sheets was exposed to a roller which had been textured using the turning process, resulting in a horizontal groove based micro-riblet structure. During these tests, the streamwise orientated riblets performed best and provided a 7.2% drag reduction at ≈40ms^{-1}. Issues were encountered with the experimental setup and equipment have been discussed in chapter 6. The issues identified were resolved with the development of a new high speed, open-loop wind tunnel with a novel workholding solution. The workholding solution allows different test sheets to be clamped down in any orientation without obstructing the flow. Whilst clamped down, a bespoke force lift/drag balance, incorporated within the workholding solution, allows for forces, resulting from the surface interacting with the flow, to be measured. Further work needs

to be carried out in order to prove the drag reductions seen and whether the surfaces perform better or worse at high-speed train velocities (above 100ms^{-1}).

The second aim of the project was to investigate the manufacturability of structured surfaces and develop a feasible manufacturing strategy using currently available University facilities. A substantial review of current technologies was presented in the literature review (chapter 2). The advantages and disadvantages for each of the manufacturing methods was summarised. Each method was also ranked in their current ability to mass manufacture textured surfaces (Table 2.3). The lowest ranked method was micro-milling and the best ranked method was rolling. One of the main problems with lower ranked processes was the speed and efficiency at which they produced a structured surface. The laser surface texturing (LST) method which was ranked 3rd, has the ability to structure a surface with a whole range of texture shapes and sizes. The repeatability and the accuracy of the process is also very good but its ability to structure large surface areas, cost effectively, prohibits it from being widely adopted and consequently, this is reflected by its ranking. Although, it was suggested that the rolling method, which is very cost-effective for large surface areas, should be combined with methods that have a lower productivity. This hybrid method would make use of a process like LST to create elaborate structures across the circumference of the roller component. To create the structure on the journal surface for the hydrodynamic bearing investigation it was decided to make use of a grinding technique that was demonstrated by Stepien (2007b). One of the texturing methods presented by Stepien produces a texture referred to as “type III”. The method for producing this particular surface structure was adapted in this investigation for the cylindrical grinding process (chapter 5). A small batch of textured components were produced with varying degrees of success. Some issues still need to be resolved with the process, mainly related to the texturing of the grinding wheel surface with the single point dressing technique. A more elaborate control system needs to be designed in order to have more control of the dressing tool movement. By improving the dressing control system over

the currently implemented one, more elaborate pattern configurations could be introduced with increased reliability. This is key in achieving a pattern that is closer to the optimised features, recommended by the numerical investigation (chapter 4). Another improvement is to have the ability of including dwell time within the cutting process. This will not only improve the roundness and sizing of the component but also improve the consistency of the textures across the component surface. In order to assess the quality of the structural functional surface, some measurable performance indicator (PI) needs to be used. The appropriate PI would enable the metrology technicians to assess the component (for potential frictional performance in a hydrodynamic bearing application). Extensive surface measurements (for roughness parameters and surface energy) have been carried out for both the non-textured samples (chapter 3) and the textured samples in chapter 5. Sku (kurtosis of the surface) has shown promise (for both non-textured and textured surfaces) as a PI with correlations and trends showing that minimising this parameter would decrease the COF for the bearing. Lowering the kurtosis of the surface reduces the *peakedness* of the surface. This would indicate that sharp, high gradient peaks are an unwanted feature. By spacing the peaks of the profile out, the gradient lowers, lowering the kurtosis of the profile. Literature has discussed (Nosonovsky and Bhushan (2008), Nosonovsky and Bhushan (2009)) the effect of surface wettability on the capillary adhesion force and the subsequent effect on the frictional forces between surfaces for bearing applications. Once the correct liquid combinations were used, the surface energy values determined in chapter 5 would indicate some correlation with the COF for each surface. Unfortunately, the amount of data generated by the small batch size is probably not enough to substantiate this as a reliable PI.

Another aim of this investigation was to demonstrate how a computational fluid dynamics (CFD) code could be used in order to predict the performance of structured surfaces. The open-source software, OpenFOAM®, was selected as the CFD solver. The CAD capabilities of SALOME-MECA was used to create the geometrical models of these surfaces. ParaView was then used to post-

process the data generated from the solver. These three software packages were combined through the use of a python (version 2.7) script developed during this investigation (chapter 4). An input matrix (containing the details of the different pattern configurations) can be submitted to the script and it automatically generates the CAD model, the mesh and the solution of the simulation. This script was used with the Taguchi optimisation technique and a set of direct effect charts were generated. This enabled for a best performing and worst performing surface to be designed. Very little drag reduction was observed for the worst performing (0.4% at 3um clearance). At the same clearance, the best performing surface was predicted to have a surface drag reduction of 16.6%. Unfortunately, one of the problems with the simulation was that no direct comparison could be made between what was simulated and the tribometer setup used. Although, some validation of the numerical investigation was performed. The validation that was carried out replaced the ellipsoidal textures with rectangular grooves by editing a single line of the script. The same meshing rules, which were determined by the mesh independence study, were used for this geometry. This simulation was then validated against well-established infinite Rayleigh step bearing theory. The results compared well, proving that the approach to the simulation setup had been correct.

It is hoped that the suggestions made for the improvement of the texturing method used (cylindrical grinding process) are implemented in the future and an alternative tribometer setup is used (such as pin-on-disk) which will allow for direct comparison between simulation and reality. Plenty of potential could be envisaged with the combination of the grinding technique with the rolling process described in chapter 6. Most drag reducing surfaces published in literature (see Table 2.1 and 2.2) have been two-dimensional structures which provide little comparison, geometrically, to naturally occurring surfaces, like shark skin. Time has been taken to develop the high-speed wind tunnel and also the instrumentation placed in the working section. This

apparatus should provide the foundation for the generation of high-quality experimental data with turbulent boundary layers over 3D micro-rib surfaces.

7.2 Future Work

Based on the conclusions of this investigation, the following points raise some potential for future research in the field of structural functional surfaces:

- In order to properly validate the full 3D Navier-Stokes simulations carried out in this thesis, changes should be made to the tribometer apparatus. A pin-on-disk tribometer setup would allow for direct comparisons to be made, due to the exclusion of the hydrodynamic lubrication mechanism and the resulting eccentricity, which is hard to both control and predict;
- Future grinding trials should make use of the suggestions in improving the cylindrical grinding process, mainly improving the control systems relating to the dressing tool movement and controlling the cutting process for the inclusion of a dwell period;
- Sku has proved to be a reliable performance indicator (PI) for both textured and non-textured components. Wettability has also showed some promise as a PI but limited batch sizes have resulted in claims that cannot be properly substantiated unless large datasets are generated. To achieve this, future components generated by further grinding trials should be measured using the static sessile technique described in this thesis but making sure the correct liquid combinations are used (see chapter 5). The use of alternative surface energy determination methods could also be explored. One suggestion would be to use an atomic force microscope (AFM) in conjunction with a Faraday cage to determine the changing adhesive forces between the surface and the AFM tip (example of this setup is seen in Awada et al. (2005));

- The wind tunnel apparatus developed during this investigation should be used to generate high-quality surface drag measurements and boundary layer profile data for the existing 2D riblet sheets. This then should be compared with CFD predictions. CFD predictions should be run using full three-dimensional large eddy simulations (LES). These will provide both spatial and temporal detail of the turbulent structures. Once validated, the simulation could then be used to then predict and optimise the riblet sizing for drag-reducing configurations;
- The textured grinding technique presented in this thesis should be used to texture the rollers for the forming process described earlier. Little literature has been presented in the fluid mechanics community which shows 3D structures with similar capabilities to that of nature. The arrangement of the micro-ribs that could be potentially manufactured onto the sheet surfaces and would potentially provide be a better comparison, geometrically, with the arrangement of denticles on naturally occurring sharkskin. The use of rolling to generate 3D structures is a very novel and cost effective technique that is worthy of further investigation.

References

ANDERSSON, P., KOSKINEN, J., VARJUS, S., GERBIG, Y., HAEFKE, H., GEORGIOU, S., ZHMUD, B. & BUSS, W. 2007. Microlubrication Effect by Laser-Textured Steel Surfaces. *Wear,* 262**,** 369-379.

ARNAL, D. & JUILLEN, J. C. 1978. Contribution experimentale a l'etude receptivite d'une couche limite laminaire, a la turbulence de l'ecoulement general. *ONERA Rapport Technique No. 1/5018*.

ARNELL, R. D., DAVIES, P. B., HALLING, J. & WHOMES, T. L. 1991. *Tribology: Principles amd Design Applications*, MacMillan Education Ltd.

AWADA, H., CASTELEIN, G. & BROGLY, M. 2005. Quantitative determination of surface energy using atomic force microscropy: the case of hydrophobic/hydrophobic contact and hydrophilic/hydrophilic contact. *Surface and Interface Analysis,* 37**,** 755-764.

BALL, P. 1999. Shark skin and other solutions. *Nature,* 400**,** 507-508.

BARON, A. & QUADRIO, M. 1993. Some preliminary results on the influence of riblets on the structure of a turbulent boundary layer. *Int. J. Heat and Fluid Flow,* 14**,** 223-230.

BECHERT, D. W., BRUSE, M. & HAGE, W. 2000. Experiments with Three-Dimensional Riblets as an Idealized Model of Shark Skin. *Experiments in Fluids,* 28**,** 403-412.

BECHERT, D. W., BRUSE, M., HAGE, W., VAN DER HOEVEN, J. G. T. & HOPPE, G. 1997. Experiments on drag-reducing surfaces and their optimization with an adjustable geometry. *Journal of Fluid Mechanics,* 338**,** 59-87.

BHUSHAN, B. & JUNG, Y. C. 2011. Natural and Biomimetic Artificial Surfaces for Superhydrophobicity, Self-Cleaning, Low Adhesion and Drag Reduction. *Progress in Materials Science,* 56**,** 1-108.

BIXLER, G. D. & BHUSHAN, B. 2013. Shark skin inspired low-drag microstructured surfaces in closed channel flow. *Journal of Colloid and Interface Science,* 393**,** 384-396.

BRADSHAW, P. & HUANG, G. P. 1995. The law of the wall in turbulent flow. *Proc.: Math. Phys. Sci.,* 451**,** 165-188.

BRANDNER, J. J., ANURJEW, E., BOHN, L., HANSJOSTEN, E., HENNING, T., SCHYGULLA, U., WENKA, A. & SCHUBERT, K. 2006. Concepts and Realization of Microstructure Heat Exchangers for Enhanced Heat Transfer. *Experimental Thermal and Fluid Science,* 30**,** 801-809.

References

BRAUN, D., GREINER, C., SCHNEIDER, J. & GUMBSCH, P. 2014. Efficiency of Laser Surface Texturing in the Reduction of Friction Under Mixed Lubrication. *Tribology International,* 77**,** 142-147.

BRITISH STANDARDS INSTITUTE 1996. BS ISO 8486-1:1996 Bonded abrasives - Determination and designation of grain size distribution - Part 1: Macrogrits F4 to F220. BSI Standards Publication.

BRITISH STANDARDS INSTITUTE 2013. BS ISO 525:2013 Bonded abrasive products - General requirements. BSI Standards Publication.

BRUZZONE, A. A. G. & COSTA, H. L. 2013. Functional Characterization of Structured Surfaces for Tribological Applications. *Procedia CIRP,* 12**,** 456-461.

BYRNE, G., DORNFIELD, D., INASAKI, I., KETTELER, G., KONIG, W. & TETI, R. 1995. Tool Condition Monitoring (TCM) - The Status of Research and Industrial Application. *CIRP Annals - Manufacturing Technology,* 44**,** 541-567.

CAMERON, A. 1981. Basic Lubrication Theory. London: Ellis Horwood Ltd.

CASTRO, I. P., SEGALINI, A. & ALFREDSSON, P. H. 2013. Outer-layer turbulence intensities in smooth- and rough-wall boundary layers. *Journal of Fluid Mechanics,* 727**,** 119-131.

CHATZIKYRIAKOU, D., BUONGIORNO, J., CAVIEZEL, D. & LAKEHAL, D. 2015. DNS and LES of turbulent flow in a closed channel featuring a pattern of hemispherical roughness elements. *Int. J. Heat and Fluid Flow,* 53**,** 29-43.

CHEN, H., RAO, F., SHANG, X., ZHANG, D. & HAGIWARA, I. 2013. Biomimetic drag reduction study on herringbone riblets of bird feather. *Journal of Bionic Engineering,* 10**,** 341-349.

CHEN, X. & ROWE, W. B. 1996. Analysis and Simulation of the Grinding Process. Part I: Generation of the Grinding Wheel Surface. *Int. J. Mach. Tools Manufact.,* 36**,** 871-882.

CHOI, H., MOIN, P. & KIM, J. 1993. Direct numerical simulation of turbulent flow over riblets. *Journal of Fluid Mechanics,* 255**,** 503-539.

CHOI, K.-S. 1989. Near-wall structure of a turbulent boundary layer with riblets. *Journal of Fluid Mechanics,* 208**,** 417-458.

CHU, D. C. & KARNIADAKIS, G. E. 1993. A direct numerical simulation of laminar and turbulent flow over riblet-mounted surfaces. *Journal of Fluid Mechanics,* 250**,** 1-42.

COLEBROOK, C. F. 1939. Turbulent Flow in Pipes, with particular reference to the Transition Region between the Smooth and Rough Pipe Laws. *Journal of the ICE,* 11**,** 133-156.

CUTHILL, E. & MCKEE, J. Reducing the bandwidth of sparse symmetric matrices. ACM Proceedings of the 1969 24th National Conference, 1969. 157-172.

DANIEL, T. L. 1981. Fish mucus: in situ measurements of polymer drag reduction. *Bio. Bull.,* 160**,** 376-382.

DEAN, B. & BHUSHAN, B. 2012. The Effect of Riblets in Rectangular Duct Flow. *Applied Surface Science,* 258**,** 3936-3947.

References

DENKENA, B., KÖHLER, J. & WANG, B. 2010. Manufacturing of functional riblet structures by profile grinding. *CIRP Journal of Manufacturing Science and Technology,* 3**,** 14-26.

DENKENA, B., LEON, L. D. & WANG, B. 2008. Grinding of Microstructured Functional Surfaces: A Novel Strategy for Dressing of Microprofiles. *Production Engineering: Research and Development,* 3**,** 41-48.

DRELICH, J. 2013. Guidelines to measurements of reproducible contact angles using a sessile-drop technique. *Surface Innovations,* 1**,** 248-254.

EL-SAMNI, O. A., CHUN, H. H. & YOON, H. S. 2007. Drag reduction of turbulent flow over thin rectangular riblets. *Int. J. Engineering Sci.,* 45**,** 436-454.

FROHNAPFEL, B. 2007. *Flow control of near-wall turbulence.* PhD Thesis, University of Erlangen-Nuremberg, Shaker Verlag, Aachen.

FROHNAPFEL, B., JOVANIC, J. & DELGADO, A. 2007. Experimental investigations of turbulent drag reduction by surface-embedded grooves. *Journal of Fluid Mechanics,* 590**,** 107-116.

GARCÍA-MAYORAL, R. & JIMÉNEZ, J. 2011. Hydrodynamic stability and breakdown of the viscous regime over riblets. *Journal of Fluid Mechanics,* 678**,** 317-347.

GOLDSTEIN, D. B. & TUAN, T.-C. 1998. Secondary flow induced by riblets. *Journal of Fluid Mechanics,* 363**,** 115-151.

GRUNEBERGER, R. & HAGE, W. 2011. Drag characteristics of longitudinal and transverse riblets at low dimensionless spacings. *Experiments in Fluids,* 50**,** 363-373.

HAGEN, G. Uber den Einfluss der Temperatur auf die Bewegung des Wassers in Rohren. Math. Abh. Akad. Wiss., 1854 Berlin. 17-98.

HERBERT, T. 1988. Secondary instability of boundary layers. *Annu. Rev. Fluid Mech.,* 20**,** 487-526.

HILGENBURG, K. & STEINHOFF, K. 2015. Texturing of skin-pass rolls by pulsed laser dispersing. *Journal of Materials Processing,* 225**,** 84-92.

HIRT, G. & THOME, M. 2008. Rolling of Functional Metallic Surface Structures. *CIRP Annals - Manufacturing Technology,* 57**,** 317-320.

HOLMBERG, K., ANDERSSON, P., NYLUND, N.-O., MÄKELÄ, K. & ERDEMIR, A. 2014. Global energy consumption due to friction in trucks and buses. *Tribology International,* 78**,** 94-114.

HUNT, J. C. R. & DURBIN, P. A. 1999. Perturbed shear layers. *Fluid. Dyn. Res.,* 24**,** 375-404.

HUNT, J. C. R., WRAY, A. A. & MOIN, P. 1988. Eddies, streams, and convergence zones in turbulent flows. *Proceedings of Summer program.* Centre for Turbulence Research, Stanford University.

JACOBS, R. G. & DURBIN, P. A. 2001. Simulations of bypass transition. *Journal of Fluid Mechanics,* 428**,** 185-212.

JUNG, Y. C. & BHUSHAN, B. 2006. Contact angle, adhesion and friction properties of micro- and nanopatterned polymers for superhydrophobicity. *Nanotechnology,* 17**,** 4970-4980.

References

JUNG, Y. C. & BHUSHAN, B. 2010. Biomimetic structures for fluid drag reduction in laminar and turbulent flows. *Journal of Physics: Condensed Matter,* 22**,** 035104 (9 pp.).

KENDALL, J. M. Studies on the laminar boundary layer receptivity to free stream turbulence near a leading edge. *In:* REDA, D. C., REED, H. L. & KOBAYASHI, R., eds. Bounary Layer Stability and Transition to Turblence, 1991. 22-30.

KIETZIG, A.-M., HATZIKIRIAKOS, S. G. & ENGLEZOS, P. 2009. Patterned superhydrophobic metallic surfaces. *Langmuir,* 25**,** 4821-4827.

KLEBANOFF, P. S., TIDSTROM, K. D. & SARGENT, L. M. 1962. The three-dimensional nature of boundary-layer instability. *Journal of Fluid Mechanics,* 138**,** 1-34.

KLOCKE, F. & FELDHAUS, B. 2007. Development of an incremental rolling process for the production of defined riblet surface structures. *Prod. Eng. Res. Devel. ,* 1**,** 233-237.

LANG, A., MOTTA, P. & HABEGGER, M. L. 2011. Shark skin separation control mechanisms. *Marine Tech. Soc.,* 45**,** 208-215.

LEE, L.-H. 1996. Correlation between Lewis Acid-Base Surface Interaction Compnents and Linear Solvation Energy Relationship Solvatochromic α and β Parameters. *Langmuir,* 12**,** 1681-1687.

LI, H. N. & AXINTE, D. 2016. Textured grinding wheels: A review. *International Journal of Machine Tools & Manufacture,* 109**,** 8-35.

LUO, Y., LIU, Y., ZHANG, D. & NG, E. Y. K. 2014. Influence of Morphology for Drag Reduction Effect of Sharkskin Surface. *Journal of Mechanics in Medicine and Biology,* 14**,** 1-16.

LUO, Y. & ZHANG, D. 2012. Experimental Research on Biomimetic Drag-Reducing Surface Application in Natural Gas Pipelines. *OIL GAS European Magazine,* 38**,** 213-214.

LUO, Y. & ZHANG, D. 2013. Investigation on Fabricating Continuous Vivid Sharkskin Surface by Bio-Replicated Rolling Method. *Applied Surface Science,* 282**,** 370-375.

MARINESCU, I. D., HITCHINER, M., UHLMANN, E., ROWE, W. B. & INASAKI, I. 2007. *Handbook of Machining with Grinding Wheel*, CRC Press.

MARTIN, S. & BHUSHAN, B. 2014. Fluid flow analysis of a shark-inspired microstructure. *Journal of Fluid Mechanics,* 756**,** 5-29.

MISHRA, S. P. & POLYCARPOU, A. A. 2011. Tribological Studies of Unpolished Laser Surface Textures Under Starved Lubrication Conditions for Use in Air-Conditioning and Refrigeration Compressors. *Tribology International,* 44**,** 1890-1901.

MOODY, L. F. 1944. Friction factors for pipe flow. *Trans. ASME,* 66**,** 671-684.

MORKOVIN, M. 1969. On the many faces of transition. Viscous Drag Reduction, September 24th and 25th 1968 LTV Research Center, Dallas, Texas. Springer Science, 1-31.

NIKURADSE, J. Strömungsgesetze in rauhen Rohren. V.D.I. Forschungsheft, 1933 Berlin. 1-22.

References

NOSONOVSKY, M. & BHUSHAN, B. 2008. Patterned Nonadhesive Surfaces: Superhydrophobicity and Wetting Regime Transitions. *Langmuir,* 24**,** 1525-1533.

NOSONOVSKY, M. & BHUSHAN, B. 2009. Multiscale Effects and Capillary Interactions in Functional Biomimetic Surfaces for Energy Conversion and Green Engineering. *Phil. Trans. Roy. Soc.,* 367**,** 1511-1539.

OEFFNER, J. & LAUDER, G. V. 2012. The hydrodynamic function of shark skin and two biomimetic applications. *Journal of Experimental Biology,* 215**,** 785-795.

ORLANDI, P., LEONARDI, S. & ANTONIA, R. A. 2006. Turbulent channel flow with either transverse or longitudinal roughness elements on one wall. *Journal of Fluid Mechanics,* 561**,** 279-305.

OWENS, D. K. & WENDT, R. C. 1969. Estimation of the free surface energy of polymers. *Journal of Applied Polymer Science,* 13**,** 1741-1747.

PAN, J., CHEN, H., ZHANG, D., ZHANG, X. & YUAN, L. 2013. Large-Scale Solvent-Swelling-Based Amplification of Microstructured Sharkskin. *Journal of Micromechanics and Microengineering,* 23**,** 1-9.

PANDE, S. J. & LAL, G. K. 1979. Effect of Dressing on Grinding Wheel Performance. *Int. J. MTDR,* 19**,** 171-179.

PARK, S. R. & WALLACE, J. M. 1994. Flow alteration and drag reduction by riblets in a turbulent boundary layer. *AIAA,* 32**,** 31-38.

PATANKAR, S. V. & SPALDING, D. B. 1972. A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-dimensional Parabolic Flows. *Int. J. Heat Mass Transfer,* 15**,** 1787.

PAWELSKI, O., RASP, W., ZWICK, W., NETTELBECK, H.-J. & STEINHOFF, K. 1994. The influence of different work-roll texturing systems on the development of surface structure in the temper rolling process of steel sheet used in the automotive industry. *J. Mater. Process. Technol.,* 45**,** 215-222.

PHILIPS, O. M. 1969. Shear-flow turbulence. *Annu. Rev. Fluid Mech.,* 1**,** 245-264.

PRANDTL, L. 1904. Über Flüssigkeitsbewegung bei sehr kleiner Reibung. *Proc. 3rd Int. Math. Congr. Heidelberg***,** 481-491.

RAMESH, A., AKRAM, W., MISHRA, S. P., CANNON, A. H., POLYCARPOU, A. A. & KING, W. P. 2013. Friction Characteristics of Microtextured Surfaces Under Mixed and Hydrodynamic Lubrication. *Tribology International,* 57**,** 170-176.

RAYLEIGH 1918. Notes on the Theory of Lubrication. *Phil. Mag.***,** 1-12.

REIF, W.-E. Squamation and the ecology of sharks. Courier Forschungsinstitut Senckenberg Nr.78, 1985 Frankfurt am Main.

REYNOLDS, O. 1883. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Phil. Trans. Roy. Soc.,* 174**,** 935-982.

References

REYNOLDS, O. 1886. On the Theory of Lubrication and Its Application to Mr. Beauchamp Tower's Experiments, Including an Experimental Determination of the Viscosity of Olive Oil. *Phil. Trans. Roy. Soc.,* 177**,** 157-234.

RICHARDSON, L. F. & GAUNT, J. A. 1927. The deferred approach to the limit. Part I. Single lattice. Part II. Interpenetrating lattices. *Phil. Trans. Roy. Soc.,* 226**,** 299-361.

ROACHE, P. J. 1994. Perspective: A method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering,* 116**,** 405-413.

ROACHE, P. J. Error Bars For CFD. AIAA 41st Aerospace Sciences Meeting, 2003 Reno, Nevada.

ROWE, W. B. 2009. *Principles of Modern Grinding Technology*, William Andrew - Elsevier.

SARAVI, S. S., CHENG, K., CHONG, T. P. & VATHYLAKIS, A. 2014. Design of Serrate-Semi-Circular Riblets with Application to Skin Friction Reduction on Engineering Surfaces. *Int. J. of Flow Control,* 6**,** 83-92.

SAYADI, T., HAMMAN, C. W. & MOIN, P. 2013. Direct numerical simulation of complete H-type and K-type transitions with implications for the dynamics of turbulent boundary layers. *Journal of Fluid Mechanics,* 724**,** 480-509.

SCHLATTER, P. & ÖRLÜ, R. 2012. Turbulent boundary layers at moderate Reynolds numbers: inflow length and tripping effects. *Journal of Fluid Mechanics,* 710**,** 5-34.

SCHLICHTING, H. 1979. Boundary layer theory. Seventh ed.: McGraw-Hill.

SCHOLZ, N. 1955. Strömungsvorgänge in Grenzschichten. *VDI-Ber.,* 9**,** 7-12.

SCHRECK, S. & ZUM GAHR, K.-H. 2005. Laser-Assisted Structuring of Ceramic and Steel Surfaces for Improving Tribological Properties. *Applied Surface Science,* 247**,** 616-622.

STACHOWIAK, G. W. & BATCHELOR, A. W. 1993. Chapter 10 Fundamentals of Contact Between Solids. *Tribology Series,* 24**,** 527-556.

STACHOWIAK, G. W. & BATCHELOR, A. W. 2006. *Engineering Tribology*, Butterworth Heinemann.

STENZEL, V., WILKE, Y. & HAGE, W. 2011. Drag-reducing paints for the reduction of fuel consumption in aviation and shipping. *Progress in Organic Coatings,* 70**,** 224-229.

STEPIEN, P. 2007a. Grinding forces in regular surface texture generation. *International Journal of Machine Tools & Manufacture,* 47**,** 2098-2110.

STEPIEN, P. 2007b. Undeformed chip sizes in grinding process of regular surface texture generation. *The archive of mechanical engineering,* 54**,** 236-258.

STEPIEN, P. 2008. Mechanism of grinding wheel surface reproduction in regular surface texture generation. *Surface Engineering,* 24**,** 219-225.

STEWART, R. W. 1968. Film Notes for TURBULENCE. *National Committee for Fluid Mechanics Films (MIT, Cambridge, Massachusetts)***,** 1-7.

References

SUTARDI & CHING, C. Y. 2003. Effect of different sized transverse square grooves on a turbulent boundary layer. *Experiments in Fluids,* 34**,** 261-274.

SYED, I. & SARANGI, M. 2014. Hydrodynamic lubrication with deterministic micro textures considering fluid inertia effect. *Tribology International,* 69**,** 30-38.

SZERI, A. Z. 1980. *Tribology: Friction, Lubrication and Wear*, McGraw Hill Higher Education.

TA, V. D., DUNN, A., WASLEY, T. J., LI, J., KAY, R. W., STRINGER, J., SMITH, P. J., ESENTURK, E., CONNAUGHTON, C. & SHEPHARD, J. D. 2016. Laser textured superhydrobic surfaces and their applications for homogeneous spot deposition. *Applied Surface Science,* 365**,** 153-159.

TAGUCHI, G. 1986. *Introduction to Quality Engineering: Designing Quality into Products and Processes*, Quality Resources.

TRITTON, D. J. 1988. Physical Fluid Dynamics. Second ed.: Oxford University Press.

VAN OSS, C. J., CHAUDHURY, M. K. & GOOD, R. J. 1988. Interfacial Lifshitz-van der Waals and polar interactions in macroscopic systems. *Chemical Review,* 88**,** 927-940.

VAN OSS, C. J., GOOD, R. J. & CHAUDHURY, M. K. 1986. The Role of van der Waals Forces and Hydrogen Bonds in "Hydrophobic Interactions" between Biopolymers and Low Energy Surfaces. *Journal of Colloid and Interface Science,* 111**,** 378-390.

VENKATESWARLU, K., RAO, N. J., VENUGOPAL, E. V. & AKELLA, S. 1990. Three-dimensional Laminar and Turbulent Lubrication in Journal Bearings. *Wear,* 136**,** 263-279.

VERSTEEG, H. K. & MALALASEKERA, W. 2007. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Pearson Education Limited.

WAKUDA, M., YAMAUCHI, Y., KANZAKI, S. & YASUDA, Y. 2003. Effect of Surface Texturing on Friction Reduction Between Ceramic and Steel Materials Under Lubricated Sliding Contact. *Wear,* 254**,** 356-363.

WALSH, M. J. Turbulent boundary layer drag reduction using riblets. 20th Aerospace Sciences Meeting, 1982 Orlando, FL, USA. Paper 82-0169.

WALSH, M. J. 1983. Riblets as a viscous drag reduction technique. *AIAA,* 21**,** 485-486.

WANG, X., KATO, K., ADACHI, K. & AIZAWA, K. 2001. The Effect of Laser Texturing of SiC Surface on the Critical Load for the Transition of Water Lubrication Mode from Hydrodynamic to Mixed. *Tribology International,* 34**,** 703-711.

WANG, X., KATO, K., ADACHI, K. & AIZAWA, K. 2003. Loads Carrying Capacity Map for the Surface Texture Design of SiC Thrust Bearing Sliding in Water. *Tribology International,* 36**,** 189-197.

WHARTON, J. T. & ALLANSON, D. R. 2015. *Tutorial 2: Numerical Diffusion - Computational Fluid Dynamics*, Liverpool John Moores University.

WHARTON, J. T., CHEN, X. & ALLANSON, D. R. 2014. Review of current structural functional surfaces manufacturing methods. *GERI Annual Research Symposium, June 2014*.

References

WHARTON, J. T., CHEN, X. & ALLANSON, D. R. 2016. A Numerical Investigation into the Frictional Performances of Ground, Micro-structured Surfaces. *Faculty of Engineering & Technology Research Week (9th - 13th May 2016).* Liverpool John Moores University.

WU, X. & MOIN, P. 2010. Transitional and turbulent boundary layer with heat transfer. *Physics of Fluids,* 22**,** 085105.

YAN, Y. Y., GAO, N. & BARTHLOTT, W. 2011. Mimicking natural superhydrophobic surfaces and grasping the wetting process: A review on recent progress in preparing superhydrophobic surfaces. *Advances in Colloid and Interface Science,* 169**,** 80-105.

YOUNG, T. 1805. An Essay on the Cohesion of Fluids. *Phil. Trans. Roy. Soc.,* 95**,** 65-87.

ZAKI, T. A. 2013. From Streaks to Spots and on to Turbulence: Exploring the Dynamics of Boundary Layer Transition. *Flow, Turbulence and Combustion,* 91**,** 451-473.

ZANOUN, E.-S. & DURST, F. 2003. Evaluating the law of the wall in two-dimensional fully developed turbulent channel flows. *Physics of Fluids,* 15**,** 3079-3089.

ZENKIEWICZ, M. 2007. Methods for the calculations of surface free energy of solids. *Journal of Achievements in Materials and Manufacturing Engineering,* 24**,** 137-145.

ZHANG, D., LUO, Y. & CHEN, H. 2011a. Application and Numerical Simulation Research on Biomimetic Drag-Reducing Technology for Gas Pipelining. *OIL GAS European Magazine,* 37**,** 85-90.

ZHANG, D., LUO, Y., LI, X. & CHEN, H. 2011b. Numerical Simulation and Experimental Study of Drag-Reducing Surface of a Real Shark Skin. *Journal of Hydrodynamics,* 23**,** 204-211.

ZHIQING, H., BO, W., WEI, W. & CHEN, X. 2014. Numerical simulation of microstructure on surface of roll forming sheet. *Forging & Stamping Technology,* 39.

ZUM GAHR, K.-H., WAHL, R. & WAUTHIER, K. 2009. Experimental Study of the Effect of Microtexturing on Oil Lubricated Ceramic/Steel Friction Pairs. *Wear,* 267**,** 1241-1251.

Appendices

Appendix A 2D Roughness Data (Chapter 3 Components)

Sample	Mean Roughness Measurement									
	Ra (µm)	Rsk	Rp (µm)	Rq (µm)	Rku	Rv (µm)	Rt (µm)	Rz (DIN) (µm)	Rz (µm)	Rpc (peaks/cm)
1B	0.206	-2.305	0.349	0.288	11.518	1.052	2.606	1.401	1.401	319.100
2B	0.210	-1.517	0.400	0.299	7.129	1.045	2.593	1.446	1.446	395.567
3B	0.218	-1.814	0.435	0.296	8.261	1.113	2.347	1.548	1.548	375.100
4B	0.267	-2.153	0.436	0.371	10.896	1.357	3.135	1.793	1.793	319.133
5B	0.161	-2.572	0.332	0.242	14.915	0.952	2.261	1.284	1.284	453.333
6B	0.270	-0.704	1.072	0.371	9.630	1.161	4.777	2.233	2.233	712.000
7B	0.168	-1.403	0.460	0.231	7.886	0.837	2.333	1.297	1.297	636.433
8B	0.173	-0.984	0.505	0.231	6.432	0.807	2.525	1.312	1.312	656.000
9B	0.224	-4.180	0.487	0.364	45.431	1.079	4.166	1.566	1.566	520.867
C1	0.180	-1.907	0.442	0.251	11.949	0.923	2.437	1.365	1.365	498.667
C2	0.081	-3.469	0.219	0.136	24.668	0.655	1.705	0.874	0.874	677.367
C3	0.208	-0.458	0.751	0.271	6.310	0.793	2.755	1.544	1.544	742.650
C4	0.307	-0.289	0.964	0.391	3.439	1.158	2.845	2.283	2.123	600.700

Appendix B 3D Height Roughness Data (Chapter 3 Components)

Sample	Mean Roughness Measurement						
	Sa (nm)	Sku	Sp (nm)	Sq (nm)	Ssk	Sv (nm)	Sz (nm)
1B	352.142	6.427	4622.09	482.776	-1.291	-4657.75	9279.85
2B	384.318	6.296	3424.21	514.451	-1.332	-4269.81	7694.01
3B	263.860	7.544	5063.94	361.935	-1.028	-3179.36	8243.30
4B	410.538	7.465	4474.79	557.822	-1.566	-4331.19	8805.98
6B	331.311	4.330	3109.62	432.742	-0.447	-4316.69	7426.32
7B	238.012	10.963	3199.11	327.105	-1.167	-4128.95	7328.06
8B	304.447	5.782	2302.94	409.755	-1.042	-3051.26	5354.20
9B	216.004	9.160	3556.59	301.618	-1.417	-4209.80	7766.39
C1	349.031	5.607	5094.47	465.276	-0.842	-3672.10	8766.58
C2	340.223	8.611	4607.15	470.095	-1.582	-5043.59	9650.74
C3	326.862	5.287	3443.57	435.065	-0.759	-5089.45	8533.02
C4	406.756	4.442	4033.56	530.733	-0.703	-3472.90	7506.46

Appendix C 3D Functional Roughness Data (Chapter 3 Components)

Sample	Mean Roughness Measurement				
	Sbi	Sc ($\mu m^3/\mu m^2$)	Sci	Sm ($\mu m^3/\mu m^2$)	Svi
1B	0.801	0.418	1.081	16.318	0.181
2B	0.816	0.447	1.060	14.068	0.169
3B	0.780	0.318	1.130	14.672	0.166
4B	0.887	0.457	0.963	12.434	0.183
6B	0.686	0.470	1.353	25.018	0.128
7B	0.712	0.322	1.284	14.841	0.145
8B	0.740	0.387	1.198	15.674	0.161
9B	0.816	0.250	1.071	11.472	0.169
C1	0.783	0.429	1.126	16.552	0.162
C2	0.858	0.382	0.997	14.985	0.178
C3	0.705	0.430	1.287	17.518	0.147
C4	0.710	0.533	1.275	21.373	0.143

Appendix D 3D Spatial and Hybrid Roughness Data (Chapter 3 Components)

Sample	Mean Roughness Measurement						
	Sal (µm)	Std (deg)	Str	Sdq (deg)	Sdr (%)	Sds (1/mm^2)	Ssc (1/ µm)
1B	47.163	88.060	0.231	16.242	3.918	13027.282	0.369
2B	43.358	89.557	0.144	18.398	5.127	12970.297	0.448
3B	22.358	88.482	0.088	17.471	4.561	13034.755	0.386
4B	36.018	89.143	0.102	16.821	4.175	11991.698	0.396
6B	5.506	87.953	0.030	25.161	9.969	12046.368	0.650
7B	12.222	86.328	0.097	18.171	4.934	13614.737	0.415
8B	31.307	88.528	0.135	17.894	4.830	13793.163	0.419
9B	13.352	88.693	0.087	15.682	3.723	13618.012	0.380
C1	25.074	89.451	0.079	20.299	6.171	13225.784	0.480
C2	9.733	89.657	0.044	21.053	6.751	13659.187	0.530
C3	15.428	88.305	0.096	19.549	5.864	14049.654	0.529
C4	6.503	86.723	0.019	21.577	7.003	13243.394	0.501

Appendix E Contact Angles from Wettability Study (Chapter 3 Components)

Sample	Contact Angle (deg) for Water				Contact Angle (deg) for Glycerol			
	Test 1	Test 2	Test 3	Average	Test 1	Test 2	Test 3	Average
1B	72.1	75.3	71.0	72.8	73.5	71.4	74.6	73.2
2B	78.2	74.4	78.9	77.2	66.1	60.1	61.7	62.6
3B	74.5	62.1	66.6	67.7	61.5	66.3	67.2	65.0
4B	72.8	66.5	67.8	69.0	61.9	67.4	66.4	65.2
6B	79.3	80.1	76.2	78.5	73.0	74.2	77.5	74.9
7B	73.1	70.4	70.5	71.3	74.3	78.1	68.9	73.8
8B	60.3	62.9	65.2	62.8	66.4	60.3	60.0	62.2
9B	81.9	81.4	79.9	81.1	64.2	67.7	65.8	65.9
C1	75.7	66.5	68.6	70.3	77.4	79.1	78.4	78.3
C2	67.1	68.4	69.7	68.4	68.6	70.4	73.0	70.7
C3	77.1	80.2	80.5	79.3	70.4	66.7	77.0	71.4
C4	91.6	85.2	87.3	88.0	72.1	73.3	66.2	70.5

Appendix F Calculated Free Surface Energies (Chapter 3)

Component	Free Surface Energy (mN/m)		
	Polar	Dispersive	Total
1B	26.3	5.1	31.4
2B	6.6	31.3	37.9
3B	24.9	9.8	34.7
4B	22.5	11.1	33.6
6B	16.9	9.3	26.2
7B	30.7	3.3	34.0
8B	31.5	7.7	39.2
9B	4.7	32.4	37.1
C1	42.3	0.3	42.6
C2	32.4	3.8	36.2
C3	11.6	16.4	28.0
C4	1.6	37.6	39.2

Appendix G Coefficient of Friction Results (Chapter 3 Components)

Sample	Coefficient of Friction at Different Speeds			
	500rpm	1000rpm	1500rpm	1850rpm
1B	0.177	0.230	0.252	0.242
2B	0.140	0.155	0.223	0.226
3B	0.174	0.238	0.268	0.253
4B	0.151	0.226	0.223	0.249
5B	0.158	0.208	0.219	0.215
6B	0.128	0.219	0.192	0.253
7B	0.158	0.234	0.253	0.260
8B	0.136	0.223	0.249	0.242
9B	0.170	0.257	0.294	0.275
C1	0.158	0.208	0.223	0.238
C2	0.121	0.181	0.234	0.272
C3	0.098	0.158	0.200	0.264
C4	0.158	0.223	0.245	0.249

Appendix H Correlation Coefficient Data (Chapter 3 Components)

Parameter	Pearson's Correlation Coefficient			
	COF (at 500rpm)	COF (at 1000rpm)	COF (at 1500rpm)	COF (at 1850rpm)
Surface Energy	0.370	0.094	0.435	-0.302
Sa	-0.222	-0.462	-0.589	-0.551
Sku	0.310	0.320	0.582	0.488
Sp	0.416	0.078	0.111	-0.016
Sq	-0.198	-0.456	-0.551	-0.534
Ssk	-0.236	-0.044	-0.448	-0.154
Sv	0.480	0.486	0.386	-0.390
Sz	0.039	-0.225	-0.141	0.216
Sbi	0.242	-0.011	0.249	-0.006
Sc	-0.324	-0.427	-0.715	-0.521
Sci	-0.278	0.020	-0.305	0.056
Sm	-0.305	-0.102	-0.571	-0.120
Svi	0.322	0.018	0.375	-0.085
Sal	0.290	-0.125	0.046	-0.694
Std	-0.209	-0.409	-0.160	-0.158
Str	0.303	-0.002	0.209	-0.437
Sdq	-0.513	-0.312	-0.678	0.010
Sdr	-0.500	-0.268	-0.668	0.036
Sds	-0.216	-0.198	0.345	0.357
Ssc	-0.691	-0.448	-0.755	0.121
Ra	0.241	0.292	-0.108	-0.240
Rsk	-0.350	-0.284	-0.568	-0.438
Rp	-0.277	0.035	-0.422	0.025
Rq	0.316	0.373	0.014	-0.138
Rku	0.198	0.323	0.536	0.626
Rv	0.481	0.420	0.033	-0.284
Rt	0.016	0.344	-0.117	0.158
Rz(DIN)	0.081	0.246	-0.246	-0.135
Rz	0.069	0.250	-0.273	-0.136
RPc	-0.700	-0.254	-0.331	0.469

Appendix I compareStudy2.py Python Script

```
00000001 import numpy as np
00000002 import math
00000003 import fileinput
00000004 import subprocess
00000005 import os
00000006
00000007 inputArray = np.genfromtxt('inputArray.csv', delimiter=',',
dtype=None)
00000008
00000009 noOfStudies = 1 # Enter the number of trials
00000010 study = 0
00000011
00000012 while study < noOfStudies:
00000013     xincrementDist = inputArray[study][4]
00000014     yincrementDist = inputArray[study][5]
00000015     xSize = inputArray[study][1]
00000016     ySize = inputArray[study][2]
00000017     zSize = inputArray[study][3]
00000018     radialClearance = inputArray[study][7]
00000019     staggerProp = inputArray[study][6]
00000020     staggerDist = staggerProp * yincrementDist
00000021     angleDeg = inputArray[study][0]
00000022     angle = math.radians(angleDeg)
00000023
00000024     os.system("python bearing_program.py %s %s %s %s %s %s %s %s"
% (xincrementDist, yincrementDist, xSize, ySize, zSize, radialClearance,
staggerDist, angle))
00000025
00000026     os.system("mkdir study%s" % (study))
00000027     os.system("cp -R ./[0-9]* ./study%s/" % (study))
00000028     os.system("rm -R [0-9]*")
00000029     os.system("cp -R ./constant ./study%s/constant" % (study))
00000030     os.system("cp -R ./system ./study%s/system" % (study))
00000031     os.system("cp -R ./log ./study%s/log" % (study))
00000032     os.system("cp -R ./bearingStagger2.py
./study%s/bearingStagger2.py" % (study))
00000033     os.system("cp -R ./postProcessing ./study%s/postProcessing" %
(study))
00000034     os.system("rm -R constant")
00000035     os.system("rm -R system")
00000036     os.system("rm -R bearingStagger2.py")
00000037     os.system("rm -R log")
00000038     os.system("rm -R postProcessing")
00000039
00000040     study += 1
```

Appendix J bearing_program.py Python Script

```
00000001 import fileinput
00000002 import subprocess
00000003 import os
00000004 import math
00000005 import argparse
00000006 import numpy as np
00000007
00000008 parser = argparse.ArgumentParser()
00000009 parser.add_argument("xSpacing", type=float)
00000010 parser.add_argument("ySpacing", type=float)
00000011 parser.add_argument("length", type=float)
00000012 parser.add_argument("width", type=float)
00000013 parser.add_argument("depth", type=int)
00000014 parser.add_argument("clearance", type=float)
00000015 parser.add_argument("stagDist", type=float)
00000016 parser.add_argument("angle", type=float)
00000017 args = parser.parse_args()
00000018
00000019 # Setup parameters
00000020
00000021 xQuantity = 2 # number of pockets in streamwise direction
00000022 yQuantity = 2 # number of pockets in spanwise direction
00000023 xincrementDist = args.xSpacing # distance between pockets in
streamwise (micron)
00000024 yincrementDist = args.ySpacing # distance between pockets in
spanwise (micron)
00000025 xSize = args.length # streamwise size of pocket (micron)
00000026 ySize = args.width # spanwise size of pocket (micron)
00000027 zSize = args.depth # depth of pocket (micron)
00000028 radialClearance = args.clearance # clearence between journal and
bearing surface (micron)
00000029
00000030 staggerDist = args.stagDist # offset for staggered rows, ignore
if false (micron)
00000031 speed = 5 # Enter the linear operational speed (m/s)
00000032
00000033 angle = args.angle # Enter the angle of the taxture between 0
and 1.57 radians
00000034 procMesh = 23 # Enter the number of processors for the mesh
generation process
00000035 procSolve = 23 # Enter the number of processors for the solving
process
00000036 textRefine = 17 #Enter the number of cells present across the
depth of the pocket
00000037
00000038 if angle == 0:
00000039    anglePattern = False
00000040 else:
00000041    anglePattern = True
00000042
00000043 if staggerDist == 0:
00000044    staggeredGrid = False
00000045 else:
00000046    staggeredGrid = True
00000047
00000048 print anglePattern
00000049 print staggerDist
```

```
00000050 print yincrementDist
00000051 print (ySize/2)
00000052 print staggeredGrid
00000053
00000054 # Copy the source salome setup file to the working directory
00000055 os.system("cp -R ./source/bearingStagger2.py
./bearingStagger2.py")
00000056
00000057 # Copies the source constant structure
00000058 os.system("cp -R ./source/constant ./constant")
00000059
00000060 # Copies the source system structure
00000061 os.system("cp -R ./source/system ./system")
00000062
00000063
00000064 # Process required to change all the variables in the salome
setup file
00000065
00000066 pythonOrigLoc = './bearingStagger2.py'
00000067 pythonTmp = './bearingStagger2new.py'
00000068
00000069 blockMeshLoc = './system/blockMeshDict'
00000070 blockMeshLocTmp = './system/blockMeshDictTmp'
00000071
00000072 oldLine1 = 'xQuantity = changeNo1'
00000073 newLine1 = 'xQuantity = %s\n' % (xQuantity)
00000074 oldLine2 = 'yQuantity = changeNo2'
00000075 newLine2 = 'yQuantity = %s\n' % (yQuantity)
00000076 oldLine3 = 'xincrementDist = changeNo3'
00000077 newLine3 = 'xincrementDist = %s\n' % (xincrementDist)
00000078 oldLine4 = 'yincrementDist = changeNo4'
00000079 newLine4 = 'yincrementDist = %s\n' % (yincrementDist)
00000080 oldLine5 = 'xSize = changeNo5'
00000081 newLine5 = 'xSize = %s\n' % (xSize)
00000082 oldLine6 = 'ySize = changeNo6'
00000083 newLine6 = 'ySize = %s\n' % (ySize)
00000084 oldLine7 = 'zSize = changeNo7'
00000085 newLine7 = 'zSize = %s\n' % (zSize)
00000086 oldLine8 = 'radialClearance = changeNo8'
00000087 newLine8 = 'radialClearance = %s\n' % (radialClearance)
00000088 oldLine9 = 'staggeredGrid = changeNo9'
00000089 newLine9 = 'staggeredGrid = %s\n' % (staggeredGrid)
00000090 oldLine10 = 'staggerDist = changeNo10'
00000091 newLine10 = 'staggerDist = %s\n' % (staggerDist)
00000092 oldLine11 = 'anglePattern = changeNo11'
00000093 newLine11 = 'anglePattern = %s\n' % (anglePattern)
00000094 oldLine12 = 'angle = changeNo12'
00000095 newLine12 = 'angle = %s\n' % (angle)
00000096
00000097 xAngleCorrect = (xSize/2)*(math.cos(angle))
00000098 yAngleCorrect = (xSize/2)*(math.sin(angle))
00000099
00000100
00000101 if anglePattern == True:
00000102    minVertexX = -(ySize/2)
00000103    if staggeredGrid == True:
00000104        maxVertexX = (xQuantity * xincrementDist) - (ySize/2)
00000105        maxVertexY = ((yQuantity -1) * yincrementDist) +
yAngleCorrect + staggerDist
00000106        minVertexY = -yAngleCorrect
00000107    else:
```

```
00000108         maxVertexX = ((xQuantity) * xincrementDist) - (ySize/2)
00000109         maxVertexY = ((yQuantity - 1) * yincrementDist) +
yAngleCorrect
00000110         minVertexY = -yAngleCorrect
00000111 else:
00000112     minVertexX = 0
00000113     maxVertexX = (xQuantity * xincrementDist)
00000114     minVertexY = -(ySize/2)
00000115     if staggeredGrid == True:
00000116         maxVertexY = ((yQuantity - 1) * yincrementDist) +
(ySize/2) + staggerDist
00000117     else:
00000118         maxVertexY = (yQuantity * yincrementDist) - (ySize/2)
00000119
00000120 vertex0Old = '(x1 y1 z1)'
00000121 vertex0New = '(%s %s -%s)\n' % (minVertexX, minVertexY, zSize)
00000122 vertex1Old = '(x2 y1 z1)'
00000123 vertex1New = '(%s %s -%s)\n' % (maxVertexX, minVertexY, zSize)
00000124 vertex2Old = '(x2 y2 z1)'
00000125 vertex2New = '(%s %s -%s)\n' % (maxVertexX, maxVertexY, zSize)
00000126 vertex3Old = '(x1 y2 z1)'
00000127 vertex3New = '(%s %s -%s)\n' % (minVertexX, maxVertexY, zSize)
00000128 vertex4Old = '(x1 y1 z2)'
00000129 vertex4New = '(%s %s %s)\n' % (minVertexX, minVertexY,
radialClearance)
00000130 vertex5Old = '(x2 y1 z2)'
00000131 vertex5New = '(%s %s %s)\n' % (maxVertexX, minVertexY,
radialClearance)
00000132 vertex6Old = '(x2 y2 z2)'
00000133 vertex6New = '(%s %s %s)\n' % (maxVertexX, maxVertexY,
radialClearance)
00000134 vertex7Old = '(x1 y2 z2)'
00000135 vertex7New = '(%s %s %s)\n' % (minVertexX, maxVertexY,
radialClearance)
00000136
00000137 # Writing separation vector values to create patch dict file
00000138
00000139 symSepDist = maxVertexY - minVertexY
00000140
00000141 xCyclicDist = maxVertexX - minVertexX
00000142
00000143 print maxVertexY
00000144 print minVertexY
00000145 print symSepDist
00000146
00000147 sepVect1Old = 'separationVector (xDist1 0 0);'
00000148 sepVect1New = '            separationVector (%se-6 0 0);\n' %
(xCyclicDist)
00000149 sepVect2Old = 'separationVector (xDist2 0 0);'
00000150 sepVect2New = '            separationVector (-%se-6 0 0);\n' %
(xCyclicDist)
00000151 sepVect3Old = 'separationVector (0 xDist3 0);'
00000152 sepVect3New = '            separationVector (0 %se-6 0);\n' %
(symSepDist)
00000153 sepVect4Old = 'separationVector (0 xDist4 0);'
00000154 sepVect4New = '            separationVector (0 -%se-6 0);\n' %
(symSepDist)
00000155
00000156 patchDictLoc = './system/createPatchDict'
00000157 patchDictLocTmp = './system/createPatchDictTmp'
00000158
```

```
00000159 with open(patchDictLoc, 'r') as input_file,
open(patchDictLocTmp, 'w') as output_file:
00000160     for line in input_file:
00000161         if line.strip() == sepVect1Old :
00000162             output_file.write(sepVect1New)
00000163         else:
00000164             output_file.write(line)
00000165
00000166 os.system('rm -R %s' % (patchDictLoc))
00000167
00000168 os.system('mv %s %s' % (patchDictLocTmp, patchDictLoc))
00000169
00000170 with open(patchDictLoc, 'r') as input_file,
open(patchDictLocTmp, 'w') as output_file:
00000171     for line in input_file:
00000172         if line.strip() == sepVect2Old :
00000173             output_file.write(sepVect2New)
00000174         else:
00000175             output_file.write(line)
00000176
00000177 os.system('rm -R %s' % (patchDictLoc))
00000178
00000179 os.system('mv %s %s' % (patchDictLocTmp, patchDictLoc))
00000180
00000181 with open(patchDictLoc, 'r') as input_file,
open(patchDictLocTmp, 'w') as output_file:
00000182     for line in input_file:
00000183         if line.strip() == sepVect3Old :
00000184             output_file.write(sepVect3New)
00000185         else:
00000186             output_file.write(line)
00000187
00000188 os.system('rm -R %s' % (patchDictLoc))
00000189
00000190 os.system('mv %s %s' % (patchDictLocTmp, patchDictLoc))
00000191
00000192 with open(patchDictLoc, 'r') as input_file,
open(patchDictLocTmp, 'w') as output_file:
00000193     for line in input_file:
00000194         if line.strip() == sepVect4Old :
00000195             output_file.write(sepVect4New)
00000196         else:
00000197             output_file.write(line)
00000198
00000199 os.system('rm -R %s' % (patchDictLoc))
00000200
00000201 os.system('mv %s %s' % (patchDictLocTmp, patchDictLoc))
00000202
00000203 # Writing parameters to geometry python build file
00000204 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000205     for line in input_file:
00000206         if line.strip() == oldLine1 :
00000207             output_file.write(newLine1)
00000208         else:
00000209             output_file.write(line)
00000210
00000211 os.system('rm -R %s' % (pythonOrigLoc))
00000212
00000213 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000214
```

```
00000215 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000216     for line in input_file:
00000217         if line.strip() == oldLine2 :
00000218             output_file.write(newLine2)
00000219         else:
00000220             output_file.write(line)
00000221
00000222 os.system('rm -R %s' % (pythonOrigLoc))
00000223
00000224 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000225
00000226 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000227     for line in input_file:
00000228         if line.strip() == oldLine3 :
00000229             output_file.write(newLine3)
00000230         else:
00000231             output_file.write(line)
00000232
00000233 os.system('rm -R %s' % (pythonOrigLoc))
00000234
00000235 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000236
00000237 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000238     for line in input_file:
00000239         if line.strip() == oldLine4 :
00000240             output_file.write(newLine4)
00000241         else:
00000242             output_file.write(line)
00000243
00000244 os.system('rm -R %s' % (pythonOrigLoc))
00000245
00000246 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000247
00000248 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000249     for line in input_file:
00000250         if line.strip() == oldLine5 :
00000251             output_file.write(newLine5)
00000252         else:
00000253             output_file.write(line)
00000254
00000255 os.system('rm -R %s' % (pythonOrigLoc))
00000256
00000257 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000258
00000259 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000260     for line in input_file:
00000261         if line.strip() == oldLine6 :
00000262             output_file.write(newLine6)
00000263         else:
00000264             output_file.write(line)
00000265
00000266 os.system('rm -R %s' % (pythonOrigLoc))
00000267
00000268 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000269
```

```
00000270 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000271     for line in input_file:
00000272         if line.strip() == oldLine7 :
00000273             output_file.write(newLine7)
00000274         else:
00000275             output_file.write(line)
00000276
00000277 os.system('rm -R %s' % (pythonOrigLoc))
00000278
00000279 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000280
00000281 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000282     for line in input_file:
00000283         if line.strip() == oldLine8 :
00000284             output_file.write(newLine8)
00000285         else:
00000286             output_file.write(line)
00000287
00000288 os.system('rm -R %s' % (pythonOrigLoc))
00000289
00000290 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000291
00000292 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000293     for line in input_file:
00000294         if line.strip() == oldLine9 :
00000295             output_file.write(newLine9)
00000296         else:
00000297             output_file.write(line)
00000298
00000299 os.system('rm -R %s' % (pythonOrigLoc))
00000300
00000301 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000302
00000303 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000304     for line in input_file:
00000305         if line.strip() == oldLine10 :
00000306             output_file.write(newLine10)
00000307         else:
00000308             output_file.write(line)
00000309
00000310 os.system('rm -R %s' % (pythonOrigLoc))
00000311
00000312 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000313
00000314 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000315     for line in input_file:
00000316         if line.strip() == oldLine11 :
00000317             output_file.write(newLine11)
00000318         else:
00000319             output_file.write(line)
00000320
00000321 os.system('rm -R %s' % (pythonOrigLoc))
00000322
00000323 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000324
```

```
00000325 with open(pythonOrigLoc, 'r') as input_file, open(pythonTmp,
'w') as output_file:
00000326     for line in input_file:
00000327         if line.strip() == oldLine12 :
00000328             output_file.write(newLine12)
00000329         else:
00000330             output_file.write(line)
00000331
00000332 os.system('rm -R %s' % (pythonOrigLoc))
00000333
00000334 os.system('mv %s %s' % (pythonTmp, pythonOrigLoc))
00000335
00000336 # Writing to blockMesh file
00000337 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000338     for line in input_file:
00000339         if line.strip() == vertex0Old :
00000340             output_file.write(vertex0New)
00000341         else:
00000342             output_file.write(line)
00000343
00000344 os.system('rm -R %s' % (blockMeshLoc))
00000345
00000346 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000347
00000348 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000349     for line in input_file:
00000350         if line.strip() == vertex1Old :
00000351             output_file.write(vertex1New)
00000352         else:
00000353             output_file.write(line)
00000354
00000355 os.system('rm -R %s' % (blockMeshLoc))
00000356
00000357 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000358
00000359 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000360     for line in input_file:
00000361         if line.strip() == vertex2Old :
00000362             output_file.write(vertex2New)
00000363         else:
00000364             output_file.write(line)
00000365
00000366 os.system('rm -R %s' % (blockMeshLoc))
00000367
00000368 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000369
00000370 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000371     for line in input_file:
00000372         if line.strip() == vertex3Old :
00000373             output_file.write(vertex3New)
00000374         else:
00000375             output_file.write(line)
00000376
00000377 os.system('rm -R %s' % (blockMeshLoc))
00000378
00000379 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000380
```

```
00000381 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000382     for line in input_file:
00000383         if line.strip() == vertex4Old :
00000384             output_file.write(vertex4New)
00000385         else:
00000386             output_file.write(line)
00000387
00000388 os.system('rm -R %s' % (blockMeshLoc))
00000389
00000390 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000391
00000392 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000393     for line in input_file:
00000394         if line.strip() == vertex5Old :
00000395             output_file.write(vertex5New)
00000396         else:
00000397             output_file.write(line)
00000398
00000399 os.system('rm -R %s' % (blockMeshLoc))
00000400
00000401 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000402
00000403 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000404     for line in input_file:
00000405         if line.strip() == vertex6Old :
00000406             output_file.write(vertex6New)
00000407         else:
00000408             output_file.write(line)
00000409
00000410 os.system('rm -R %s' % (blockMeshLoc))
00000411
00000412 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000413
00000414 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000415     for line in input_file:
00000416         if line.strip() == vertex7Old :
00000417             output_file.write(vertex7New)
00000418         else:
00000419             output_file.write(line)
00000420
00000421 os.system('rm -R %s' % (blockMeshLoc))
00000422
00000423 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000424
00000425 # calculation of blockMesh grading and topoSet settings
00000426
00000427 zSizeAbs = np.multiply(zSize,0.000001)
00000428 radialClearanceAbs = np.multiply(radialClearance,0.000001)
00000429
00000430 zCellSize = np.multiply(4,np.divide(zSizeAbs,textRefine))
00000431 Nz =
int(np.ceil(np.divide(np.add(radialClearanceAbs,zSizeAbs),zCellSize)))
00000432 xyCellSize = 2 * zCellSize
00000433 Nx =
int(np.ceil(np.divide(np.multiply(xCyclicDist,0.000001),xyCellSize)))
00000434 Ny =
int(np.ceil(np.divide(np.multiply(symSepDist,0.000001),xyCellSize)))
```

```
00000435
00000436 meshGradOld = 'hex (0 1 2 3 4 5 6 7) (x y z) simpleGrading (1 1
1)'
00000437 meshGradNew = '    hex (0 1 2 3 4 5 6 7) (%s %s %s)
simpleGrading (1 1 1)\n' % (Nx, Ny, Nz)
00000438
00000439 with open(blockMeshLoc, 'r') as input_file,
open(blockMeshLocTmp, 'w') as output_file:
00000440     for line in input_file:
00000441         if line.strip() == meshGradOld :
00000442             output_file.write(meshGradNew)
00000443         else:
00000444             output_file.write(line)
00000445
00000446 os.system('rm -R %s' % (blockMeshLoc))
00000447
00000448 os.system('mv %s %s' % (blockMeshLocTmp, blockMeshLoc))
00000449
00000450
00000451 topoSet1z = np.divide(radialClearance,2)
00000452 topoSet2z = 0
00000453
00000454 topoSet1Old = 'box (x1 y1 z1) (x2 yz z2);'
00000455 topoSet1New = '            box (%se-6 %se-6 -%se-6) (%se-6 %se-6
%se-6);\n' % (minVertexX, minVertexY, zSize, maxVertexX, maxVertexY,
topoSet1z)
00000456 topoSet2Old = 'box (x1 y1 z1) (x2 yz z2);'
00000457 topoSet2New = '            box (%se-6 %se-6 -%se-6) (%se-6 %se-6
%se-6);\n' % (minVertexX, minVertexY, zSize, maxVertexX, maxVertexY,
topoSet2z)
00000458
00000459
00000460 topoSet1Loc = './system/topoSetDict.1'
00000461 topoSet1LocTmp = './system/topoSetDictTmp.1'
00000462 topoSet2Loc = './system/topoSetDict.2'
00000463 topoSet2LocTmp = './system/topoSetDictTmp.2'
00000464
00000465
00000466 with open(topoSet1Loc, 'r') as input_file, open(topoSet1LocTmp,
'w') as output_file:
00000467     for line in input_file:
00000468         if line.strip() == topoSet1Old :
00000469             output_file.write(topoSet1New)
00000470         else:
00000471             output_file.write(line)
00000472
00000473 os.system('rm -R %s' % (topoSet1Loc))
00000474
00000475 os.system('mv %s %s' % (topoSet1LocTmp, topoSet1Loc))
00000476
00000477 with open(topoSet2Loc, 'r') as input_file, open(topoSet2LocTmp,
'w') as output_file:
00000478     for line in input_file:
00000479         if line.strip() == topoSet2Old :
00000480             output_file.write(topoSet2New)
00000481         else:
00000482             output_file.write(line)
00000483
00000484 os.system('rm -R %s' % (topoSet2Loc))
00000485
00000486 os.system('mv %s %s' % (topoSet2LocTmp, topoSet2Loc))
```

```
00000487
00000488
00000489 # Write to parameters to snappyHexMeshDict
00000490
00000491 snappyLoc = './system/snappyHexMeshDict'
00000492 snappyLocTmp = './system/snappyHexMeshDictTmp'
00000493
00000494 searchMinOld = 'min (x1 y1 z1);'
00000495 searchMinNew = '        min (0 -%se-6 -%se-6);\n' % ((ySize/2),
zSize)
00000496 searchMaxOld = 'max (x2 y2 z2);'
00000497 searchMaxNew = '        max (%se-6 %se-6 0);\n' % (maxVertexX,
maxVertexY)
00000498 locInMeshOld = 'locationInMesh (x y z);'
00000499 locInMeshNew = '    locationInMesh (%se-6 %se-6 %se-6);\n'%
((maxVertexX/2), (maxVertexY/2), (radialClearance/2))
00000500
00000501 with open(snappyLoc, 'r') as input_file, open(snappyLocTmp, 'w')
as output_file:
00000502     for line in input_file:
00000503         if line.strip() == searchMinOld :
00000504             output_file.write(searchMinNew)
00000505         else:
00000506             output_file.write(line)
00000507
00000508 os.system('rm -R %s' % (snappyLoc))
00000509
00000510 os.system('mv %s %s' % (snappyLocTmp, snappyLoc))
00000511
00000512 with open(snappyLoc, 'r') as input_file, open(snappyLocTmp, 'w')
as output_file:
00000513     for line in input_file:
00000514         if line.strip() == searchMaxOld :
00000515             output_file.write(searchMaxNew)
00000516         else:
00000517             output_file.write(line)
00000518
00000519 os.system('rm -R %s' % (snappyLoc))
00000520
00000521 os.system('mv %s %s' % (snappyLocTmp, snappyLoc))
00000522
00000523 with open(snappyLoc, 'r') as input_file, open(snappyLocTmp, 'w')
as output_file:
00000524     for line in input_file:
00000525         if line.strip() == locInMeshOld :
00000526             output_file.write(locInMeshNew)
00000527         else:
00000528             output_file.write(line)
00000529
00000530 os.system('rm -R %s' % (snappyLoc))
00000531
00000532 os.system('mv %s %s' % (snappyLocTmp, snappyLoc))
00000533
00000534 # Write parameters for mesh decomposePar
00000535
00000536 decompMesh = './system/decomposeParDict'
00000537 decompMeshTmp = './system/decomposeParDictTmp'
00000538
00000539 decompMeshLineOld = 'numberOfSubdomains procs;'
00000540 decompMeshLineNew = 'numberOfSubdomains %s;\n' % (procMesh)
00000541
```

```
00000542 with open(decompMesh, 'r') as input_file, open(decompMeshTmp,
'w') as output_file:
00000543     for line in input_file:
00000544         if line.strip() == decompMeshLineOld :
00000545             output_file.write(decompMeshLineNew)
00000546         else:
00000547             output_file.write(line)
00000548
00000549 os.system('rm -R %s' % (decompMesh))
00000550
00000551 os.system('mv %s %s' % (decompMeshTmp, decompMesh))
00000552
00000553
00000554 # Calls salome in the terminal and performs the bearing geometry
creations
00000555 subprocess.call('/home/david/salome_meca/appli_V2015_2/salome -t
python bearingStagger2.py --ns-port-log=salomePort.txt', shell=True)
00000556
00000557 # Copies the stl file to the trisurface folder
00000558 os.system("cp -R bearingSurface.stl
./constant/triSurface/bearingSurface.stl")
00000559 os.system("rm -R bearingSurface.stl")
00000560
00000561 # Extracts the surfaces off the stl file
00000562 os.system("surfaceFeatureExtract")
00000563
00000564 # Creates the original domain before bearing geometry is used to
refine
00000565 os.system("blockMesh")
00000566
00000567 meshRefInc = 1
00000568 meshRefEnd = 3
00000569
00000570 os.system("decomposePar")
00000571
00000572 while meshRefInc < meshRefEnd:
00000573     os.system("foamJob -p -s topoSet -dict system/topoSetDict.%s"
% (meshRefInc))
00000574     os.system("foamJob -p -s refineHexMesh c0 -overwrite")
00000575     meshRefInc += 1
00000576
00000577
00000578 os.system("foamJob -p -s snappyHexMesh")
00000579
00000580 os.system("reconstructParMesh -time 2")
00000581
00000582 #Copies the snappyHexMesh generated files into the constant
folder
00000583 os.system("rm -R ./constant/polyMesh")
00000584 os.system("cp -R ./2/polyMesh ./constant/polyMesh")
00000585 os.system("rm ./constant/cellLevel")
00000586 os.system("rm ./constant/pointLevel")
00000587 os.system("rm -R ./2")
00000588 os.system("rm -R ./1")
00000589 os.system("rm -r 0")
00000590 os.system("rm -r processor*")
00000591
00000592 os.system("createPatch -overwrite")
00000593
00000594
00000595 # Copies the 0 file
```

```
00000596 os.system("cp -R ./source/0 ./0")
00000597
00000598 # Edits the U document
00000599 ULoc = './0/U'
00000600 ULocTmp = './0/UTmp'
00000601
00000602 speedOldLine = 'value           uniform (speed 0 0);'
00000603 speedNewLine = '        value           uniform (%s 0 0);\n' %
(speed)
00000604 intOldLine = 'internalField   uniform (speed 0 0);'
00000605 intNewLine = 'internalField   uniform (%s 0 0);\n' % (speed)
00000606
00000607 with open(ULoc, 'r') as input_file, open(ULocTmp, 'w') as
output_file:
00000608     for line in input_file:
00000609         if line.strip() == speedOldLine :
00000610             output_file.write(speedNewLine)
00000611         else:
00000612             output_file.write(line)
00000613
00000614 os.system('rm -R %s' % (ULoc))
00000615
00000616 os.system('mv %s %s' % (ULocTmp, ULoc))
00000617
00000618 with open(ULoc, 'r') as input_file, open(ULocTmp, 'w') as
output_file:
00000619     for line in input_file:
00000620         if line.strip() == intOldLine :
00000621             output_file.write(intNewLine)
00000622         else:
00000623             output_file.write(line)
00000624
00000625 os.system('rm -R %s' % (ULoc))
00000626
00000627 os.system('mv %s %s' % (ULocTmp, ULoc))
00000628
00000629 # write parameters for solve decomposePar
00000630
00000631 decompSolveLineOld = 'numberOfSubdomains %s;' % (procMesh)
00000632 decompSolveLineNew = 'numberOfSubdomains %s;\n' % (procSolve)
00000633
00000634 with open(decompMesh, 'r') as input_file, open(decompMeshTmp,
'w') as output_file:
00000635     for line in input_file:
00000636         if line.strip() == decompSolveLineOld :
00000637             output_file.write(decompSolveLineNew)
00000638         else:
00000639             output_file.write(line)
00000640
00000641 os.system('rm -R %s' % (decompMesh))
00000642
00000643 os.system('mv %s %s' % (decompMeshTmp, decompMesh))
00000644
00000645 # Runs decomposePar and pimpleFoam
00000646 os.system("decomposePar")
00000647 os.system("mpirun -np %s renumberMesh -overwrite -parallel" %
(procSolve))
00000648 os.system("mpirun -np %s simpleFoam -parallel > log" %
(procSolve)) #run solving process
00000649 os.system("reconstructPar")
00000650 os.system("rm -R processor*")
```

Appendix K bearingStagger2.py Python Script

```
00000001 import salome
00000002 salome.salome_init()
00000003 import GEOM
00000004 from salome.geom import geomBuilder
00000005 geompy = geomBuilder.New(salome.myStudy)
00000006 gg = salome.ImportComponentGUI("GEOM")
00000007 import math
00000008
00000009 xincrementNo = 0
00000010 yincrementNo = 0
00000011 totalIncrementNo = 0
00000012 xQuantity = changeNo1
00000013 yQuantity = changeNo2
00000014 xincrementInitial = 0
00000015 xincrementDist = changeNo3
00000016 yincrementInitial = 0
00000017 staggeredGrid = changeNo9
00000018 staggerDist = changeNo10
00000019 yincrementDist = changeNo4
00000020 xSize = changeNo5
00000021 ySize = changeNo6
00000022 zSize = changeNo7
00000023 radialClearance = changeNo8
00000024 d={}
00000025 even = 0
00000026 odd = 1
00000027 anglePattern = changeNo11
00000028 angle = changeNo12
00000029
00000030 xAngleCorrect = (xSize/2)*(math.cos(angle))
00000031 yAngleCorrect = (xSize/2)*(math.sin(angle))
00000032
00000033 print yAngleCorrect
00000034
00000035 if anglePattern == True:
00000036
00000037    pointMin = geompy.MakeVertex((-1 - (ySize/2)), (-1 -
yAngleCorrect), 0)
00000038    xDomainDist = (xQuantity * xincrementDist) - (ySize/2)
00000039    xincrementInitial = xAngleCorrect - (xSize/2)
00000040    if staggeredGrid == True:
00000041        yDomainDist = (yincrementInitial + (yincrementDist *
(yQuantity)) + (staggerDist)) + (yAngleCorrect) #even command 2nd 4th etc
00000042    else:
00000043        yDomainDist = (yincrementInitial + (yincrementDist *
(yQuantity - 1))) + (yAngleCorrect) #odd command 1st 3rd etc
00000044 else:
00000045    pointMin = geompy.MakeVertex(-1, (-1 - (ySize/2)), 0)
00000046    xDomainDist = (xQuantity * xincrementDist)
00000047    if staggeredGrid == True:
00000048        yDomainDist = (yincrementInitial + (yincrementDist *
(yQuantity - 1)) + (staggerDist)) + (ySize/2) #even command 2nd 4th etc
00000049    else:
00000050        yDomainDist = (yincrementInitial + (yincrementDist *
yQuantity)) - (ySize/2) #odd command 1st 3rd etc
00000051
00000052 print yDomainDist
```

```
00000053 print 'yDomainDist'
00000054
00000055 pointMax = geompy.MakeVertex((xDomainDist+1), (yDomainDist+1),
(radialClearance+1))
00000056 domainBox = geompy.MakeBoxTwoPnt(pointMin, pointMax)
00000057
00000058
00000059 while xincrementNo < xQuantity :
00000060
00000061     if staggeredGrid == True and xincrementNo % 2 == 0:
00000062         Stagger = staggerDist
00000063     else:
00000064         Stagger = 0
00000065     print Stagger
00000066     while yincrementNo < yQuantity :
00000067         xDist = xincrementInitial + (xincrementDist *
xincrementNo) + (xSize/2)
00000068         yDist = yincrementInitial + (yincrementDist *
yincrementNo) + (Stagger)
00000069         # create a vertex
00000070         d["p{0}".format(yincrementNo)] = geompy.MakeVertex(xDist,
yDist, 0)
00000071         # create radius
00000072         d["radius{0}".format(yincrementNo)] = 1
00000073         # create sphere
00000074         d["sphere{0}".format(yincrementNo)] =
geompy.MakeSpherePntR(d["p{0}".format(yincrementNo)],
d["radius{0}".format(yincrementNo)])
00000075         # scale sphere into ellipsoid
00000076         d["scale{0}".format(yincrementNo)] =
geompy.MakeScaleAlongAxes(d["sphere{0}".format(yincrementNo)],
d["p{0}".format(yincrementNo)], (xSize/2), (ySize/2), zSize)
00000077         if anglePattern == True:
00000078             d["pVector{0}".format(yincrementNo)] =
geompy.MakeVertex(xDist, yDist,  1)
00000079             d["vector{0}".format(yincrementNo)] =
geompy.MakeVector(d["p{0}".format(yincrementNo)],
d["pVector{0}".format(yincrementNo)])
00000080             d["angle{0}".format(yincrementNo)] =
geompy.MakeRotation(d["scale{0}".format(yincrementNo)],
d["vector{0}".format(yincrementNo)], angle)
00000081             if xincrementNo == 0 and yincrementNo == 0:
00000082                 d["fuse{0}".format(even)]=
geompy.MakeFuseList([d["angle{0}".format(yincrementNo)],domainBox])
00000083             else:
00000084                 if totalIncrementNo % 2 == 0:
00000085                     d["fuse{0}".format(even)]=
geompy.MakeFuseList([d["angle{0}".format(yincrementNo)],d["fuse{0}".forma
t(odd)]])
00000086                 else:
00000087                     d["fuse{0}".format(odd)]=
geompy.MakeFuseList([d["angle{0}".format(yincrementNo)],d["fuse{0}".forma
t(even)]])
00000088             print "Finished y increment %d at an y distance of
%s" % (yincrementNo, yDist)
00000089         # fuse the domain with the ellipsoid shape
00000090         else:
00000091             if xincrementNo == 0 and yincrementNo == 0:
00000092                 d["fuse{0}".format(even)]=
geompy.MakeFuseList([d["scale{0}".format(yincrementNo)],domainBox])
00000093             else:
```

```
00000094                if totalIncrementNo % 2 == 0:
00000095                    d["fuse{0}".format(even)]=
geompy.MakeFuseList([d["scale{0}".format(yincrementNo)],d["fuse{0}".forma
t(odd)]])
00000096                else:
00000097                    d["fuse{0}".format(odd)]=
geompy.MakeFuseList([d["scale{0}".format(yincrementNo)],d["fuse{0}".forma
t(even)]])
00000098            print "Finished y increment %d at an y distance of
%s" % (yincrementNo, yDist)
00000099        #loop increment + 1
00000100        yincrementNo += 1
00000101        totalIncrementNo += 1
00000102
00000103    print "Finished x increment %d" % (xincrementNo)
00000104    # resets the y iteration
00000105    yincrementNo = 0
00000106    # adds one to the x increment
00000107    xincrementNo += 1
00000108
00000109 if totalIncrementNo % 2 == 0:
00000110    geompy.addToStudy(d["fuse{0}".format(odd)],"domainBox")
00000111    geompy.Export(d["fuse{0}".format(odd)], "bearingSurface.stl",
"STL")
00000112 else:
00000113    geompy.addToStudy(d["fuse{0}".format(even)],"domainBox")
00000114    geompy.Export(d["fuse{0}".format(even)],
"bearingSurface.stl", "STL")
00000115
```

Appendix L Parametric Study Results

Table L.1. Parametric Study Results.

Trial No.	Area (x $10^{-6}m^2$)	Shear Force (mN)	Pressure Drag Force (mN)	Texture-Induced Lift (N)
0	0.060	7.982	0.105	0.110
1	0.140	10.822	0.478	0.136
2	0.300	16.806	0.687	0.126
3	0.140	8.308	0.008	0.007
4	0.300	39.200	1.502	0.047
5	0.083	4.512	1.392	0.176
6	0.128	7.365	0.028	0.009
7	0.100	9.166	3.255	0.262
8	0.200	16.056	0.182	0.023
9	0.325	43.967	1.302	0.045
10	0.255	21.054	0.182	0.030
11	0.200	11.369	0.389	0.057
12	0.325	26.810	0.402	0.151
13	0.210	11.627	0.875	1.197
14	0.350	47.197	1.091	0.175
15	0.360	29.152	1.031	0.038
16	0.363	21.567	0.086	0.005
17	0.270	33.178	3.691	0.229

Table L.2. Parametric Study Results (continued…).

Trial No.	Total Drag (mN)	Shear Stress (kPa)	Lift Pressure (MPa)	Lift/Drag	Drag Red. (%)
0	8.086	133.026	1.840	13.650	3.819
1	11.300	77.304	0.971	12.034	3.996
2	17.493	56.019	0.421	7.220	2.905
3	8.316	59.346	0.047	0.786	1.086
4	40.702	130.667	0.156	1.153	3.178
5	5.905	54.692	2.129	29.752	14.873
6	7.393	57.765	0.071	1.227	3.441
7	12.422	91.663	2.622	21.107	11.352
8	16.238	80.281	0.116	1.431	3.430
9	45.269	135.283	0.137	0.985	0.596
10	21.236	82.565	0.118	1.421	0.948
11	11.758	56.843	0.287	4.886	2.106
12	27.212	82.493	0.464	5.546	0.411
13	12.501	55.365	5.698	95.720	0.872
14	48.287	134.847	0.501	3.634	1.542
15	30.183	80.978	0.104	1.243	0.277
16	21.653	59.496	0.014	0.235	0.535
17	36.869	122.880	0.847	6.202	2.551

Table L.3. Dimensionless roughness height and spacing for parametric study.

Trial No.	Friction Velocity (ms^{-1}), u_τ	Roughness Height, k^+	Spacing, s^+	k/s Ratio
0	12.340	0.130	6.495	0.020
1	9.550	0.302	10.053	0.030
2	8.117	0.427	12.816	0.033
3	8.193	0.086	8.624	0.010
4	12.382	0.391	19.550	0.020
5	8.993	0.473	4.733	0.100
6	8.095	0.256	12.781	0.020
7	11.847	0.624	6.235	0.100
8	9.578	0.101	10.082	0.010
9	12.546	0.660	13.206	0.050
10	9.700	0.102	15.317	0.007
11	8.150	0.257	4.290	0.060
12	9.727	0.307	5.119	0.060
13	8.202	0.432	8.633	0.050
14	12.486	0.131	19.714	0.007
15	9.733	0.512	15.368	0.033
16	8.215	0.086	4.324	0.020
17	12.422	0.392	13.075	0.030

Table L.4. Data for Direct Effect Charts.

Parameter Level	Shear Stress (kPa)	Lift Pressure (MPa)	Lift/Drag Ratio	Drag Reduction (%)
A1	82.307	0.930	9.818	5.34
A2	90.083	0.908	13.319	1.09
B1	90.173	0.629	6.699	2.40
B2	86.235	1.499	22.765	3.66
B3	82.177	0.629	5.240	3.60
C1	91.482	0.444	3.906	1.61
C2	82.843	1.597	21.945	3.48
C3	84.260	0.717	8.854	4.57
D1	91.593	0.439	3.526	1.89
D2	87.992	0.466	5.174	2.61
D3	79.000	1.852	26.005	5.15
E1	84.382	1.784	24.662	4.42
E2	83.497	0.755	7.282	3.39
E3	90.706	0.218	2.762	1.84
F1	79.702	1.226	12.529	5.52
F2	88.410	1.303	19.526	2.09
F3	90.473	0.229	2.650	2.05
G1	89.527	1.367	19.680	2.28
G2	86.564	0.637	7.978	4.16
G3	82.494	0.753	7.047	3.21
H1	124.728	1.017	7.788	3.84
H2	76.385	0.651	8.571	3.99
H3	57.472	1.090	18.346	1.82

Appendix M Confirmation Study Results

Table M.1. Data from Confirmation Study Simulations.

Study	Shear Force (mN)	Pressure Force (mN)	Total Drag Force (mN)	Clearance (um)	Non-Textured Drag (mN)	Drag Reduction (%)
0	5.930	3.717	9.647	3	11.560	16.55%
1	4.512	1.392	5.904	5	6.936	14.88%
2	3.623	0.663	4.286	7	4.954	13.50%
3	3.017	0.367	3.384	9	3.853	12.19%
4	2.580	0.223	2.803	11	3.153	11.09%
5	48.483	0.365	48.847	3	49.044	0.40%
6	29.213	0.119	29.332	5	29.426	0.32%
7	20.900	0.060	20.960	7	21.019	0.28%
8	16.274	0.038	16.313	9	16.348	0.22%
9	13.323	0.028	13.351	11	13.376	0.19%

Table M.2. Dimensionless Roughness Height and Spacing.

Study	Friction Velocity (ms^{-1}), u_τ	Roughness Height, k^+	Spacing, s^+	k/s Ratio
0	11.495	0.605	6.050	0.100
1	8.992	0.473	4.733	0.100
2	7.661	0.403	4.032	0.100
3	6.808	0.358	3.583	0.100
4	6.196	0.326	3.261	0.100
5	12.558	0.132	19.828	0.007
6	9.731	0.102	15.365	0.007
7	8.226	0.087	12.988	0.007
8	7.257	0.076	11.458	0.007
9	6.565	0.069	10.366	0.007

Appendix N 2D Roundness Profiles

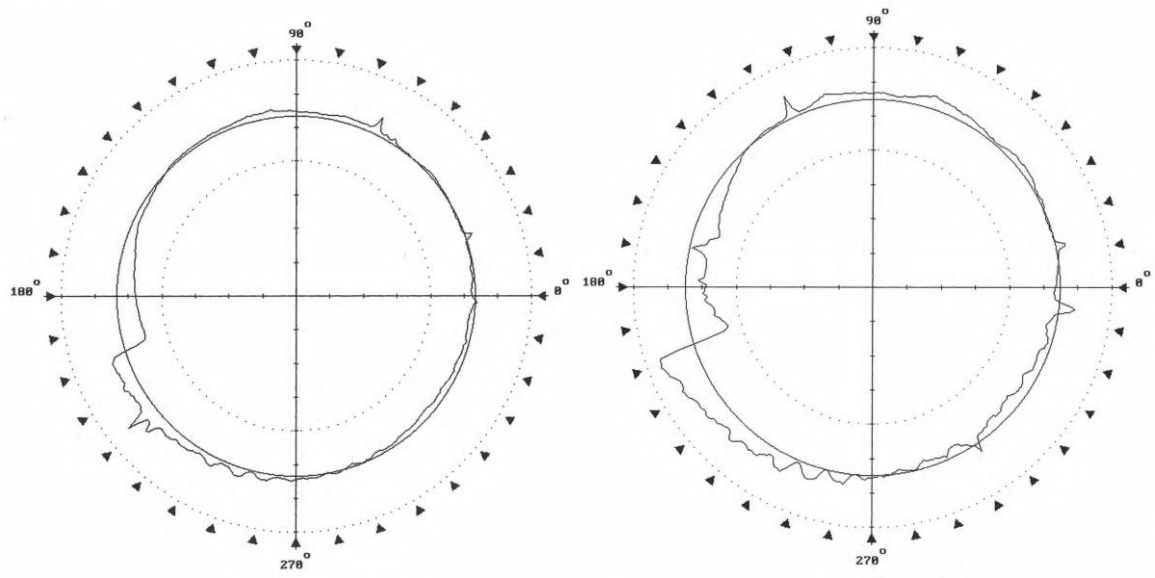


Figure N.1. Left: component 1, test 1 (division is 2µm). Right: component 1, test 2 (division is 1µm).

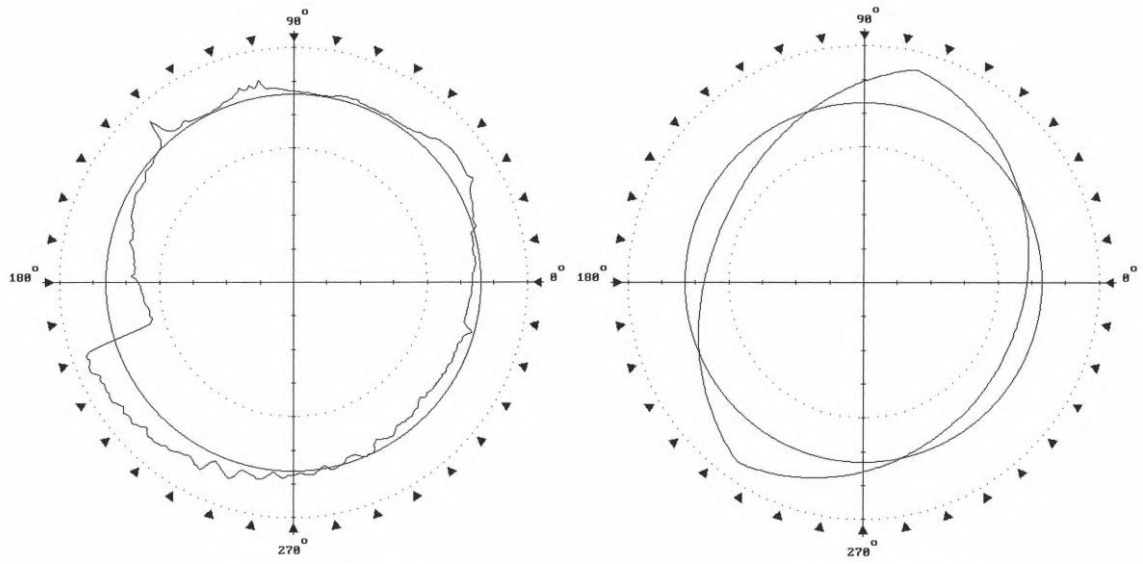


Figure N.2. Left: component 1, test 3 (division is 1µm). Right: component 2, test 1 (division is 50µm).

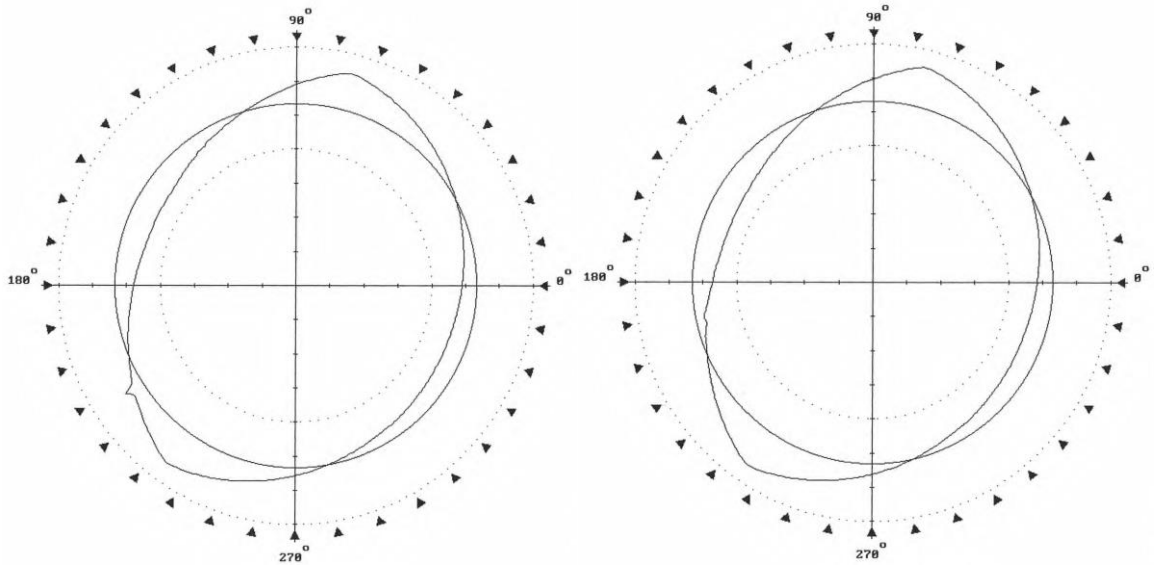


Figure N.3. Left: component 2, test 2 (division is 50µm). Right: component 2, test 3 (division is 50µm).

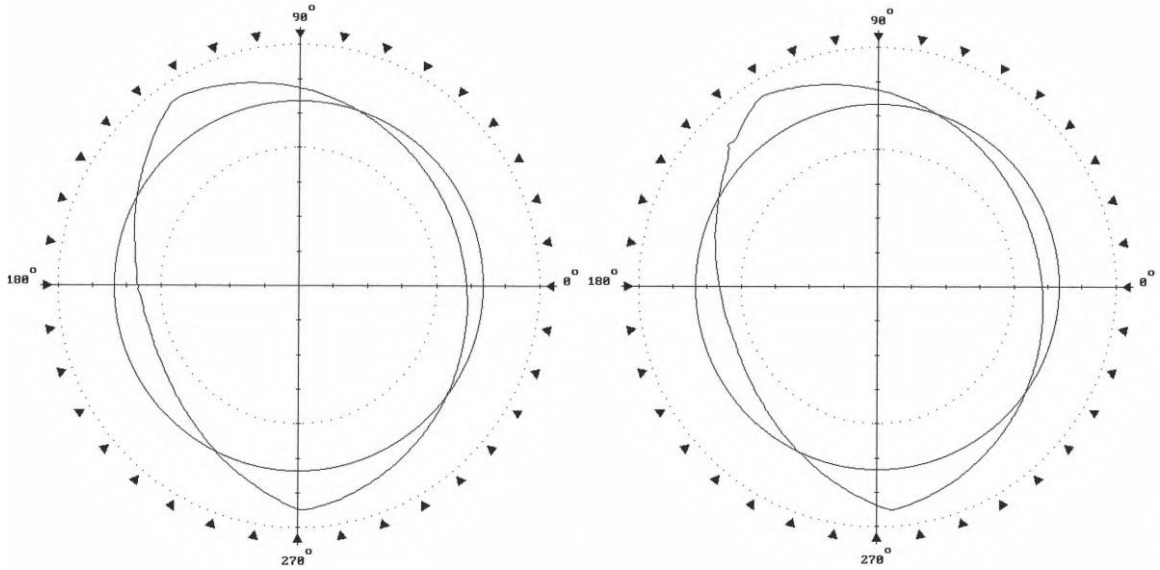


Figure N.4. Left: component 3, test 1 (division is 50µm). Right: component 3, test 2 (division is 50µm).

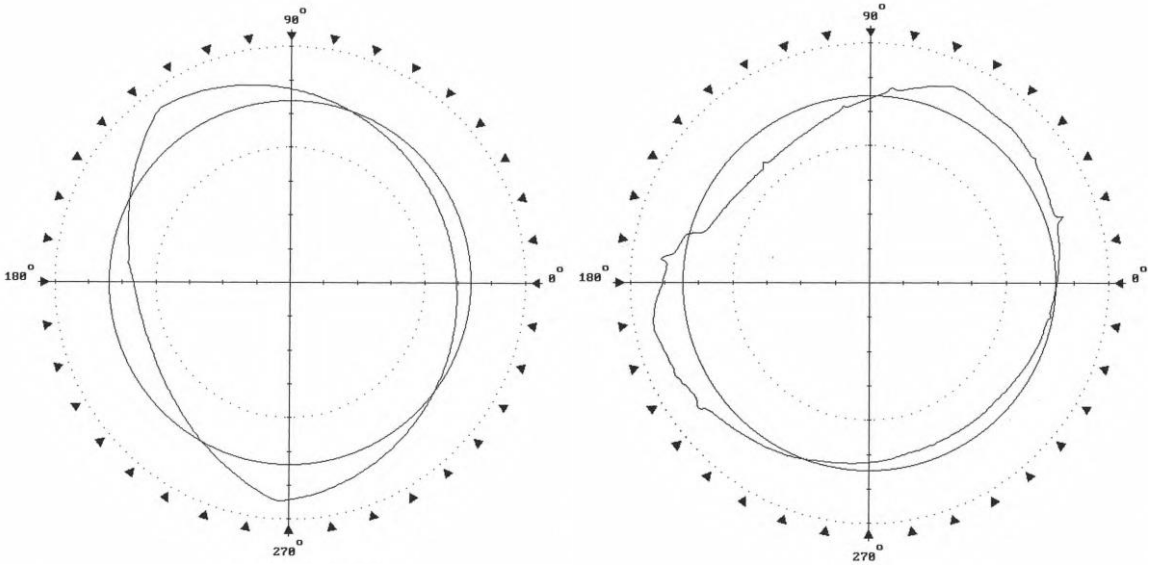


Figure N.5. Left: component 3, test 3 (division is 50µm). Right: component 4, test 1 (division is 5µm).

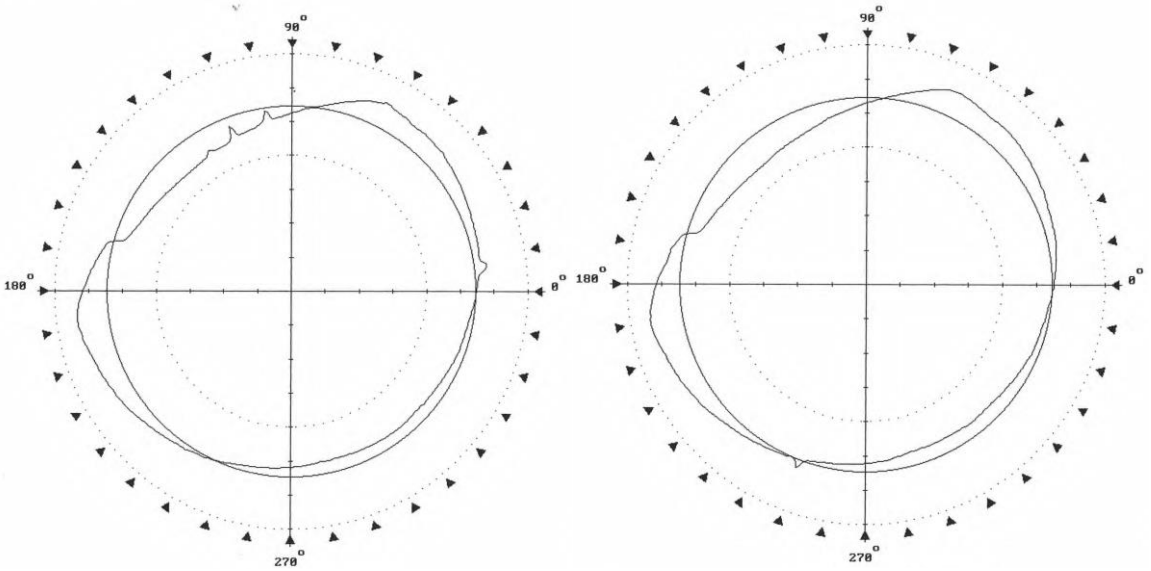


Figure N.6. Left: component 4, test 2 (division is 5µm). Right: component 4, test 3 (division is 5µm).

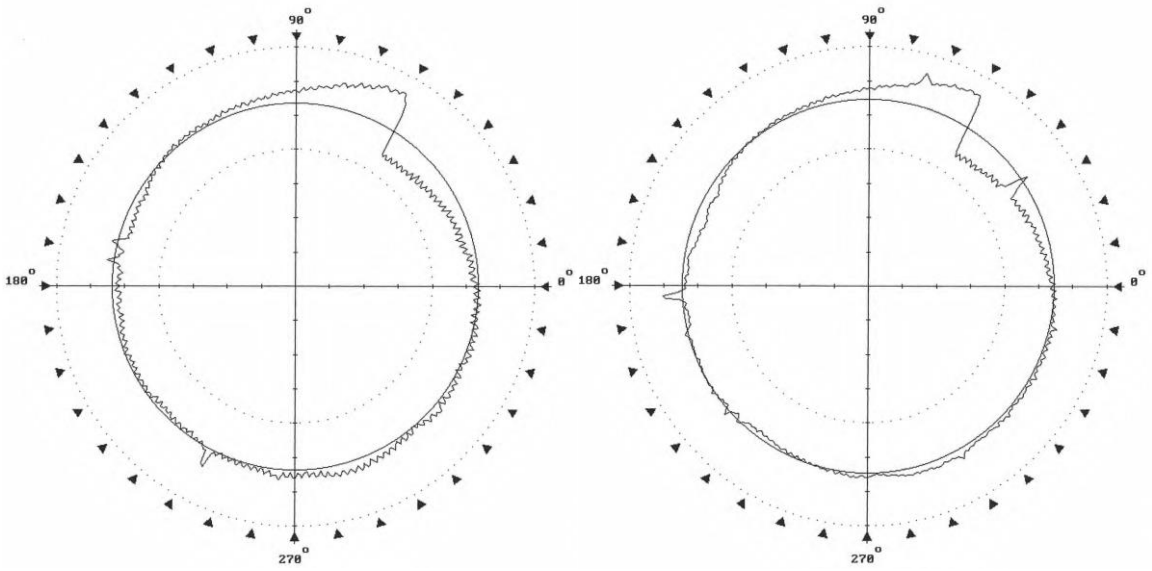


Figure N.7. Left: component 5, test 1 (division is 1µm). Right: component 5, test 2 (division is 1µm).

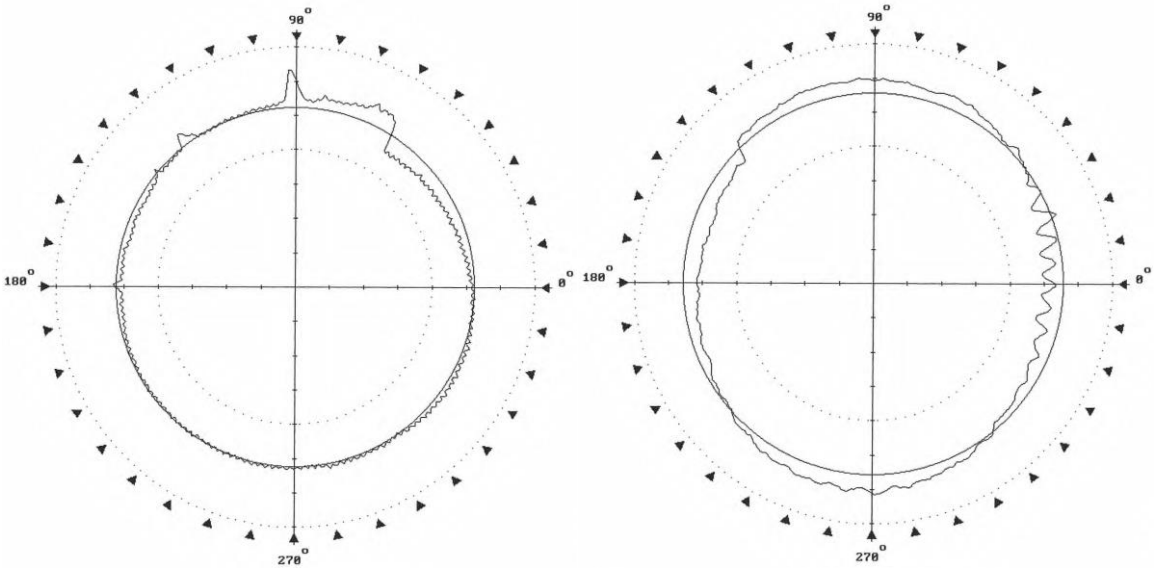


Figure N.8. Left: component 5, test 3 (division is 2µm). Right: component 6, test 1 (division is 0.5µm).

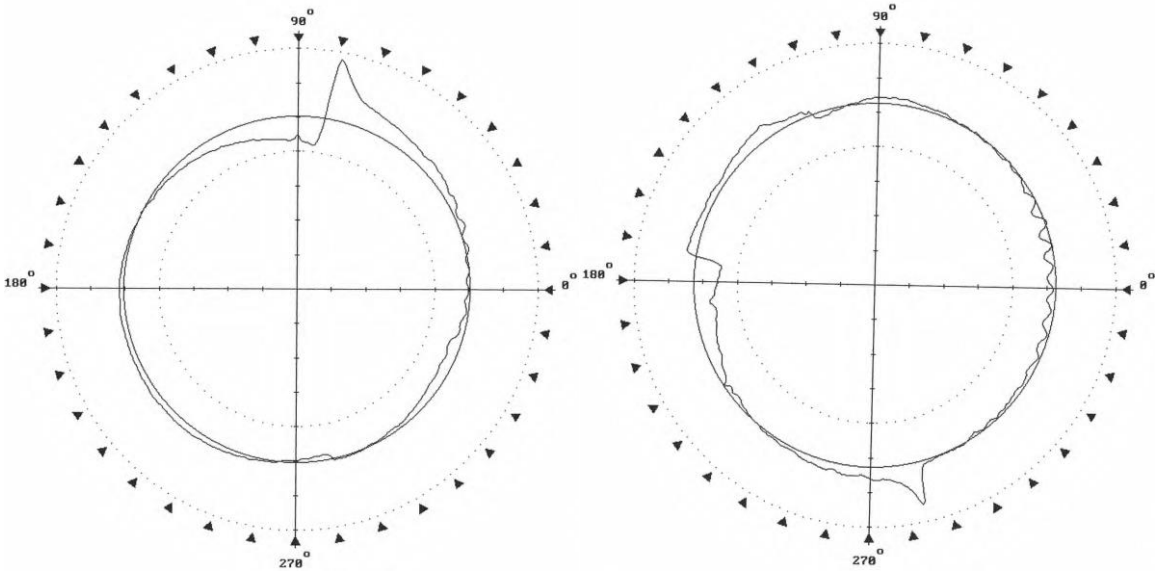


Figure N.9. Left: component 6, test 2 (division is 2µm). Right: component 6, test 3 (division is 1µm).

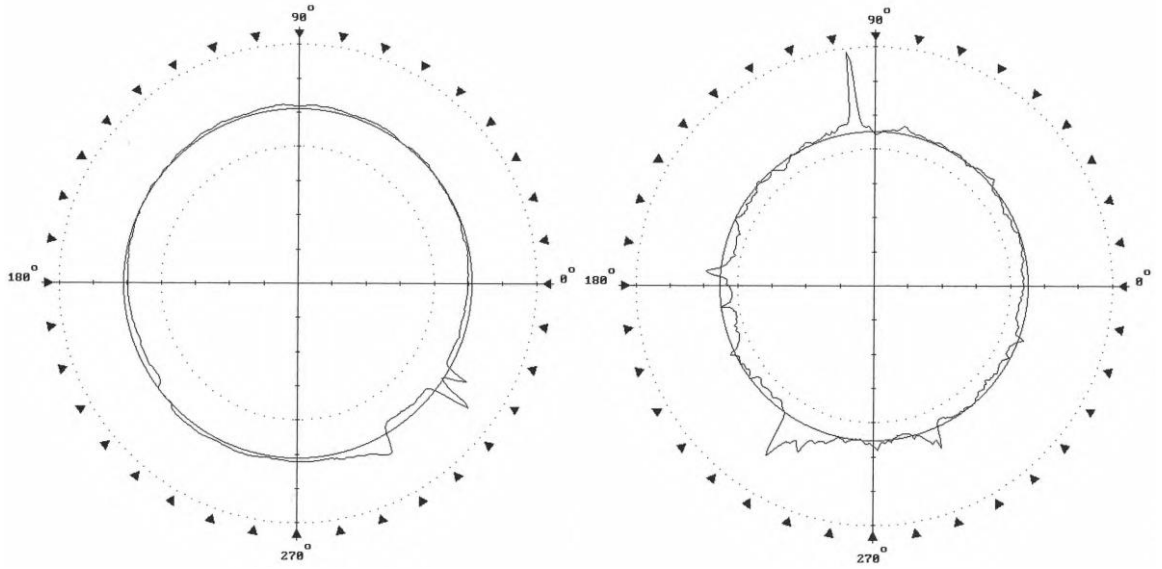


Figure N.10. Left: component 7, test 1 (division is 5µm). Right: component 7, test 2 (division is 1µm).

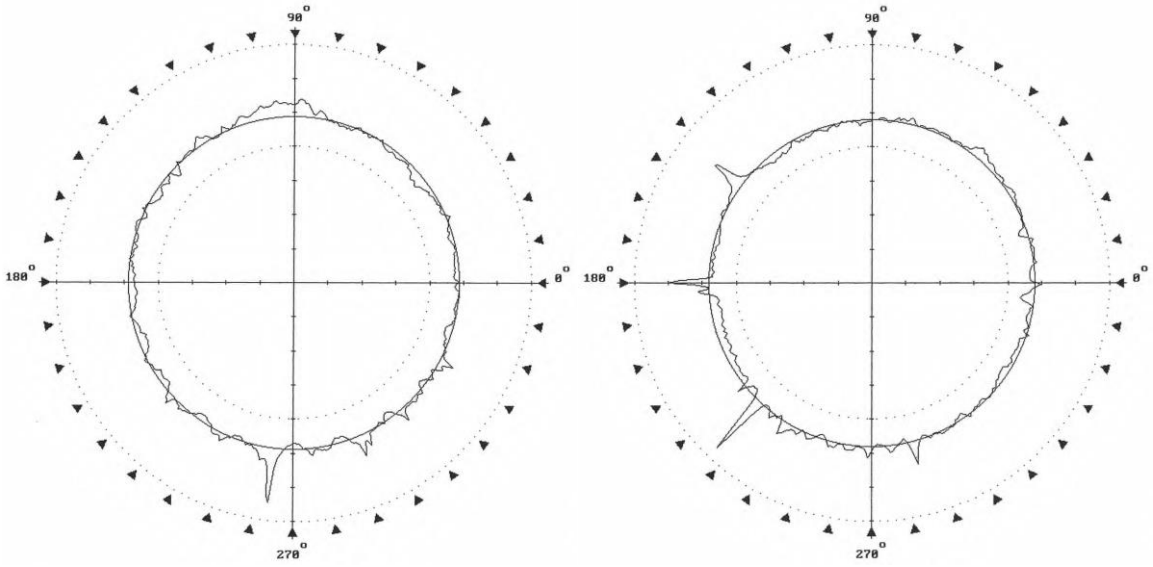


Figure N.11. Left: component 7, test 3 (division is 1µm). Right: component 8, test 1 (division is 1µm).

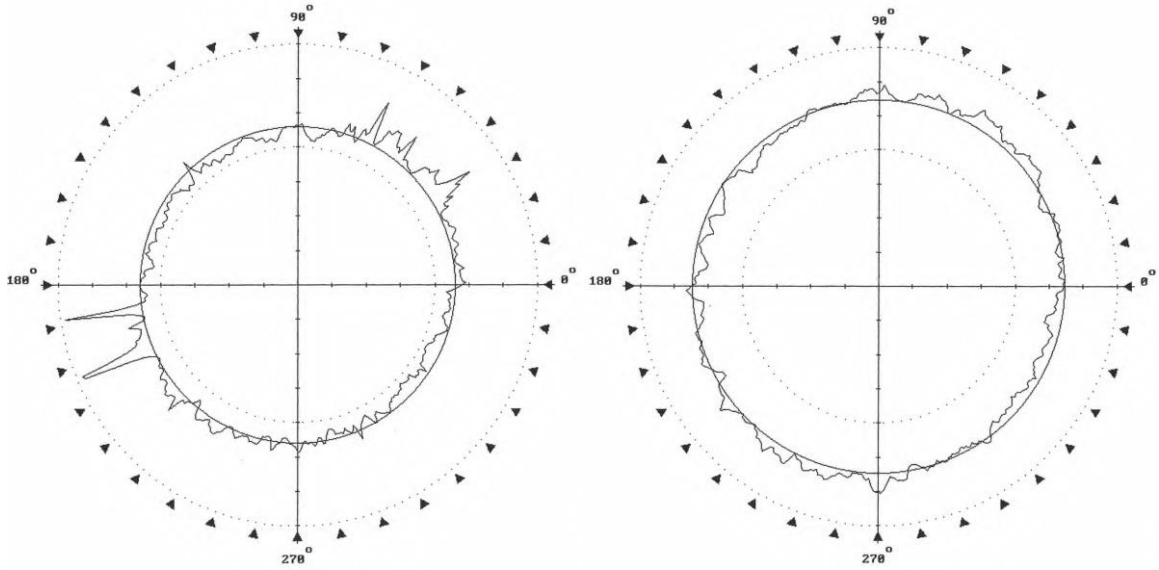


Figure N.12. Left: component 8, test 2 (division is 0.5µm). Right: component 8, test 3 (division is 0.5µm).

Appendix O 3D Height Roughness Data (Chapter 5 Components)

Magnification levels other than 2.5X are highlighted as yellow for 5X, turquoise for 10X and red for 40X.

Component	Trial No.	Sa (µm)	Sku	Sp (µm)	Sq (µm)	Ssk	Sv (µm)	Sz (µm)
1	1	0.279	10.479	1.649	0.379	-1.301	-9.216	10.866
	2	0.272	8.707	3.738	0.365	-1.110	-7.225	10.964
	3	0.262	7.014	3.009	0.353	-0.560	-4.533	7.542
	Average	0.271	8.733	2.799	0.366	-0.990	-6.991	9.791
2	1	0.226	11.112	1.222	0.303	-1.141	-6.038	7.260
	2	0.183	10.936	1.423	0.247	-0.591	-4.202	5.625
	3	0.217	6.993	1.589	0.285	-0.558	-3.840	5.429
	Average	0.209	9.680	1.411	0.278	-0.763	-4.693	6.105
3	1	0.102	38.022	1.797	0.162	-3.543	-2.928	4.726
	2	0.098	23.170	2.308	0.149	-1.958	-2.148	4.455
	3	0.131	10.449	2.963	0.184	0.184	-2.553	5.516
	Average	0.110	23.880	2.356	0.165	-1.772	-2.543	4.899
4	1	0.261	3.783	1.549	0.333	-0.294	-3.753	5.302
	2	0.263	3.681	1.684	0.337	-0.331	-2.267	3.952
	3	0.187	3.589	1.371	0.241	-0.090	-1.309	2.679
	Average	0.237	3.684	1.535	0.304	-0.238	-2.443	3.978
5	1	0.263	5.535	1.554	0.337	-0.305	-5.053	6.607
	2	0.257	39.982	3.467	0.364	-2.319	-6.245	9.712
	3	0.227	11.271	3.456	0.302	0.650	-3.157	6.613
	Average	0.249	18.929	2.826	0.334	-0.658	-4.818	7.644
6	1	0.104	9.856	1.023	0.139	-0.347	-3.132	4.155
	2	0.121	18.108	1.466	0.177	-1.919	-4.133	5.599
	3	0.137	22.912	0.995	0.203	-2.366	-4.792	5.787
	Average	0.121	16.959	1.161	0.173	-1.544	-4.019	5.180
7	1	0.254	3.478	1.352	0.327	0.015	-1.402	2.755
	2	0.234	4.813	2.444	0.306	-0.467	-2.333	4.777
	3	0.235	5.059	2.816	0.309	-0.204	-2.667	5.483
	Average	0.241	4.450	2.204	0.314	-0.219	-2.134	4.338
8	1	0.188	3.790	2.217	0.253	-0.500	-2.936	5.153
	2	0.166	6.433	1.236	0.212	-0.457	-1.284	2.519
	3	0.206	22.774	5.253	0.321	0.521	-3.303	8.557
	Average	0.187	10.999	2.902	0.262	-0.145	-2.508	5.410

Appendix P 3D Functional Roughness Data (Chapter 5 Components)

Magnification levels other than 2.5X are highlighted as yellow for 5X, turquoise for 10X and red for 40X.

Component	Trial No.	Sbi	Sc ($\mu m^3/\mu m^2$)	Sci	Sm (nm^3/nm^2)	Sv (nm^3/nm^2)	Svi
1	1	0.689	0.392	1.319	14.420	52.745	0.139
	2	0.685	0.382	1.328	13.771	50.762	0.139
	3	0.702	0.356	1.291	15.164	49.268	0.140
	Average	0.692	0.377	1.313	14.452	50.925	0.139
2	1	0.679	0.321	1.347	12.502	39.520	0.130
	2	0.614	0.279	1.526	14.487	27.949	0.133
	3	0.639	0.320	1.454	13.730	33.522	0.118
	Average	0.644	0.307	1.442	13.573	33.664	0.127
3	1	0.844	0.128	1.044	6.985	24.665	0.152
	2	0.794	0.129	1.144	7.896	20.492	0.137
	3	0.699	0.182	1.337	12.554	22.675	0.123
	Average	0.779	0.146	1.175	9.145	22.611	0.137
4	1	0.640	0.377	1.453	15.276	39.422	0.118
	2	0.643	0.378	1.444	15.289	41.601	0.124
	3	0.635	0.271	1.460	12.333	29.694	0.123
	Average	0.639	0.342	1.452	14.299	36.906	0.122
5	1	0.678	0.376	1.382	19.930	39.591	0.109
	2	0.619	0.396	1.510	15.936	37.923	0.113
	3	0.675	0.324	1.388	16.822	32.590	0.108
	Average	0.657	0.365	1.427	17.563	36.701	0.110
6	1	0.648	0.150	1.451	8.431	15.740	0.113
	2	0.760	0.158	1.183	8.267	26.132	0.147
	3	0.767	0.181	1.164	8.851	29.856	0.147
	Average	0.725	0.163	1.266	8.516	23.909	0.136
7	1	0.623	0.369	1.505	18.953	37.305	0.114
	2	0.659	0.330	1.395	14.539	39.689	0.130
	3	0.651	0.332	1.425	16.903	38.540	0.125
	Average	0.644	0.344	1.442	16.798	38.511	0.123
8	1	0.679	0.257	1.354	13.897	33.789	0.134
	2	0.725	0.215	1.257	9.069	28.833	0.136
	3	0.850	0.256	1.073	21.091	44.703	0.139
	Average	0.751	0.243	1.228	14.686	35.775	0.136

Appendix Q 3D Spatial and Hybrid Roughness Data (Chapter 5 Components)

Magnification levels other than 2.5X are highlighted as yellow for 5X, turquoise for 10X and red for 40X.

Component	Trial No.	Sal (µm)	Std (deg)	Str	Sdq (deg)	Sdr (%)	Sds (1/mm²)	Ssc (1/mm)
1	1	15.957	0.963	0.059	6.198	0.576	981.278	34.336
	2	21.483	0.773	0.046	6.109	0.559	988.310	34.909
	3	15.957	1.078	0.034	5.749	0.493	977.552	31.420
	Average	17.799	0.938	0.046	6.019	0.543	982.380	33.555
2	1	23.261	1.012	0.233	4.871	0.356	863.081	27.308
	2	17.650	1.086	0.190	5.877	0.521	2213.248	57.309
	3	19.733	0.364	0.192	6.651	0.669	2646.113	69.161
	Average	20.215	0.821	0.205	5.800	0.515	1907.481	51.259
3	1	13.214	3.847	0.170	7.630	0.851	6126.019	137.000
	2	9.908	3.821	0.196	8.182	0.977	8077.329	154.000
	3	6.517	1.667	0.092	13.348	2.623	11018.394	311.000
	Average	9.880	3.112	0.153	9.720	1.484	8407.247	200.667
4	1	27.925	1.321	0.175	4.901	0.360	900.550	27.045
	2	29.042	0.025	0.165	4.861	0.353	912.863	26.615
	3	21.373	0.920	0.086	10.337	1.581	11487.709	198.000
	Average	26.113	0.755	0.142	6.700	0.765	4433.707	83.887
5	1	13.601	1.358	0.189	20.821	6.357	12042.178	423.000
	2	19.946	7.699	0.139	5.392	0.438	945.161	30.597
	3	20.402	1.494	0.137	11.812	2.036	9437.234	223.000
	Average	17.983	3.517	0.155	12.675	2.944	7474.858	225.532
6	1	12.615	0.257	0.096	2.299	0.080	671.324	11.125
	2	11.968	2.046	0.111	3.004	0.136	737.009	14.857
	3	19.946	0.065	0.128	3.288	0.162	719.465	15.953
	Average	14.843	0.789	0.112	2.864	0.126	709.266	13.978
7	1	7.772	3.190	0.056	14.631	3.218	13163.465	308.000
	2	7.772	0.247	0.074	14.419	3.107	11989.240	312.000
	3	8.011	0.390	0.127	15.374	3.520	11907.710	336.000
	Average	7.852	1.276	0.086	14.808	3.282	12353.472	318.667
8	1	17.379	0.665	0.229	14.695	3.141	14619.402	332.000
	2	6.910	1.147	0.351	35.937	21.470	249000.000	4838.000
	3	1.577		0.658	52.050	61.260	184000.000	8448.000
	Average	8.622	0.906	0.413	34.227	28.624	149206.467	4539.333

Appendix R 3D Measurements of the Textured Component Surfaces (Chapter 5 Components)

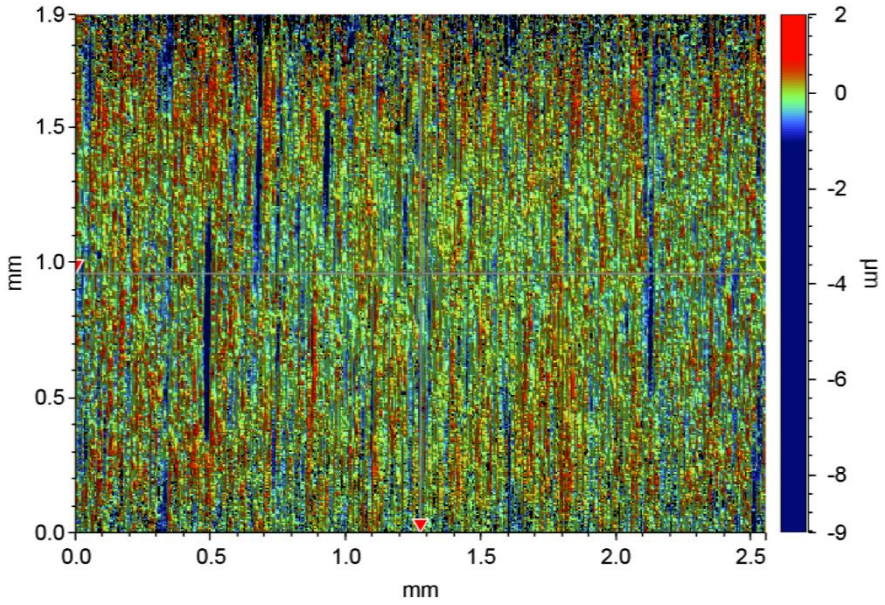


Figure R.1. Component 1, trial 1 (2.5X magnification).

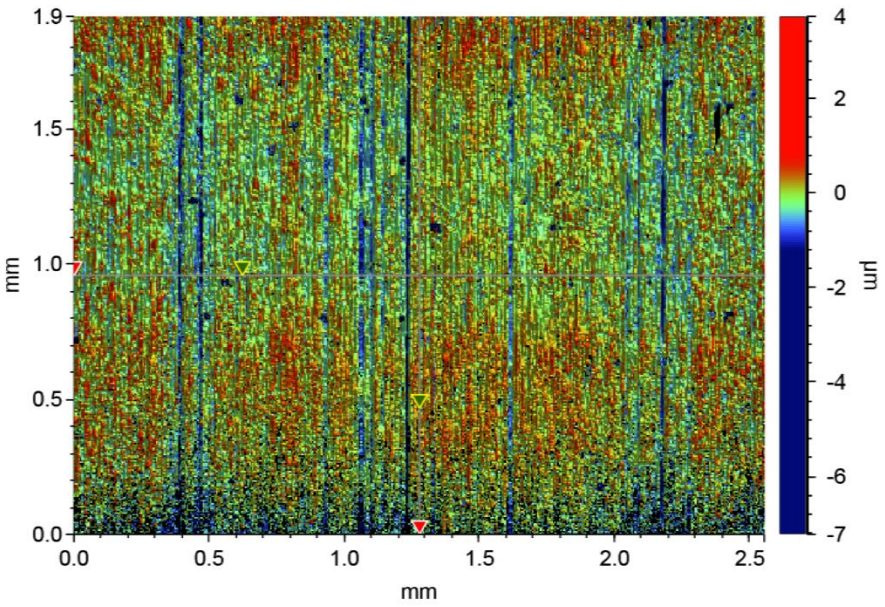


Figure R.2. Component 1, trial 2 (2.5X magnification).

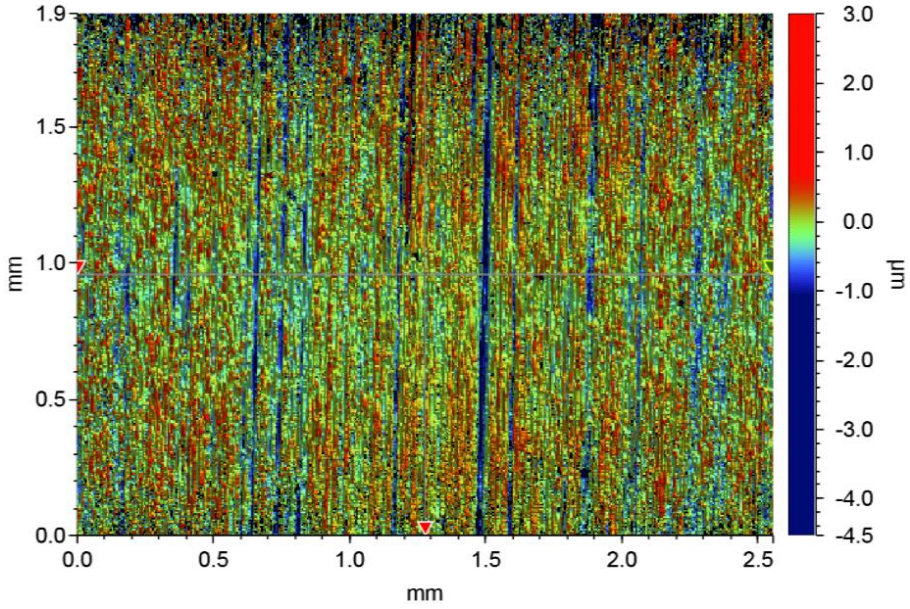


Figure R.3. Component 1, trial 3 (2.5X magnification).

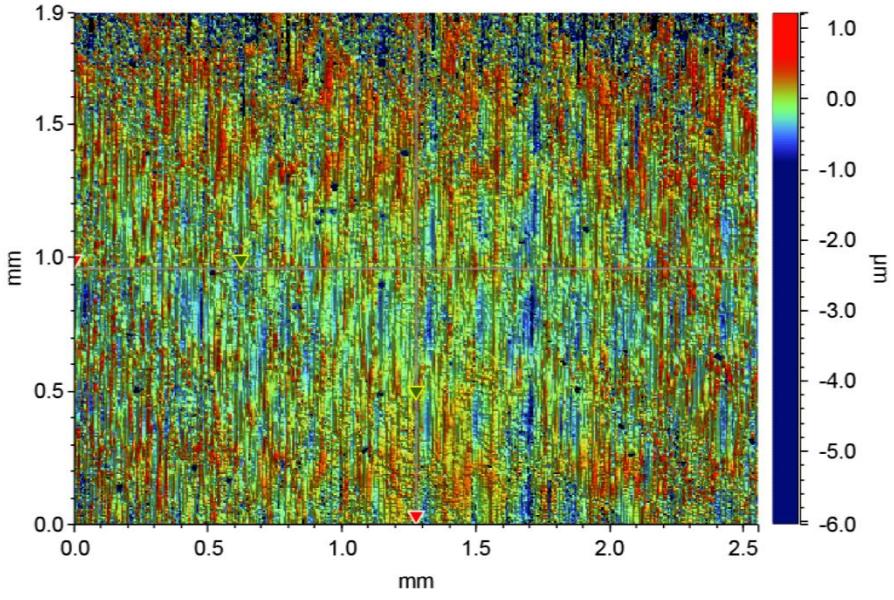


Figure R.4. Component 2, trial 1 (2.5X magnification).

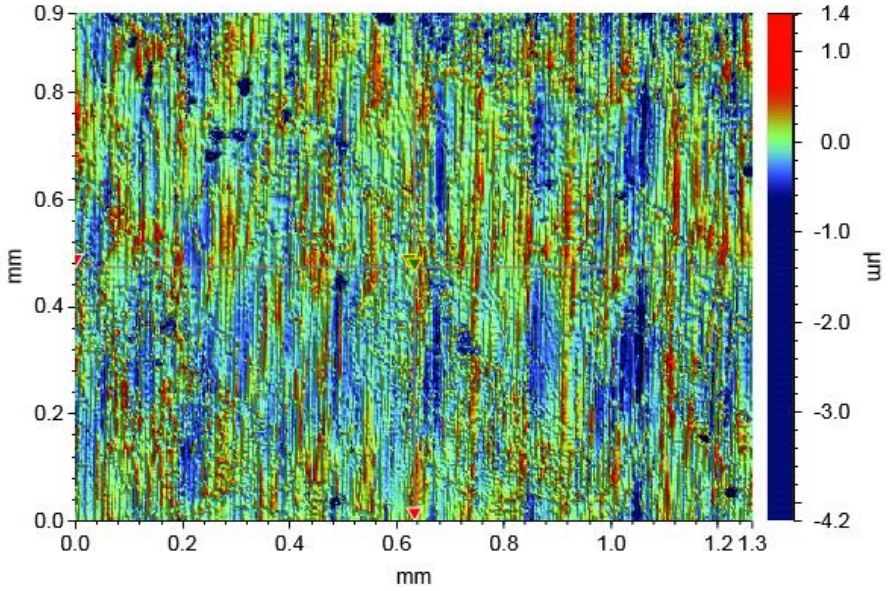


Figure R.5. Component 2, trial 2 (5X magnification).

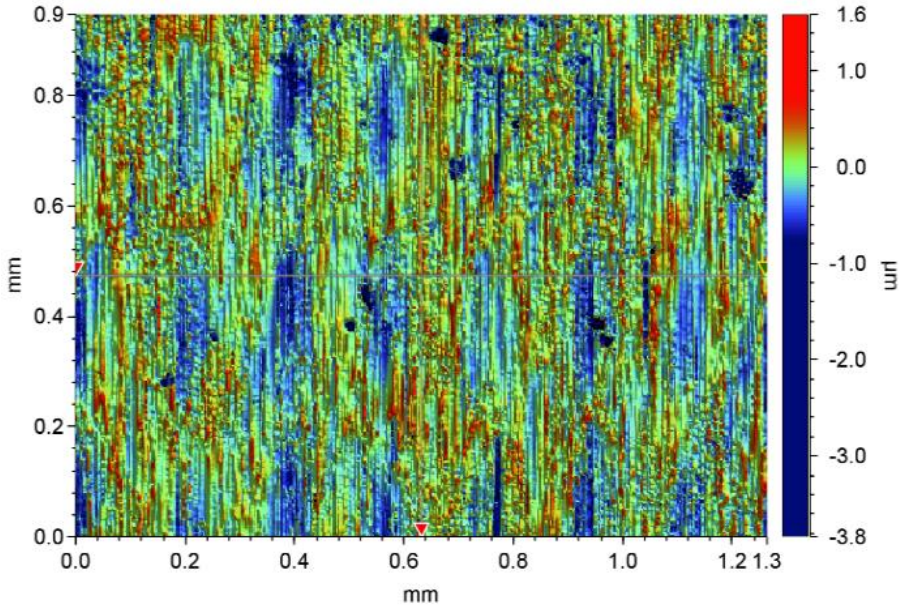


Figure R.6. Component 2, trial 3 (5X magnification).

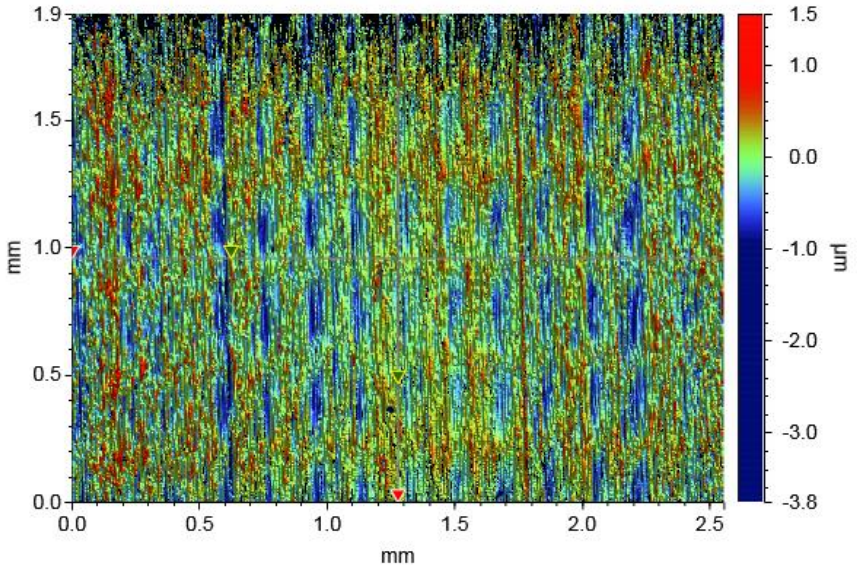


Figure R.7. Component 4, trial 1 (2.5X magnification).

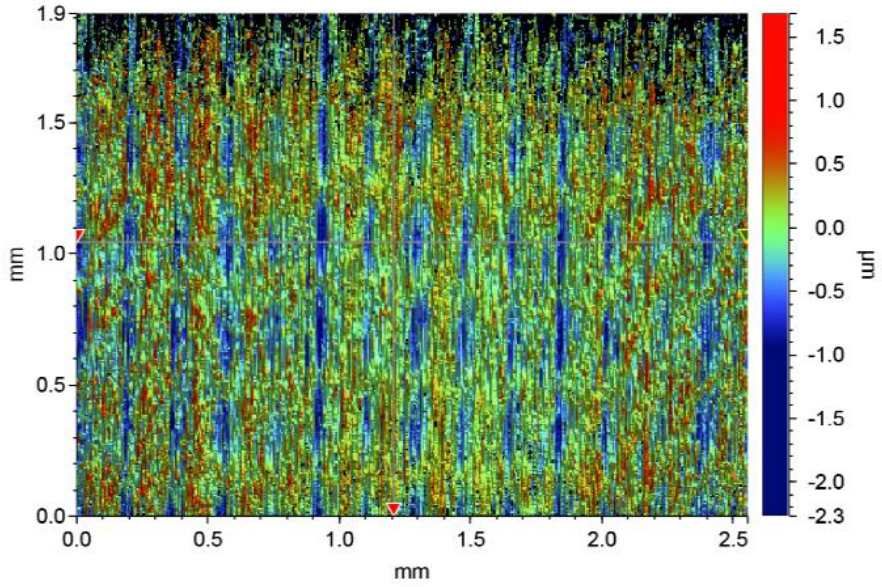


Figure R.8. Component 4, trial 2 (2.5X magnification).

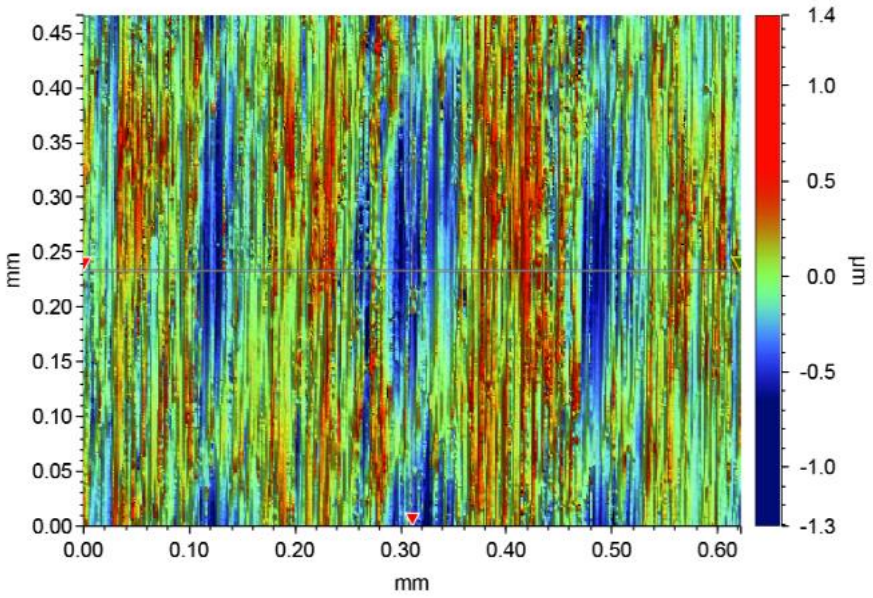


Figure R.9. Component 4, trial 3 (10X magnification).

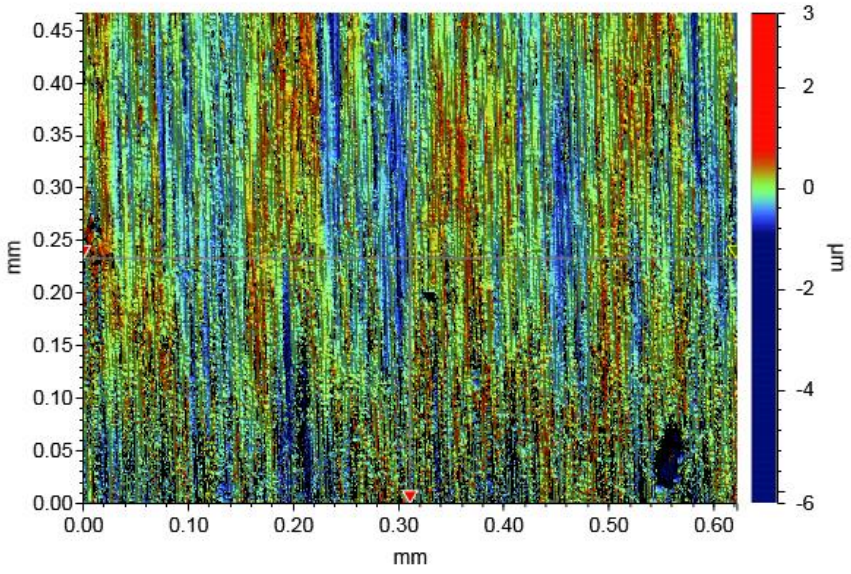


Figure R.10. Component 5, trial 1 (10X magnification).

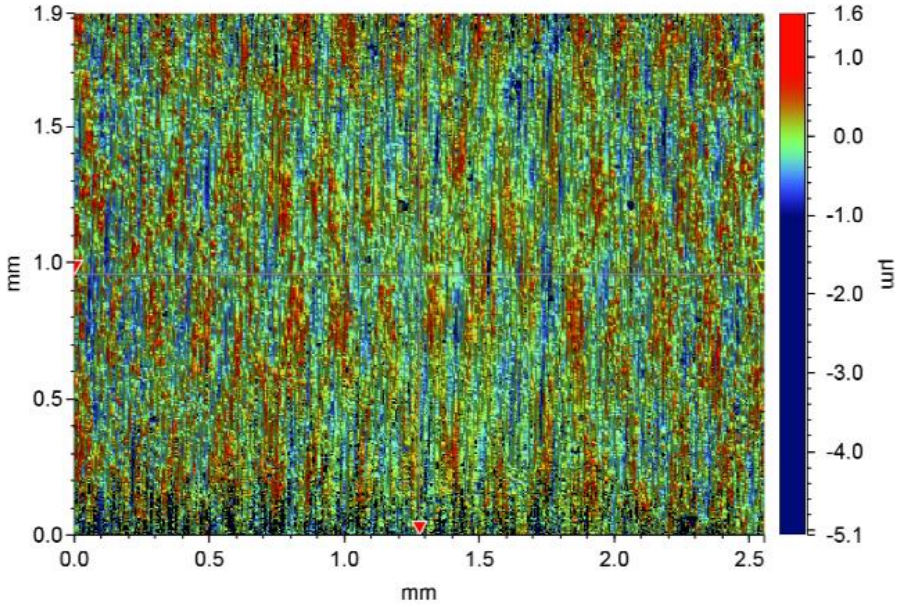


Figure R.11. Component 5, trial 2 (2.5X magnification).

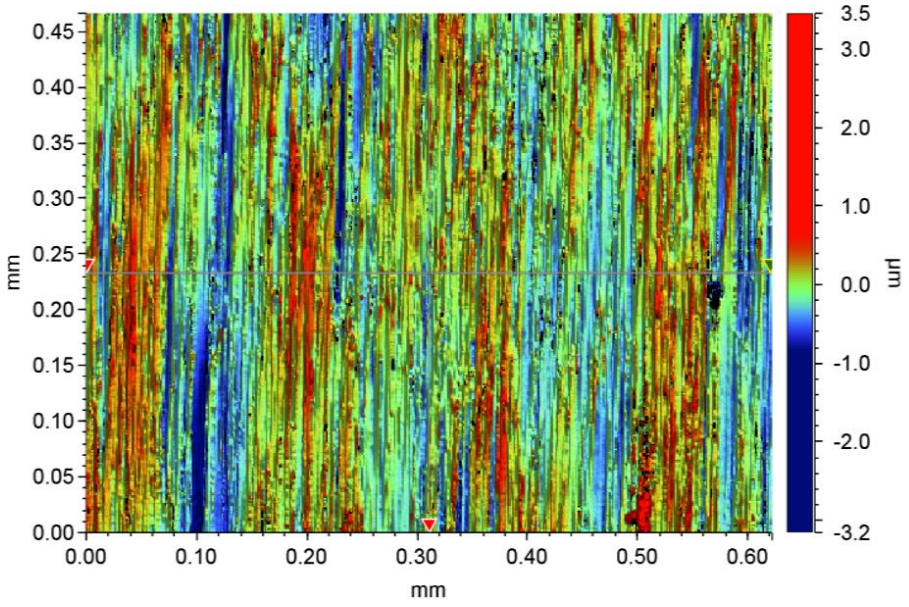


Figure R.12. Component 5, trial 3 (10X magnification).