Fredrik Cappelen

3D Printing and Carbonization of High Performing Polyimide Formulations

Master's thesis in Mechanical Engineering
Supervisor: Jan Torgersen
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Kunnskap for en bedre verden

Fredrik Cappelen

3D Printing and Carbonization of High Performing Polyimide Formulations

Master's thesis in Mechanical Engineering
Supervisor: Jan Torgersen
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

Stereolithographic 3D printing (SLA) is a mature additive manufacturing technology and is gaining traction for the production of certain end-use parts. In this thesis, the possibility of using SLA to produce the gas diffusion layers found in proton exchange membrane fuel cells is explored. A study of the existing literature is presented, which indicates that the resolution of SLA is compatible with the design requirements for gas diffusion layers. A technique for efficiently designing lattice structures with sufficient resolution and a large number of features have been developed. This has been done by avoiding 3D modeling and instead directly producing 2D cross sections with a Python script. Comparisons show that this technique could be orders of magnitudes faster than conventional 3D modeling. Further, the technique allows for precise tuning of the energy dosage through pixel-based intensity control, this enables 3D printing of lattice structures with beams only one pixel wide (50 µm), separated by only three pixels (150 µm) gaps on a commercially available a DLP-SLA printer. The thesis also investigates two polyimide-based materials as candidate materials for the 3D printed gas diffusion layers. The results indicate that it is possible to carbonize UV-curable polyimide, but also that more work is needed to determine if the material is suitable.

Sammendrag

Stereolitografisk 3D-printing (SLA) er en moden form for additiv tilvirkning, og blir i økende grad brukt til produksjon av deler som inngår i et ferdig produkt. I denne oppgaven utforskes muligheten for å bruke SLA tilå produsere gassdiffusjonslagene som inngrår i proton-utvekslende brensel-celler. En studie av eksisterende litteratur presenteres, og denne studien indikerer at oppløsningen som er oppnåelig med SLA er kompatibel med designkravene for gassdiffusjonslag. Teknikker for effektiv generering av gitterkonstruksjoner med slik oppløsing har blitt utviklet. Dette er gjort ved å hoppe over 3D modellering som et steg i prosessen og isteden produsere 2D tverrsnitt med et python script. Sammenligninger som er utført viser at denne teknikken er størrelsesordener raskere enn vanlig 3D-modellering. Videre tillater teknikken nøyaktig innstilling av energidoseringen gjennom pikselbasert intensitetskontroll. Dette muliggjorde utskrift av gitterkonstruksjoner med bjelker bare en piksel brede (50uM), adskilt av mellomrom kun tre piksler (150uM) brede på en kommersielt tilgjengelig DLP-SLA 3D-printer. Denne masteroppgaven undersøker også to polyimidbaserte materialer som kandidatmaterialer for de 3D-printede gassdiffusjonslagene. Resultatene indikerer at det er muligå karbonisere UV-herdbar polyimid, men også at mer arbeid er nødvendig for å avgjøre om materialet er egnet.

Preface

This thesis marks the final stretch on a long and interesting journey to become an engineer. After completing my (second) Bachelor's degree I still had an urge to learn new subjects and become qualified for interesting jobs in my professional life. Throughout this education, additive manufacturing technologies has stood out as an especially exciting field. While we are still in the early stages of understanding and utilizing this technology, its endless possibilities are being demonstrated at an increasingly rapid rate over several industries. At first I was intrigued by how additive manufacturing removes certain obstacles between a designer and a product. This enables creativity, focusing on performance and and help bring ideas into life which previously stranded as a sketch in a notebook. I am also excited that additive manufacturing can boost competitiveness of high cost countries such as Norway, as it levels the playing field in terms of labor costs. I am lucky to have met people who share my interest in additive manufacturing. They have made it possible to write both my Bachelor and Master thesis within this field.

First and foremost I want to express great appreciation to my supervisor Jan Torgersen. He has been an inspiration and my biggest supporter since I started working on this project. Jan always have a great advice or know people that can help. Among those people are my co-supervisor Robert Bock, which has been investing in the project since day one, with his time, advice and resources for the carbonization experiments in this thesis, thank you Robert. Thanks to the guys from W2P and especially Simon Gruber which came all the way to Trondheim to teach us their 3D printer and Sebastian Schüller who answered every question I have sent his way in record time. Thank you to Liang Wang for conducting the TGA experiments and to Sulalit Bandyopadhyay for helping me in the chemistry lab.

On a personal level I would like to thank my friends both at NTNU and HiOA. I have relied on your assistance and friendship to get through so many years in university. Thank you to my family for providing support and security for me to pursue my interests, and last but not at all least, thank you to my girlfriend Silje, which has been there for me for more than a decade, and for the past two years has been home waiting for me on while I have been studying in Trondheim.

The thing that differentiates scientists is purely an artistic ability to discern what is a good idea, what is a beautiful idea, what is worth spending time on, and most importantly, what is a problem that is sufficiently interesting, yet sufficiently difficult, that it hasn't yet been solved, but the time for solving it has come now.

Savas Dimopoulos

Table of Contents

List of Tables

List of Figures

Abbreviations

AM	=	additive manufacturing
CL	=	catalyst layer
Cd	=	cure depth
DLP	=	digital light processing
Ed	=	energy dosage
GDL	=	gas diffusion layer
IPA	=	isopropyl alcohol
MPL	=	microporous layer
MPS	=	macroporous substrate layer
PAA	=	polyamic acid
PEM	=	proton exchange membrane
PEMFCs	=	proton-exchange membrane fuel cells
PI	=	polyimide
PMDA-ODA	=	pyromellitic dianhydride oxydianaline = Kapton
SLA	=	stereolithography
SLA-DLP	=	stereolitography based on DLP technology
TGA	=	thermograviemtric analysis
uSLA	=	microstereolitography
3D	=	three-dimensional

Chapter 1

Introduction

1.1 Background

Fuel cell technology is a promising candidate to replace fossil fuels in several applications due to their lack of harmful tailpipe emissions in combination with much higher energy density than current battery technology. If fuel cells are to become a viable alternative to both fossil fuels and battery technology, several technical challenges must be solved in order to lower their production cost and increase efficiency and longevity.

The gas diffusion layer (GDL) is an integral part of a proton exchange membrane fuel cell (PEMFCs) and consists of a 100-500 µm thick porous membrane inside the membrane electrode assembly. It governs gas diffusion into the catalyst layer while providing protection and electrical conductivity. Its design and production is cumbersome as it has to fulfill many requirements. For now, there is limited control over the design, which could be overcome by the use of high-resolution stereolithography (SLA) 3D printing of high performing polymers.

Like most AM processes, SLA is useful to produce 3D parts directly by joining materials layer by layer, which offers the possibility to build parts with geometric complexities that could not be formed by other traditional processes. SLA is known for its high resolution and can be designed to have reasonable throughput. The limitation of SLA has been the material catalog, which has been limited to commodity plastics. This is now about to change, as chemists and engineers are finding ways to print high-end engineering grade polymers such as polyimide (PI) with SLA. PIs are among the highest performing engineering polymers and has been documented by several researchers to become a good electric conductor after successful carbonization, which makes it a very exciting candidate material for a 3D printed GDL.

1.2 Objectives and Thesis Structure

1.2.1 Scope

The motivation of this thesis is to explore the feasibility and challenges of creating a 3D printed GDL. Because this is a broad challenge, the scope is narrowed down to:

- Understanding and improving the achievable resolution of DLP-SLA technology
- Evaluating if 3D printed polyimide can be carbonized, to obtain the desired electrical conductivity necessary for a gas diffusion layer.

1.2.2 Goals

The scope has been broken down to a set of sub-goals:

1. Finding techniques for designing and producing lattice structures with features sizes of 50μm with the departments SLA 3D printer.
2. Explore the maximum unsupported distance of beams with 50 µm rod width printed in BASF x004m photo-resin on the same printer .
3. Investigate and carry out carbonization of two polyimide based materials.
4. (Bonus) Measurement of electrical conductivity of carbonized materials.

1.2.3 Structure

The thesis is organized into six chapters:

1. Chapter one is an introduction which explains and summarizes the most significant aspects of fuel cell and SLA technology. Several of the sections in this chapter is reused from the literature study performed in the specialization project.[1] These sections will be marked with a reference in the **section title**.[1]
2. Chapter two describes techniques that have been specifically developed by the author for generating lattice structures that are suitable for a GDL.
3. Chapter three is an experimental section that documents methods and results related to goal 1.
4. Chapter four is an experimental section that documents methods and results related to goal 2
5. Chapter five contains an experimental section related to goal 3.
6. Chapter six contains the conclusion and recommendations for further work.

1.3 Fuel Cells

1.3.1 Fuel Cells in General[1]

A fuel cell is a device that generates electricity by converting chemical energy into electric energy by electrochemical reactions. Since the fuel cell process does not involve combustion, it is more efficient and quieter than the equivalent-power thermal generator.[2] It is also a clean technology because the by-products of the electrochemical reaction are only water and heat when pure hydrogen is used as fuel. Unlike batteries, there is no chemical transformation of any component of the fuel cell device during operation, and it can generate power without recharging as long as it is being fed with fuel. There exist many types of fuel cells, but they all consist of two electrodes called the *anode* and *cathode* separated by an electrolyte as illustrated in Figure 1.1.

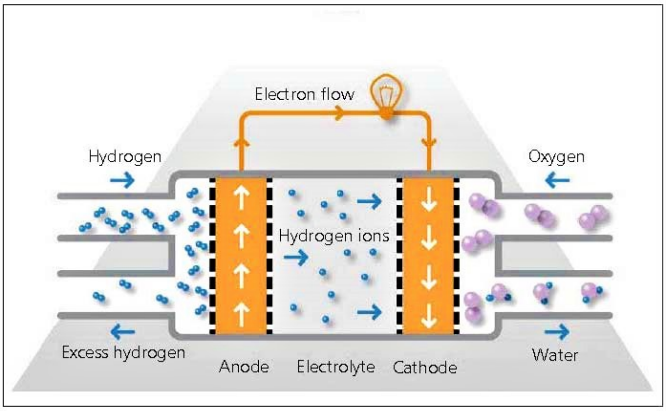


Figure 1.1: Fuell Cell Conceptual Illustration

1.3.2 Proton Exchange Membrane Fuel Cells (PEMFCs)[1]

PEM technology was invented by General Electric in the 1960s through the work of Thomas Grubb and Lenoar Niedrach. Early uses of the technology were NASAS piloted space program, where they tried to replace batteries to accommodate the longer missions.[3]

PEMFCs work with a polymer electrolyte in the form of a thin, proton conducting sheet. Usually, PEMFCs operate at temperatures of about 100°C, and efficiency is ranging from about 30-60%, and the specific power ($\frac{W}{kg}$) is higher than any other type of fuel cell.[4] The solid electrolyte means they are not subject to leaks, and the relatively low temperature and fast start-up time make them the most promising candidate for replacing gasoline and diesel as a power source for cars. However, PEMFCs require purified fuel, and a platinum catalyst is used on both sides, which make them expensive.

MEA[1]

The heart of a PEMFC is the Membrane Electrode Assembly (MEA), which is illustrated in Figure 1.2). The MEA is the part of the fuel cell where power is produced. It consists of a multi-layer sandwich composed of catalyst layer (CL); where electrochemical reactions occur, a gas-diffusion layer (GDL); enabling fuel and oxidant transport, water management and electron/heat conduction, and a proton-exchange membrane (PEM); responsible for conduction of protons and separating the reactions occurring in the anode and cathode. The membrane electrode assembly (MEA) is the governing factor for a PEMFCs efficiency, durability and cost.[5] The more efficient a fuel cell is, the smaller it can be, which in turn lowers the usage of the expensive platinum catalyst. In the next sections, the main factors that affect efficiency in PEMFCs will be described.

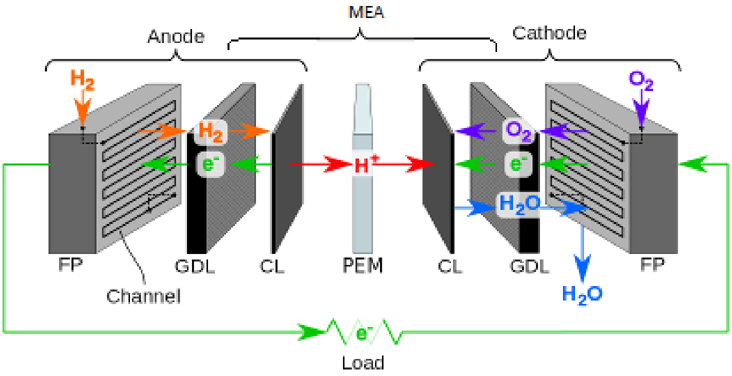


Figure 1.2: Membrane Electrode Assembly

1.3.3 Basic Fuel Cell Theory[1]

At the atomic scale, high energy reactants bonds are broken, and lower energy product bonds are formed, which is an energetically favorable (more stable) state. PEMFCs reacts hydrogen (H_2) with oxygen (O_2) to produce electricity, and water as a by-product. The reaction goes like this:

$$\mathrm{H_2} + {}_{1/2}\mathrm{O_2} \longrightarrow \mathrm{H_2O}$$

By separating the hydrogen and oxygen reactants, the overall reaction is split into two electrochemical reactions:

$$\mathrm{H_2} \longrightarrow 2\mathrm{H}^+ + 2\mathrm{e}^-$$

$${}_{1/2}\mathrm{O_2} + 2\mathrm{H}^+ + 2\mathrm{e}^- \longrightarrow \mathrm{H_2O}$$

These bonds are broken and formed by the transfer of electrons. The separation is accomplished by placing an electrolyte between the anode and the cathode. An electrolyte is a material that allows ions to flow, but not electrons. The electrons instead flow through an external circuit, which can be harnessed as an electrical current to perform work.

The exploitable energy potential of the fuel cell reaction is the Gibbs free energy (also known as free enthalpy), which is the enthalpy difference between the reactants and the product, minus the energy lost to the surrounding environment.[6]

$$\Delta G = \Delta H - T\Delta S \tag{1.1}$$

The Gibbs free energy is related to the fuel cells potential to do electrical work:

$$W_{elec} = -\Delta G \tag{1.2}$$

$$W_{elec} = U \times Q \tag{1.3}$$

where U is voltage and Q is charge in coulombs. For electrons, the charge is equal to the number of moles of electrons and F is the Faraday's constant:

$$Q = nF \tag{1.4}$$

Combining 1.2, 1.3 and 1.4 we get:

$$\Delta G = -nF\Delta U \tag{1.5}$$

$$U = -\frac{\Delta G}{nF} \tag{1.6}$$

For the hydrogen-oxygen reaction the Gibbs free energy change is -237kJ/mol.[7] The theoretical, electrical potential/voltage is then:

$$U = \frac{-237kj/mol}{(2mol\, e^{-}/mol\, reactant)(96,485kj/Vmol)} = -1.23V \tag{1.7}$$

Fuel Cell Losses[1]

From a purely thermodynamic point of view, a fuel cell would supply any amount of current, as long as it is supplied with enough fuel, in reality, fuel cells fall short of this theoretical limit based on thermodynamics, due to several kinetic barriers. This behavior is visualized in Figure 1.3 that shows voltage output for a given current output (solid) compare to the theoretical ideal (dashed). The current output is normalized by the area of the fuel cell to be able to compare fuel cells of any size since larger fuel cells can produce more electricity.[6]

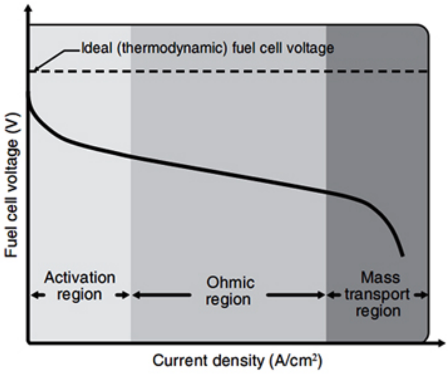


Figure 1.3: IV-curve for typical fuel cell

There are three major types of losses in fuel cells which gives the iv-curve it's characteristic shape:

1. Activation losses due to electrochemical reaction
2. Ohmic loss from electrical and ionic resistance
3. Concentration losses due to mass transport limitations.

The real-life output voltage can then be written as:

$$V = E_{theoretical} - \eta_{activation} - \eta_{ohmic} - \eta_{concentration} \tag{1.8}$$

These losses are paramount to understand the design of fuel cells and especially gas diffusion layers so a brief explanation of these concepts will be provided in the following sections.

Activation Loss[1]

Activation loss arises from the fact that to get the reaction going, one has to push the reactants through a higher energy (non-favourable) state both on the anode and cathode side of the fuel cell. This creates a hump in the overall downhill (Figure 1.4) reaction, which must be overcome and is known as activation overvoltage.[8]

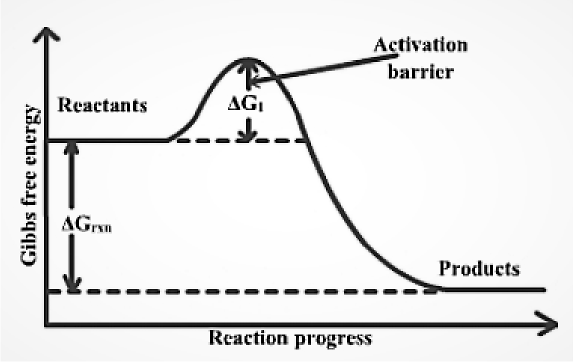


Figure 1.4: Illustration of the activation barrier

The rate of which it is overcome is described by equation 1.9 which is derived from the ”Buttler-Equation”[9]

$$J_0 = nFC_R t_1 e^{\frac{-\Delta G_1}{RT}} \tag{1.9}$$

where C_r is the reactant contentration, t_1 is the decay rate to products, R is the gas constant, ΔG_1 is the activation barrier, T is temperature. A high J_0 indicates a facile reaction, while a low J_0 indicates a sluggish reaction. To increase J_0 there are essentially four options[10]:

1. Increase reactant concentration, C_r (pressurize reactants)
2. Increase reaction temperature, T (limited by polymer membrane)
3. Decrease the activation barrier, ΔG_1 (employing a catalyst)
4. Increase the number of reaction sites (increase surface area)

The three first can be read directly from the equation, while number 4 requires to know that J0 represents current density, meaning reaction per unit area. By fabricating 3D structured electrodes with a high degree of roughness, one can increase the surface area several orders of magnitude for a given size and thereby increase the total reaction rate.

Ohmic loss[1]

Ohmic losses in fuel cells are straight forward: transporting charges induces a penalty that manifests as a loss in cell voltage. This is because fuel cell conductors are not perfect conductors and hence have a conduction resistance. It is called Ohmic loss because the amount of voltage loss follows Ohm's law[11]: $V = iR$, where R can be rewritten as $L/A\sigma$ intot:

$$V_{ohmic} = i \times \Big(\frac{L}{A\sigma}\Big) \tag{1.10}$$

From this equation, it is easy to see that to minimize this type of loss, one can essentially do three things:

1. Increase conductivity, σ
2. Increase area, A
3. Reduce conductor length, L

For this reason, fuel cells are generally made as thin as possible. The ohmic resistance is additive and comes from the combined resistance of the electrodes, electrolyte, interconnects, etc. However, it is usually dominated by the electrolytic resistance because ionic conductivity in good electrolytes generally is orders of magnitude lower than electron conductivity in good electrodes. Usually, the ionic conductivity of electrolytes used in PEMFCs (Nafion) is dominated by water content, meaning they must be kept wet to work well.[12]

Concentration Loss[1]

To produce electricity a fuel cell must have a continuous supply of reactants (H_2 and O_2). Simultaneously, products (H_2O) must be removed to avoid blocking the fuel cells access to reactant. As mentioned in the section on activation loss, fuel cell performance is affected by the reactant concentrations. Depletion of reactant or accumulation of products at the reaction sites will lead to a loss in performance, which is known as concentration or mass transport loss. It primarily occurs when one tries to extract large amounts of current from the fuel cell.

Mass transport loss in fuel cells occurs in the flow channels (FP in Figure 1.2) and the gas diffusion layers, but because they are so different in scale, they are usually treated separately. The flow channels consist of well-defined channels that are on the millimeter and centimeter scale, and therefore follow the laws of fluid dynamics. The GDL, which is the focus of this project is usually 100-500 μm thick and consists of pores and structures on a micro or nanoscale that are full of twists and bends. These structures insulate gas molecules from convective forces, which means that transport is dominated by diffusion, meaning that the gas molecules move from a region of higher concentration to a region of lower concentration by a gradient in chemical potential.[13]

The diffusion process can be described as:

$$J_{diff} = -D\frac{dc}{dx} \tag{1.11}$$

where J_{diff} is diffusion flux, D is the diffusion coefficient and d $\frac{dc}{dx}$ is the change in reactant concentration as one moves through the thickness of the diffusion layer. The pore walls of the GDL impedes the diffusion process, and hence the diffusion flux is lower than indicated by the difference in chemical potential. This is compensated for by calculating what is called an "effective diffusivity":

$$D^{eff} = \varepsilon^{\tau} * D_{ij} \tag{1.12}$$

Where ε is porosity, τ is how tortuous (twisted) the structure is and D_{ij} is the diffusive coefficients which is governed by the molecular weight of the gasses involved, temperature and pressure.[13] The diffusive flux is therefore a product of the concentration gradient and the effective diffusivity. This can be expressed as:

$$J_{diff} = -D^{eff}\frac{c_{R_*} - c_{R_0}}{\delta} \tag{1.13}$$

where c_{R_*} is reactant concentration at the catalyst layer (CL, Figure 1.2), c_{R_0} is the reactant concentration in the flow channel and δ is thickness. If we rearange to solve for c_{R_*}

$$c_{R_*} = c_{R_0} - \frac{\delta \times J_{diff}}{D^{eff}} \tag{1.14}$$

From this, one can see that in order to keep the catalyst layer from depletion, one can:

1. Ensure that c_{R_0} is high, by ensuring good flow and even distribution in flow channel.
2. Increase D^{eff} by optimizing the porosity, tortuousness and conditions such as pressure and temperature.
3. Reduce δ, the thickness of the GDL.

1.3.4 Electrode Design[1]

Fuel cell electrodes serves three main purposes in the fuel cell: they must efficiently transport electrons, reactant gasses and water in and out of the fuel cell. In addition, high catalytic activity is desired in the vicinity of the electrode - electrolyte interface. To meet the described requirements with a minimum of losses as discussed in section 1.2, PEMFCs usually utilize a dual-layer or triple-layer configuration.

Dual Layer Configuration[1]

The dual-layer configuration consists of the CL and a GDL as illustrated in Figure 1.2. The CL is typically $10 - 30\mu m$ thick and is made from nano-sized carbon particles coated in a platinum catalyst. This layer is usually deposited directly on the surface of the electrolyte. The CL is where the electrochemical reaction takes place, and more specifically, it happens in the boundary between the catalyst layer, the PEM and the reactant, known as the ”triple phase boundary”. A much thicker (100-500 µm) porous and electrically conductive layer without catalyst is then bonded on top of the catalyst layer. This layer is usually what is referred to as the GDL. Its purpose is to permit gas to diffuse into the catalyst layer while providing protection and electrical connectivity. The GDL also plays a significant role in determining the removal of liquid water from the cathode side of the fuel cell. For this reason, the GDL is sometimes referred to as the ”porous transport layer”, which more accurately reflects its physical shape and combined role of transporting liquids and electrons in addition to gases.

Triple layer Configuration[1]

A ”triple layer configuration” (Figure1.5) splits the GDL into a macroporous substrate (MPS) layer, usually made from carbon cloth or carbon paper and a microporous layer (MPL). The MPS is in contact with the gas flow channels and primarily serves as a gas distributor and current collector, while the MPL which consists of fine-grained carbon powder coated in a hydrophobic coating primarily controls water flow. Adding an MPL has two main benefits: It reduces ohmic contact resistance by creating a flat and uniform surface that is not permeable to the CL’s platinum particles. Secondly and most importantly according to literature, it improves water management because the MPL acts as a valve, keeping excess water out of the CL on the cathode side and hence improve access to oxygen during high loads.[14]

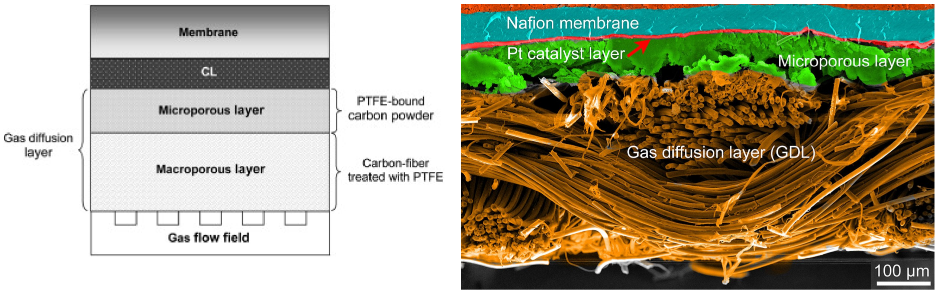


Figure 1.5: Conceptual triple layer configuration left and real life implementation right

GDL materials, production and characteristics[1]

Commercially available GDLs are made from carbon-based products such as a woven carbon cloth or non-woven paper. The primary characteristics that make carbon cloth and paper suitable for a GDL are: [15]

1. Stable in an acidic environment
2. Can provide high gas permeability
3. Good electronic conductivity
4. Elastic on compression. This means it can allow expansion and shrinkage without generating too much stress on the catalyst layer and electrolyte.
5. Can to some degree provide controlled porosity

Carbon GDLs are made from a precursor fiber called polyacrylonitrile (PAN). PAN fibers are formed using a spinning process and subsequent carbonization step involving temperatures in the range of 1000-1700°. Also, the GDL is usually wet-proofed so that the surface and pores are not clogged with liquid water.

The exact manufacturing process will be slightly different for a woven cloth and a non-woven carbon paper, but both require multiple steps of sizing operations, further carbonization/graphitization treatments, different coatings and bonding steps which drives up cost and lead time. In Figure 1.6 the production process of a commercially available GDL is shown. [16]

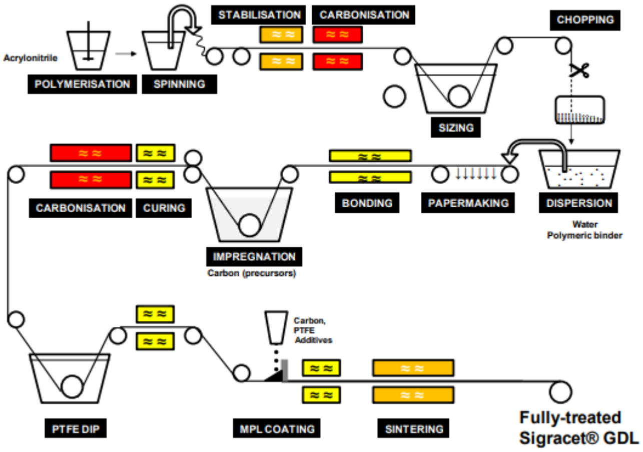


Figure 1.6: traditional carbon based GDL manufacturing process [16]

For the GDL, metal-based versions in the form of meshes, foams, micro-machined sheets, and even 3D printed structures have been researched [17,18,19] and while they have potential benefits, oxidization in the low pH water environment of a fuel cell is still a problem.

1.3.5 Improving the GDL

What an ”optimal” design of a GDL actually should look like is a subject of undergoing research and not a settled issue. What can be said is that the GDL will have the most influence on ohmic loss and mass transport loss. Ohmic loss is simply a function of conductivity, thickness, and area. In this regard, the GDL should be thin and have large cross-sectional area. 3D printed fuel cells could, potentially, have an advantage in this regard because it will be manufactured as a continuous material instead of separate strands of carbon fiber. This reduces the number of contact points and will not rely as much on contact pressure to obtain proper conductivity.

Mass transport optimization is more challenging to optimize for as water can exist both as a liquid and gas depending on temperature and pressure in the MEA. A study by Benziger et al. concluded that water transport across the GDL mainly happens through the largest pores and that only a few perfect of the void fraction is necessary to facilitate this transport.[20] This leads them to a recommendation of a bi-modal design that features a few large channels for water transport, which keeps the smaller pores upon for gas transport. To maximize the effect, they recommend that the smaller pores should be highly hydrophobic in nature, while the larger pores should have a contact angle near 90°and straight, smooth walls to minimize the pressure needed to push water through the pores. Other researchers have documented the benefits of a porosity gradient near the CL in order to create a gradient in capillary pressure that creates a driving force for water flow.[21] [22] While the optimum design of a GDL is not a settled subject, it is clear that having a production process that can produce all the necessary variations in geometry in one single process would be advantageous both for reaching optimal performance and from a cost perspective.[23] In Figure1.7 a conceptual model combining a porosity gradient with a larger channels is shown. High-resolution SLA-3D printing in suitable materials could be an answer to this challenge.

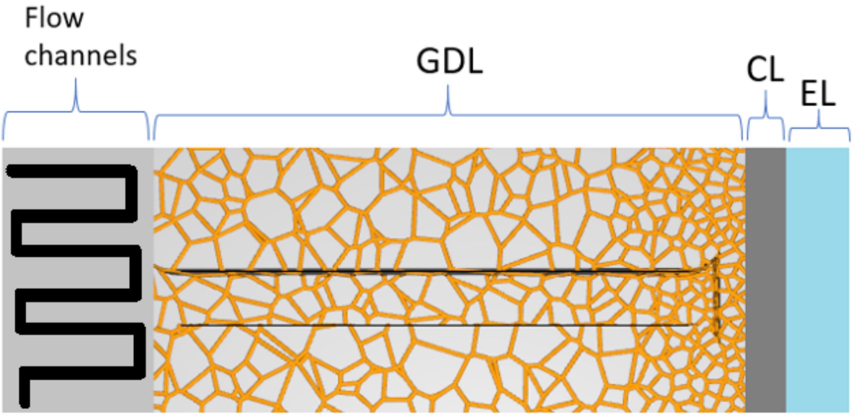


Figure 1.7: Interpretation of an ”optimal” GDL (orange) featuring a porosity gradient and a large smooth channel for water transport (black)

1.3.6 Summary of Fuel Cells and GDLs

From this introduction of fuel cell technology, the fundamental principles of fuel cells and GDLs have been outlined. These principles explain why a GDL is made from thin porous structures predominantly made from types of carbon. If we also look at the specifications of commercially available GDLs[24] [25] [26] one can describe some approximate specifications a 3D printed GDL should target to be competitive:

1. A thickness in the range of 100-500 $\mu\mathrm{m}$. Thinner is usually better.
2. A porosity in the range of 50-90% depending on the application.
3. Mean pore diameter in the range of 20-60 $\mu\mathrm{m}$ (excluding the MPL)
4. Electrical through-plane resistance below $20m\Omega cm^2$
5. Ability to vary pore diameter to produce a porosity gradient
6. Ability to incorporate larger smoother channels

With those targets as a backdrop, the next part of the introduction will evaluate the capabilities of SLA based 3D printing.

1.4 Stereolithography

1.4.1 SLA Working Principle[1]

Stereolithography (SLA) is an additive manufacturing (AM) technology that converts liquid materials (resins) into solid parts through a process called photopolymerization. Layer by layer, a cross-section of the part is cured by selective exposure from a light source. The resins incorporate photo initiator molecules which breaks into free radicals when they absorb photons. The radicals work as a catalyst which makes the oligomers in the resin combine and forms long chains (polymers). In the process the chains become cross-linked, and the material solidifies.[27]

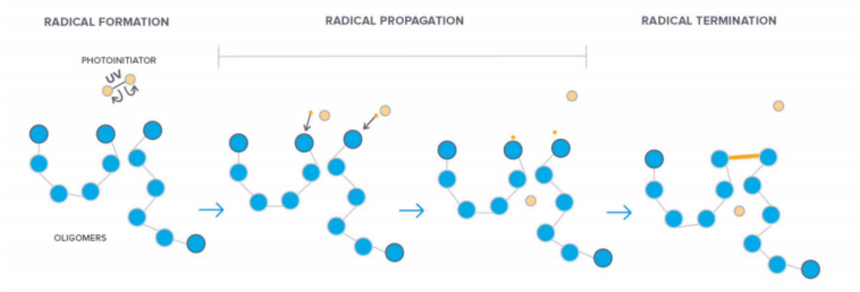


Figure 1.8: Free radical photopolymerisation[28]

1.4.2 Common Usage of SLA

SLA technology has been successfully utilized in industries with a need for mass-customization such as the hearing aid and dental industry, where it has become a dominating manufacturing technology.[29] Other products which benefit from customization such as shoes and shoe soles have recently be introduced in limited quantities and are expected to follow a similar trend.[30]

1.4.3 SLA Technologies[1]

Three main categories within the SLA technology exist today (Figure 1.9)

- Laser-based SLA, or point-vise approach: a laser beam is scanned over the layer curing one point at the time. The benefit of the point-wise method is that the resolution is independent of the scanning area or said in other words: the size of the printed part.
- DLP-SLA: a UV projector much like those found in a cinema projects an image of an entire layer across the build platform. Because the image is composed of square pixels, each layer will be formed from small rectangular bricks called voxels

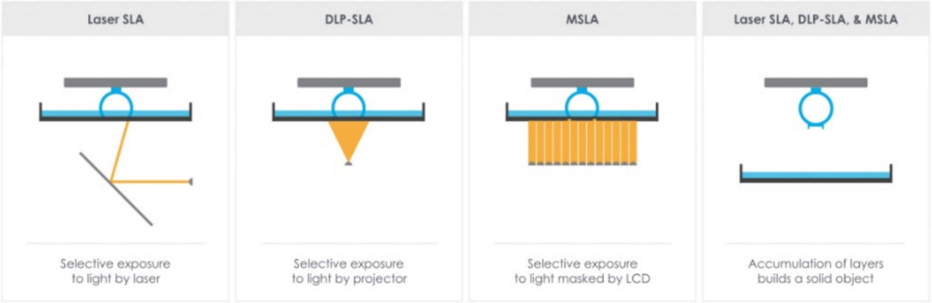

Laser SLA
DLP-SLA
MSLA
Laser SLA, DLP-SLA, & MSLA
Selective exposure to light by laser
Selective exposure to light by projector
Selective exposure to light masked by LCD
Accumulation of layers builds a solid object

Figure 1.9: Main types of SLA [31]

that are stacked up along the z-axis as illustrated in Figure 1.10. Controlling the area to be exposed in this manner obviates the need for any scanning at all, thus increasing throughput and reducing the number of moving parts, but resolution gets worse when projecting over a large area.

- Mask-projection (MSLA) A photo masking device (usually an LCD display) are used to control the light exposure from a large UV-LED array. As with DLP, the pixels are square, but unlike the DLP one cannot zoom or change the focus of the lens, so the resolution is fixed to the resolution of the LCD.

One will often see the Laser-based SLA method is referred to as ”SLA” and the other two as ”DLP-printer” and ”Masked projection printer”. It is also worth noting that DLP printers capable of moving the light source exist. This means it can focus on several smaller areas and stitch them together and maintain very high resolution also for large parts. This happens to be the case for the printer used in this project.

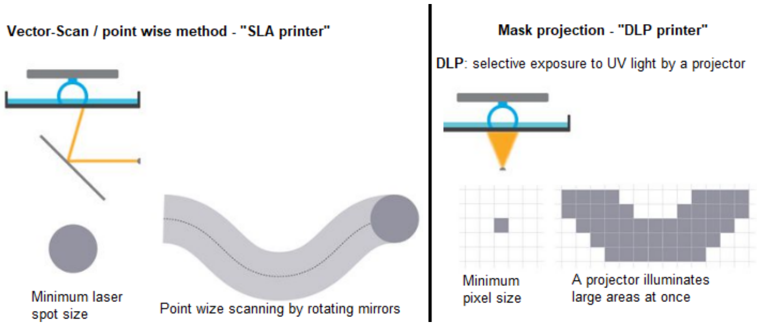

Vector-Scan / point wise method - "SLA printer"
Minimum laser spot size
Point wize scanning by rotating mirrors
Mask projection - "DLP printer"
DLP: selective exposure to UV light by a projector
Minimum pixel size
A projector illuminates large areas at once

Figure 1.10: point-vise vs mask projection method fror SLA [31]

SLA machines can be configured in a top-down configuration or bottom-up (inverted) configuration (Figure1.11). The top-down configurations are advantageous in that the parts are not subject to peel forces each time a new layer is formed. This gives higher throughput and makes part orientation straight forward. Unfortunately, top-down machines require a significantly larger volume because the resin container must be large enough to fit the entire part also in the vertical direction. This makes both the initial investment and running cost high, which limits their use to high-end industrial applications. The bottom-up design has a benefit in more precise Z-axis control as the gap distance between the build platform almost solely depends on the Z-actuator accuracy while a top-down configuration also must consider the fluid properties of the resin.

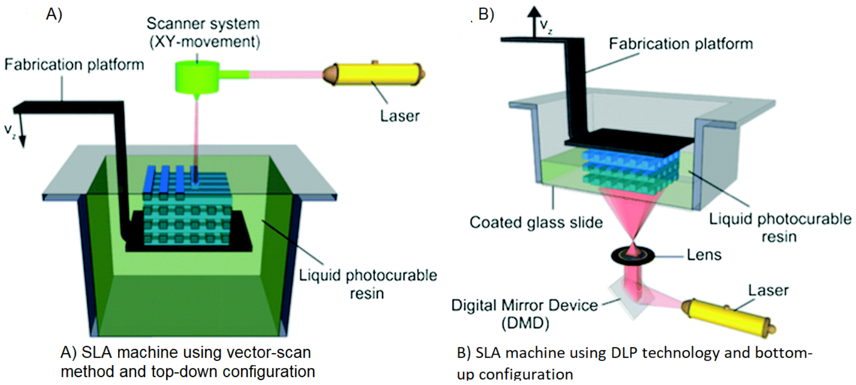

A)
Scanner system
(XY-movement)
Fabrication platform
Laser
v_z
Liquid photocurable resin
A) SLA machine using vector-scan method and top-down configuration
B)
v_z
Fabrication platform
Coated glass slide
Liquid photocurable resin
Lens
Laser
Digital Mirror Device (DMD)
B) SLA machine using DLP technology and bottom-up configuration

Figure 1.11: Top-down vs Bottom-up configuration[31]

1.4.4 Peel Forces

A drawback of the bottom-up design is that forces applied to the part during the peeling step may cause it to detach from the build platform or break the part itself. Equation 1.15 is simplified model (Stefan Adhesion) for approximating the peel force of a cylindrical cross section[32]:

$$F = \int_0^R 2\pi P dr = \frac{3\pi\mu V}{2h^3} R^4 \tag{1.15}$$

we can from the equation see that peel forces increase with a linear relationship to the resin's viscosity (μ), the separation velocity (V) and much faster as a function of increased part dimensions (R) or reduced distance (h) between the bottom of the cured part and the constraining surface. This creates clear limitations in terms of maximum part size and build speed. Numerous techniques are used to mitigate this issue. Among the most common is to orient parts at an angle to the constraining surface, to reduce the cross-sectional area. This requires more support material and in some cases, degrades the accuracy of the part. In addition bottom up printers usually either have a tilting mechanism or a flexible VAT to lower the forces during peeling (see Appendix A.1), and also by using an oxygen permeable constraining surface, as oxygen inhibits polymerization and hence can provide a gap (h) between the part and the constraining surface.[33]

1.4.5 The Basic Physics of SLA[1]

The most fundamental model for understanding SLA technology is the ”working curve.” This curve describes the thickness of the polymer that forms as a result of light absorption. This model is based on two well-known phenomenons; The ”Beer-Lambert Law,” which states that propagation of light decays exponentially when it travels through an absorbing media[34]. The second phenomenon is that a photopolymer undergoes a very rapid transition from liquid to a solid when a certain amount of energy in the form of light has been absorbed. This is known as the critical energy (Ec). The curing begins at the surface of the resin and then grows in thickness away from the light source. This behavior was described mathematically by Dr. Paul F. Jacobs in 1992[35] when he was Director of Research & Development at 3D systems, and resulted in a beautifully simple equation which is known as the Jacobs equation:

$$Cd = D_p ln\left(\frac{Ed}{Ec}\right) \tag{1.16}$$

where Cd is the cure depth, Dp is the depth of light penetration into the resin, Ed is the energy dosage from the UV laser and Ec is the critical exposure energy that initiates polymerization. From the equation, we can learn that:

- The cure depth should scale as the natural logarithm of the laser exposure.
- A plot of Cd on the y-axis and ln(Ed) on the x-axis should result in a straight line relationship, this is known as the Working Curve.
- The slope of the Working Curve is the Dp.
- Since ln(1) = 0, the value of Ed where Cd = 0 is precisely the ciritcal exposure Ec.
- Since Dp and Ec are material properties of the resin, both the slope of the working curve and the value of Ed where Cd = 0 is independent of laser power, laser spot size or the scan velocity.

Jacobs also describe how cured width (Lw) is related to the Cd, Ed and resin properties:

$$Lw = Wo\sqrt{2\,Cd/Dp} \tag{1.17}$$

or

$$Lw = \sqrt{2}\,Wo\,\left[ln(Ed/Ec)\right]^{\frac{1}{2}} \tag{1.18}$$

Where Wo is the radius of the beam, defined as the point where the local irradiance is $1/e^2$ or about 13.5% of the peak irradiance. This tells us that Lw should grow much slower than Cd when increasing Ed.

While these concepts are important for understanding SLA technology, they are based on the theoretical case of a laser with a Gaussian power distribution as illustrated in Figure1.12. For a real-life implementation such as the printer of the type used in this project, the distribution most likely differs from the theoretical Gaussian model due to the shape of the light source and refraction properties of both the resin and the constraining surface in the case of bottom-up machines. DLP printers will also simultaneously illuminate several points in space. This means stray light from one pixel can interfere with neighboring pixels.

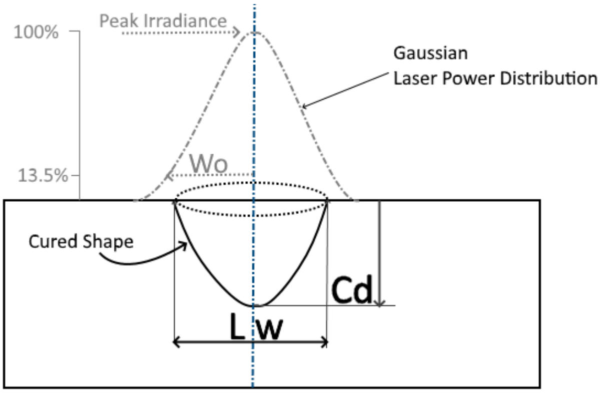


Figure 1.12: Illustration of cured depth and width of a gaussian beam distribution

An example of a working curve is shown in Figure1.13 . Samples exposed to various amounts of energy are printed and their thickness measured. Then a logarithmic approximation is used to generate a working curve. The curve lets one determine needed exposure time for a given laser power or vice versa. Another example is attached in appendix A.2

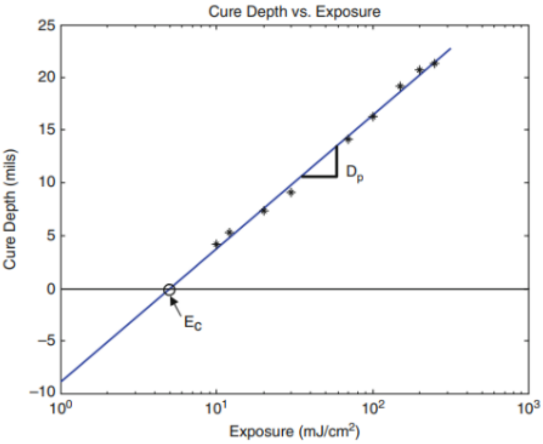


Figure 1.13: An example of a working curve and measured values as stars

1.4.6 DLP-Technology

The DLP technology has emerged as especially promising for certain applications because its speed is only depending on the exposure time and not the cross-sectional area. This is because there is no scanning or moving of the light source, as long as the parts fit within the projected image. The exposure time depends on the type of resin and layer thickness. This is a huge benefit when building parts with large cross-sections, but also a benefit for very feature dense parts such as a GDL which would require the point-wise SLA systems to scan and rapidly change direction to cover the thousands of microscopic features. Due to physical constraints such as the inertia of the scanning mechanism and available laser power, there are clear limitations on achievable scanning speed.

The achievable resolution with DLP-SLA is also very high as it can be increased by focusing the available pixels over a smaller area. The pixel count mainly decides how large objects can be made with a given resolution. This is controlled by the number of micromirrors that make up the digital micromirror device (DMD). A DMD consists of millions of electro-mechanically actuated mirrors that can be individually addressed and tilted between two angles (states). DMD devices specifically designed for DLP printers that have 2560x1600 pixels (4million mirrors) are already available[36] and there is little reason to doubt that this will increase as DMD devices with true 4K (8million mirrors) resolution already is available for other applications. Increasing the number of mirrors will enable larger build areas, or reduce the minimum pixel size in applications where this is important such as a gas diffusion layer.

Because the projected image of a DLP printer consists of square pixels, the orientation of the structure is important, especially when the features are not much larger than the pixel size. If the alignment of the feature is at an angle in respect to the DMD-mirrors, one pixel will in reality span multiple mirrors. This is illustrated in Figure 1.14 where the features A and B are oriented orthogonal, and the mirrors (red) are oriented in a diamond pattern leading to a distribution of light over several pixels after the software has tried to determine what mirrors should be activated.[37] The orientation of the mirrors in the DMD can be both orthogonal or diamond-shaped depending on the DLP chipset used.[36] This effect is already accounted for in most printers in the way that the picture is rotated before it gets projected, so simply rotating the part before the slicing stage could have the opposite effect of what one is trying to achieve. This is a lossy process and the mapping will not all ways be perfect as shown in Figure 1.14, where the structure is smeared out over several mirrors distorting the end result. Through experiments with orientation in the semester project, it was found that orienting the printed lattices in an orthogonal orientation and let the printer itself handle the rotation to work by far the best in the case of the 3D printer utilized in this thesis.

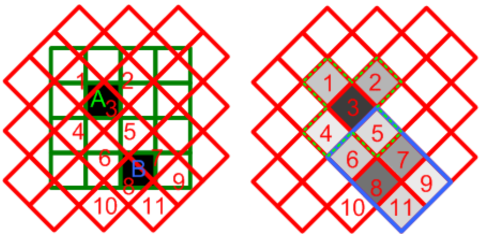


Figure 1.14: The effect of misaligned parts in regards to mirror orientation[37]

1.4.7 State of The Art Micro Scale SLA

We end the introduction with a look at what is currently possible in terms of micro-scale resolution with SLA technology; In 2015 Gong et al. published a study where they through experimental work achieved consistently open flow channels with size of 60 µm x 108 µm on a commercial available DLP-SLA printer with a pixel size of 27 µm and z-axis resolution of 1 µm.[38] In the same paper, Gong et al. created an optical model that predicts that the minimum reliable channel is 3.5-5.5 the penetration depth of the resin. Subsequent research by the same authors published in 2017 achieves flow channels with cross sections as small as 18x20uM.[39] This was achieved using a custom built DLP printer based on a DMD which features 2560 x 1600 resolution. By projecting this image over an area of 19.35mm x 12mm a pixel size of 7.6 µm was achieved. They conclude that in-plane (xy) resolution is mainly a function of the pixel size, and that out of plane resolution (z-direction) mainly is controlled by the resins absorption properties. They uphold that the minimum opening channel has to be designed 4 pixels wide to remain open consistently.

There is also an entire branch of SLA technology known as Micro Stereolithography(uSLA) that has been researched and utilized since the late 90's mainly in the prototyping and fabrication of advanced micro electro-mechanical systems (MEMS). uSLA is based on the same principle as normal SLA and but target even higher resolution[40]. As with normal SLA, DMD based uSLA devices exist and are often preferred due to their much higher throughput. As early as 1999, a study be Belize et al.[41] showed a DMD based device capable of resolution below 5 µm in all directions. The results obtained by Belize et al. support the conclusion that the pixel size controls resolution in the XY plane, while the resin absorption properties control the Z-axis resolution.

The results from published literature indicate that the resolution capabilities of SLA already can meet the requirements of a GDL in (illustrated in Figure 1.15) terms of required feature size. It is more questionable whether the throughput is high enough at these resolutions. In addition, most of the structures found in literature differ from the GDL requirements in terms of porosity, which has to be in the 60-90% range in the case of a GDL. This means that we essentially are targeting a free-standing lattice which is much more delicate structure than that of a flow channel in a solid block of material. Nevertheless, such structures have been demonstrated with uSLA.[42]

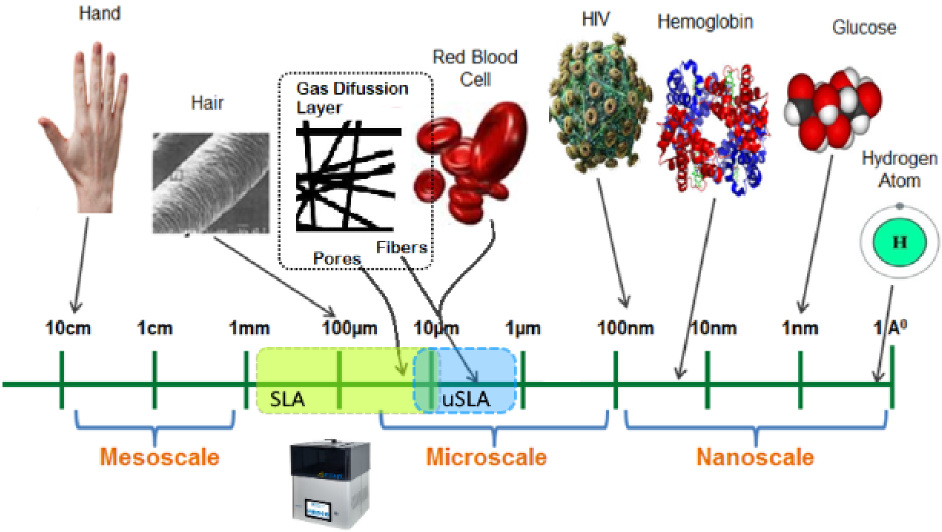


Figure 1.15: Scale of things compared to current SLA resolution

Chapter 2

Lattice Generation

2.1 Introduction

As discussed in the section on Fuel Cells, we need to represent a very high number of features per volume unit to compete with the pore size and surface area of commercially available GDLs. During the work with the project assignment, it became clear that generating such feature dense 3D lattice structures with traditional CAD software is very inefficient. Even a basic rectangular beam will consist of at least 6 faces, 8 vertices, and 12 edges. The number of beams grows at a cubic rate as part dimensions or the feature density doubles. This quickly leads to long processing times and crashing the software due to exceeding the available memory. For rapid and iterative prototyping, traditional CAD software is not well suited. With the advance of AM, advanced lattice structures are more frequently being used, and several new software suits are emerging to meet such needs.

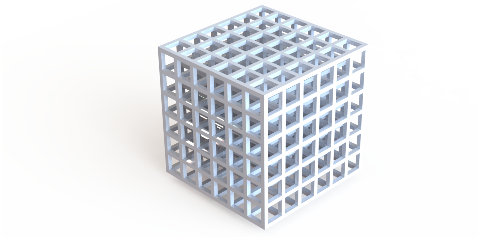


Figure 2.1: A cubic lattice

Of the commercially available alternatives tested in the semester project,[1] Element by nTopology[43] was by far the preferred option. This software is specifically designed to generate complex lattice structures with additive manufacturing in mind. Its speed comes from the fact that structures are represented as mathematical functions in the design stage. Still, the mathematical representation has to be turned in to a solid which later is converted to an STL before it can be sliced and printed. Because of this, memory requirements and file sizes quickly gets out of hand when we move from the macro universe and into the micro universe, also in the case of nTopology.

2.2 Lattice Generation with Python

To solve this problem, a script that bypasses the 3D stage and instead directly outputs 2D PNG files was written in Python. This approach seems logical, as we are working with a technology that projects 2D images and the default input format is PNG.

Cubic lattices, as seen in Figure2.1 became the natural starting point in order to maximize the available printing resolution. This is because it is possible to align the beams of a cubic lattice with the pixels of the printer so that one beam only spans one pixel (see Figure1.14).

A cubic lattice is essentially just a collection of straight beams joined at certain intervals. Hence it is straight forward to draw them even with basic image processing libraries available for Python such as PIL.[44] The script draws a collection of lines and rectangles of a certain thickness and length at certain intervals dictated by the user input, one layer at the time. Each layer is then saved as a PNG and a corresponding XML file that tells the printer what height the current layer is gets appended for each layer. When all layers have been generated, the images and the XML file is compressed to a zip archive which is directly readable by the printer. No further processing is necessary, and the whole process of generating a cubic lattice is done in a matter of seconds.

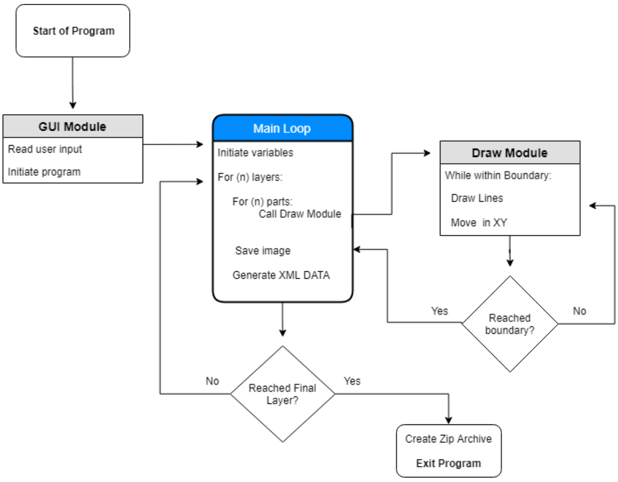


Figure 2.2: Flow chart of the Python Script

2.3 Methods

Two experiments were performed to benchmark the performance of the Python script. In the first experiment, a cubic lattice with fixed dimensions of 12x12x2 mm (X,Y,Z) was generated while the space between beams as illustrated in Figure2.3 was halved for each iteration. With decreasing space between beams, the number of beams pr volume unit increases, which increases processing time, memory requirements, and file size.

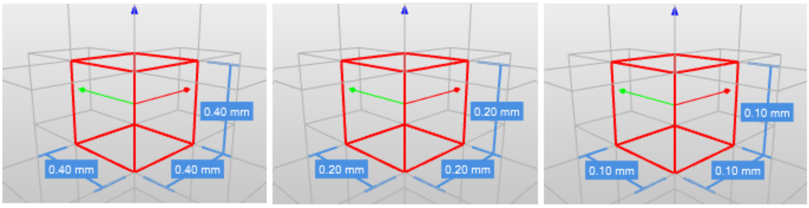


Figure 2.3: Illustration of beam spacing variation in experiment 1

In the second experiment, the spacing between beams was kept constant at 400 µm and instead the part dimensions were doubled in all directions from 12x12x2 to 24x24x4 up to 48x48x8 mm. This gives a cubic increase in volume, and hence a cubic increase in number of features.

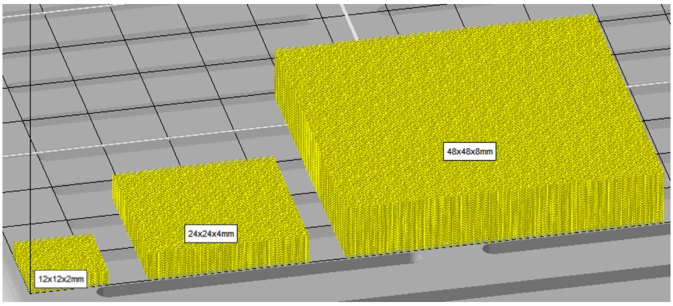


Figure 2.4: Illustration of volume increase between runs in experiment 2

The Python script was compared to nTopology and Netfab Premium, where nTopology generates the structures while Netfab provides the slicing necessary to turn the structure into PNG files. The processing time was recorded with a stopwatch and rounded down to the nearest second. Only the main operations; Generate solid, Convert to STL and Slicing were recorded. A timer was implemented into the Python script. The timer starts once the user has finished giving input and click the run button, it stops when the process is completely finished and the zip file ready for printing has been generated.

2.4 Results and Discussion

Results from experiment 1:

Elapsed Time vs Beam Spacing Element nTopology + NetFab Premium			
Spacing Operation	400 µm	200 µm	100 µm
Generate solid	15s	110s	779s
Convert to STL	1s	13s	99s
Slicing	17s	25s	58s
Total	33s	148s	936s

Table 2.1: Elapsed time for generating lattice of varying beam spacing using nToplogy and Netfab

Elapsed Time vs Beam Spacing Python Script						
Spacing Operation	400 µm	200 µm	100 µm	50 µm	25 µm	12 µm
Total	15.26s	15.3s	18.38s	21.88s	34.4s	107s

Table 2.2: Elapsed time for generating lattice of varying beam spacing with Python Script

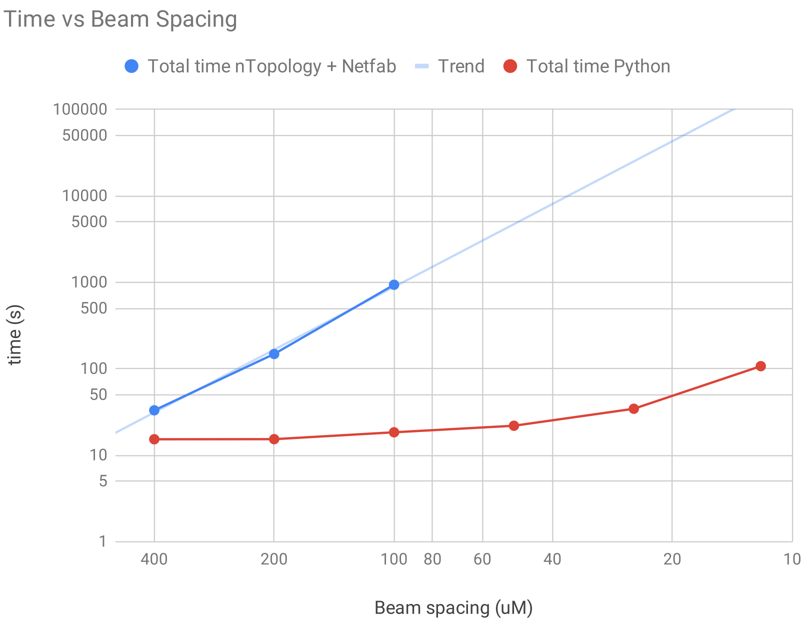


Figure 2.5: Data from table 2.1 & 2.2 plotted on a log/log axis

Results from experiment 2:

Elapsed Time vs Volume Element nTopology + NetFab Premium			
Volume Operation	288 mm^3	2304 mm^3	18432 mm^3
Generate solid	15s	117s	863s
Convert to STL	1s	13s	104s
Slicing	17s	36s	155s *
Total	33s	166s	1122s

Table 2.3: Elapsed time for generating lattice of varying volume with nTopology & Netfab

Elapsed Time vs Volume Custom Python Script				
Volume Operation	288 mm^3	2304 mm^3	18432 mm^3	147 456 mm^3
Total	15.3s	31.8s	75.6s	154s

Table 2.4: Elapsed time for generating lattice of varying volume with Python script

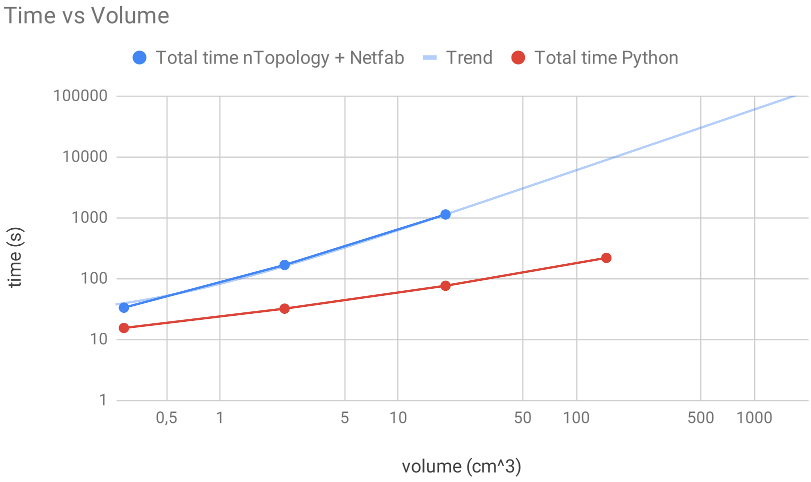


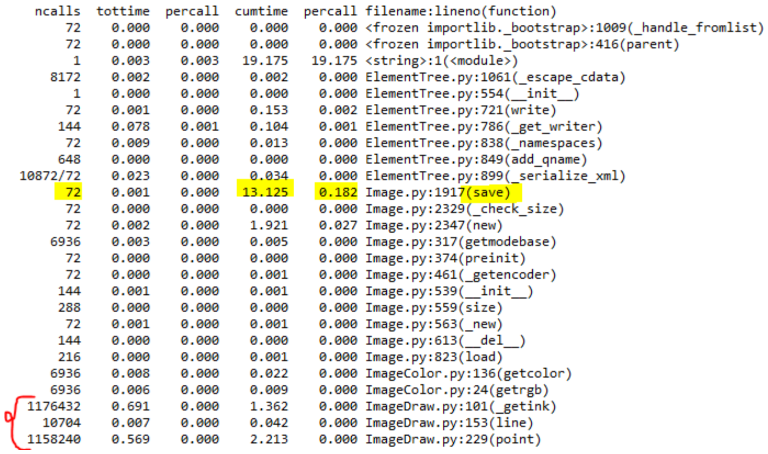
Figure 2.6: Data from table 2.3 & 2.4 plotted on a log/log axis

In Figure 2.5 it can be seen that the Python script has a large advantage as the spacing between beams decrease (number of features increase). At 100 µm spacing, it is 50 times faster and for a beam spacing comparable to a GDLs pore size it is approximately a thousand times faster. Keep in mind that both the x and the y-axis is on a logarithmic scale for readability.

In Figure 2.6 one can see that the increase in total processing time for nTopology + Netfab is close to a 1:1 ratio to the volume increase, which is expected. For the Python Script, the solving time appears to increase much more slowly.

2.4.1 Discussion

The explanation to the observed trends is that the drawing of the features have very little impact on the solving time in the Python script until the feature count becomes extremely high. This can be seen by running a ”profiler” on the code, which produces a timing log for all functions within the script.



```
   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
       72    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:1009(_handle_fromlist)
       72    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:416(parent)
        1    0.003    0.003   19.175   19.175 <string>:1(<module>)
     8172    0.002    0.000    0.002    0.000 ElementTree.py:1061(_escape_cdata)
        1    0.000    0.000    0.000    0.000 ElementTree.py:554(__init__)
       72    0.001    0.000    0.153    0.002 ElementTree.py:721(write)
      144    0.078    0.001    0.104    0.001 ElementTree.py:786(_get_writer)
       72    0.009    0.000    0.013    0.000 ElementTree.py:838(_namespaces)
      648    0.000    0.000    0.000    0.000 ElementTree.py:849(add_qname)
 10872/72    0.023    0.000    0.034    0.000 ElementTree.py:899(_serialize_xml)
       72    0.001    0.000   13.125    0.182 Image.py:1917(save)
       72    0.000    0.000    0.000    0.000 Image.py:2329(_check_size)
       72    0.002    0.000    1.921    0.027 Image.py:2347(new)
     6936    0.003    0.000    0.005    0.000 Image.py:317(getmodebase)
       72    0.000    0.000    0.000    0.000 Image.py:374(preinit)
       72    0.000    0.000    0.001    0.000 Image.py:461(_getencoder)
      144    0.001    0.000    0.001    0.000 Image.py:539(__init__)
      288    0.000    0.000    0.000    0.000 Image.py:559(size)
       72    0.001    0.000    0.001    0.000 Image.py:563(_new)
      144    0.000    0.000    0.000    0.000 Image.py:613(__del__)
      216    0.000    0.000    0.001    0.000 Image.py:823(load)
     6936    0.008    0.000    0.022    0.000 ImageColor.py:136(getcolor)
     6936    0.006    0.000    0.009    0.000 ImageColor.py:24(getrgb)
  1176432    0.691    0.000    1.362    0.000 ImageDraw.py:101(_getink)
    10704    0.007    0.000    0.042    0.000 ImageDraw.py:153(line)
  1158240    0.569    0.000    2.213    0.000 ImageDraw.py:229(point)
```

Figure 2.7: Timing log from Profiler

A screen-shot of the timing log is shown in Figure2.7. Here one can see that factor that dominates the solving time for the Python script is the saving operation (yellow highlight) and not the drawing operations (red bracket). Each saving operation takes 0.182 seconds and happens one time per layer. This explains why the solving time for the Python script is not as affected by the number of features pr volume unit, but more by the number of layers as each layer is saved as a PNG file. A large factor of the saving time of a PNG image is from the compression process. By turning down the compression factor to the lowest setting, the speed of the Python script is further increased with about 40% over the presented results, at the cost of increased file size, which increases from approx 40kB to 110kB per image in this case.

In summary, we can conclude that the Python Script is much faster than its 3D counterpart. Its solving time is most heavily affected by the number of layers which is a result of the parts Z-axis dimension and the chosen layer thickness. For a realistically sized GDL, the Z-axis dimension would be constrained to about 500uM, and the layer-height count would therefore not become extremely high. Hence very dense detailed and large cross sections can be designed efficiently with this technique.

Chapter 3

Pixel-Based Dosage Control

3.1 Introduction

During the work with the semester project, it was discovered that the overall energy dosage (Ed) had a large effect on the printed lattices. At the recommended dosage levels, the designed voids in the lattices solidified, and the result was an essentially solid piece of plastic instead of a lattice structure. This lead to the conclusion that there must be energy also outside the intended areas that cure what was supposed to remain voids. By lowering the Ed, either by reducing exposure time or the laser power, it was observed that the voids became more and more open and the part started resembling the designed lattice. Unfortunately, the strength of the printed parts also was reduced when lowering the Ed, which leads to parts delaminating during printing, or the part collapsing during post-processing. To mitigate this, a reinforcing structure of thicker beams was added to the structures to hold the part together, much like the steel reinforcement of a concrete building. This made the parts more robust and likely to survive at lower dosage levels.

While the main motivation behind the Python script originally was speed and precision, another important feature was unlocked by generating the lattices as PNGs; the ability to control the intensity of individual pixels through manipulation of the assigned RGB value. Every pixel is represented with an RGB value that can be typically be adjusted with 8-bit precision (values between 0-255). This opens up the possibility to alter cure depth, cure width, and modulus for different areas within a layer. This will from now on be referred to as *pixel-based dosage control* (PDC). In addition, this feature allows that many values can be tested in the same print as one is no longer adjusting the global intensity of the printer, which is very time saving when trying to find the perfect balance of parameters.

3.2 Methods

To study the effectiveness of PDC, two different experiments will be presented. The first is PCD's effect on printing a lattice structure with a beams designed with a width of 50 µm, which corresponds to one pixel in the case of our printer. The centre of each beam is spaced 200 µm apart, leaving an intended 150x150x150 µm void between. The layer height used is 25 µm, which means that each beam consists of two layers stacked on top of each other. Vertical beams are represented by illuminating a single pixel. The surrounding frame and large white beams seen in Figure 3.8 is the reinforcing structure mentioned earlier.

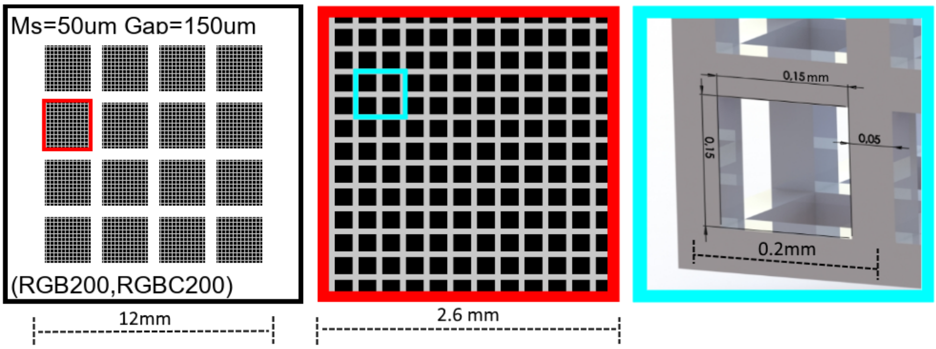


Figure 3.1: Geometry used for testing lattice resolution

The second experiment is a study of PDC's effect on single beams. A beam is again drawn one pixel wide. Each beam is 1 mm long and freely suspended over a channel which is 1 mm deep. The effect of adjusting RGB value is compared to the effect of adjusting the Energy Dosage through changing laser power and exposure time. Layer height is still 25 µm. This means that the first layer prints freely into the resin, while the second layer is constrained by the first layer.

The term default setting will be referred to and means the Ed resulting from the laser power and exposure time recommended by the manufacturer for printing 25 µm layers. The settings are shown in Equation 3.1.

$$8mw/cm^2 * 900ms = 7.2mJ/cm^2 \tag{3.1}$$

3.2.1 Equipment

The printer that is used is a Solflex 650 from W2P (W2P Engineering GmbH).[45] The printer is a bottom-up (ref Figure 1.11) DLP based SLA printer. It has a maximum build volume of 128x120x130 mm in total, which is divided into six printing zones. The projector light up one printing zone at the time. If a part span multiple zones, the projector is moved in the XY plane and a slight seam in the part will be visible where one zone is

”stitched” to the next. The benefit of this arrangement is that both high resolution and support for large parts can be had in the same machine. The pixel size is 50 µm and essentially represents the limit of how small features that is possible to produce with this printer. The printer supports two types of VATs. In the presented work, an FEP foil type VAT has been used. The FEP sheet is 127 µm thick and was bought from FEPshop.[46] A datasheet for the printer can be found in Appendix D.1.

The pictures and measurements are taken using using an Olympus BX53M light microscope. Measurements were one with the digital measuring tool available in the Microscope’s software.

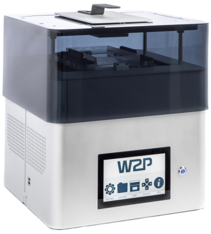


Figure 3.2: Solflex 650

The material that has been utilized in the testing is BASF Photo-Resin X004M. During the work with the semester project, it became clear that it’s combination of high tensile strength combined with relatively high elongation made it tough and hence better allowed micro-scale features to survive peel forces. A datasheet for the resin can be found in Appendix D.3.

3.3 Results and Discussion

3.3.1 Effect of PDC on $50\,\mu\mathrm{m}$ lattices

Figure 3.3, where cubic lattices of 12x12x3 mm with $50\,\mu\mathrm{m}$ members have been printed at various RGB values . In addition the supporting structure (large transparent beams and surrounding frame) have been printed at higher RGB than any of the $50\,\mu\mathrm{m}$ members. It appears to be a sweet spot around RGB 200 in combination with the default settings from Equation 3.1. (fourth sample from the left in Figure 3.3) for the $50\,\mu\mathrm{m}$ lattice where the voids remain open and the overall structure remains solid. A comparison between RGB 200 and 255 can found in Appendix B.2

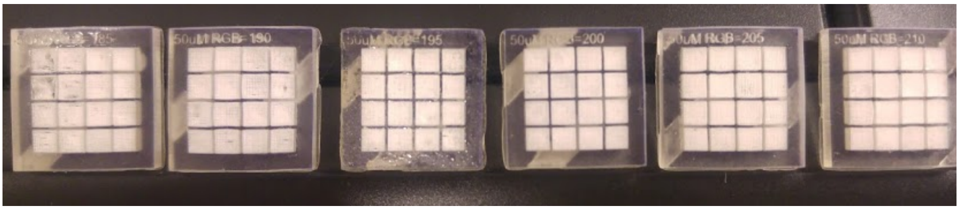


Figure 3.3: $50\,\mu\mathrm{m}$ lattices with varying RGB and supporting structure

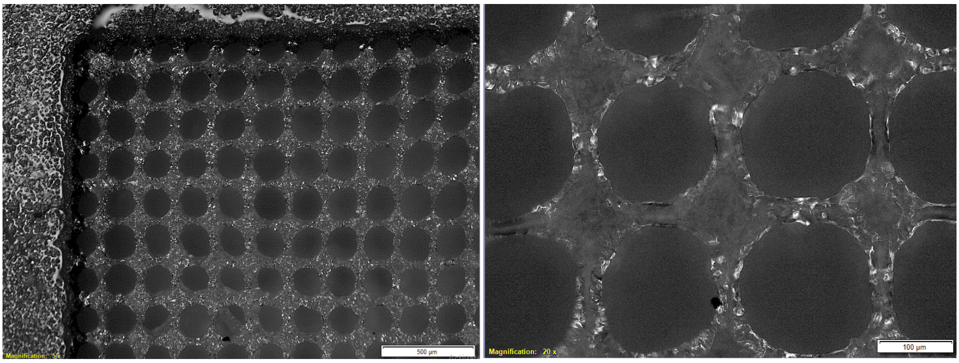


Figure 3.4: $50\,\mu\mathrm{m}$ cubic lattice 5x magnification left and 20x magnification right printed with RGB(200,200,200)

Figure 3.4 is the same $50\,\mu\mathrm{m}$ RGB 200 latice studied with a light microscope. It is clear that the main features are intact and about correct thickness, with the exception that the corners of each unit cell is rounded. It was hypothesized that this is due to the fact that the stray light from several pixels overlap in this area and hence reach a higher level than the middle of each beam.

To mitigate this, a filter function that reduced intensity near the corners of each unit cell was created in the Python script. In Figure 3.5 this filter is visualized as a heat map. The brightest pixels are in red colour and the dimmest pixels in blue colour. As we can see, the corners of the lattice unit cells are represented with low intensity to account for the stray light from the four neighboring pixels. The edges of the structure are the brightest as there are no fine details to represent and we instead print at maximum intensity to enhance the overall structural rigidity of the part.

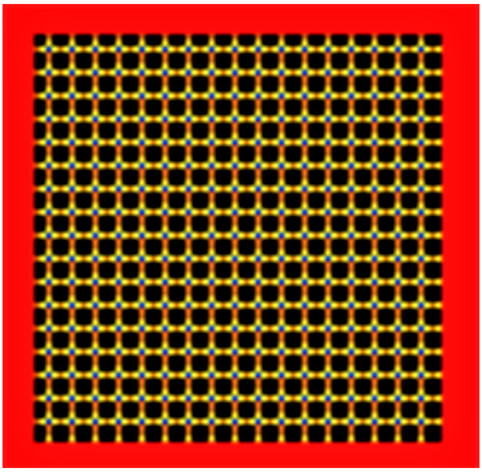


Figure 3.5: Pixel based intensity control visualized as a heat map

In reality colours are not used as only the total intensity matters and all of the RGB components are adjusted by equal amounts, this is known as gray scaling. An illustration is shown in Figure 3.6.

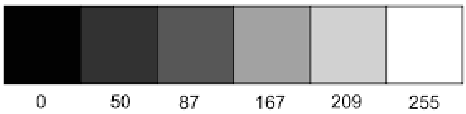


Figure 3.6: Gray-scale as a function of a selected RGB values

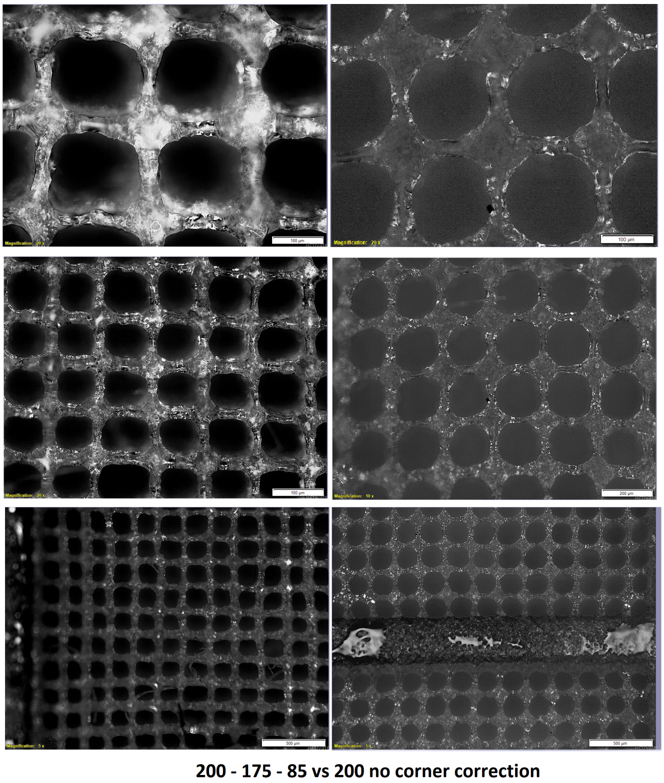


Figure 3.7: With a corner correction filter left vs uniform distribution right

The printed result from applying a gray scaling filter with the distribution from Figure 3.5 can be seen in Figure 3.7. On the left side is a lattice where the pixel in each corner of a unit cell has been printed with an RGB value of 85, neighboring pixels to the corner has the value 175 and the pixels in the middle of each beam has a RGB value of 200. To the right is the same structure as in Figure 3.4 at the same magnification for direct comparison. One can observe that the unit cells have become slightly less rounded and more rectangular and in this regard closer to the designed cubic lattice. Unfortunately, the variation in size and shape is larger than before and there is also some bulging in the overall structure. This indicates that the structure has become weaker and less stiff, resulting in more deformation and errors during printing and post-processing. While the sharpness could be further improved by reducing the intensity of each of the corner pixels, it soon came to a point where the lattices started collapsing or delaminating in certain areas. It appears to be a very delicate balance of parameters that will both yield the highest resolution and stable results.

3.3.2 Effect of RGB Intensity on Single Beams

To get a deeper understanding of the effect of pixel-based intensity control, a study on single beams was conducted. The test geometry consists of six vertical channels with seven horizontal beams spanning each channel. Each horizontal beam is one pixel wide and printed in two layers of 25 µm. All beams within a channel have the same RGB value. The cured linewidth was calculated as the was average of the seven beams found in one channel. The RGB value was reduced by a value of 35 for each instance of a channel along the x-axis as shown in Figure 3.8

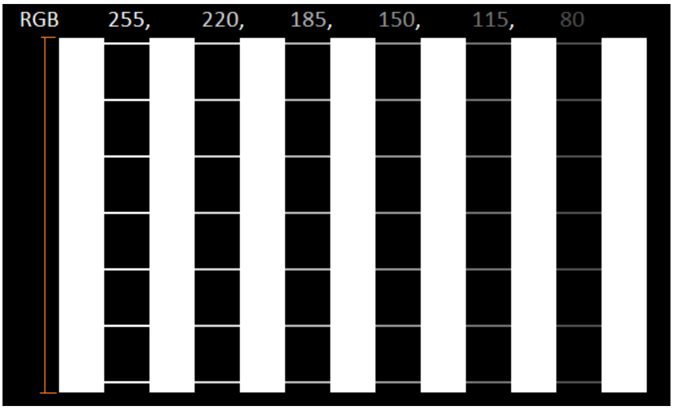


Figure 3.8: Test geometry for RGB vs Line Width testing. RGB ranging from 255-80 along x-axis. Line Width values averaged over each column along y-axis (orange)

Initially, variations of RGB values in the range of RGB (200-255) was tested as that range made a pronounced effect on the cubic lattices as discussed in the previous section. Surprisingly the test revealed no significant difference in beam width between RGB values in that range. The variation in thickness between beams of the same RGB value was larger than any observable difference between the different RGB values. The RGB range was therefore increased to 80-255, approximately a 300% increase from the lowest to the highest RGB value. The test was repeated using different laser power and exposure times, as presented in Table 3.1. Each input setting represents a multiplication of the default setting for a 25 µm layer in terms of Ed. The results from the default setting is not in the graph as single beams with less Ed than the default generally did not survive the printing process.

Input matrix of RGB vs single beam Line Width						
3x laser power	RGB=255	RGB=220	RGB=185	RGB=150	RGB=115	RGB=80
3x exposure time	RGB=255	RGB=220	RGB=185	RGB=150	RGB=115	RGB=80
2x laser power	RGB=255	RGB=220	RGB=185	RGB=150	RGB=115	RGB=80
2x exposure time	RGB=255	RGB=220	RGB=185	RGB=150	RGB=115	RGB=80

Table 3.1: Input values used in single Beam RGB test

The results presented in Figure 3.9 show that the cured line width follows a near logarith-

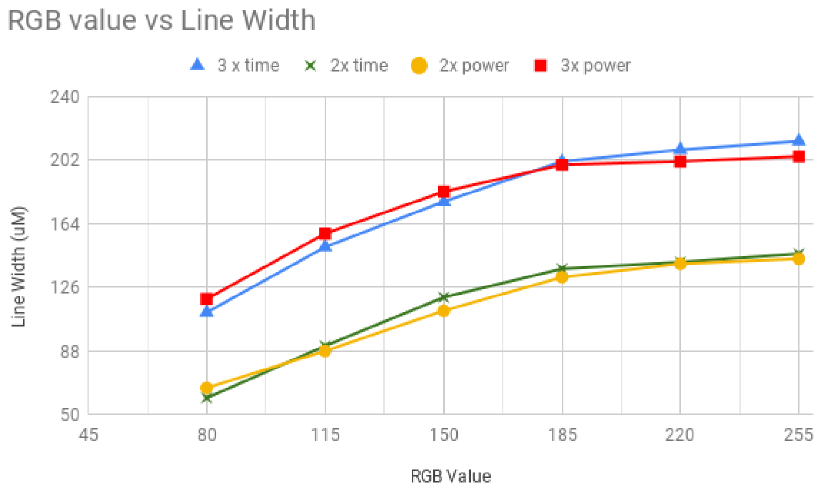


Figure 3.9: RGB value vs Line Width and four different energy settings

mic relationship to the RGB input. Also, we can see that increasing the laser power or the exposure time by an equal factor has the same net result. What is more surprising is that the line width increases slightly faster than the cure depth when increasing energy over the tested range as shown 3.10. This contradicts the Jacobs equation which states that line width should only grow with the square root of the cure depth.

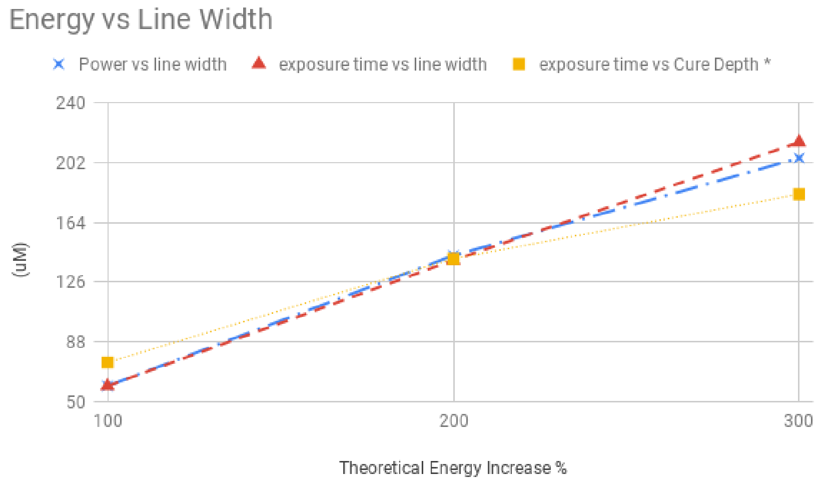


Figure 3.10: Change in Line Width vs Change in Energy

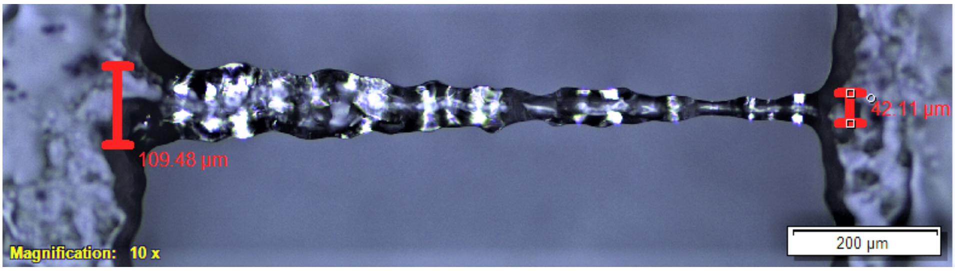


Figure 3.11: A 1 mm long single pixel beam printed with RGB values falling from 255 on the left to 50 on the right. Printed in a single layer with 1500ms exposure and 8mw laser power

3.3.3 Discussion on Pixel-based Intensity Control

The results presented in Figure 3.9 indicates that RGB does not have a linear relationship to the output energy. In the range of RGB 200-255, there is hardly any change to the width of the beams and those differences drown in the naturally occurring variations present at low dosage levels (see Appendix B.4). This explains why the initial testing showed no appreciable difference over the range of 200-255 at the default Ed setting. At higher RGB intervals the effect becomes very clear as seen in Figure 3.11 where the RGB value gradually changed from 255 to 50 over the length of the beam.

This means that the changes in the cubic lattices from small changes in RGB value can not be explained by the change in line width. Instead, my understanding is that through small adjustments of pixel intensity one can find a level that brings the unintended exposure (stray light) below Ec. This is illustrated in Figure 3.12 where three features are printed near to each other. When we reduce the intensity, the energy in between the features fall below Ec and hence the three features remain separate. This almost binary behavior explains why small adjustments can have such a profound effect on lattice structures and at the same time make little difference to a single feature or pixel. Of course, in the process of lowering the energy dosage one also reduces the Cd, which reduces the degree of cross linking between layers and the margin one has to overcome variations in the light distribution caused by imperfections in the FEP-film or the printer itself. This explains why the risk of delamination and printing errors becomes higher when power is reduced.

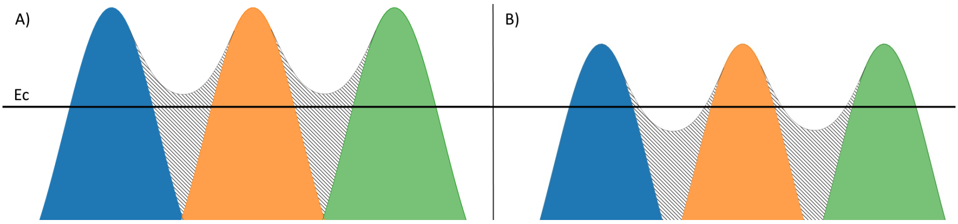


Figure 3.12: The effect of high intensity (A) vs low intensity (B) on stray light (slashed) and curing depth

Further, it has been found that increasing the exposure time had the same effect as increasing the laser power by an equal amount has the same net effect over the tested range. This is in agreement with the theory presented by Jacobs. More surprisingly, the growth in linewidth is much faster than the Jacobs equation would suggest.1.17 While no conclusive evidence can be presented, an explanation could be that the beams in this experiment was printed as two layers, and when the first layer is in place, the second layer is constrained by the previous layer, and the voxel growth becomes constrained to the XY-plane. The Jacobs equation considers a single Gaussian beam distributed in a medium. It seems logical that when the light hits the solidified material from the previous layer its direction of growth will be diverted sideways.

With this knowledge, one can deduce that printing the vertical members found in the cubic lattice can and should be printed with much higher intensity than the rest of the lattice as they are isolated single pixel features, which means the change in RGB value must be much larger to have a significant effect on it's width. There is also no real risk of curing beyond the intended depth as the vertical beams are continuous along the Z-axis. Using a higher intensity here should reduce the chance of defects and increase the strength of the lattice.

With the promising results of using gray scaling to enhance the resolution of lattice structures, it was not surprising to discover examples of gray scaling manipulation all ready being used by SLA-DLP companies[47] in the form of anti-aliasing where gray pixels are put in place along edges and in between layers. This reduces jaggedness from the rectangular pixels approximating a continuous slope and hence reduce surface roughness in printed parts as seen in Fig3.13. Autodesk also has shown how it can be used to offset an edge with distances smaller than one pixel.[48] At the same time it can be said that this is not an identical technique as it is used for different purposes and implemented differently. One could not replace the other all tough both take advantage of gray scaling.

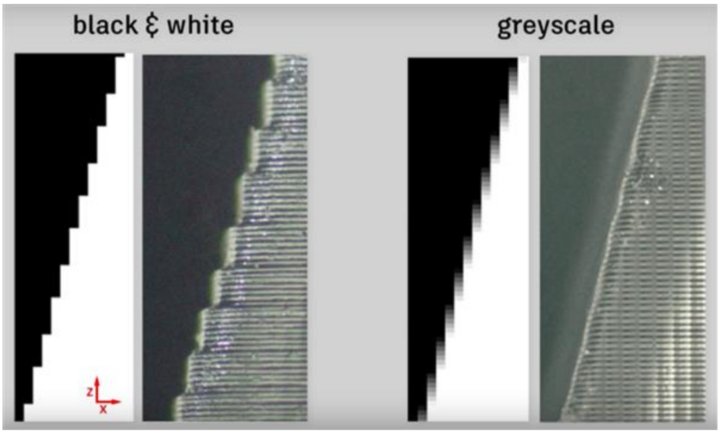


Figure 3.13: Effect on Anti-aliasing on printed part[48]

Future Potential of PDC

As the pixels of an existing PNG can be read and stored as a variable, one can manipulate it in retrospect. One could then design an algorithm that dims or brightens pixels, based on feature topology or the accumulated energy dosage over an area. This could make the technique applicable to all kinds of structures, regardless of the software used during design, and perhaps even run in real time during printing. This would likely require a more sophisticated model for predicting the correct dosage, than what the working curves based on the Jacobs equation provide.

Chapter 4

Unsupported 50uM Beam Length

4.1 Introduction

To understand more about the possible failure modes in regards to printing small features, a study of maximum unsupported beam length has been conducted. In the world of 3D printing, this is commonly referred to as bridging. It means the ability of the printer in combination with a given material to produce horizontal overhangs without support beneath. In engineering terms a this is a called a fixed beam and will have a moment distribution as shown in Figure 4.1 if loaded by a uniform force such as gravity.

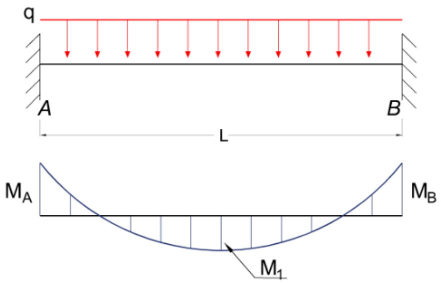


Figure 4.1: Fixed beam with evenly distributed force

On the face of it, this appears to be a straight forward task, but as it turns out, it is anything but straight forward. First of all, consider the beam itself; A 50 µm beam produced on a printer with a pixel size of 50 µm is printed with a single pixel, yet it can be printed in several ways. A 50 µm beam can be approximated using a single layer of 50 µm depth, but as discussed in section 1.4.2 the cured profile for a pixel is not a cubic voxel, but something closer to a parabola or a cone. By slicing the beam into several layers, the beam will take on a different shape than a beam printed in a single layer, but probably, it will never look like a 50x50 µm rectangle or a round rod.

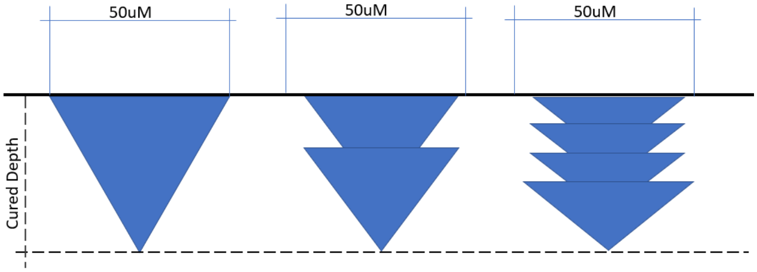


Figure 4.2: Illustration of variation in cured cross section of single pixel beam based on layer height

A beam printed as a stack of four 12 µm layers would perhaps most closely resemble a rectangular beam. On the other hand, the Ed would have to be extremely low, increasing the chance that a pixel falls below the critical limit. Secondly, it would result in an extremely thin structure after the first layer, which would not have enough strength and rigidity to survive the peeling process and remain sufficiently flat. A single layer is the safer and faster option as there is only one peel action, but results in triangular cross-section, which poorly represents the beams that has been utilized in the lattice structures so far. With this reasoning it was decided to continue to print 50 µm features using two layers of 25 µm using the default settings from Equation 3.1. From the RGB testing this method is known to produces beams that are 50-60 µm wide x 80-90 µm thick and span gaps of at least 1 mm with reasonable consistency. It is also known that the beam width and thickness at this setting show large variations which will affect the results as length increases (see Appendix B.4).

Further, the beams are not only subject to gravitational forces, but a peel force from separating the printed structures from the FEP sheet. This will wary depending on according to the variables shown in 1.15s, but also the placement on the build plate, the part orientation and neighbouring objects. These variables are clearly affecting the achieved lengths which will be shown and explained during the results and discussion section. To my knowledge there is no existing standard that will account for these effects and allow comparison across printers and materials. More important than the actual numbers are therefor the findings about the failure modes and relation to the mentioned variables.

4.2 Methods

The test geometry that was chosen after testing many different versions is a 3.5 mm thick rectangle with a V-shaped cut. Three different sizes was used, as shown in Figure 4.3 to cover a range from 0.5 mm to 12 mm spans. Beams are printed across this V-shaped cut using the default settings as described in Equation 3.1. The material is as before the BASF Photo-resin x004M. Data was recorded after a 10-15 second rinse in IPA without the use of ultra-sonic vibrations, to make the process as gentle as possible.

Sample size:

- 18 of the small V-specimens were printed in batches of six at a time. The six specimens were placed in Zone 1 illustrated in 4.4. The maximum span of this specimen is 2.2 mm. Each beam is spaced 1 mm apart.
- 36 of medium V-specimens were printed and placed across the left side of the build platform exactly as shown in Figure 4.4. These specimens was labeled to identify their locations. The maximum span of this specimen is 5 mm. Each beam is spaced 1mm apart.
- 20 of the large V-specimens were printed in batches of four at the time. They were placed in each corner of the build platform, as shown in Appendix C.2. The maximum span of this specimen is 12mm. Each beam is spaced 3 mm apart.

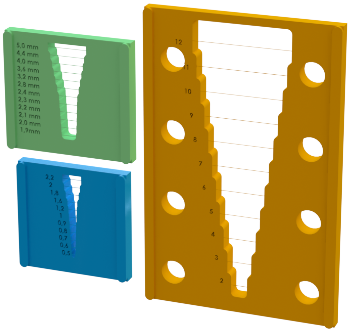


Figure 4.3: Unsupported beam length V-specimen in size small, medium and large versions

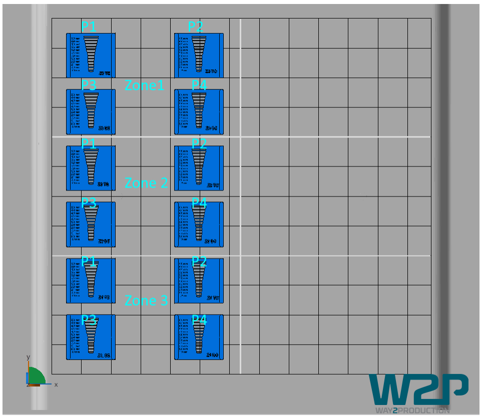


Figure 4.4: Placement on the build platform. Shown placement was used for the medium sized specimens. For the small, all was placed in Zone 1.

4.3 Results and Discussion

4.3.1 Small V-specimens

The recorded data from the small specimens are presented in Figure 4.5. The results are quite stable with of 88% of the specimens reaching 2 mm or more. With this result, it was assumed that longer spans are possible and larger V-specimen was tested.

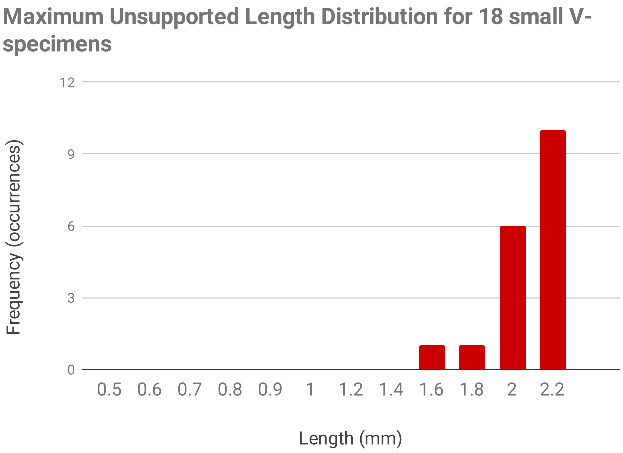


Figure 4.5: Distribution of obtained lengths after printing 18 small V-specimens)

4.3.2 Medium V-Specimens

The data from printing 36 medium sized V-specimens with placement as described in Figure 4.4 is presented in Figure 4.6. As one can see this yielded a totally different result than the small specimens. Here the majority of the specimens recorded a maximum length of 0, meaning that none of the beams were intact in those specimens.

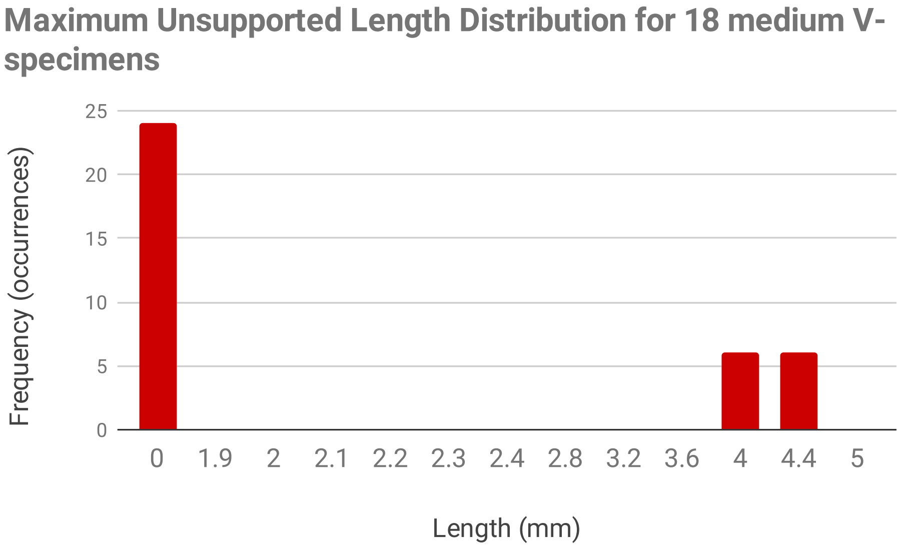


Figure 4.6: Distribution of obtained lengths after printing 36 medium V-specimens)

In table 4.1 the recorded placement and recorded length of each specimen in Figure 4.6 can be found. The results appears to have a pattern when we take into account the position of the specimens:

- None of the beams in any of the specimens in Zone 2 survives printing in any of the three iterations.
- The same goes for the specimens Z1P3 and Z1P4, and Z3P1 and Z3P2.
- This leaves Z1P1,Z1P2,Z3P3 and Z4P4 which consistently produces beams of 4-4.4 mm length. These are the specimens located along the top and bottom edge of the build platform in Figure 4.4.
- For the Z1P1,Z1P2,Z3P3 and Z4P4 specimens a repeating failure mode is appearing, where the longer beams are attached to each other, forming a cross as shown in 4.7. In a few cases, it was observed that the beams detach again during the drying phase after washing the samples in IPA while other times they are destroyed.

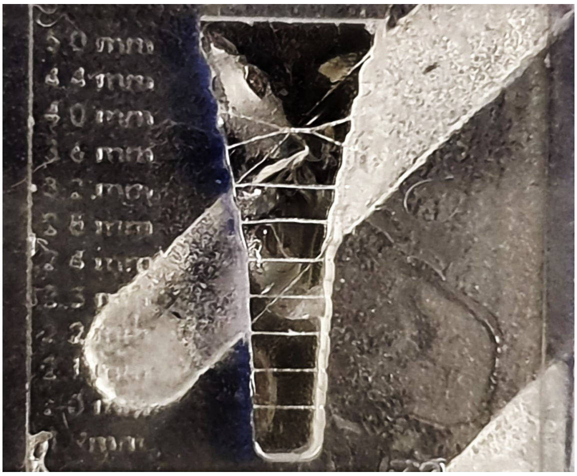


Figure 4.7: Repeatedly observed failure mode of beams approaching 4 mm in length

Unsupported beam length type A (1.9-5 mm version)			
Iteration Placement	1	2	3
Zone 1			
P1	4.4 mm	4 mm	4.4 mm
P2	4.4 mm	4.4 mm	4 mm
P3	0 mm	0 mm	0 mm
P4	0 mm	0 mm	0 mm
Zone 2			
P1	0 mm*	0 mm	0 mm*
P2	0 mm*	0 mm	0 mm
P3	0 mm	0 mm	0 mm
P3	0 mm	0 mm	0 mm
Zone 3			
P1	0 mm*	0 mm	0 mm*
P2	0 mm*	0 mm	0 mm
P3	4.4mm mm	4.4 mm	4.4 mm
P4	4.4 mm	4 mm	4 mm

Table 4.1: Recorded maximum length from 36 type A specimens printed over three iterations

4.3.3 Capillary Failure Modes

The failure mode in the medium specimens has been observed throughout the project also sometimes in the case of the cubic lattices from chapter 3, mostly upon washing and drying. While several washing regimes have been tried, the reason was not clear to the author. Upon further research a paper by Wu et.Al describes similar behaviour.[49] The authors explain that the problem is typically referred to as ”stiction”, and that the reason for this type of failure mode are capillary forces mainly occurring during the evaporating drying which bends and often collapses the beams. This is not a problem when printing macro-scale features as this is a weak and short-range type of force, but when thin beams are placed in near vicinity, the forces become large enough to bend the beams. The Wu et al. paper shows how this stiction effect is a weak function of surface energy of the solids and liquids, and more strongly by the thickness and spacing of the beams. This explains why this effect appears only as we approach the micro-scale with SLA.

A test was performed to validate that capillary force was the reason for the stiction in the V-Specimens. 1 mm long and 50 µm thick beams with a separation distance ranging from 450 µm down to 200 µm was printed. The resulting specimens are presented in Figure 4.10. This experiment show that the amount of beams that becomes attached grows quickly as the separation distance is reduced. At 450 µm none of the beams are subject to permanent stiction (they pop back during drying), while at 200 µm almost all the beams are attached to each other due to the capillary forces overcoming the elastic energy of the beams. Microscope photos of the effect is found in Appendix B.5

4.3.4 Large V-specimens Results

A larger version of the V-specimen where the distance beams were increased from 1 mm to 3 mm. Also, the maximum span was increased to 12 mm. Five iterations with four specimens each were printed (20 in total). The specimens in each iteration were placed in each corner of the print platform as the previous result indicated this was the optimal placement. The specimens were printed on 5 mm tall supports to avoid stiction to the build platform. The observed distribution of maximum length pr specimen is presented in Figure 4.8. The unsupported beam length increased to approximately 7 mm on average. At distances above 6 mm the stiction problem reoccurred in some of the specimens, and it was clear that post-processing was a big factor as several specimens reached lengths up to 11 mm before collapsing during removal, washing or drying. See Appendix C.3 for an example.

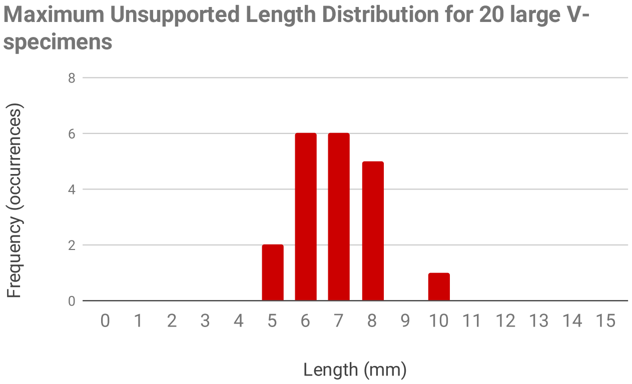


Figure 4.8: Distribution of obtained beam length of 20 specimens of enlarger type A specimen)

4.3.5 Discussion

Through the experiments in this section, it has been substantiated that capillary induced buckling and variations in peel forces with placement are dominating factors for the achievable maximum unsupported length. Hence, quoting an exact length seem of little value, as the results will vary greatly depending on the parts placement, geometry and type of post-processing. Other geometries, such as a traditional bridge structure shown in Figure 4.9 has also been tried, but such free-standing structures are found to be even more inconsistent and generally not being able to survive the print process at all.

A solution to capillary induced buckling and stiction is to change the washing technique to a process that does not involve the liquid phase, such as sublimation drying or supercritical drying. This is well described in the paper by Wu et al.[49] Attempting to solve the problem by lowering the surface tension in the utilized resin and washing agent seems not like a feasible strategy. This is because stiction is more strongly related to the scale of the structures, so any incremental reduction in surface tension will be insignificant when the scale approaches that of a GDL. Secondly, it can be mentioned that IPA, which was used in this work, already is on the lower end of the spectrum for applicable washing agents.[50] Peel forces can be avoided by using a top-down type SLA-printer, which should also would reduce the effect of imperfections in the FEP film as a result of wear and fogging occurring after usage. It seems likely that this can significantly affect single pixel features printed with low energy dosages. This knowledge seems more valuable for anyone attempting to print micro-scale features with SLA, than the achieved obtained unsupported lengths.

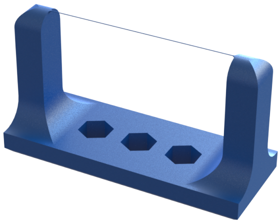


Figure 4.9: Different type of unsupported beam yields different results

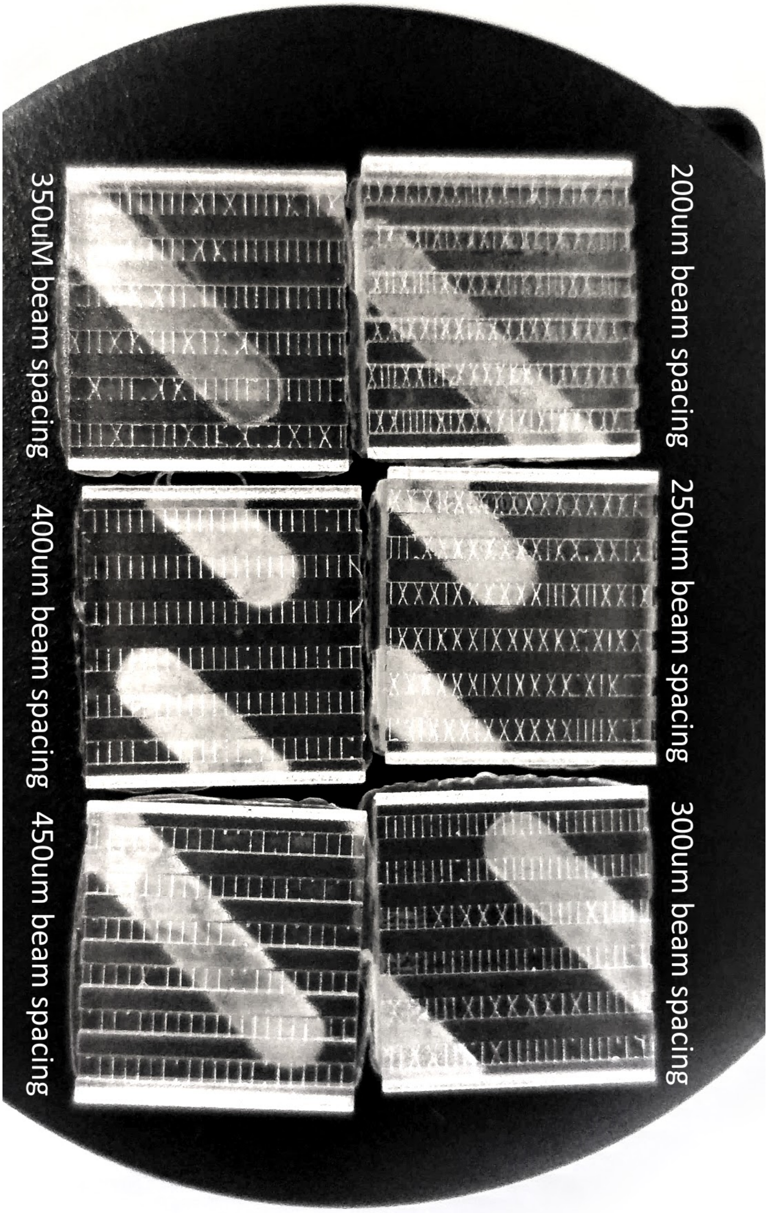


Figure 4.10: 50 µm beams spanning a 1 mm wide channel. Note how the number of beams collapsing due to capillary forces increase as the spacing between the beams decrease

Chapter 5

3D printing and Carbonization of Polyimide

This chapter will evaluate polyimide (PI) as a candidate material for a 3D printed GDL.

5.1 Introduction

PIs are incredibly strong synthetic polymers that are also very heat and chemically resistant. As the name suggests polyimide consists of chains of imide monomers. An imide group is a structure that consists of nitrogen, which is bound to two carbon atoms. Each carbon atom is bound to one oxygen atom.[51]

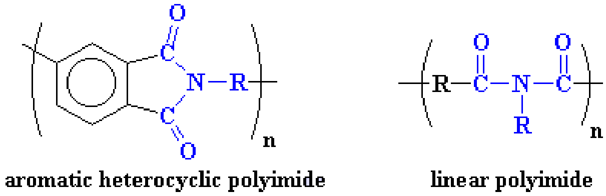


Figure 5.1: Aromatic and linear repeating imide groups

Polyimides can take on two different forms, a circular (aromatic) configuration, or a linear (aliphatic) configuration (Figure 5.1). Aromatic polyimdes are typical in most commercial polyimides such as Ultem from Sabic and Kapton from Dupont. These polymers have remarkable mechanical and thermal properties as a result of strong inter molecular forces between the polymer chains.[52] The first commercial polyimide was Kapton made by DuPont in 1955. Kapton's chemical structure is called pyromellitic dianhydride oxydianaline, or PMDA-ODA for short. PMDA-ODA does not burn or melt, remains stable from -269°C up to 400°C and exhibits little to no creep under continuous loading.[53]. Also at ambient temperatures, the tensile strength and compressive strength of polyimide is higher than other engineering-grade polymers, such as Polyether ether ketone (PEEK) and Polyamide-imide (PAI).[54]

Because PMDA-ODA can not be melt processed, and is resistant to all organic solvents, which makes it very difficult to process. There exists a polyamic acid (PAA) that has a very similar chain architecture to PMDA-ODA, but instead of the refractory imide group one has an acid moiety (see Figure 5.2) which gives the chain much more flexibility, and also makes it much more soluble in relatively benign solvents, such as N-methy-pyrrolidone (NMP).[55] Thus, one first casts a film of the PAA solved in NMP, and then after drying, the PAA can be converted to the polyimide. This is done by baking the resin at temperatures in the range of 300-400°C. This releases the solvent along with water through condensation, and forms the ring shaped imid-group that turns it into an aromatic-poly*imide*. This process is known as imidization.[56]

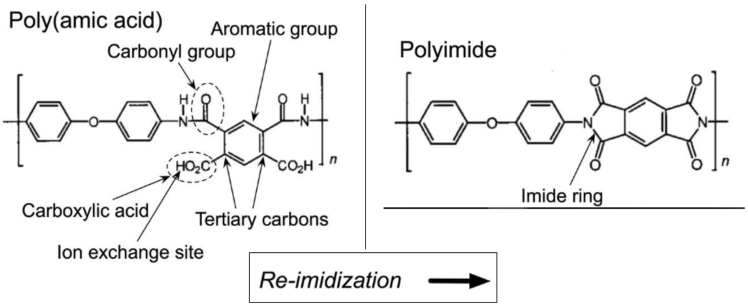


Figure 5.2: PAA as a precursor to PMDA-ODA (kapton)

5.1.1 3D printed Polyimide

3D printed polyimide can open up new applications for AM and also make polyimide more attractive for applications where it was previously disregarded. Aromatic polyimides are not well suited for injection molding, which limits geometrical complexity. It is costly in combination with subtractive manufacturing, due to high material waste, in combination with the high material cost. 3D printing would allow both complex gemoetries and reduce material waste.

3D printing of aromatic-polyimides is a very recent advancement, and the first paper on this was published in 2017 by a team of researchers from the "DREAMS" Laboratory of Virginia Tech (VT) USA.[57] In April 2018, the VT group published a second paper which describes a way of creating UV reactive PAA[58] in a relatively simple manner from commercially available ingredients. The way the team at VT accomplished this was by adding acrylate groups in the form of (2-(dimethylamino)ethyl methacrylate (DMAEMA)), to the

PAA which is solvable in NMP. In this procedure, the DMAEMA reacted with the carboxyl groups of the dissolved PAA and provided the UV-sensitive groups which are necessary for SLA.

The authors tested for various molar ratios of DMAEMA to PAA, which turns out to be critically important for fast photo-cross-linking and to obtain sufficient mechanical integrity. At the same time, the introduced DMAEMA groups only act as sacrificial cross-links to enable selective photo-curing which must be released during conversion from PAA to PMDA-ODA (Kapton). Consequently, they recommend keeping the number of transient cross-links at a minimum to facilitate their release upon imidization. The addition of DMAEMA raised the solution viscosity from 6 Pa·s to 12-18 Pa·s, which is very viscous for a resin meant to be 3D-printed. For comparison, the resin used in chapter 3 has viscosity of 0.2 Pa·s, and is described as a medium viscosity resin. The authors solved this challenge by using a top-down SLA machine (see Figure 1.11) with a re-coating blade which helps push the resin back between each layer. This also eliminate the increase in peel forces due to using high viscosity resins with a bottom-up machine.

After successful 3D printing, thermal imidization was performed which lead to an isotropic shrinkage of about 48%. For many applications this will be considered negative as the maximum build volume is almost halved. But from the perspective of a 3D printed GDL, a halving of feature size could be a step in the right direction in obtaining the small features required. Please see VT production instructions[59] for further details on the processing of this material.

5.1.2 Carbonization of Polyimide

In its normal state, PI would not be usable as a GDL material due to its low conductivity. In fact, it is one of the best dielectrics there is with a resistivity of $1.45 - 2 \cdot 10^{17} \Omega \cdot cm.$[60] But PIs in general and especially PMDA-ODA, are known for their ability to undergo carbonization. After the carbonization process it becomes a very good electric conductor. Carbonization is a process where solid residues with increasing carbon content is formed from organic polymers. At temperatures above 500°C, PI looses much of its non-carbon content as gasses and change to forms of carbon. The amount of heat applied, controls the degree of carbonization and the residual content of foreign elements. Complete carbonization is not achieved before temperatures exceed 1000°C. Below 1000°C, the material will have intermediate properties between polymers and carbon.[61]

The changes in electrical properties was documented by Brom et al., by heating samples of Kapton film under a nitrogen atmosphere at temperatures ranging from 500-800°C. They found that the resistivity dropped 20 orders of magnitude, down to about 0,01$\Omega \cdot cm$ for the samples heated to to more than 700°C.[62] The reason polyimide is a good carbon precursor is that it is extremely stable and decomposes before it reaches any type of melting point. This means it can retain its physical shape when subjected to extreme temperatures.

Burger et.al published one of the first studies on carbonization of polyimide.[63] They conducted thermograviemtric analysis (TGA) which confirmed the thermal stability of Kapton up to 400c, followed by a rapid weight loss and a corresponding shrinkage around 550°C.

At 1000°C, a carbon yield of 60%, and shrinkage of 39-45% was found for four different versions of polyimide. Carbonization of PI in both foil type (Kapton) and thicker sheets (Vespel) were performed. They found that Vespel sheets of up to 0.8 mm could be carbonized without formations of cracks, and even up to 2 mm in the case of Vespel containing 30% glass filler.[63] Later research by Ingaki et.al[64] confirms the results from the Burger et.al study. Ingaki et.al worked with thin films (Kapton) and studied heating rates ranging from 60 to 400c/hr, and also the effect of holding stable temperature in the region of rapid weight loss, but found very little difference between them. However, they found that applying light pressure to the samples resulted in remarkably less deformation. Sufficient flushing with nitrogen, before heating was also necessary to avoid cracks.

Researchers have also documented the possibility to graphitize Kapton.[64] Graphitization is performed by increasing the temperature beyond 2500°C and transforms the disordered carbon structure into three-dimensional crystals of graphite. The possibility to graphitize Kapton was originally a surprising result, since thermosetting resins are carbonized in a solid (no melt phase) state with the polymers oriented in a isotropic manner, resulting in carbon structures that usually cannot be graphitized. The explanation for the ability to graphitize Kapton is that thin films achieve a ordering of the polymer chains along the film surface and that the starting molecule is relatively flat.[65] For this reason, polyimides with greater thickness to a lesser degree will be able to undergo graphitization. Pure graphite is an even better electrical conductor than the carbon obtained after carbonization, but it also involves a highly anisotropic change in dimensions as the carbon atoms orientate themselves in the layer like structure. For the same reason, also the electrical conductivity is highly anisotropic with much poorer performance perpendicular to the layers.[66] If 3D printed Kapton achieve the same type of ordering that allow graphitization is not known and could be a topic for future research, but is not in the scope of this thesis.

5.2 Methods

The purpose of this research is to assess if existing 3D printable PIs can be carbonized. Two PI based materials have been tested: Cubicure Thermoblast[67] and our self produced PAA-resin. This means we are interested in their mass loss over a temperature range up to 1000°C.

Mass loss during carbonization of the materials will be measured using thermogravimetric analysis (TGA). The resulting mass loss as a function of temperature will be compared to results from existing literature on PMDA-ODA, which suggests stable behaviour to about 500°C followed by a rapid mas loss up to about 600°C, and a final carbon yield of about 60%.

For the PAA-resin, imidization treatment must be performed before the carbonization, to obtain PMDA-ODA (see 5.2). The imidization process has been performed on seven specimens in total, in two separate runs. In the first round three rectangular specimens approximately 2 mm thick was used. The focus was on documenting the weight loss, and obtaining usable samples for the following carbonization experiment. The specimens starting weight was 0.14g, 1g, 2g. The second round consisted of larger specimens with thickness ranging from 3-11 mm with a mass of 0.3g, 0.5g, 11g and 15.1g. In the second round, dimensions were measured before and after to investigate the shrinkage behaviour. Specimens were suspended on a stainless steel mesh during the imidization process.

5.2.1 Materials

Thermoblast was kindly supplied by Cubicure in the form of printed parts in various sizes and geometries. A picture of the parts can be found in Appendix E.8. A small piece was cut from one of the parts due to the size limitations of the TGA. The formula and production process is proprietary, but is assumed to be based on a type of polyimide.

PAA-resin was synthesized at NTNU's chemical department, according to the production instructions made by VT.[59] The resin is made from of PAA (Appendix E.1), metha-acrylate (Appendix E.2) and photo-initator (Appendix E.3) which was bought from Sigma Aldrich. The PAA contains 15-16% of pyromelltici dianhydride-co-4,4', which is the precursor molecule to the PMDA-ODA while the rest is NMP (solvent). The resulting resin (Appendix E.5) had much slower reaction rates then reported by Virginia Tech, but could be cured by using very long exposure time.[1]

5.2.2 Equipment

TGA was performed on a Mettler Toledo AG - TGA/SDTA851e. A nitrogen flow of 100ml/min was utilized to provide an inert atmosphere.

Imidization was performed in a Nabertherm LV 15/11 lab furnace that features a 18L chamber and a maximum temperature of 1100C. The oven was modified with a N_2 inlet at the bottom and the air inlets was blocked to provide an inert atmosphere. Gas flow was controlled by a Alicat MC-50SLPM gas flow controller, which has a maxi-mum volumetric flow of 50 l/min which means that the furnace could be efficiently flushed with nitrogen at start up. A picture of the setup can be found in Appendix E.9. The weight measurements before and after imidization was made with a normal laboratory grade scale with 0.1mg resolution. External dimensions was measured using a hand-held caliper.

The experiments were conducted at the Department of Energy and Process Engineering.

5.2.3 Heat rates

For imidization of the PAA resin, a drying stage was performed by holding at 60°C for 24 hours. This was followed by heating to 100°C and holding for 30 min, heating to 150°C and holding for 30min, heating to 200°C and holding for 30min and finally under N_2 flow heating up to 400°C and holding for 30 mins which is illustrated in Figure 5.3.

Carbonization was done by heating to 100C°at a rate of 1°C/min and holding for 1hour, then heating to 1000°C at a rate of 1°C/min.

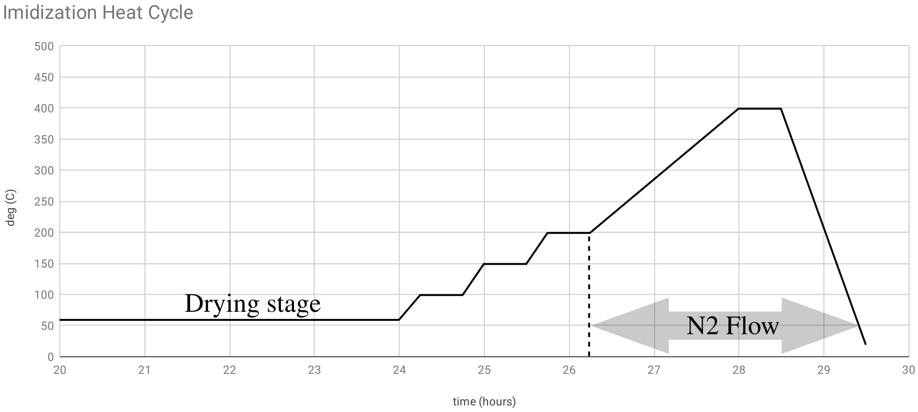


Figure 5.3: Illustration of heat cycle used for imidization of PAA-resin

5.3 Results and Discussion

5.3.1 Imidization Round one

Presented in Figure 5.4 is the weight loss during the Imidization process in round one. The samples were taken out and weighed after each holding stage in the heat cycle. It can be seen that the final weight at 400°C is 15-16% of the starting weight. This equals the amount of PMDA-ODA precursor in the PAA-resin and the results obtained by the researchers at Virginia Tech. Simultaneously, it is evident that the weight loss in our specimens are much larger in the region below 60°C than what was reported by the researchers from Virginia Tech. The two largest samples see a slightly larger weight loss than the smallest sample.

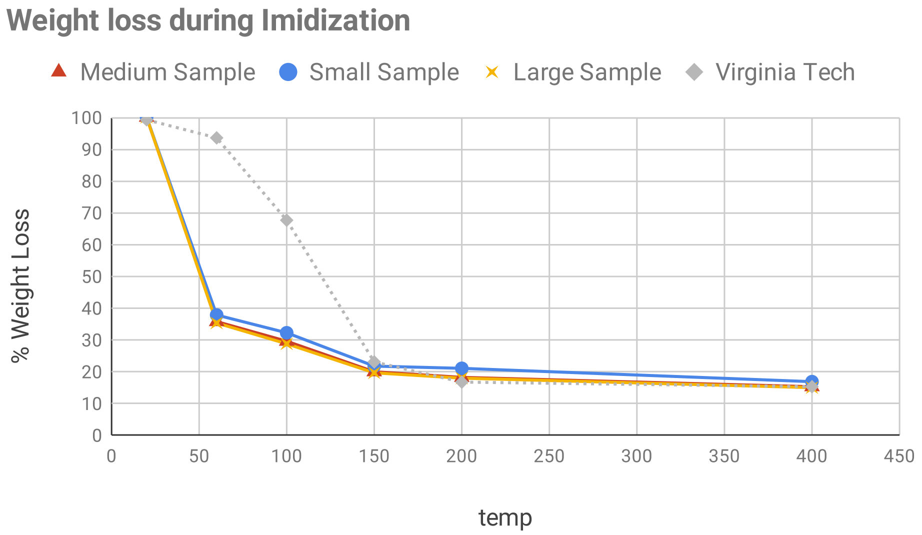


Figure 5.4: Mass loss during imidization for the three samples (red,blue,yellow) in round one compared to Virginia Techs published results (stapled gray)

In Figure 5.5 a picture of the specimens in round one before the drying stage is shown. By comparing this to Figure 5.6 which is a picture of the samples after each stage in the heat cycle one can see that there is a large and anisotropic change in shape during the drying stage. This change is much more severe in case of the two larger samples which appears to have floated outwards under its own weight. This behaviour was not reported by the researchers at Virginia Tech and is problematic, as maintaining dimensional accuracy is crucial to obtain usable parts. The smaller samples to a larger degree retains it original geometry with nice sharp corners.

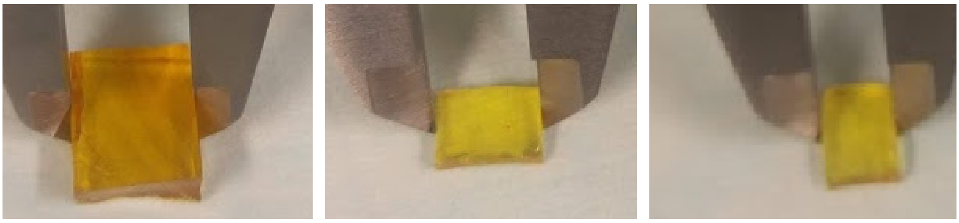


Figure 5.5: From left: Large, Medium and Small sample before drying stage. The pictures have been scaled for visibility so size is not directly comparable

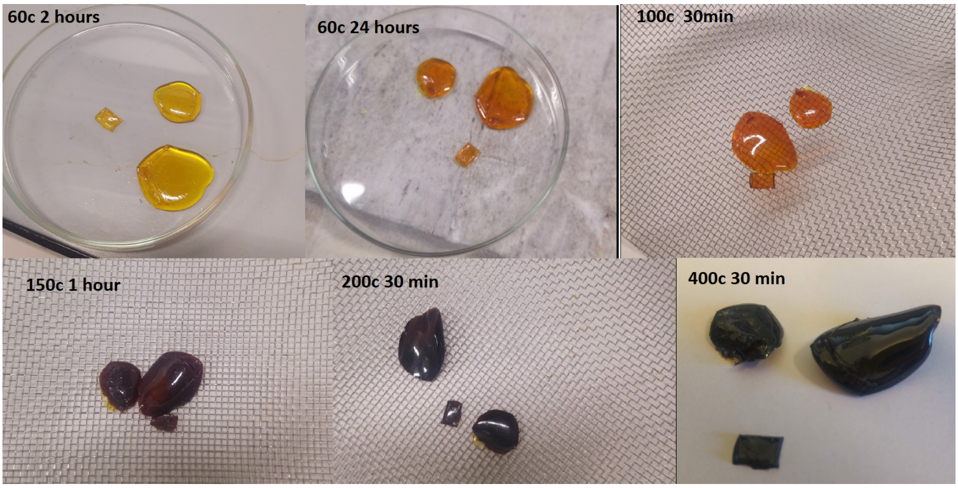


Figure 5.6: Colour and shape during the Imidization process

5.3.2 Imidization Round Two

In the second round the heat cycle from Figure 5.3 was kept, but the four specimens were not taken out until the entire process was completed to eliminate any effects this could have on the process. This results in only two data points pr type of measurement which are the before and after measurements. Therefore the result are this time presented as a column chart in Figure 5.7. One column represent one type of measurement and the values are quoted as $\frac{finalvalue}{initialvalue} * 100\%$ pr specimen. What one would hope to see is that all the specimens show the same remaining weight % (blue column) and equal isotropic shrinkage along the X,Y and Z direction (red, yellow, green).

As one can see in Figure5.7, this is not the case. Specimen three and four which are the largest, only retained 1.65% of their original mass. In addition, the shrinkage along the Z-direction is much larger than XY-direction for all the specimens, but especially for specimen three which was the largest and thickest of the specimens. Observing the change visually (Figure 5.8), show that the specimen three has totally collapsed. The photo of specimen four before imidization was unfortunately not taken.

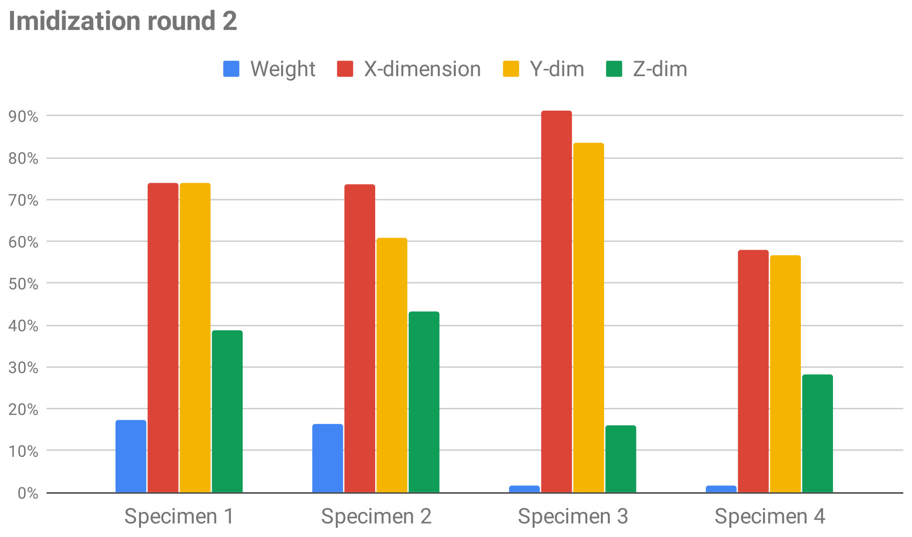


Figure 5.7: Post-Imidization weights and dimension as percent of initial values.

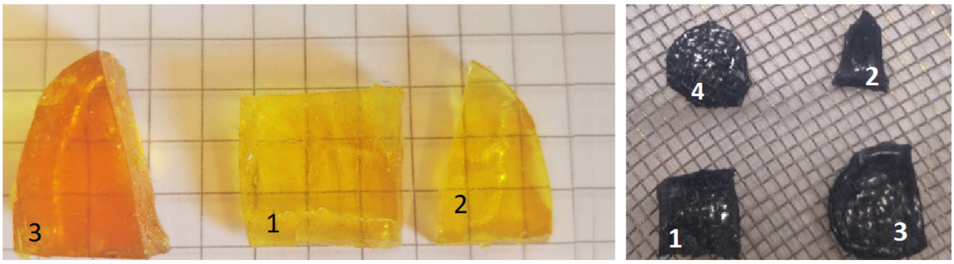


Figure 5.8: Round two: pre imidization (left) and post imidization (right).

5.3.3 Carbonization Results

As discussed in the introduction to this chapter, we expect PMDA-ODA to remain stable up to about 500°C followed by a rapid weight loss around 550°C and a have a yield of approximately 60% (40% mass loss) at 1000°C.

Cubicure Thermoblast TGA Results

The TGA plot of the CubiCure Thermoblast in Figure 5.9 does not show this behavior. At 300°C the mass drops quickly and a spike appears around 600°C. Upon post-inspection, the sample appears to have melted which correlates with the TGA behaviour.

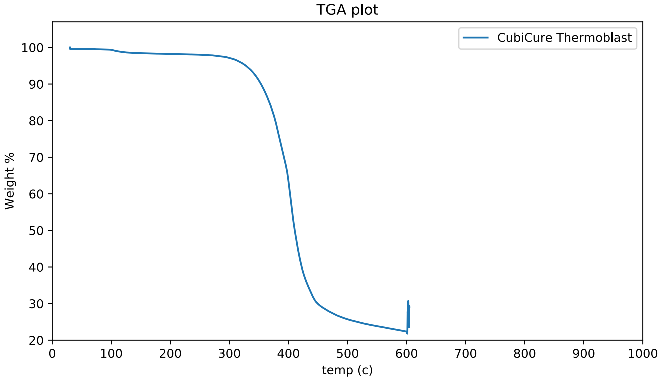


Figure 5.9: TGA of the Cubicure specimen

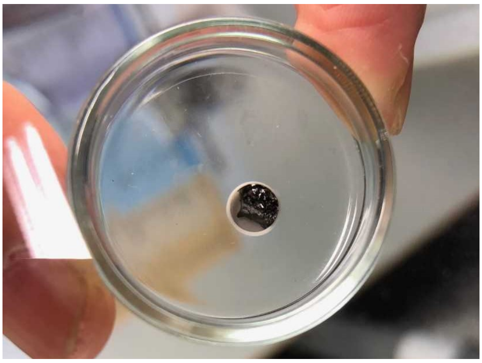


Figure 5.10: Cubicure Thermoblast after attempted carbonization

Imidized PAA TGA Results

The TGA results from the Imidized PAA are presented in Figure 5.11.

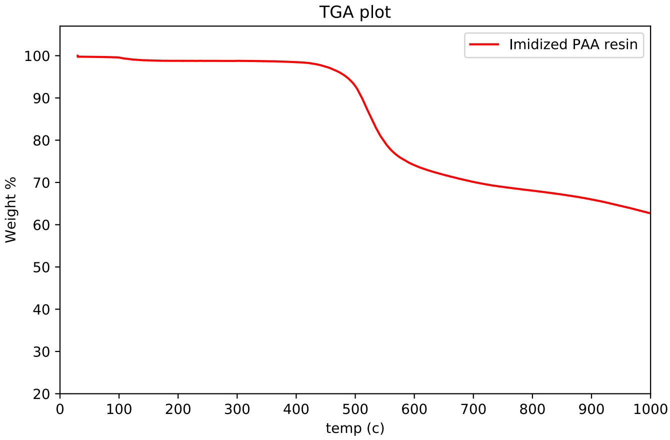


Figure 5.11: TGA curve imidized PAA resin

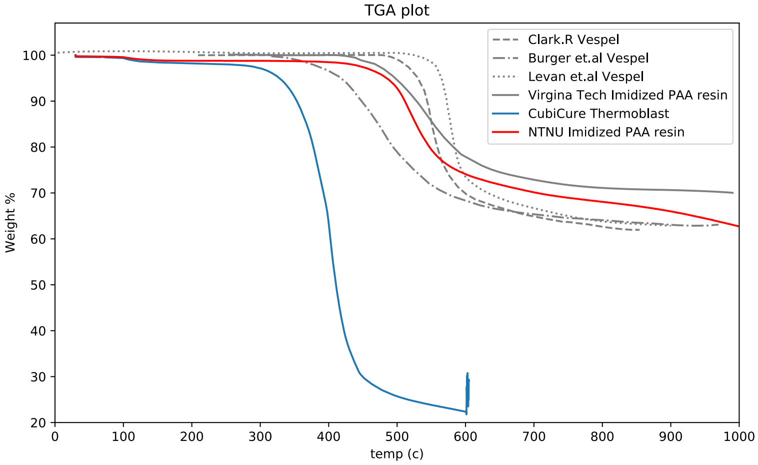


Figure 5.12: Our result compared to results from published literature [68] [69] [63] [59]

5.3.4 Discussion

The temperature stability up to 500°C indicates that aromatic PMDA-ODA was achieved from the imidization of the PAA resin. Otherwise the part would have decomposed at lower temperatures, as was seen in the Cubicure specimen. Also in respect to the total weight loss at 1000°C, the results are comparable to what is found in published literature. This comparison is shown in Figure 5.12

There is a deviance at higher temperature, where the mass loss continues all the way to 1000°C in the specimen. The reason for this behaviour is not clear at this time. It could could be a result of a undetected difference in synthesis procedure or imidization process. Both the Cubicure and the PAA resin show some small weight loss at low temperature which should not occur. This means there might be a calibration issue with the TGA. Our TGA was calibrated using an empty crucible, while at least one of the cited studies[69] state that they calibrated by using an already carbonized specimen as a dummy.

The Virginia Tech results are slightly different from commercial variants of PMDA-ODA by showing significantly less weight loss (70% yield). Since it has been noted a significant differences, both at the Imidization stage (Figure 5.4) and at the UV-curing stage[1], between their results and ours, it is a possible there is a consequential error following from the synthesis procedure or even the materials used to produce our PAA resin. A different brand of polyamic acid was use due to the one used by Virginia Tech was no longer available . This would contradict the opinion of two individually consulted chemists that confirmed that the molecular structure in the datasheet was identical same. Also the solvent, amount of solvent, and viscosity was identical for the two products according to the datasheet (E.1) and what was documented in the Virginia Tech supporting information.[58]

In Figure 5.13 the specimen used for the TGA is shown. Cracks in the specimen have developed, which also was reported at Burger.et al[63] for bulk (non film) specimens. This could be a result of the heat rate being too fast for the thickness of the specimen.

Finally, the bonus goal for this thesis was to measure the electrical conductivity of the resulting specimen. This was unfortunately not achieved within the thesis deadline. The specimen from the conducted TGA run was too small to allow proper measurement, and there was not enough time left to run the experiment again with a larger specimen and perform the measurements. This should be done in future work.

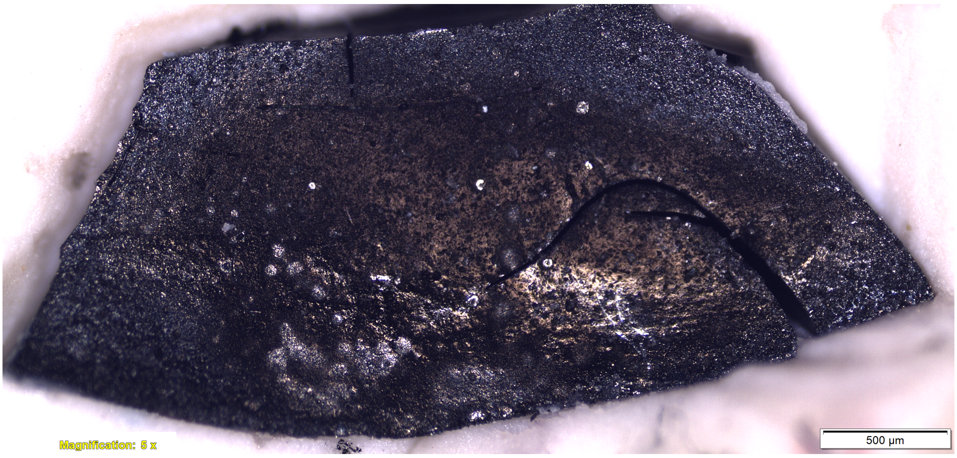


Figure 5.13: Imidized PAA resin after heating to 1000°

Chapter 6

Conclusion and Future work

In this thesis, the feasibility and challenges of creating 3D printed gas diffusion layers. The conclusion will follow the chapters in chronological fashion:

6.1 Conclusion

- The studied literature indicates that the feature size required for a GDL is obtainable through the use of SLA.
- Efficient generation of very feature dense lattices can be obtained by bypassing the 3D step, and directly produce 2D cross sections readable by a 3D printer.
- The Python script should be considered a proof of concept, not an end use software.
- Bitmap-based formats such as PNG enable the possibility of detailed Energy Dosage control in SLA printers by manipulating the RGB value of individual pixels.
- Pixel-based Intensity Control can produce profound effects on dense lattice structures both in resolution and mechanical properties.
- Pixel-based Intensity Control is useful for fine tuning and extracting the last bit of detail from a DLP-SLA printer, but in general, the minimum feature size is determined by the pixel size of the printer.
- Pixel-based Intensity Control can not control the resolution in the XY and Z plane independently. Reducing the intensity to enhance the in-plane resolution will weaken the structure due to lack of cross-linking. Increasing the intensity to cure either thicker or wider lines will reduce resolution.
- Adjusting RGB values does not provide a linear change in cured width and depth. Small adjustments can have big effects on lattice structures, while single beams need

larger adjustments in RGB, to significantly alter the dimensions.

- Over the energy range and type of geometry tested, the line width apperars to grown faster that suggested by the Jacobs equation.
- The maximum unsupported beam length of a single pixel (50 µm) beam using BASF x004m appears to be limited by capillary forces or peel forces.
- Near 50 µm features can be produced in certain situations, but can not be said to be robust or very repeatable. Especially in the case of single beams.
- Capillary induced stiction and collapse has been observed repeatedly and must be solved to make use of the resolution capabilities of SLA. This can be solved either by sublimation drying or super critical drying.[49]
- Peel forces also have a big affect on parts with delicate features, as changes in placement and orientation has a large influence on the single beam results. The top-down projection layout would remove many of the variables involved in peeling and increase throughput. Because of this it seems better suited for this application.
- It has been shown that UV-curable polyimide has similar temperature stability as commercial versions up to 500 °C, and similar mas loss up to 1000°C. It is therefore likely that it can be used as a carbon precursor.
- The promising results from Virginia Tech[58] in terms of 3D printing and maintaining dimensional accuracy, has not been reproduced in this project. Unreasonably long curing times and low modulus was experienced during the specialization project.[1] In this thesis, a highly anisotropic shrinkage during imidization was observed. This is problematic, and must be solved for this material to become useful. The high viscosity of this type of resin is also problematic.

6.2 Future Work

For the continuation of this project the author would like to purpose the following recommendations:

Separating the scope: The results from the studied literature indicate that SLA should be capable of high enough resolution for a GDL. The main challenge of SLA has been, and still is, the material catalog. Therefore it is the authors' recommendation to focus on the materials. Successful 3D printing of aromatic polyimide would be very useful, not only in the case of fuel cells.

With the results from the TGA, it seems very likely that the specimen has been successfully carbonized, and therefore also has similar electrical properties as that obtained by other researchers.[62] If future research finds this to be true, a study of the synthesis procedure, acrylate content and photoinitiator content seems justified. In parallel, a search for other material candidates should be performed.

It is worth consideration that since 2006, the cost of fuel cells has declined by 60% , and is expected to fall slightly in the coming years.[70] With large production volumes, it is estimated that the GDL only make up 6% of the cost, while the CL and bipolar plates becomes an increasingly dominant cost factor.[70] That means that a 3D printed could only would provide incremental reductions to the total cost of a fuel cell.

It could be interesting to look into combining the bipolar plates, the GDL and the MPL into one component. This would have a more significant impact as the bipolar plates make up as much as 80% of the total mass and a much larger part of the cost of a fuel cell.[70] Bi-polar plates have challenges related to oxidization which limits longevity. Polymers with high graphite or carbon content have been evaluated by researchers for the bi-polar plates.[71]

Thank you for reading,
Fredrik Cappelen

Bibliography

[1] Fredrik Cappelen. Investigating the possibility of using stereolithographic 3d printing to manufacture gas diffusion layers for hydrogen fuel cells. Unpublished specialization project NTNU, 2018.

[2] University of Thessaly Department of Mechanical Engineering. Fuel Cell Effiency and Performance. URL http://www.mie.uth.gr/ekp_yliko/ThermodynamicsFuelCells.pdf. Visited: 2018-12-19.

[3] J. M. Andjar and F. Segura. Fuel cells: History and updating. A walk along two centuries. *Renewable and Sustainable Energy Reviews*, 13(9):2309–2322, December 2009. ISSN 1364-0321. doi: 10.1016/j.rser.2009.03.015. URL http://www.sciencedirect.com/science/article/pii/S1364032109001336.

[4] F. Barbir and T. Gmez. Efficiency and economics of proton exchange membrane (PEM) fuel cells. *International Journal of Hydrogen Energy*, 22(10):1027–1037, October 1997. ISSN 0360-3199. doi: 10.1016/S0360-3199(96)00175-9. URL http://www.sciencedirect.com/science/article/pii/S0360319996001759.

[5] T. Maiyalagan and Sivakumar Pasupathi. Components for pem fuel cells: An overview. *Materials Science Forum*, 657:143–189, 11 2010. doi: 10.4028/www.scientific.net/MSF.657.143.

[6] R. O'Hayre, S.W. Cha, W Colella, and F.B Prinz. *Fuel cell fundamentals*. Wiley, Hoboken, N.J, 2nd ed. edition, 2009. ISBN 9780470258439. pages: 37-42.

[7] *Lange's handbook of chemistry*. McGraw-Hill standard handbooks. McGraw-Hill, New York, 16th ed. edition, 2005. ISBN 0071432205.

[8] NPTEL :: Chemical Engineering - Fuel Cell Technology, . URL https://nptel.ac.in/courses/103102015/7. Visited: 2018-12-18.

[9] NPTEL :: Chemical Engineering - Fuel Cell Technology, . URL https://nptel.ac.in/courses/103102015/1. Visited: 2018-10-12.

[10] R. O'Hayre, S.W. Cha, W Colella, and F.B Prinz. *Fuel cell fundamentals*. Wiley, Hoboken, N.J, 2nd ed. edition, 2009. ISBN 9780470258439. pages: 71-106.

[11] NPTEL :: Chemical Engineering - Fuel Cell Technology, . URL `https://nptel.ac.in/courses/103102015/9`. Visited: 2018-12-18.

[12] R. O'Hayre, S.W. Cha, W Colella, and F.B Prinz. *Fuel cell fundamentals*. Wiley, Hoboken, N.J, 2nd ed. edition, 2009. ISBN 9780470258439. pages: 111–127.

[13] R. O'Hayre, S.W. Cha, W Colella, and F.B Prinz. *Fuel cell fundamentals*. Wiley, Hoboken, N.J, 2nd ed. edition, 2009. ISBN 9780470258439. pages 161-193.

[14] Adam Z. Weber and John Newman. Effects of Microporous Layers in Polymer Electrolyte Fuel Cells. *Journal of The Electrochemical Society*, 152(4):A677–A688, April 2005. ISSN 0013-4651, 1945-7111. doi: 10.1149/1.1861194. URL `http://jes.ecsdl.org/content/152/4/A677`.

[15] Sehkyu Park, Jong-Won Lee, and Branko N. Popov. A review of gas diffusion layer in PEM fuel cells: Materials and designs. *International Journal of Hydrogen Energy*, 37(7):5850–5865, April 2012. ISSN 0360-3199. doi: 10.1016/j.ijhydene.2011.12.148. URL `http://www.sciencedirect.com/science/article/pii/S0360319911028825`.

[16] (PDF) SIGRACET Gas Diffusion Layers for PEM Fuel Cells, Electrolyzers and Batteries (White Paper). URL `https://www.researchgate.net/publication/295859224_SIGRACETR_Gas_Diffusion_Layers_for_PEM_Fuel_Cells_Electrolyzers_and_Batteries_White_Paper`. Visited: 2018-2-18.

[17] Feng-Yuan Zhang, Suresh G. Advani, and Ajay K. Prasad. Performance of a metallic gas diffusion layer for PEM fuel cells. *Journal of Power Sources*, 176(1):293–298, January 2008. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2007.10.055. URL `http://www.sciencedirect.com/science/article/pii/S0378775307023324`.

[18] Kazuyoshi Fushinobu, Daishi Takahashi, and Ken Okazaki. Micromachined metallic thin films for the gas diffusion layer of PEFCs. *Journal of Power Sources*, 158(2):1240–1245, August 2006. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2005.10.080. URL `http://www.sciencedirect.com/science/article/pii/S0378775305014758`.

[19] Arunkumar Jayakumar, Sarat Singamneni, Maximiano Ramos, Ahmed M Al-Jumaily, and Sethu Sundar Pethaiah. Manufacturing the Gas Diffusion Layer for PEM Fuel Cell Using a Novel 3d Printing Technique and Critical Assessment of the Challenges Encountered. *Materials*, 10(7), July 2017. ISSN 1996-1944. doi: 10.3390/ma10070796. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5551839/`.

[20] Jay Benziger, James Nehlsen, David Blackwell, Tom Brennan, and Johannah Itescu. Water flow in the gas diffusion layer of PEM fuel cells. *Journal of Membrane*

Science, 261(1):98–106, September 2005. ISSN 0376-7388. doi: 10.1016/j.memsci.2005.03.049. URL http://www.sciencedirect.com/science/article/pii/S0376738805002620.

[21] Haolin Tang, Shenlong Wang, Mu Pan, and Runzhang Yuan. Porosity-graded microporous layers for polymer electrolyte membrane fuel cells. *Journal of Power Sources*, 166(1):41–46, March 2007. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2007.01.021. URL http://www.sciencedirect.com/science/article/pii/S0378775307000912.

[22] Pranay Shrestha. *Custom Microporous Layers for Polymer Electrolyte Membrane Fuel Cells*. Thesis, June 2018. URL https://tspace.library.utoronto.ca/handle/1807/89510. Visited: 2019-05-13.

[23] Silas Towne, Vish Viswanathan, James Holbery, and Peter Rieke. Fabrication of polymer electrolyte membrane fuel cell MEAs utilizing inkjet print technology. *Journal of Power Sources*, 171(2):575–584, September 2007. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2007.07.017. URL http://www.sciencedirect.com/science/article/pii/S037877530701498X.

[24] A. ElKharouf, N. V. Rees, and R. SteinbergerWilckens. Gas Diffusion Layer Materials and their Effect on Polymer Electrolyte Fuel Cell Performance Ex Situ and In Situ Characterization. *Fuel Cells*, 14(5):735–741, 2014. ISSN 1615-6854. doi: 10.1002/fuce.201300247. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fuce.201300247.

[25] Gas Diffusion Layer (GDL) Comparison Chart, May 2013. URL https://fuelcellsetc.com/helpful-tools/gas-diffusion-layer-gdl-comparison-chart/. Visited: 2019-04-01.

[26] (PDF) SIGRACET Gas Diffusion Layers for PEM Fuel Cells, Electrolyzers and Batteries (White Paper). URL https://www.researchgate.net/publication/295859224_SIGRACETR_Gas_Diffusion_Layers_for_PEM_Fuel_Cells_Electrolyzers_and_Batteries_White_Paper. Visited: 2018-12-18.

[27] Ramji Pandey. Photopolymers in 3d printing applications, 2014. URL http://www.theseus.fi/handle/10024/80083. Visited: 2019-05-15.

[28] FormLabs. The Ultimate Guide to Stereolithography (SLA) 3d Printing. URL https://formlabs.com/blog/ultimate-guide-to-stereolithography-sla-3d-printing/. Visited: 2018-10-16.

[29] Rakesh Sharma. The 3d Printing Revolution You Have Not Heard About. URL https://www.forbes.com/sites/rakeshsharma/2013/07/08/the-3d-printing-revolution-you-have-not-heard-about/. Visited on: 2019-02-4.

[30] Andria Cheng. How Adidas Plans To Bring 3d Printing To The Masses. URL https://www.forbes.com/sites/andriacheng/2018/05/

22/with-adidas-3d-printing-may-finally-see-its-mass-retail-potential/. Visited: 2019-04-13.

[31] Laser sla vs dlp vs masked sla 3d printing technology, March 2017. URL https://theorthocosmos.com/laser-sla-vs-dlp-vs-masked-sla-3d-printing-technology-compared/. Visited: 2019-05-15.

[32] Yayue Pan, Haiyang He, Jie Xu, and Alan Feinerman. Study of separation force in constrained surface projection stereolithography. *Rapid Prototyping Journal*, 23 (2):353–361, March 2017. ISSN 1355-2546. doi: 10.1108/RPJ-12-2015-0188. URL https://www.emeraldinsight.com/doi/full/10.1108/RPJ-12-2015-0188.

[33] bad-zima. SLA/DLP Basics. URL https://www.instructables.com/id/SLADLP-Basics/. Visited: 2018-10-02.

[34] George Wypych. *1 - PHOTOPHYSICS*. Elsevier, Oxford, fifth edition edition, 2013. ISBN 978-1-895198-62-1. doi: https://doi.org/10.1016/B978-1-895198-62-1.50004-4. URL http://www.sciencedirect.com/science/article/pii/B9781895198621500044.

[35] F.P Jacobs. Fundamentals of stereolitography, 1992. URL http://sffsymposium.engr.utexas.edu/Manuscripts/1992/1992-23-Jacobs.pdf. Visited: 03.10.2018.

[36] Texas Instruments. TI DLP Technology. URL http://www.ti.com/lit/sg/dlpt019e/dlpt019e.pdf. Visited: 2019-04-13.

[37] Karl_Guttag. TI DLP Diamond Pixel, February 2012. URL https://www.kguttag.com/2012/02/09/ti-dlp-diamond-pixel/. Visited: 2019-04-13.

[38] Hua C. Gong, Michael J. Beauchamp, Steven F. Perry, Adam T. Woolley, and Gregory Nordin. Optical Approach to Resin Formulation for 3d Printed Microfluidics. *RSC advances*, 5(129):106621–106632, 2015. doi: 10.1039/C5RA23855B.

[39] Hua Gong, Bryce P. Bickham, Adam T. Woolley, and Gregory P. Nordin. Custom 3d printer and resin for 18 m 20 m microfluidic flow channels. *Lab on a Chip*, 17(17):2899–2909, August 2017. ISSN 1473-0189. doi: 10.1039/C7LC00644F. URL https://pubs.rsc.org/en/content/articlelanding/2017/lc/c7lc00644f.

[40] (1) (PDF) Microstereolithography: A review, . URL https://www.researchgate.net/publication/37422063_Microstereolithography_A_review. Visited: 2019-03-04.

[41] (1) Microstereolithography: A new process to build complex 3d objects | Request PDF, . URL https://www.researchgate.net/publication/37422018_Microstereolithography_A_new_process_to_build_complex_3D_objects. Visited:2019-03-04.

[42] Xiaoyu Zheng, Howon Lee, Todd H. Weisgraber, Maxim Shusteff, Joshua DeOtte, Eric B. Duoss, Joshua D. Kuntz, Monika M. Biener, Qi Ge, Julie A. Jackson, Sergei O. Kucheyev, Nicholas X. Fang, and Christopher M. Spadaccini. Ultralight, ultrastiff mechanical metamaterials. *Science*, 344(6190):1373–1377, June 2014. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1252291. URL `https://science.sciencemag.org/content/344/6190/1373`.

[43] ntpology: Element, . URL `https://www.ntopology.com/element/`. Visited: 2018-12-22.

[44] Pillow Pillow (PIL Fork) 6.0.0 documentation. URL `https://pillow.readthedocs.io/en/stable/`. Visited: 2019-05-07.

[45] Products W2p | SolFlex 3d Printers. URL `https://www.way2production.at/en/products/`. Visited: 2019-04-03.

[46] FEP 127hd - A4 Size. URL `https://www.fepshop.com/shop/fep-film/fep-127/fep-127hd-a4/`. Visited: 2019-03-01.

[47] EnvisionTEC. Advanced DLP For Superior 3d Printing. URL `https://envisiontec.com/wp-content/uploads/2017/04/Why-EnvisionTEC-DLP-3D-Printing-is-Better-rebranded.pdf`. Visited: 2019-05-22.

[48] Autodesk offers grayscale trick for DLP 3d printing at sub-pixel resolution. URL `http://www.3ders.org/articles/20160815-autodesk-offers-grayscale-trick-for-dlp-3d-printing-at-sub-pixel-resolution.html`. Visited: 2019-05-22.

[49] Dongmin Wu, Nicholas Fang, Cheng Sun, and Xiang Zhang. Stiction problems in releasing of 3d microstructures and its solution. *Sensors and Actuators A: Physical*, 128(1):109–115, March 2006. ISSN 0924-4247. doi: 10.1016/j.sna.2005.12.041. URL `http://www.sciencedirect.com/science/article/pii/S0924424705007739`.

[50] unknown. Surface tension values of some common test liquids for surface energy analysis. URL `http://www.surface-tension.de/`. Visited: 2019-06-2.

[51] Polymer Science Learning Center. Polyimides. URL `https://www.pslc.ws/macrog/imide.htm`. Visited: 2018-12-19.

[52] Paul M. Hergenrother. The Use, Design, Synthesis, and Properties of High Performance/High Temperature Polymers: An Overview. *High Performance Polymers*, 15 (1):3–45, March 2003. ISSN 0954-0083. doi: 10.1177/095400830301500101. URL `https://doi.org/10.1177/095400830301500101`.

[53] Chang-Sik Ha and Stella Mathews. *Polyimides and High Performance Organic Polymers*. 01 2011. ISBN 978-3-642-19076-6. doi: 10.1007/978-3-642-19077-3_1.

[54] Regal Plastics. Properties Comparison | Regal Plastics. URL https://www.regal-plastics.com/other-products/technical-information/properties-comparison/. Visited: 2019-05-07.

[55] MATERIALS SCIENCE AND TECHNOLOGY NEWSLETTER Vol. 2, No. 2. URL http://www.mstconf.com/Vol2No2-2005.htm. Visited: 2018-12-18.

[56] Shi-Yong Yang. *Advanced Polyimide Materials: Synthesis, Characterization, and Applications*. Elsevier, April 2018. ISBN 978-0-12-812641-7. Google-Books-ID: QilYDwAAQBAJ.

[57] Maruti Hegde, Viswanath Meenakshisundaram, Nicholas Chartrain, Susheel Sekhar, Danesh Tafti, Christopher B. Williams, and Timothy E. Long. 3d Printing All-Aromatic Polyimides using Mask-Projection Stereolithography: Processing the Non-processable. *Advanced Materials (Deerfield Beach, Fla.)*, 29(31), August 2017. ISSN 1521-4095. doi: 10.1002/adma.201701240.

[58] Jana Herzberger, Viswanath Meenakshisundaram, Christopher B. Williams, and Timothy E. Long. 3d Printing All-Aromatic Polyimides Using Stereolithographic 3d Printing of Polyamic Acid Salts. *ACS Macro Letters*, 7(4):493–497, April 2018. doi: 10.1021/acsmacrolett.8b00126. URL https://doi.org/10.1021/acsmacrolett.8b00126.

[59] Jana Herzberger, Viswanath Meenakshisundaram, Christopher B. Williams, and Timothy E. Long. 3d Printing All-Aromatic Polyimides Using Stereolithographic 3d Printing of Polyamic Acid Salts. *ACS Macro Letters*, 7(4):493–497, April 2018. doi: 10.1021/acsmacrolett.8b00126. URL https://doi.org/10.1021/acsmacrolett.8b00126.

[60] DUPONT. DUPONT Kapton - Summary of properties. URL http://www.dupont.com/content/dam/dupont/products-and-services/membranes-and-films/polyimde-films/documents/DEC-Kapton-summary-of-properties.pdf. Visited: 2018-12-19.

[61] Gwyn Morgan Jenkins. *Polymeric carbons : carbon fibre, glass and char*. Cambridge University Press, Cambridge, 1976. ISBN 0521206936.

[62] H. B. Brom, Y. Tomkiewicz, A. Aviram, A. Broers, and B. Sunners. On a new conducting polymer-pyrolyzed Kapton. *Solid State Communications*, 35(2): 135–139, July 1980. ISSN 0038-1098. doi: 10.1016/0038-1098(80)90230-6. URL http://www.sciencedirect.com/science/article/pii/0038109880902306.

[63] A. Brger, E. Fitzer, M. Heym, and B. Terwiesch. Polyimides as precursors for artificial carbon. *Carbon*, 13(3):149–157, June 1975. ISSN 0008-6223. doi: 10.1016/0008-6223(75)90225-0. URL http://www.sciencedirect.com/science/article/pii/0008622375902250.

[64] Michio Inagaki, Sunao Harada, Tetsuhito Sato, Tsuyoshi Nakajima, Yuji Horino, and Kenji Morita. Carbonization of polyimide film Kapton. *Carbon*, 27(2): 253–257, January 1989. ISSN 0008-6223. doi: 10.1016/0008-6223(89)90131-0. URL http://www.sciencedirect.com/science/article/pii/0008622389901310.

[65] M. Inagaki, Y. Hishiyama, and Y. Kaburagi. Effect of heating rate during carbonization on graphitization of carbon films derived from aromatic polyimides. *Carbon*, 32 (4):637–639, January 1994. ISSN 0008-6223. doi: 10.1016/0008-6223(94)90083-3. URL http://www.sciencedirect.com/science/article/pii/0008622394900833.

[66] T. Takeichi, Y. Eguchi, Y. Kaburagi, Y. Hishiyama, and M. Inagaki. Carbonization and graphitization of Kapton-type polyimide films prepared from polyamide alkyl ester. *Carbon*, 36(1):117–122, January 1998. ISSN 0008-6223. doi: 10.1016/S0008-6223(97)00155-3. URL http://www.sciencedirect.com/science/article/pii/S0008622397001553.

[67] cubicure - Cubicure ThermoBlast. URL http://www.cubicure.com/portfolio/materials/thermoblast. Visited: 2019-05-26.

[68] R. P. Clark. The thermal decomposition of poly-4,4'-oxydiphenylene pyromellitimide. *Thermochimica Acta*, 6(5):473–480, July 1973. ISSN 0040-6031. doi: 10.1016/0040-6031(73)85079-8. URL http://www.sciencedirect.com/science/article/pii/0040603173850798.

[69] L. Levan, Semprimoschnig, and Van Eesbeek. A DETAILED STUDY IN THE THERMAL ENDURANCE OF POLYIMIDE RESIN MATERIALS. Technical report, ESA-ESTEC, Materials Space Evaluation & Radiation Effects Section. URL http://esmat.esa.int/materials_news/isme09/pdf/3-Ground/Poster%20Ground%20Testing%20-%20Levan.pdf.
Visited:2019-05-21.

[70] Fact of the Month April 2018: Fuel Cell Cost Decreased by 60% since 2006. URL https://www.energy.gov/eere/fuelcells/fact-month-april-2018-fuel-cell-cost-decreased-60-2006.

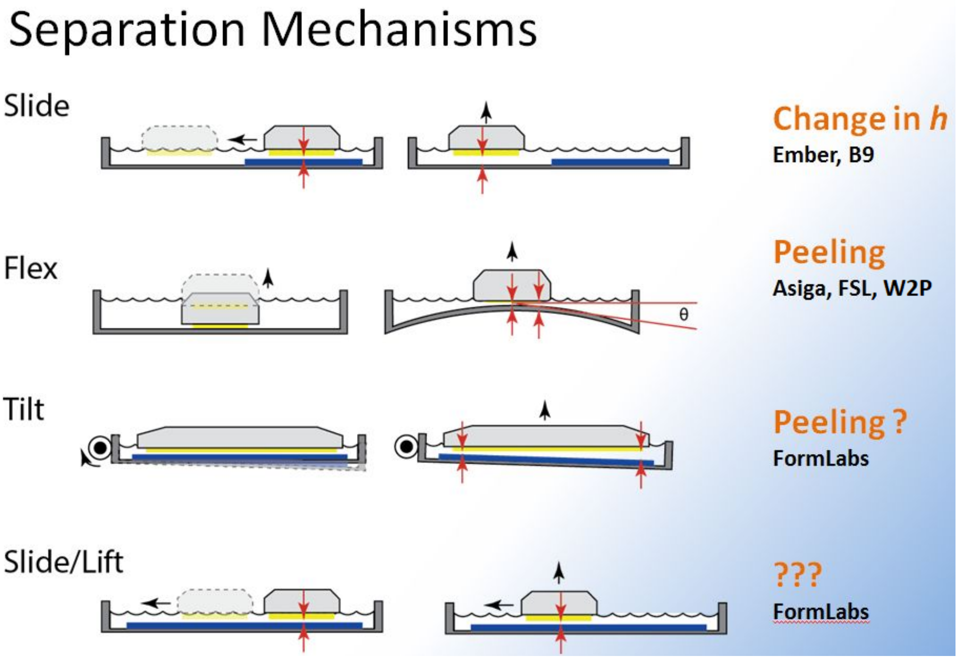
[71] E. Planes, L. Flandin, and N. Alberola. Polymer Composites Bipolar Plates for PEMFCs. *Energy Procedia*, 20:311–323, January 2012. ISSN 1876-6102. doi: 10.1016/j.egypro.2012.03.031. URL http://www.sciencedirect.com/science/article/pii/S1876610212007618.

Appendix

Appendix A

Background

A.1 Peeling Mechanisms



Separation Mechanisms

Slide

Change in *h*

Ember, B9

Flex

Peeling

Asiga, FSL, W2P

Tilt

Peeling ?

FormLabs

Slide/Lift

???

FormLabs

Figure A.1: Different peeling mechanisms for bottom up printers to minimize peel forces

A.2 Working Curve of BASF x004 and Virginia Tech PAA-resin

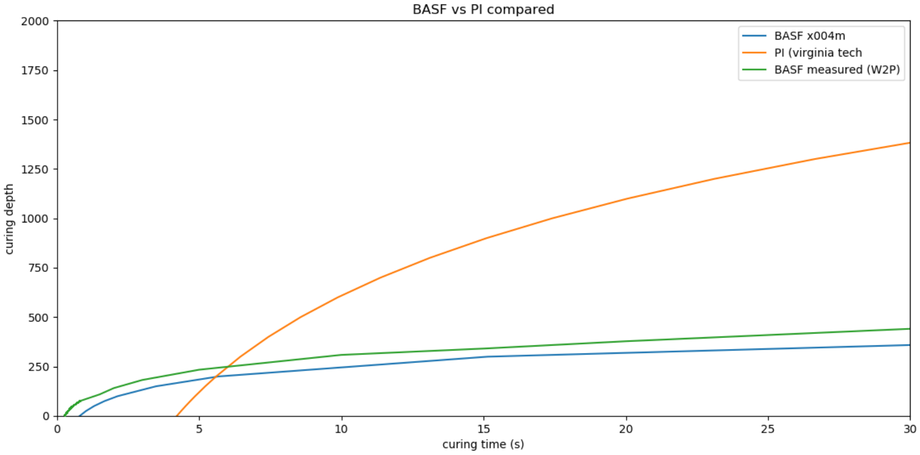


Figure A.2: Working curves: orange PI plotted based on VT results, green is BASF x004m calculated from data sheet, blue is BASF x004m as measured by W2P. The x-axis is in this case kept linear to show the logarithmic nature of the Cd vs Ed

Appendix B

Microscope Pictures

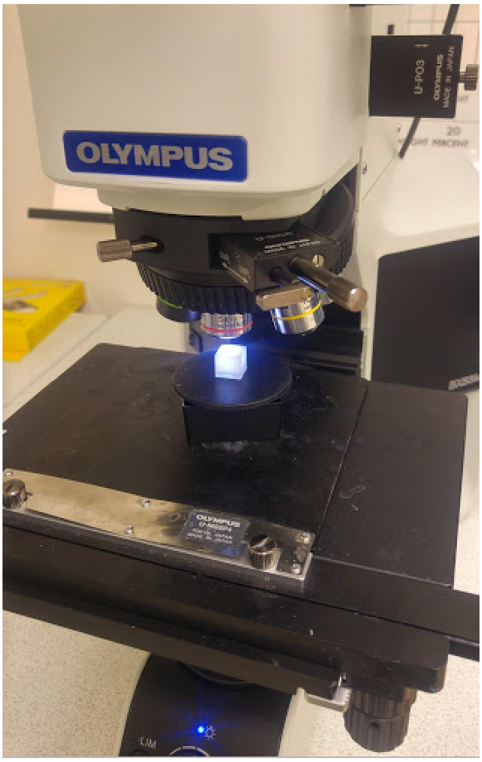


Figure B.1: Olympus microscope setup used throughout the thesis

B.1 Cubic lattices printed at RGB 200 and RGB 255

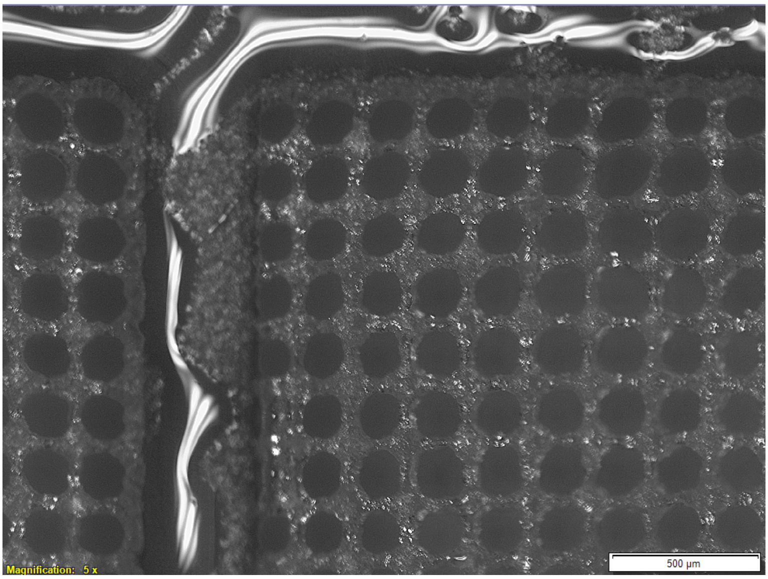


Figure B.2: 50uM mesh printed with default settings and RGB value of 200 for the small features. Compare it to the picture below.

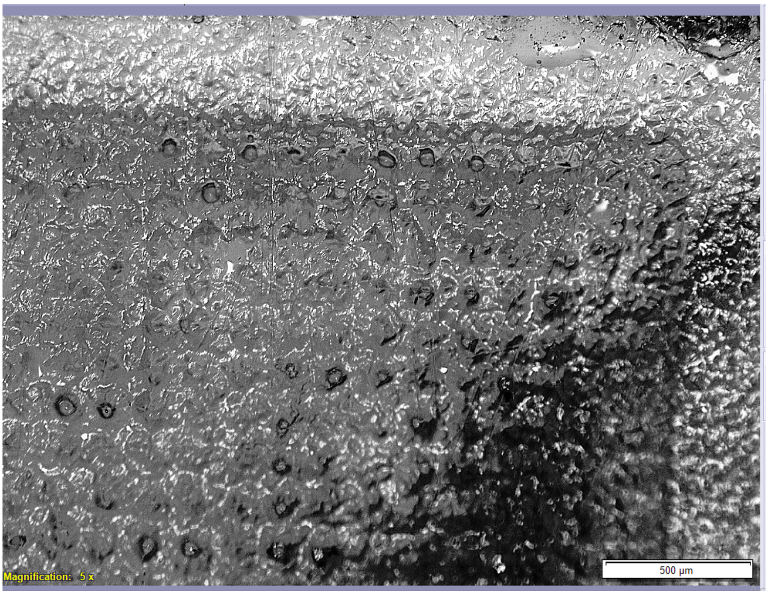


Figure B.3: 50uM mesh printed with default settings and RGB value of 255 for the small features

B.2 Variation in 50uM beams

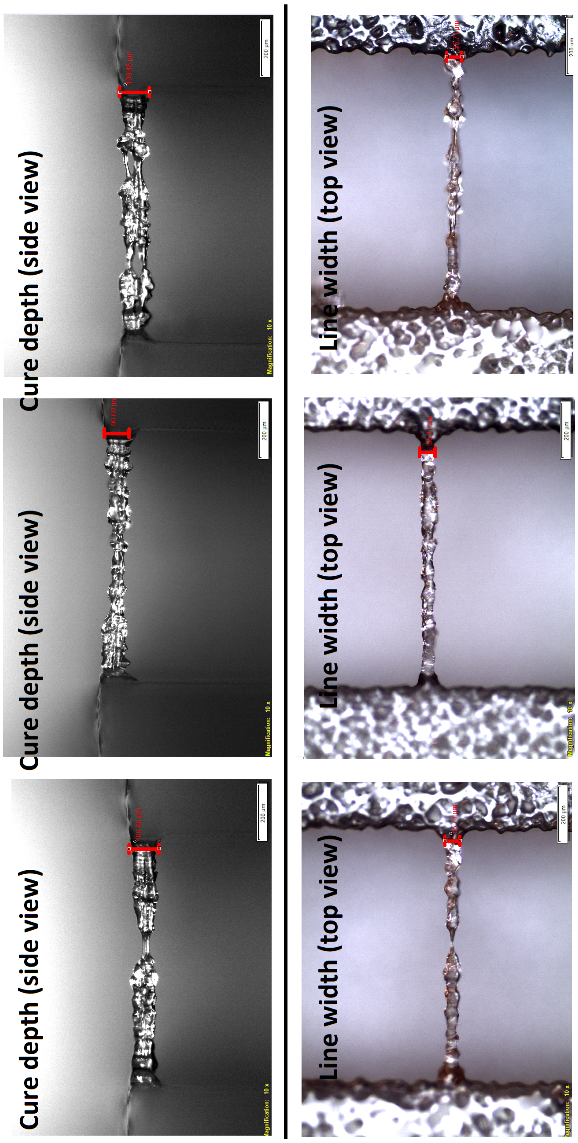


Figure B.4: Three single pixel (50uM) beams printed as two 25uM layers at ”default” settings (Equation 3.1) seen from two angles. Note the variation in the cured profile along the length of the beams

B.3 Effect of Capillary Forces on 50um Beams

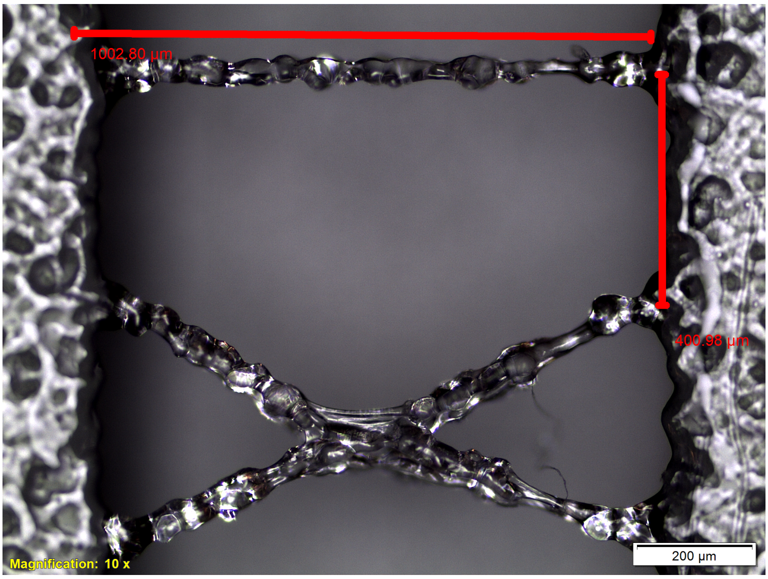


Figure B.5: Capillary collapse (buckling) of 50um beams spaced 400um apart

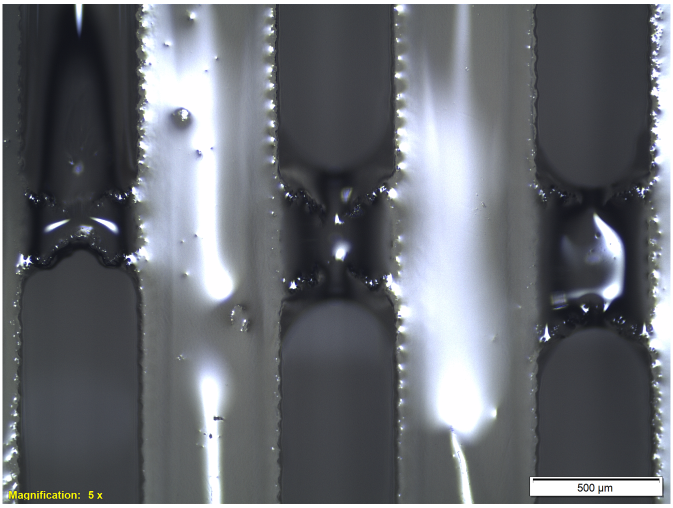


Figure B.6: Capillary forces during drying bending the beams, and IPA sticking to the surface (surface tension)

B.4 Vertical Resolution Testing

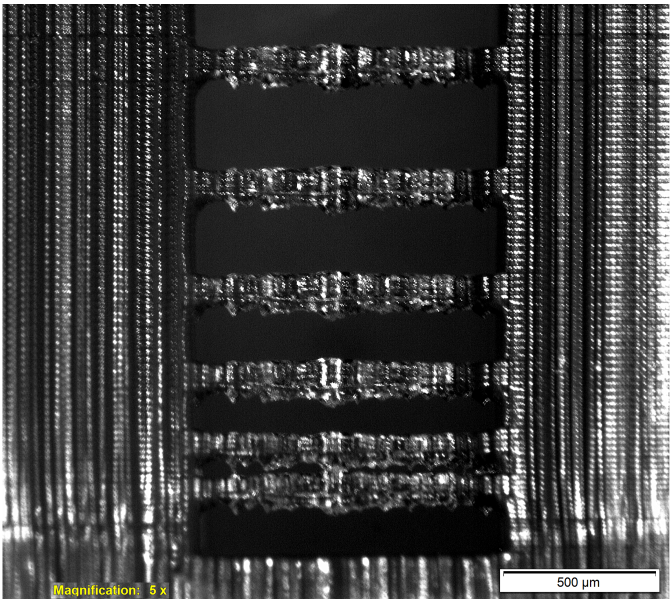


Figure B.7: 100uM beams made from four 25uM layers. We see that vertical gaps as small as 150uM can be printed without the beams fusing together.

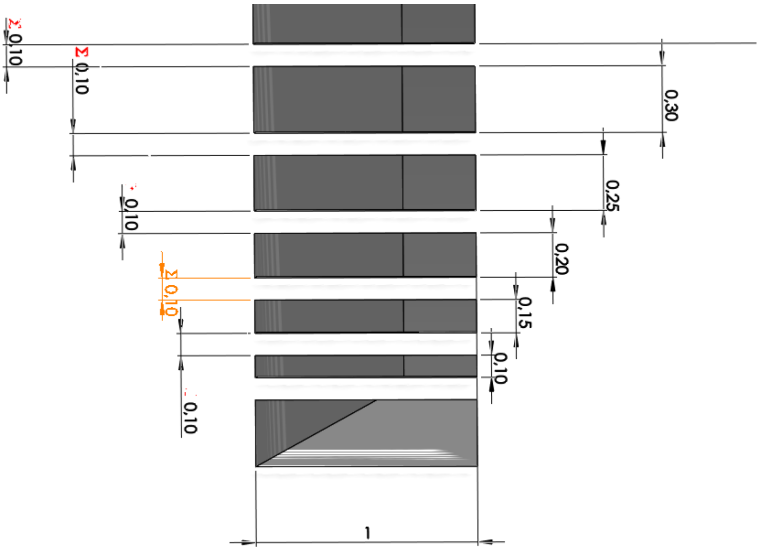


Figure B.8: The CAD file with dimensions for the part in Figure B.7

Appendix C

Beam Length Specimens and Observations

C.1 Test Geometry of Large V-Specimens

C.2 Placement of Large V-Specimens

C.3 Large Specimen Collapsing During Post Processing

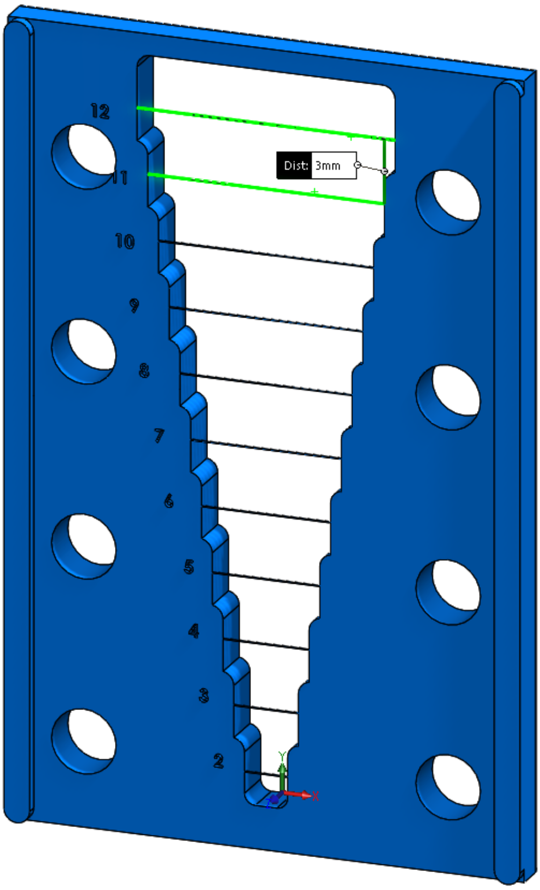


Figure C.1: Large V-Type test specimen with 3mm beam spacing and maximum span of 12mm

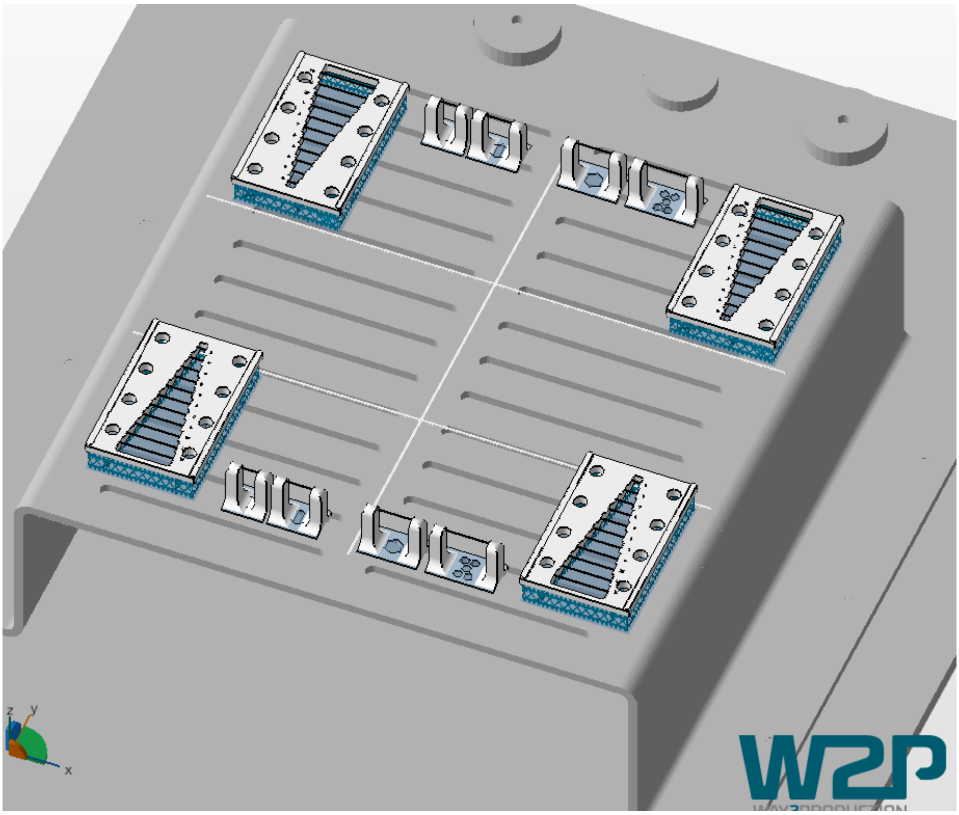


Figure C.2: Placement used to test enlarged specimens which data was presented in 5.7

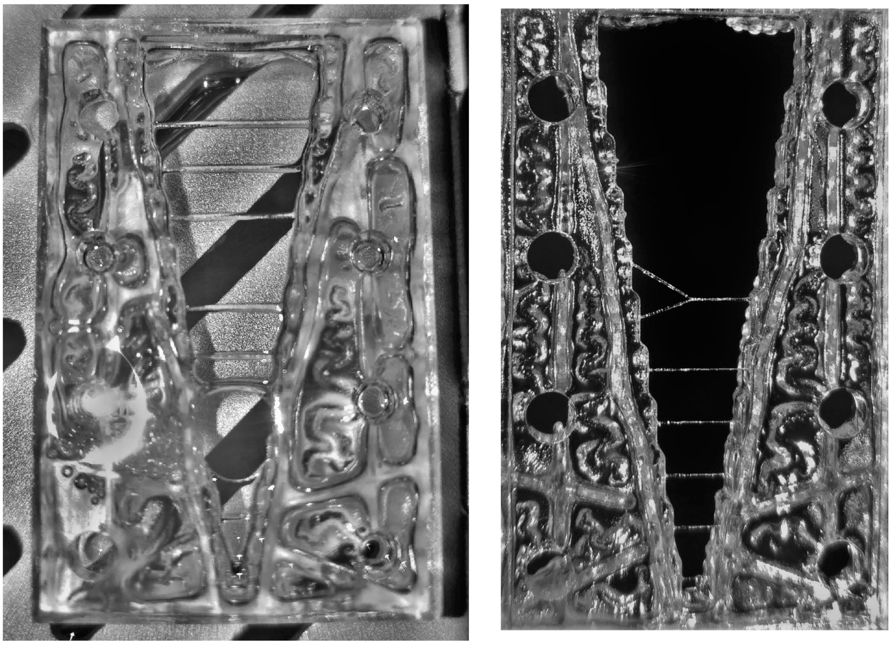


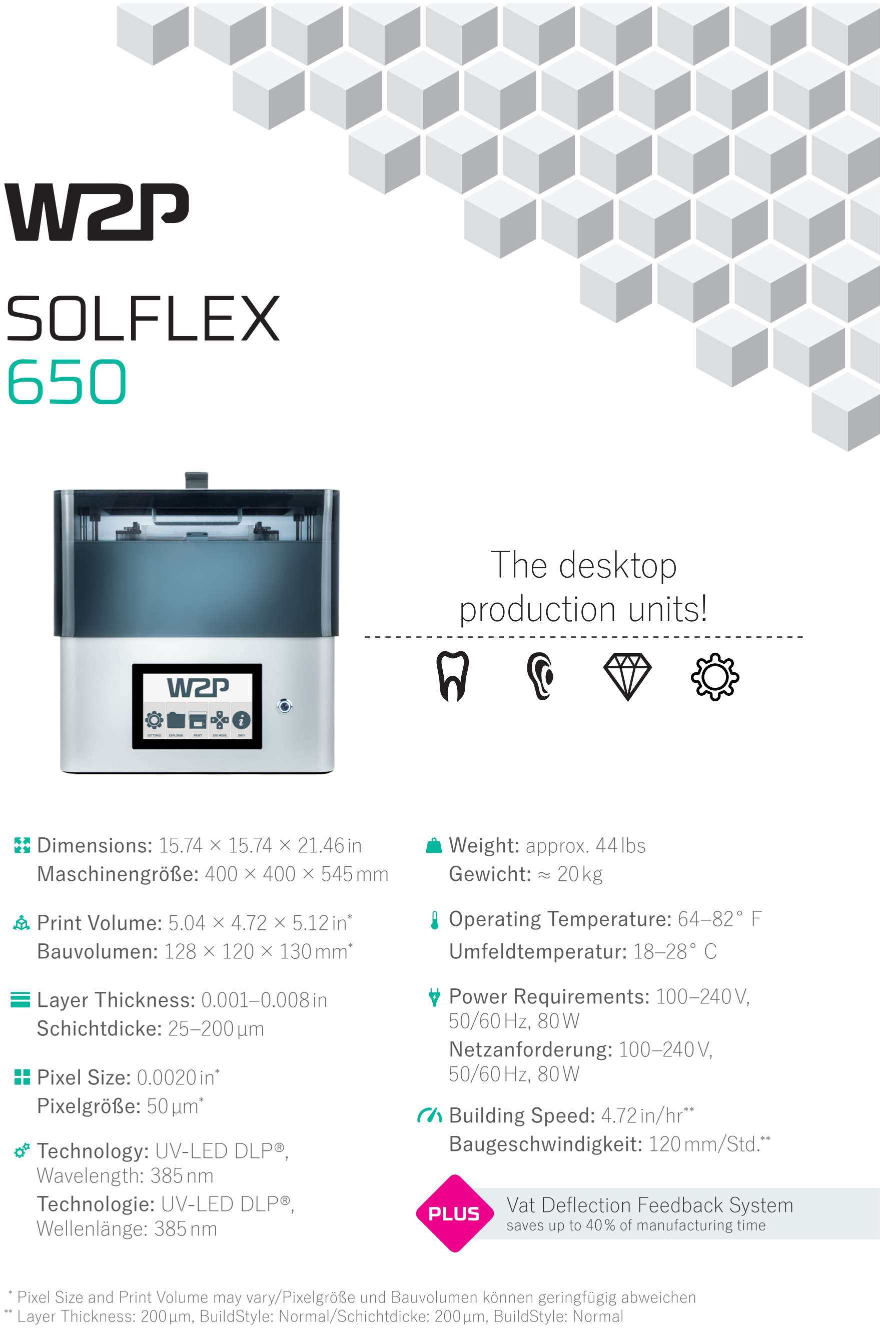
Figure C.3: A beam length specimen directly after printing left with beams up to 11mm surviving the print process and the same specimen quickly rinsed in IPA right. We see how several beams collapse in this process

Appendix D

Datasheets

D.1 Solflex 650

D.2 Basf Photo-Resin X004M



W2P

SOLFLEX
650

The desktop production units!

Dimensions: 15.74 × 15.74 × 21.46 in
Maschinengröße: 400 × 400 × 545 mm

Print Volume: 5.04 × 4.72 × 5.12 in*
Bauvolumen: 128 × 120 × 130 mm*

Layer Thickness: 0.001–0.008 in
Schichtdicke: 25–200 µm

Pixel Size: 0.0020 in*
Pixelgröße: 50 µm*

Technology: UV-LED DLP®, Wavelength: 385 nm
Technologie: UV-LED DLP®, Wellenlänge: 385 nm

Weight: approx. 44 lbs
Gewicht: ≈ 20 kg

Operating Temperature: 64–82° F
Umfeldtemperatur: 18–28° C

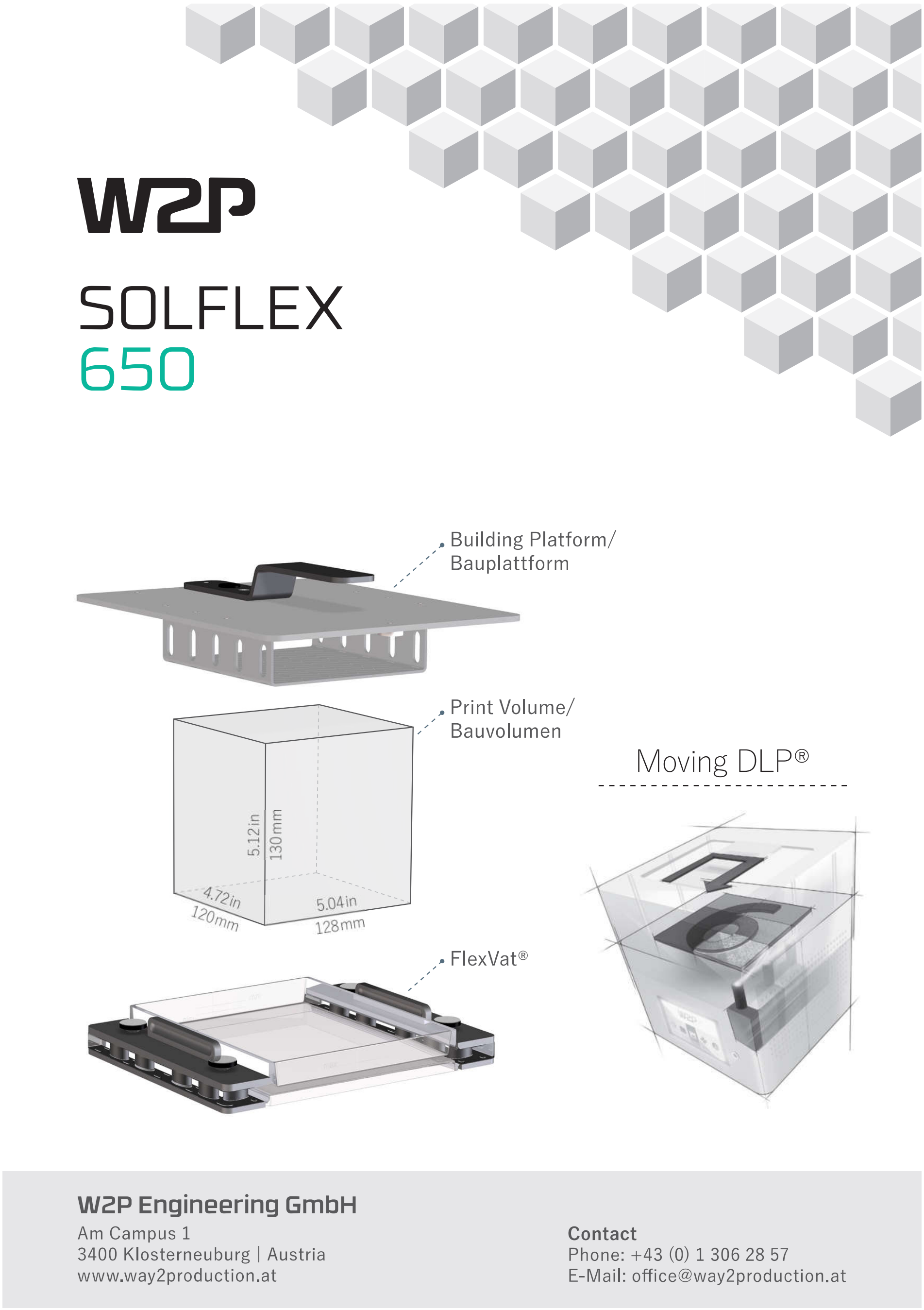
Power Requirements: 100–240 V, 50/60 Hz, 80 W
Netzanforderung: 100–240 V, 50/60 Hz, 80 W

Building Speed: 4.72 in/hr**
Baugeschwindigkeit: 120 mm/Std.**

PLUS Vat Deflection Feedback System
saves up to 40 % of manufacturing time

* Pixel Size and Print Volume may vary/Pixelgröße und Bauvolumen können geringfügig abweichen
** Layer Thickness: 200 µm, BuildStyle: Normal/Schichtdicke: 200 µm, BuildStyle: Normal

Figure D.1: Solflex 650 datasheet p1



W2P

SOLFLEX
650

Building Platform/
Bauplattform

Print Volume/
Bauvolumen

5.12 in
130 mm

4.72 in
120 mm

5.04 in
128 mm

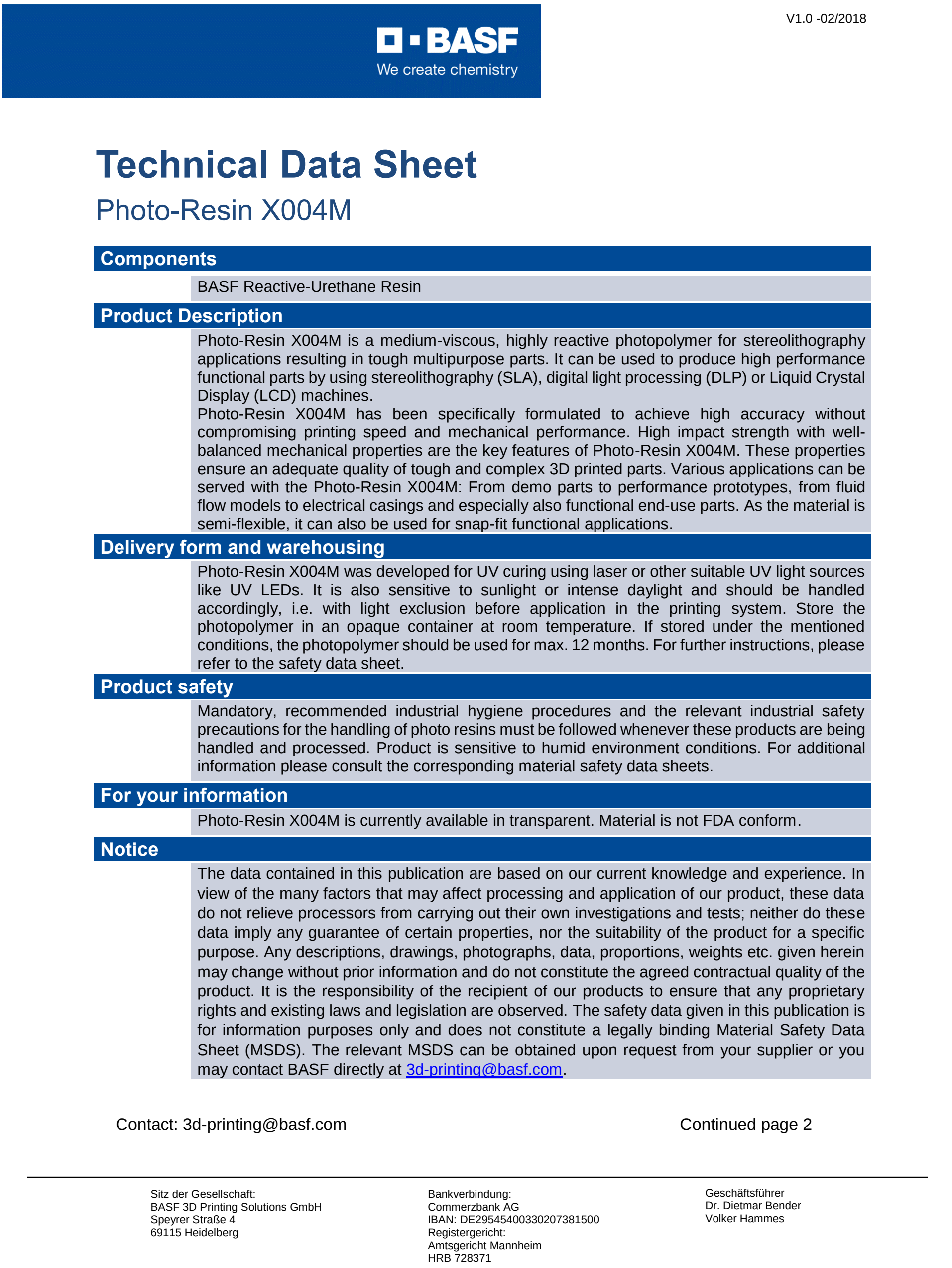
FlexVat®

Moving DLP®

W2P Engineering GmbH
Am Campus 1
3400 Klosterneuburg | Austria
www.way2production.at

Contact
Phone: +43 (0) 1 306 28 57
E-Mail: office@way2production.at

Figure D.2: Solflex 650 datasheet p2



BASF
We create chemistry

V1.0 -02/2018

Technical Data Sheet

Photo-Resin X004M

Components

BASF Reactive-Urethane Resin

Product Description

Photo-Resin X004M is a medium-viscous, highly reactive photopolymer for stereolithography applications resulting in tough multipurpose parts. It can be used to produce high performance functional parts by using stereolithography (SLA), digital light processing (DLP) or Liquid Crystal Display (LCD) machines.

Photo-Resin X004M has been specifically formulated to achieve high accuracy without compromising printing speed and mechanical performance. High impact strength with well-balanced mechanical properties are the key features of Photo-Resin X004M. These properties ensure an adequate quality of tough and complex 3D printed parts. Various applications can be served with the Photo-Resin X004M: From demo parts to performance prototypes, from fluid flow models to electrical casings and especially also functional end-use parts. As the material is semi-flexible, it can also be used for snap-fit functional applications.

Delivery form and warehousing

Photo-Resin X004M was developed for UV curing using laser or other suitable UV light sources like UV LEDs. It is also sensitive to sunlight or intense daylight and should be handled accordingly, i.e. with light exclusion before application in the printing system. Store the photopolymer in an opaque container at room temperature. If stored under the mentioned conditions, the photopolymer should be used for max. 12 months. For further instructions, please refer to the safety data sheet.

Product safety

Mandatory, recommended industrial hygiene procedures and the relevant industrial safety precautions for the handling of photo resins must be followed whenever these products are being handled and processed. Product is sensitive to humid environment conditions. For additional information please consult the corresponding material safety data sheets.

For your information

Photo-Resin X004M is currently available in transparent. Material is not FDA conform.

Notice

The data contained in this publication are based on our current knowledge and experience. In view of the many factors that may affect processing and application of our product, these data do not relieve processors from carrying out their own investigations and tests; neither do these data imply any guarantee of certain properties, nor the suitability of the product for a specific purpose. Any descriptions, drawings, photographs, data, proportions, weights etc. given herein may change without prior information and do not constitute the agreed contractual quality of the product. It is the responsibility of the recipient of our products to ensure that any proprietary rights and existing laws and legislation are observed. The safety data given in this publication is for information purposes only and does not constitute a legally binding Material Safety Data Sheet (MSDS). The relevant MSDS can be obtained upon request from your supplier or you may contact BASF directly at 3d-printing@basf.com.

Contact: 3d-printing@basf.com

Continued page 2

Sitz der Gesellschaft:
BASF 3D Printing Solutions GmbH
Speyrer Straße 4
69115 Heidelberg

Bankverbindung:
Commerzbank AG
IBAN: DE29545400330207381500
Registergericht:
Amtsgericht Mannheim
HRB 728371

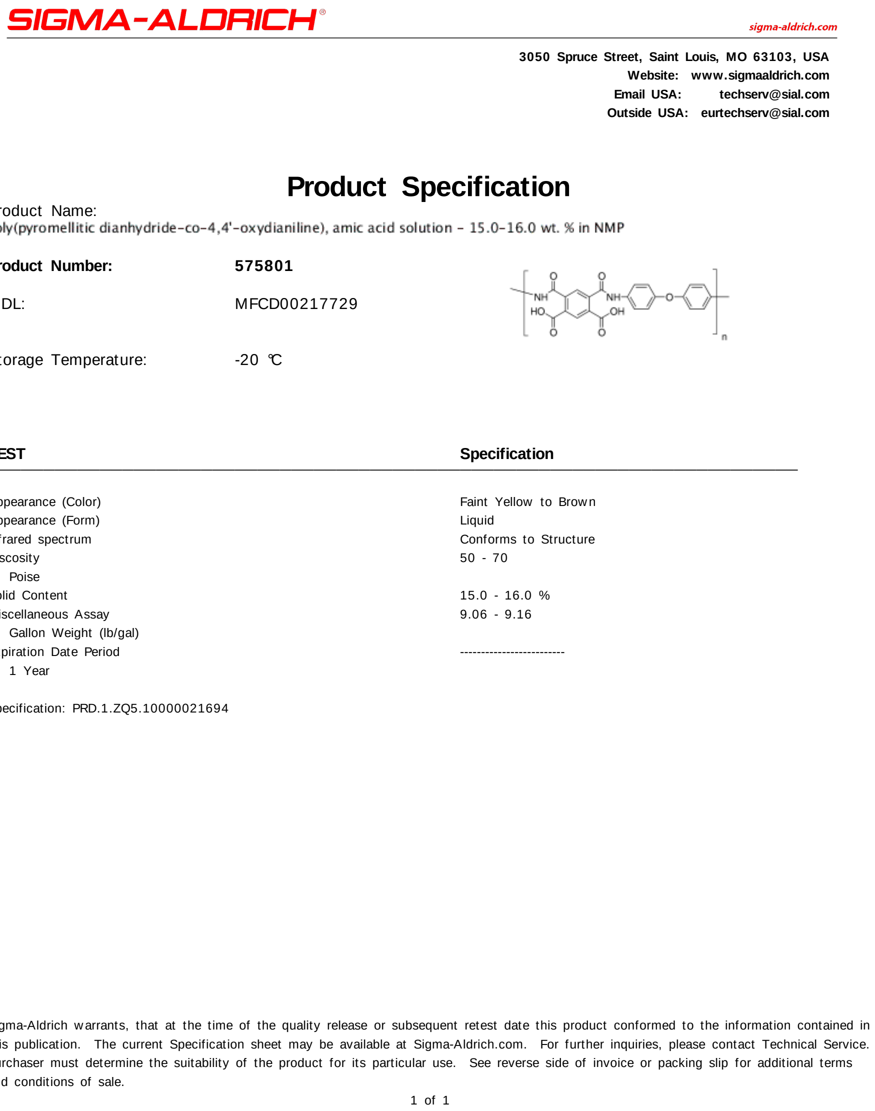
Geschäftsführer
Dr. Dietmar Bender
Volker Hammes

Figure D.3: BASF x004m datasheet

Appendix E

Polyimide: Materials, Instructions and Carbonization

E.1 Polyamic Acid (PMDA-ODA precursor) used to produce UV-Curable PAA-resin


NH
HO
O
O
NH
OH
O
O
O
n

SIGMA-ALDRICH®

sigma-aldrich.com

3050 Spruce Street, Saint Louis, MO 63103, USA
Website: www.sigmaaldrich.com
Email USA: techserv@sial.com
Outside USA: eurtechserv@sial.com

Product Specification

Product Name:
Poly(pyromellitic dianhydride-co-4,4'-oxydianiline), amic acid solution – 15.0-16.0 wt. % in NMP

Product Number: **575801**

MDL: MFCD00217729

Storage Temperature: -20 ℃

TEST	**Specification**
Appearance (Color)	Faint Yellow to Brown
Appearance (Form)	Liquid
Infrared spectrum	Conforms to Structure
Viscosity	50 - 70
Poise	
Solid Content	15.0 - 16.0 %
Miscellaneous Assay	9.06 - 9.16
Gallon Weight (lb/gal)	
Expiration Date Period	-------------------------
1 Year	

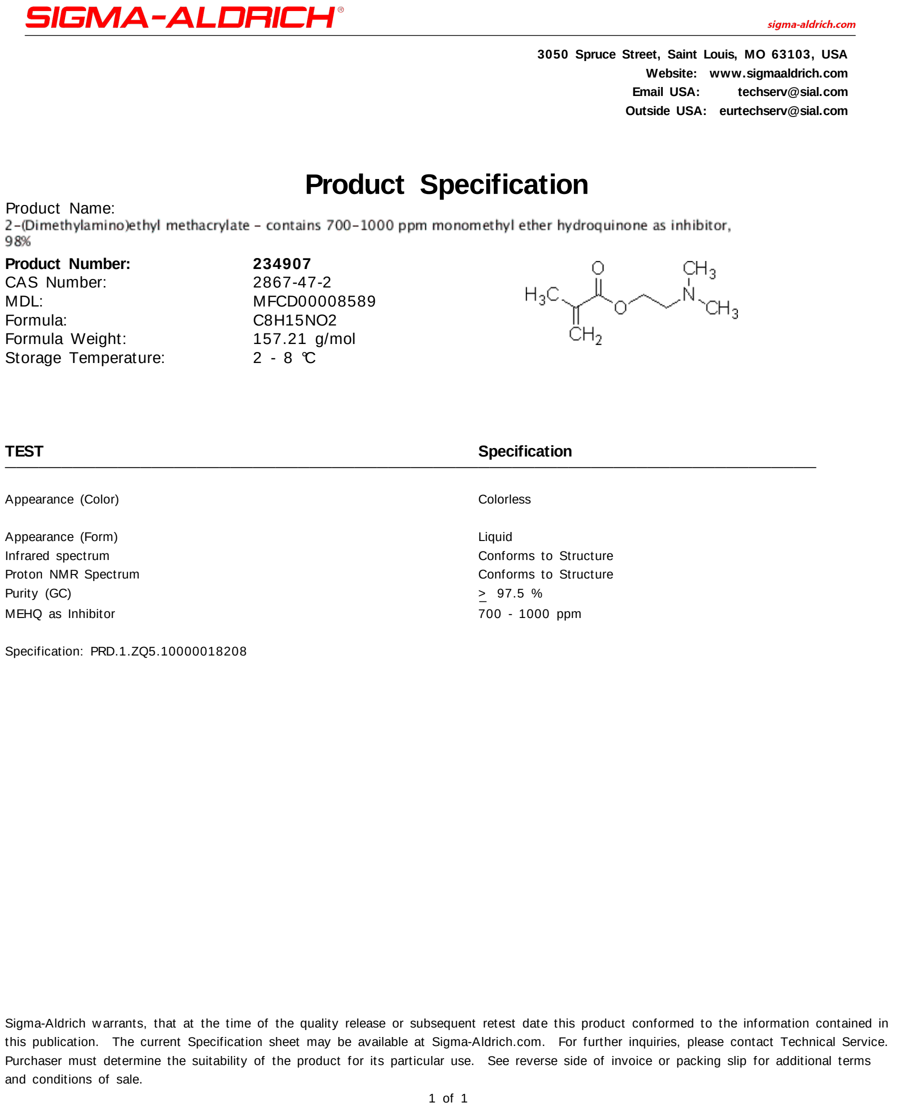
Specification: PRD.1.ZQ5.10000021694

Sigma-Aldrich warrants, that at the time of the quality release or subsequent retest date this product conformed to the information contained in this publication. The current Specification sheet may be available at Sigma-Aldrich.com. For further inquiries, please contact Technical Service. Purchaser must determine the suitability of the product for its particular use. See reverse side of invoice or packing slip for additional terms and conditions of sale.

1 of 1

Figure E.1: Polyamic-Acid datasheet (PMDA-ODA precursor. Also known under brand name Pyre M.L RC5019)

E.2 Methacrylate (DMAEMA) used to produce the UV-Curable PAA-resin



SIGMA-ALDRICH®

sigma-aldrich.com

3050 Spruce Street, Saint Louis, MO 63103, USA
Website: www.sigmaaldrich.com
Email USA: techserv@sial.com
Outside USA: eurtechserv@sial.com

Product Specification

Product Name:
2-(Dimethylamino)ethyl methacrylate - contains 700-1000 ppm monomethyl ether hydroquinone as inhibitor, 98%

Product Number:	**234907**
CAS Number:	2867-47-2
MDL:	MFCD00008589
Formula:	C8H15NO2
Formula Weight:	157.21 g/mol
Storage Temperature:	2 - 8 ℃

TEST	Specification
Appearance (Color)	Colorless
Appearance (Form)	Liquid
Infrared spectrum	Conforms to Structure
Proton NMR Spectrum	Conforms to Structure
Purity (GC)	≥ 97.5 %
MEHQ as Inhibitor	700 - 1000 ppm

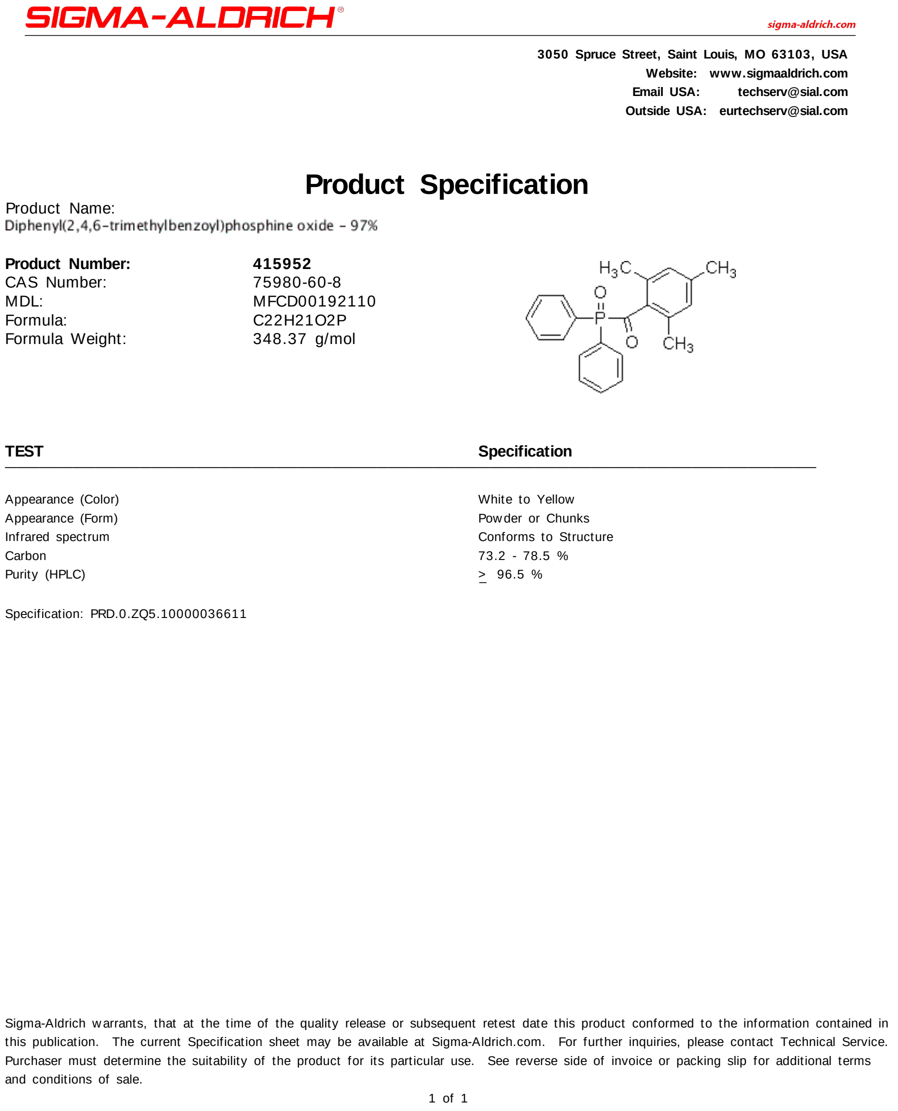
Specification: PRD.1.ZQ5.10000018208

Sigma-Aldrich warrants, that at the time of the quality release or subsequent retest date this product conformed to the information contained in this publication. The current Specification sheet may be available at Sigma-Aldrich.com. For further inquiries, please contact Technical Service. Purchaser must determine the suitability of the product for its particular use. See reverse side of invoice or packing slip for additional terms and conditions of sale.

1 of 1

Figure E.2: Methacrylate datasheet)

E.3 Photoinitatior used to produce the UV-Curable PAA-resin



SIGMA-ALDRICH®

sigma-aldrich.com

3050 Spruce Street, Saint Louis, MO 63103, USA
Website: www.sigmaaldrich.com
Email USA: techserv@sial.com
Outside USA: eurtechserv@sial.com

Product Specification

Product Name:
Diphenyl(2,4,6-trimethylbenzoyl)phosphine oxide - 97%

Product Number:	**415952**
CAS Number:	75980-60-8
MDL:	MFCD00192110
Formula:	C22H21O2P
Formula Weight:	348.37 g/mol

TEST	**Specification**
Appearance (Color)	White to Yellow
Appearance (Form)	Powder or Chunks
Infrared spectrum	Conforms to Structure
Carbon	73.2 - 78.5 %
Purity (HPLC)	≥ 96.5 %

Specification: PRD.0.ZQ5.10000036611

Sigma-Aldrich warrants, that at the time of the quality release or subsequent retest date this product conformed to the information contained in this publication. The current Specification sheet may be available at Sigma-Aldrich.com. For further inquiries, please contact Technical Service. Purchaser must determine the suitability of the product for its particular use. See reverse side of invoice or packing slip for additional terms and conditions of sale.

1 of 1

Figure E.3: Photo Initiator (TPO) datasheet

E.4 Production Procedure PAA

Production sequence for UV-curable PAA resin:

1. A three-neck round bottomed flask was equipped with a mechanical overhead stirrer, a N2 supply from a schlenk line and a septum.

2. 281.2 g PAA (Pyre ML 5019) in NMP was added to the flask, followed by 18.6 mL DMAEMA.

3.Subsequently, 0.89g TPO was dissolved by in 0.7mL NMP by shaking vigorously for 2 minutes and added to the flask using a syringe.

4.The mixture was stirred overnight under N2 atmosphere and the flask was wrapped in aluminum foil to block any UV light from reacting with the resin.

Note: The resin must be stored in a fridge to avoid thermally induced polymerization. The synthesis was performed with equipment and assistance from the Department of Chemical Engineering. Our contact was: Sulalit Bandyopadhyay

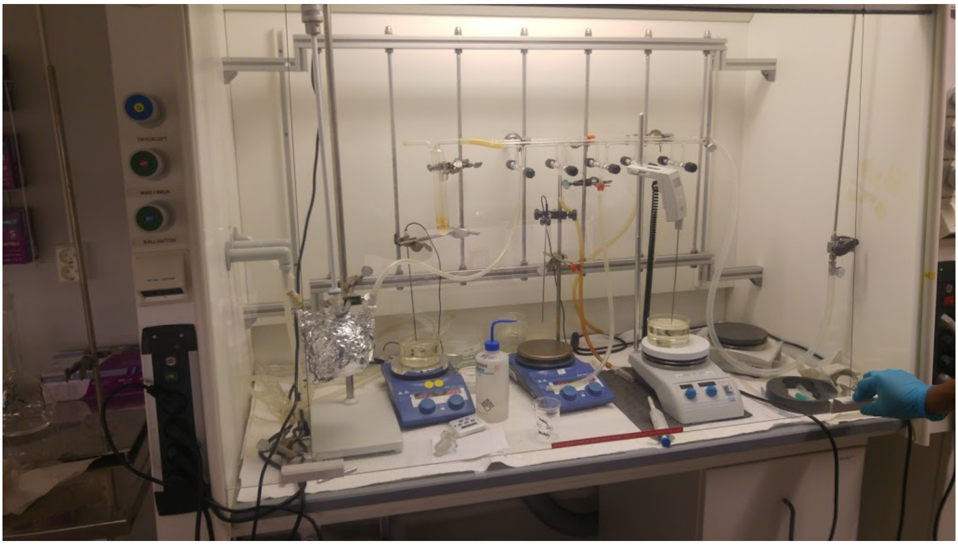


Figure E.4: Schlenk line setup used for synthesis

E.5 Synthesized PAA resin

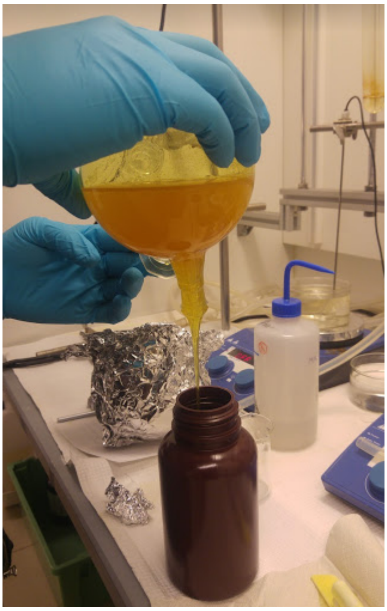


Figure E.5: UV-curable PAA-resin after synthesis

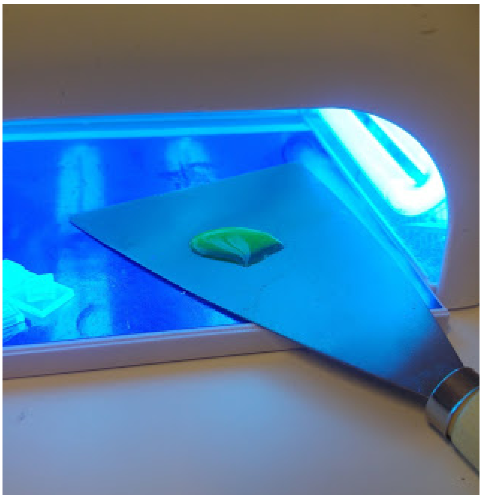


Figure E.6: UV-curable PAA resin during curing

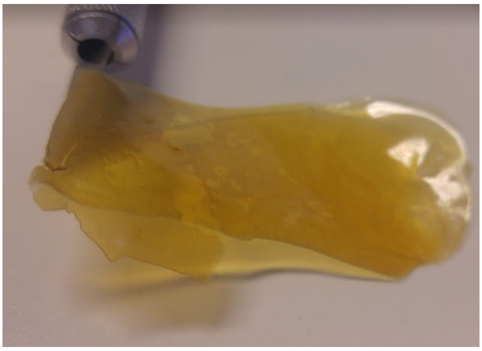


Figure E.7: PAA-resin after curing

E.6 Cubicure Thermoblast Specimens

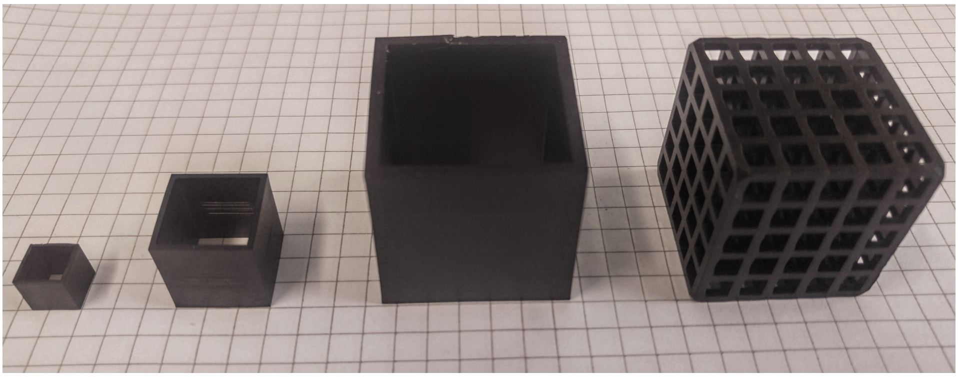


Figure E.8: Samples as delivered from CubiCure

E.7 Setup used for Imidization

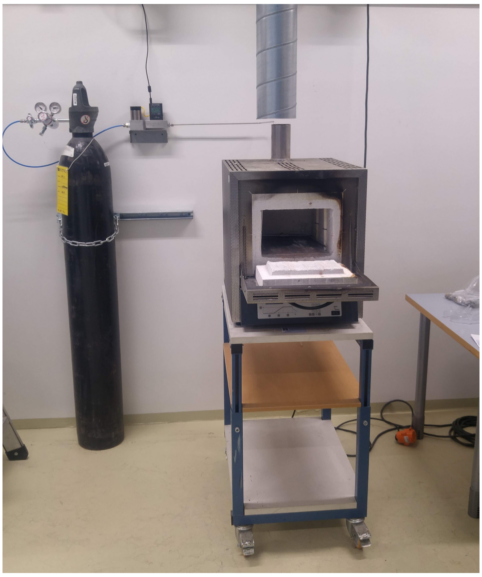
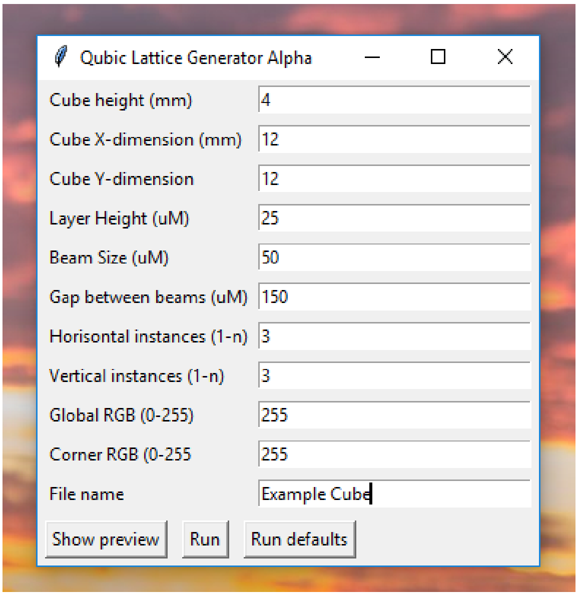


Figure E.9: The modified Lab Furnace used for imidization with attached Nitrogen and flow controller on the left.

Appendix F

Lattice Generation examples and source Code

F.1 Usage Example



Qubic Lattice Generator Alpha

Cube height (mm)	4
Cube X-dimension (mm)	12
Cube Y-dimension	12
Layer Height (uM)	25
Beam Size (uM)	50
Gap between beams (uM)	150
Horisontal instances (1-n)	3
Vertical instances (1-n)	3
Global RGB (0-255)	255
Corner RGB (0-255	255
File name	Example Cube

Figure F.1: The input form (GUI) for the Python Script containing the basic variables

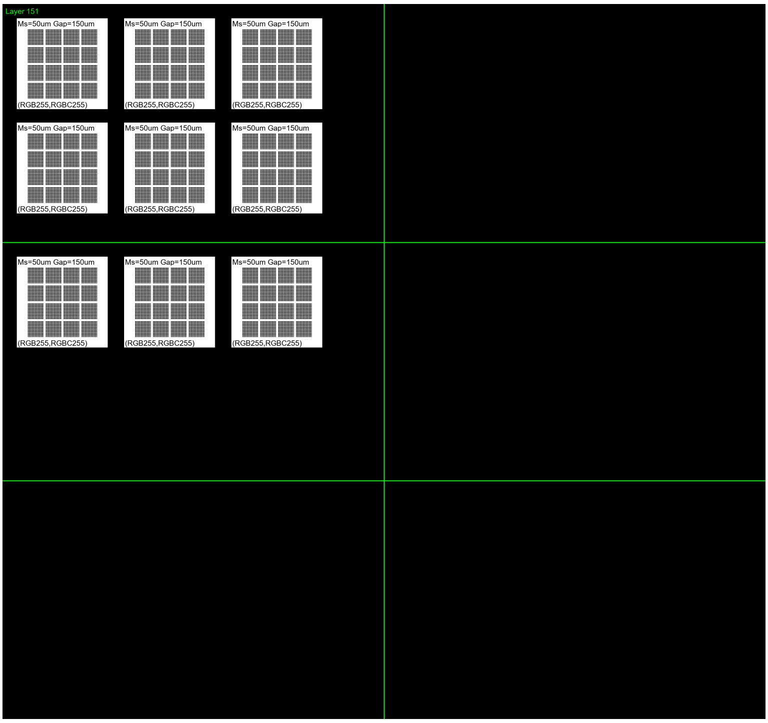
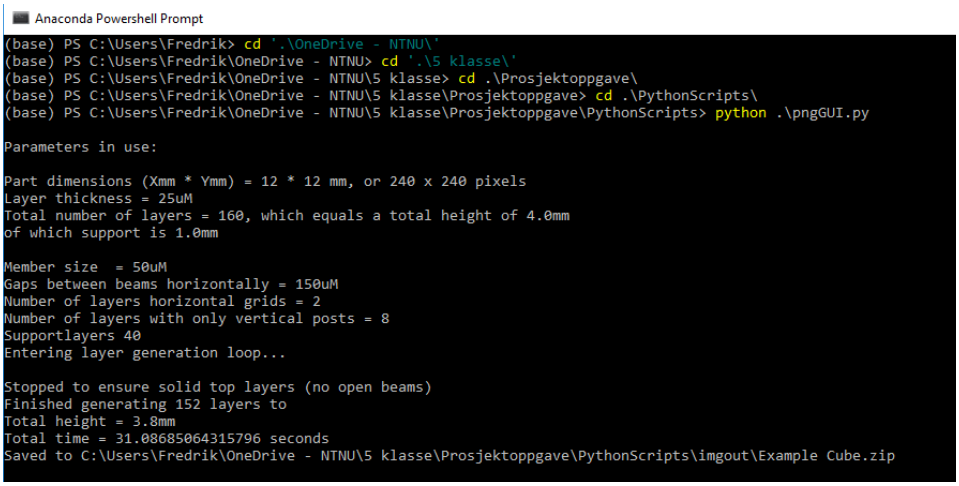


Figure F.2: Given the input in F.1, the program will produce structures that looks like this seen from above. The green lines are just to illustrate the 6 zones of the Solflex 650.



```
(base) PS C:\Users\Fredrik> cd '.\OneDrive - NTNU\'
(base) PS C:\Users\Fredrik\OneDrive - NTNU> cd '.\5 klasse\'
(base) PS C:\Users\Fredrik\OneDrive - NTNU\5 klasse> cd .\Prosjektoppgave\
(base) PS C:\Users\Fredrik\OneDrive - NTNU\5 klasse\Prosjektoppgave> cd .\PythonScripts\
(base) PS C:\Users\Fredrik\OneDrive - NTNU\5 klasse\Prosjektoppgave\PythonScripts> python .\pngGUI.py

Parameters in use:

Part dimensions (Xmm * Ymm) = 12 * 12 mm, or 240 x 240 pixels
Layer thickness = 25uM
Total number of layers = 160, which equals a total height of 4.0mm
of which support is 1.0mm

Member size  = 50uM
Gaps between beams horizontally = 150uM
Number of layers horizontal grids = 2
Number of layers with only vertical posts = 8
Supportlayers 40
Entering layer generation loop...

Stopped to ensure solid top layers (no open beams)
Finished generating 152 layers to
Total height = 3.8mm
Total time = 31.08685064315796 seconds
Saved to C:\Users\Fredrik\OneDrive - NTNU\5 klasse\Prosjektoppgave\PythonScripts\imgout\Example Cube.zip
```

Figure F.3: Command line output from running the input in F.1. The file is saved as a zip file. An unzipped version is outputted to a folder ”imgout” which is generated in same folder as the program.

F.2 Input GUI Source Code

```
#!/usr/bin/python3

from tkinter import *
from PIL import Image, ImageDraw, ImageFont
import pngtes as p
import drawLines as dl
import hashlib
import cProfile


fields = 'Cube height (mm)', 'Cube X-dimension (mm)', 'Cube Y-dimension', 'Layer Height (uM)','Beam Si
showimg = False

def preview(x):
    global showimg

    if (showimg == True):
        showimg = False
        print("Preview Off")
    else:
        showimg = True
        print("Preview On")

def run(entries):
    global showimg
    cbHeight = entries[0][1].get()
    cbXdim=entries[1][1].get()
    cbYdim=entries[2][1].get()
    layHeight =entries[3][1].get()
    mbSize = entries[4][1].get()
    spacing = entries[5][1].get()
    xn = entries[6][1].get()
    yn = entries[7][1].get()
    RGBval = entries[8][1].get()
    RGBcorner = entries[9][1].get()
    file = entries[10][1].get()

    #Start mainLoop with parameters read from entries
    p.mainLoop(showimg,int(cbHeight),int(cbXdim),int(cbYdim),int(layHeight), int(mbSize),int(xn),int(y

def fetch(entries):
   print("hei entries")
   for entry in entries:
      field = entry[0]
      text  = entry[1].get()
      print('%s: "%s"' % (field, text))


def defaults(entries):
    global showimg
    #cProfile is used for counting the time to run methods in the program
    cProfile.run('p.mainLoop(showimg,2,12,12,25,50,3,2,150,200,200,str("default"))')
    #p.mainLoop(showimg,2,12,12,25,50,3,2,150,200,200,str("default"))
```

```


def makeform(root, fields):
   entries = []

   for field in fields:
      row = Frame(root)
      lab = Label(row, width=20, text=field, anchor='w')
      ent = Entry(row)
      row.pack(side=TOP, fill=X, padx=5, pady=3)
      lab.pack(side=LEFT)
      ent.pack(side=RIGHT, expand=YES, fill=X)
      entries.append((field, ent))
   return entries


if __name__ == '__main__':

   root = Tk()
   root.title("Qubic Lattice Generator Alpha")
   ents = makeform(root, fields)
   root.bind('<Return>', (lambda event, e=ents: fetch(e)))
   variable = False

   b3 = Button(root, text='Show preview',
               command=(lambda t=ents: preview(t)))
   b3.pack(side=LEFT, padx=5, pady=5)
   b4 = Button(root, text='Run',
               command=(lambda t=ents: run(t)))
   b4.pack(side=LEFT, padx=5, pady=5)

   b5 = Button(root, text='Run defaults',
               command=(lambda t=ents: defaults(t)))
   b5.pack(side=LEFT, padx=5, pady=5)

   root.mainloop()
```

F.3 Main Loop Source Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 6 10:10:04 2018
@author: fc_ca
"""
from PIL import Image, ImageDraw, ImageFont
import os, time
import drawLines, shutil, supportGeneration, debugOptions #local files

from xml.etree.ElementTree import ElementTree
from xml.etree.ElementTree import Element
import xml.etree.ElementTree as etree

def mainLoop(previewOn,cubeHeight,xDim,yDim,layerheight,memberSize,cubesXdir,cubesYdir,spacingUM,RGBval

    start = time.time() #start stop watch

    #----User defined input----------------------------------------------------
    cubeZheiht = cubeHeight
    cubeXsize = int(xDim*1000/50) #convert size in pixels, where on pixel = 50uM
    cubeYsize = int(yDim*1000/50) #convert size in pixels, where on pixel = 50uM
    spacing = int(((spacingUM)+(memberSize))/50)

    #------Initial calculations----------------------------------------------------
    totalLayers = int(cubeZheiht*1000//layerheight) #Converts mm to a round number of layers based on
    solidLayer = int(memberSize/layerheight)
    postLayer =  2*spacing*(-1) #vertical spacing 2solidlayers(25um)*5spc = 10 post layers = 250uM z-o
    namespace = str(totalLayers) #used for generating leading zeroes in picture name
    cubeSize = max(cubeXsize,cubeYsize)# cubeXYsize*1000//50


    #--------Support options-----------------------------------
    supportLayers = (((postLayer*(-1)+solidLayer))%40+(postLayer*(-1)+solidLayer))*2#support layers nu
    platformOnlyLayers = supportLayers
    platformType = "frame" #choose solid or square (frame)
    frameThickness = 1.6 #mm

    #---------Info print at program strart-------------------------------------
    print("\nParameters in use:\n")
    print("Part dimensions (Xmm * Ymm) = "+str(xDim)+" * "+str(yDim)+" mm, or "+str(cubeXsize)+(" x ")
    print("Layer thickness = " +str(layerheight)+"uM \nTotal number of layers = " + str(totalLayers)+"
    print("of which support is "+str(platformOnlyLayers*layerheight/1000)+"mm")
    print("\nMember size  = " +str(memberSize)+"uM")
    print("Gaps between beams horizontally = "+str(spacing*50-memberSize)+"uM")
    print("Number of layers horizontal grids = " + str(solidLayer))
    print("Number of layers with only vertical posts = " +str(postLayer*-1))
    print("Supportlayers " + str(supportLayers))

    #--------Font options for labels on parts----------------------------------
    fontsize = 25 #40 = 1mm tall letters
    fntSmall=ImageFont.truetype("arial.ttf",fontsize)
    fntLarge = ImageFont.truetype("impact.ttf",fontsize*4)
    currentFont = fntSmall
    #Output request
    saveImageFiles = True #True = on, False = off |
    showImageFiles = previewOn #True = on, False = off |

    #---------Initial coordinates of cubes on buildplatform--------------------
    origXstart = 80 #Starting point in x-direction
    origYstart = 80 # starting point in y-direction
    xspacing = 120 #horizontal spacing between cubes
    yspacing = 110 #vertical spacing between cubes

    #----------XML-file-structure----------------------------------------------
    root=Element("root")
    tree=ElementTree(root)

    name=Element('name')
    root.append(name)
    name.text='neutral'

```

```
    data0=Element('data0')
    root.append(data0)
    data0.text='1'

    data1=Element('data1')
    root.append(data1)
    data1.text='1'

    data2=Element('data2')
    root.append(data2)
    data2.text='0'

    #------------Main-drawing-loop-------------------------------------------------
    sLayer = solidLayer #used for counting horizontal mesh layers and between layers with just vertica

    print("Entering layer generation loop...\n") # Nested for/while loops to draw mesh

    for layer in range (0, (totalLayers)):

        img = Image.new('RGB', (2560,2400), color = 'black')
        d = ImageDraw.Draw(img)

        xStart = origXstart; #Initialize position variables
        yStart = origYstart; #Initialize position variables

        xEnd = xStart + cubeXsize #Initialize position variables
        yEnd = yStart + cubeYsize #Initialize position variables

        debugOptions.drawZones(d,layer,currentFont) #Activate to show print zones and layer number

        for y in range(0, cubesYdir): #Vertical offsetting loop

            for x in range(0,cubesXdir): #draw x number of cubes horisontaly

                if layer>=platformOnlyLayers: #draw structures
                    supportGeneration.drawSupport(d,layer,supportLayers,platformType,frameThickness,xSt
                    drawLines.whiteLines(d,cubeSize, memberSize,RGBval,spacing,xStart, yStart, xEnd, y
                    drawLines.whiteLines(d,cubeSize, 300,255,int(3000/50),xStart, yStart, xEnd, yEnd,s

                else: #draw only support structure
                    supportGeneration.drawSupport(d,layer,supportLayers,platformType,frameThickness,xSt

                #---next---cube---movement------between x-loop iterations-----------
                xStart = xStart + cubeXsize+xspacing #traverse horisontaly
                xEnd = xEnd + cubeXsize+xspacing #traverse horsintaly

                if (1280 - (xStart%1280))< cubeSize+xspacing: #a check if object colides
                    xStart = 1280+origXstart #if yes, move to next zone
                    xEnd = xStart + cubeXsize

                # RGBval -= 5; #Activaty to gradually reduce RGB intensity pr cube
                #RGBcorner -=5 #Activaty to gradually reduce RGB intensity pr cube
                #-----------------y-movement--------------------------------------------

            yStart = yStart + cubeSize + yspacing #traverse build platform vertically
            yEnd  = yEnd + cubeSize + yspacing # same

            if      (800-(yStart%800)) < cubeSize+yspacing: # check if near a cell edge (every 800 pix

                    if(yStart < 800): #if colission in first sone, move to zone 2 (starts at 800 pixel
                        yStart = 800 + origYstart
                        yEnd  = yStart +  cubeSize

                    else: #if we colission in zone 2, move to zone 3 (starts at 1600 pixel)
                        yStart = 1680
                        yEnd  = yStart +  cubeSize

            xStart = origXstart #reset x-position after a vertical iteration
            xEnd = origXstart+cubeXsize

        #RGBval = 200;#Activaty to reset RGB after each layer
        #RGBcorner = 200; #Activaty to reset RGB after each layer

            #-------------count solid vs post layers-------------------------------
        if (sLayer-1) == postLayer:#(-spacing*solidLayer)+(solidLayer+1):   #print posts from 0,-1,-2,
            #print("restet solidlayer = {}, slayer{}".format(solidLayer,sLayer))

            sLayer =solidLayer #* multiGrid; #reset solid layer counter
        else: #continue countdown of solid layers
            sLayer -=1

        #---------------------show & save current png-------------------------------
        number = str(layer).zfill(len(namespace)) #generate number to add to name
        #r = range(supportLayers,supportLayers+solidLayer)
        if showImageFiles == True and (supportLayers+solidLayer-1) <= layer <= supportLayers+solidLaye
            img.show() #show first normal layer image
        if saveImageFiles == True:
            curdir= str(os.getcwd())
            #print(curdir+'\imgout\layer_{}.png'.format(number))
            #img.rotate(45)
            img.save(curdir+'\imgout\layer_{}.png'.format(number),compress_level=1) #save layer as imag

        #------------------------------XML-data-appended-pr.layer------------------------

        zv = round((layer+1)*layerheight/1000,3) #store current z-value (layerheight) in mm rounded to
        zvs = format(zv, '.3f') #store zvalue as a string with 3 digits behind comma allways: 0.100, 0

        xml_layer=Element('layer')
        root.append(xml_layer)

        index=Element('index')
        xml_layer.append(index)
        index.text=str(layer)

        zvalue=Element('zvalue')
        xml_layer.append(zvalue)
        zvalue.text=str(zvs)

        filename=Element('filename')
        xml_layer.append(filename)
        filename.text='layer_{}.png'.format(number)

        #print(etree.tostring(root)) #printing tree structure to console
        tree.write(curdir+"\imgout\index.xml")

        if totalLayers - layer < ((postLayer*-1)+solidLayer) and sLayer == 0: #ensure top side of grid
          print("Stopped to ensure solid top layers (no open beams)")
          break;
        #---------------------------END OF LAYER FOR LOOP--------------------------

    shutil.make_archive(zipfileName, 'zip', curdir+'\imgout')

    end = time.time()
    print("Finished generating {} layers to ".format(layer+1) +"\nTotal height = " + str((layer+1)*laye
    print("Total time = " +str(end-start)+" seconds")
    print("Saved to "+curdir+"\imgout\\"+ zipfileName+".zip \n") #end of program
```

F.4 Drawing Module Source Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 12 16:00:30 2018
@author: Fredrik
"""
import pngtes as p

def whiteLines(d,cubeSize, memberSize, RGBval, spacing, xStart, yStart, xEnd, yEnd,sLayer,spLid, platf


    i = 0; #pixelcounter
    mbSize=int(memberSize/50)#50uM = 1 pixel, 100 = 2 etcssol

    #--------------------Drawing of horizontal features (grids)------------------------------------

    if sLayer> 0 or spLid>0: #if a horizontal layer (cubic grid)
        while i <= cubeSize:

                if yStart+i <= yEnd: #IF yStart of horizontal line is withing y boundary, draw line
                  d.line((xStart, yStart+i, xEnd,yStart+i), width=mbSize, fill=(RGBval,RGBval,RGBval))
                if xStart+i <= xEnd:  #IF xStart of vertical line is withing x boundary, draw line
                  d.line((xStart+i, yStart, xStart+i, yEnd), width =mbSize, fill=(RGBval,RGBval,RGBval

                i+=spacing#*mbSize #jump i pixels between each line

        cornerCorection(d,RGBval,RGBcorner,xStart, yStart,xEnd,yEnd,spLid,spacing,mbSize,sLayer) #cont

        i = 0; #reset pixel counter before wiping out with black on post layers

    #-------------------------------------Drawing of vertical features--------------------------------
    elif sLayer < 1:# then only draw vertical beams
        RGBvalV=255;
        while i<= cubeSize:
            if i%(spacing) == 0:
                for instances in range (0,cubeSize+spacing,spacing): #draw dots = nInstances with spac
                    #if(spLid ==2 or spLid == 1):#the thick supporting posts
                    if(mbSize>1):
                        d.rectangle((xStart-mbSize/2+instances,yStart-mbSize/2+i, xStart+mbSize/2+insta
                        d.point((xStart+instances,yStart+i),fill=(255,255,255))
                    else: #the normal small posts
                        d.point((xStart-mbSize*spacing/2+instances,yStart-mbSize*spacing/2+i),fill=(RG
                #d.point((yStart,i),fill=(RGBval,RGBval,0))
            #------------------------------------------------------------
            i+=1*spacing
        #print(i)
        i = 0

    if xEnd-xStart<yEnd-yStart and (spLid==1 or spLid ==2):
      #print("trim edges")
      trimVerticalEdges(d,xStart,xEnd,yStart,yEnd,frameThickness, sLayer,spLid)

    elif yEnd-yStart < xEnd-xStart and (spLid==1 or spLid ==2):
      trimHorizEdges(d,xStart,xEnd,yStart,yEnd,frameThickness, sLayer,spLid)

#Optional reduction of RGB intensity in the corner of each unit cell
def cornerCorection(d,RGBval,RGBcorner,xStart, yStart, xEnd,yEnd,spLid,spacing,mbSize,sLayer):
    if spLid == 0: #not for post olny layers and not for support grids
      for y in range (yStart, yEnd,int(spacing*mbSize)):
        for x in range (xStart, xEnd,int(spacing*mbSize)):
          if sLayer>0:
              d.point((x,y),fill=(RGBcorner,RGBcorner,RGBcorner))
```

F.5 Support Generation Module

```
        # -*- coding: utf-8 -*-
"""
Created on Fri Jan 18 10:39:54 2019

@author: Fredrik
"""
import drawLines
import pngtes as png


def drawSupport(d,layer,supportLayers,platformType,frameThickness,xStart,yStart,xEnd,yEnd,RGBval,RGBco



    if layer < supportLayers and platformType == "frame":
        if layer < 5: #acount for burn in effect due to overexposure
            #d.rectangle(((xStart,yStart),(xEnd,yEnd)), fill="white")
            ft = frameThickness*1000/50
            d.rectangle(((xStart-ft,yStart-ft),(xEnd+ft,yEnd+ft)), fill=(255,255,255))
            d.rectangle(((xStart,yStart),(xEnd,yEnd)), fill="black")
            #drawLines.whiteLines(d,cubeSize, 200,255,int(3000/50),xStart, yStart, xEnd, yEnd,sLayer,2,
            #drawLines.whiteLines(d,cubeSize, 500,255,spacing*2.5,xStart, yStart, xEnd, yEnd,sLayer,1,

        else:
            ft = frameThickness*1000/50
            d.rectangle(((xStart-ft,yStart-ft),(xEnd+ft,yEnd+ft)), fill=(255,255,255))
            d.rectangle(((xStart,yStart),(xEnd,yEnd)), fill="black")
            drawLines.whiteLines(d,cubeSize, 300,255,int(3000/50),xStart, yStart, xEnd, yEnd,sLayer,2,p
        #'''else:
        #    #drawLines.whiteLines(d,cubeSize, 300,255,spacing*2,xStart, yStart, xEnd, yEnd,sLayer,3,p
         #   drawLines.whiteLines(d,cubeSize, 500,255,spacing*2,xStart, yStart, xEnd, yEnd,sLayer,3,pl
       # '''
    #else:

    elif layer < supportLayers and platformType == "solid": #if support platform is activated, a solid
                       d.rectangle(((xStart,yStart),(xEnd,yEnd)), fill="white")
                       #print(layer)
                       ft = frameThickness*1000/50

                       if layer <= platformOnlyLayers-10:
                           d.text((xStart+ft*0.8,yStart+ft*0.8), "{}".format(RGBval), font=curFont,fil

    elif platformType == "frame":    #creates a frame/border around the mesh
                        ft = frameThickness*1000/50
                        d.rectangle(((xStart-ft,yStart-ft),(xEnd+ft,yEnd+ft)), fill=(255,255,255))
                        d.rectangle(((xStart,yStart),(xEnd,yEnd)), fill="black")

                        #Write text on support, optional
                        d.text((xStart-ft*0.9,yStart-ft*0.86), "Ms={}um Gap={}um".format(memberSize,spa
                       #d.text((xStart-ft*0.9,yEnd+ft*0.15),  "({}/{}/{})".format(RGBval,RGBval-25,RGB
                        d.text((xStart-ft*0.9,yEnd+ft*0.15),  "(RGB{},RGBC{})".format(RGBval,RGBcorner
```

Fredrik Cappelen

NTNU
Norwegian University of
Science and Technology