Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Ciudad de México

School of Engineering and Sciences

Improving the path planning and the printing time for an optimized infill of 3D objects by reducing sharp angles and having a continuous path

A dissertation presented by

Cesar David Betancourt Adame

A thesis submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master

In

Engineering Science

Tlalpan, Ciudad de México, 14th June, 2021

Abstract

Improving the path planning and the printing time for an optimized infill of 3D objects by reducing sharp angles and having a continuous path

by Cesar David Betancourt Adame

Purpose – Three-dimensional printing is a technology that can provide one of the most efficient methods for product design, prototyping and production being positive cost-effective due to the efficiency of the design, the customization of the objects and variety of materials. However, contemporary computer-aided design (CAD) and computer-aided manufacturing (CAM) systems use different infill patterns that have the same similarity, they usually contain sharp angles and non-continuous trajectories. A new algorithm is used to create an infill that minimizes the sharp angles in the infills and having a continuous path in order to generate the necessary tool-path information. In this thesis, we propose a new algorithm to create a new type of infill that reduces the amount of time and material used in each layer of an object printed with the Fused Deposition Modeling (FDM) technology.

Design/methodology/approach – In the proposed algorithm, a grid is generated in a layer with the specific shape that corresponds to a 3D object, it consists of a percentage according to the one chosen by the user, being 20% the most used in this technology. The infill is created with a continuous path and minimizing the sharp angles in the whole layer, the optimization is accomplished by using simulated annealing.

Findings – By creating and running different experiments in various models of FDM 3D printers, we proved the base of our algorithm, that by having sharp angles in the infill, the total printing time is increased due to the positive and negative acceleration of the printing head, altogether with the non-continuous path that increases time when stopping extruding material and staring again. Applying the proposed algorithm, this information can be used to create a new path for an infill giving as result the reduction of time and material in each layer of a 3D printed object.

Research limitations/implications – The proposed methodology can be applied to create a new infill for objects that will be printed with the FDM technology. However, algorithm works for optimizing one layer at a time. In the future, we would like to investigate the results between fill patterns of consecutive layers, where consecutive layers can't be identical to provide good resiliency to the object.

Originality/value – The proposed algorithm is a novel development for creating a new type of infill that reduces the amount of time and material employed in the fabrication of 3D objects using the Fused Deposition Modeling (FDM) technology.

Acknowledgements

I would like to express my deepest gratitude to all those who have been side by side with me through all this process of learning. People and others to than:
- My family, friends and my girlfriend
- Tecnológico de Monterrey
- Dra. Julieta Noguéz
- Dr. Bedrich Benes
- Dr. Sergio Ruiz Loza
- CONACyT
- Purdue University
- HPCG laboratory at Purdue University
- The committee
- The department of Science at ITESM CCM

Contents

List of Figures

List of Tables

Thanks for all your unconditional confidence, support, patience, and encouragement. You were my main motivation for pushing through this work. People to thank: My Family, my friends, my Advisor Dra. Julieta Noguez, Dr. Bedrich Benes, Tecnológico de Monterrey and Purdue University.

Chapter 1

Introduction

1.1 Justification

Three-dimensional printing is a technology that has become more available in recent years, and it could provide one of the most efficient methods for product design, prototyping and production (manufacturing) in a manner that it is cost-effective due to the efficiency of the design, the customization and the materials that can be used in the process.
The additive manufacturing (AM) it is also known as three-dimensional printing [2], and it has the growing necessity of new libraries and algorithms to make the process of modeling and printing geometrical objects in a faster and more optimal way considering parameters like the precision, accuracy of the model, the outer and inner quality of the model.
The amount of time and material to print an object are the two important factors in the printing process [1] of a model that translates into the cost of a prototype in additive manufacturing, being the time factor very high in most of the processes in AM. Therefore, there is the growing necessity of printing in a faster way in additive manufacturing to reduce the cost of a printed model in terms of material and time.
The term infill in additive manufacturing (also known as 3D printing) refers to the interior structure of an object. It often has a regular pattern, which can be selected by the user in the slicing software, along with a specific volume percentage. An essential problem we focused on is to resolve the way an infill of an object is designed and printed due to its significantly influence in the printing process time, amount of material used and in the physical properties of the object.

1.2 Motivation

The creation of new algorithms that allow additive manufacturing to be a faster and more optimal process, focusing in the optimization of the infill of a 3D printed object, is the motivation of this project. With these optimizations we can improve the time that an object takes to be printed and consequently in some cases the usage of less material, while maintaining the physical properties of the objects. It is also important to have control of the infill generation and its translation to gcode through the entire process of the object. This will help us to optimize the printing process of an object by creating a new continuous path that the printer machine will use to print the object.

1.3 Problem

Additive manufacturing (AM) [1] is the process of having a model generated by a Computer Aided Design software and can be fabricated in something that is tangible without the need of process planning. Essentially the way that AM work is that each model that is meant to be printed is made by adding material in layers that are calculated in a slicing software. Each layer is a cross-section derived from the original model having a finite thickness in height and width. An example of this can be seen in Figure 1.1 where different factors must be considered as how the layers are created, the type of material used in the process, how the layer are bonded together, the type of machine and its scale used in the process, and the post-process required for a printed model, all of this lead to an overall cost of the machine and the process (material and time).

The main types of processes that use different techniques for AM are Selective Laser Sintering (SLS), Stereolithography (SLA), Direct Metal Laser Sintering (DMLS), Paper Lamination Technology (PLT), Thermojet, Shaped deposition Manufacturing (SDM), and the one we are focused in this project that is Fused Deposition Modeling (FDM) that extrudes material trough a heated nozzle that has a defined diameter where the material comes out in a semi-solid state at a certain speed rate and solidifies with the material that has been extruded and creating a solid object of a model as a result. All of these techniques will be further explained in chapter two of this thesis.

One of the main problems in the process of additive manufacturing is the amount of time and material it takes to print an object. This could be due to the path planning of some parts of the process like the contours, the walls, the level of detail or the infill. The infill of an object refers of the interior structure that gives support to all the object while printing the object. This infills often have regular patterns like the ones we can see in Figure 1.1, that are often selected by the user in existing slicing software. The users can choose between different patterns and the percentage of the infill they want. The process of printing the infill influence in the

printing time and physical properties of the object, like the resilience. This gives purpose to the project, that is to create an infill that is optimal with the implementation of a new algorithm to generate a infill with less sharp turns and having a continuous path; this means to create an infill that fits an object, it uses as less material if possible and keep the physical properties of the object, giving as a result a faster printing time of an object.

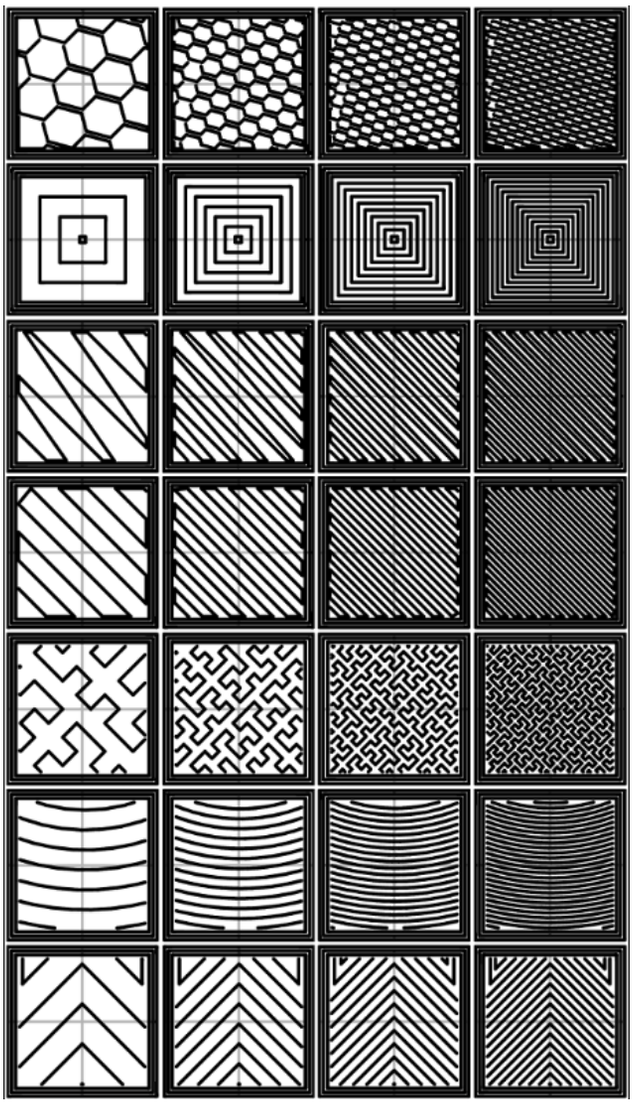

FIGURE 1.1: Example of different patterns with different percentage of covering area. [1]

1.4 General Objectives

The purpose of this investigation is to create and define an architecture, use cases, components, classes, activities and a new infill proposal that optimize the infill of a geometric model that is meant to be printed in a 3D printer that uses the printing technique of Fused Deposition Modeling (FDM).

1.5 Specific Objectives

The general objective will be achieved with the creation of an specific algorithm that will create an infill for a certain layer of a model and optimize this continuous path (infill) in order to have a new infill that avoids the usage of sharp turns and that continuous path prevails; also another algorithm that will convert the points we have in the outer walls and the infill generated into gcode instructions that the printer can understand. As a result, we will get a .gcode file that a printer can understand and will use to get the instructions of printing the model. In Table 1.1 we can see a plan that details the different sections of the project, having the theoretical framework, state of art, solution proposal, implementation, experimental design divided in two (angles experiment and infill generation), results, discussion of results and conclusions.
The experimental section of the project (Chapter 6) can be divided in four main sections to achieve its completion:
1. The definition of the models we want to print. These models should be in an .stl file, so that the model can be sliced in layers by a third party software. The result of the slices will be used to generate the proposed infill.
2. The creation of an algorithm to generate an new type of infill for an object that will be 3D printed. This will be done in python where we are going to manipulate the slices and create the infill for each slice of the object with the implementation of a novel algorithm that will be discussed further in this paper.
4. As we create the infill of a specified model, we will optimize the infill generated with a PSO heuristic (Particle Swarm Optimization). We use this heuristic because it allows us to find a new path after a defined number of iterations, that fulfills three objectives: it is a continuous path, it avoids sharp turns as much as possible, and it is faster than a previous iteration. After having the infill generated, we create the gcode necessary, using python, to print the model in the 3D printer. The gcode are a set of instructions that contain all the information of the model in coordinates that the machine understands, so it knows where to move and the amount of material it must extrude in each section of the layer in order to print the model correctly.

1.6 Hypothesis

The implementation of using the principle of having a continuous path and avoiding sharp turn to optimize the infill of a model that will be printed will improve the printing process of a model by reducing the printing time and in some cases the material used in the process. This will be done by running the experiments and measuring the printing time of each experiment to differentiate the patterns that are faster to print. This patterns that come out of the result of the experiments would be used to resolve an area coverage problem to fill the inside of a printed

object (infill) with a continuous pattern in an optimal way. This optimization will be compared to the currently infill patterns used in the market to demonstrate an improvement in the printing time and usage of the process. We will use a t-test to get how significant the differences between the patterns are, this will help us to validate if our tests are correct according to the following d. The greater the t-score, the more difference there is between patterns, the smaller the t-score, the more similarity there is between the patterns. In other words it will let us know if those differences could have happened by chance; and to prove this we will use a p-value of 5% or 0.05.

1.7 Initial Proposal of the solution

The creation of a novel algorithm that allow us to optimize the infill of a model using the principle of having a continuous path and avoiding sharp turns as much as possible, is our principal method for the generation and optimization with the validation of getting a faster printing process, by minimizing the acceleration and deceleration of the printing head and turning on and off the extrusion of material.

Section	Description
Theoretical framework	Explanation of all the technical details needed to read the entire paper.
State of Art	We explain in detail other works (and their comparison) that are related to this work.
Solution Proposal	We propose the solution to the problem described.
Implementation	In this section we can see all the architecture of the solution proposal
Experimental Design	The design of the experiments to get the percentage of time that sharp angles cause in the printing. The generation and design of experiments that will be use to test the infill generation.
Results and Discussion of Results	Here we show and discuss both results of the experiments.
Conclusions	We talk about the obtained results of the purpose of the project, and the future work that can be done.

Table 1.1: Details of the different chapters we can read through this thesis.

Chapter 2

Theoretical Framework

2.1 Additive Manufacturing

The process of creating rapidly a system, an object or a part of it can be referred as Rapid Prototyping or in most recent years in the industry as Additive Manufacturing (AM) where the output is a tangible object with specific physical properties as a result from a model [1]. The basic principle of this process is to take a model generated in a Computer Aided Design software (CAD) [1], get an .stl file out of the model and convert it into a set of instructions that a machine, in this case a 3D printer, understands and prints the model. This process is based on layers that are derived from the model; each layer has a thickness in height and width. This can change due to the material and technique used in the process, having an influence in the accuracy, material, physical and mechanical properties, the post-processing required and how much material and time it takes to complete the process.
The additive manufacturing process has a generic number of steps to print an object out of a model, these steps helps us to understand more the process of AM [1]:

1. As we have stated, the first thing is to create a model in some CAD software and get a file that contains the model.
2. The file must be processed to generate all the slices, contours, shells, supports and infills (the main objective of optimization in this work). We get a file with a set of instructions that the machine can understand, this process of slicing will be discussed in detail further in this document in Chapter 3.
3. The machine must be prepared prior to the building process with setting like the temperature, material constraints, layer thickness, timings, etc., this will vary according to the printing technique used.
4. As we have the set of instructions and the printer ready the building of the object begins, because it is a mainly automated process it can be done without supervision. It only requires

monitoring to make sure there are no errors during the run time mostly regarding material issues like running out of material or stuck material.
5. As the building process has been completed the parts must be removed from the machine.
6. In some occasions the objects could require some post-processing done by the user in order to complete the process, like gluing some parts together, removing supports or giving some priming and painting to the object to get a better surface texture and finish. We have to consider that all of this post-processing takes time an effort and should also be considered as part of the am process.

2.2 3D Printing

We can talk about 3D printing also as referring to additive manufacturing. In this world of creating objects out of a model to create fast prototypes or objects that users can use for educational purposes, research, industry, medicine or entertainment there are different techniques used and some are more commercial than the others. The development of the process of printing objects can be divided into direct part printing and binder printing technologies [3]. Direct printing refers to the printing process where all the material that conforms the object is dispensed from a print head that pours the material, and the binder process is where the binder or another material is printed into a powder bed. Each process has advantages and disadvantages, both processes have advantages over other additive manufacturing techniques as lower costs (particularly compared to laser machines), high printing speed, scalability, printing in different colors and usage of various materials. Also, the building materials of the machine are not as expensive as other machines, they can be built form standard components as drives, stages, print head, nozzles that also provide the capability of having different materials to use, compatibility and the resolution of a printed object.
The techniques used in AM can be classified by the way they print and use different materials to create an object, and how many channels they need to create the object can be taken into consideration [1] as we can see in Figure 2.1.

The most used techniques in AM are the printing techniques that use one channel to create the object, these techniques are:

• Fused Deposition Modeling (FDM)
It is the most common extrusion-based technology used in additive manufacturing. It uses a heated chamber to liquefy a polymer and with a built-in mechanism that pushes this polymer filament into a nozzle to extrude the material into a printing bed and thus generating each layer of the model and creating a 3D printed object [1]. This is the technique used in this work and will be explained in detail in section 2.3.

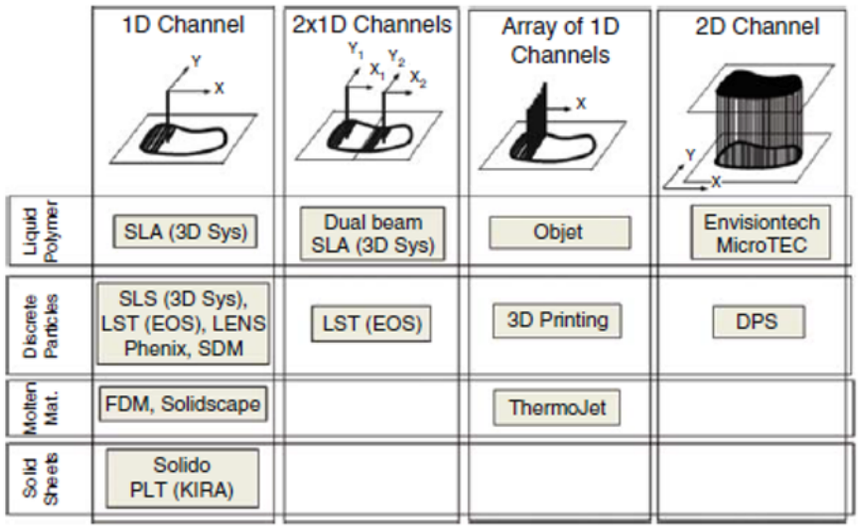

FIGURE 2.1: Division of 3D printing techniques of 3D printing by process and number of channels [1]

• Sintering (SLS)

This technique is mostly used in applications like prototyping, architecture models, hardware, sculptures, etc. The sintering process uses the slicing technique to fabricate slices (usually 0.1 mm thick) and print them one on top of another by solidifying a liquid photopolymer resin using a laser. The laser fuses each layer of powder that is preheated just before the melting point, it is spread across the printing area by a powder leveling roller that adds more powder material from some preheated feed cartridges [1]. The printing area can also be preheated to minimize the power required by the laser to fuse the powder and prevent the object to expand or contract creating curling in the object. All the process is done inside a closed chamber filled with nitrogen gas to prevent the powder from oxidation. An example of this machine can be seen in Figure 2.2.

• Stereolithography (SLA)

This technology of additive manufacturing is done by the solidification of a liquid polymer that is photosensitive to a light source [4], in a similar way that is done in the previous technology of SLS.

• Direct Metal Laser Sintering (DMLS)

This technique is focused in creating rapid prototypes with the difference that these objects are based on metal materials. It prints the object by solidifying a metal powder with a layer by layer technique [5] as it can be seen in Figure 2.3. This provides the possibility of creating complex shaped parts and with better physical properties that can't be achieved in other techniques.

• Paper Lamination Technology (PLT)

It is a technology of additive manufacturing based on a two-step process focused in printing big three-dimensional objects as tools like molds, with a high accuracy and stability. This process

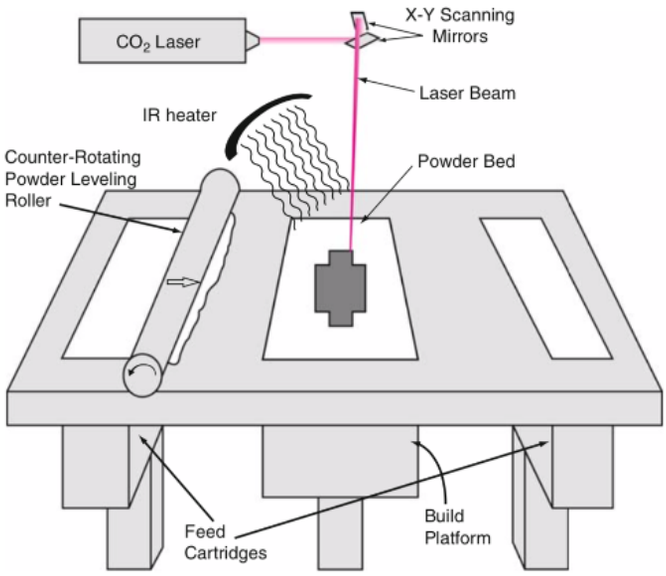

FIGURE 2.2: Example of sintering (SLS) machine and its process [1].

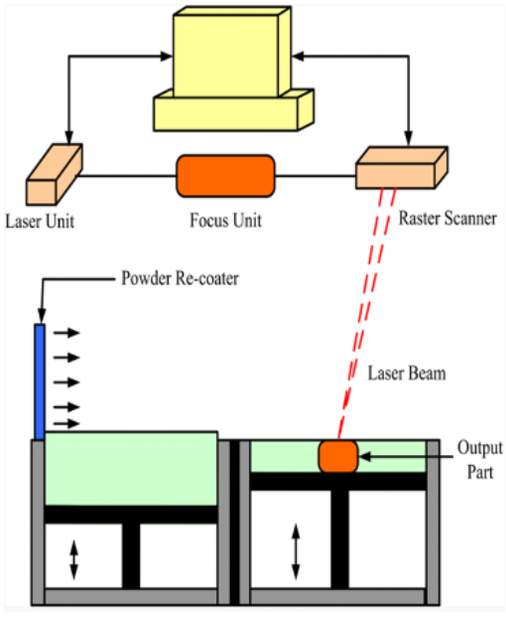

FIGURE 2.3: Example of direct metal laser sintering (DMLS) [1].

is done with a laser system that stacks the metal foils with each other in a layer by layer process and the use of a laser beam for fixing and generate the contours of the object combined with the second part of the process to enhance the mechanical properties [6].

2.3 Fused Deposition Modeling (FDM)

This technique is the one used in the printing process that is targeted to be optimized. Here we will explain how this process works, its principles and some of the commercial machines used in

the process od FDM including the machine used this this project that are the Ultimaker 2+™ and Ultimaker 3™ from the company Ultimaker™, and Ender 3 Pro from Creality™.

In this technique we follow the principle of having an model created in a CAD software and saved as an .stl file, after this it is processed by an slicing software that will provide the machine the necessary instructions to print the model by extruding material and moving across the three different axis int the printer (x, y, and z).

The printing process starts with the selection of a material that it is admitted by the printer in use due to every material has different properties like the melting point. The material comes out through the printing head that is composed of different parts, mainly where the material is contained in a reservoir and a nozzle where a certain pressure is applied resulting in the extrusion of the material. The extrusion will remain constant as the pressure remains constant and the diameter of the extruded material also depends on the diameter of the nozzle and for it to be constant as well. The material that is extruded must remain in a semi-solid state when it comes out of the nozzle, this is achieved by heating the nozzle to a determined temperature (this will vary according to the material in use) and maintaining the temperature through all the process. This molten material must be in this state so it can flow out through the nozzle and bond which each previous layer deposited in the printing bed, once a layer has been completed the machine will move the bed downwards so the next layer can be printed and bond to each other before solidifying.The key parts to this process are:

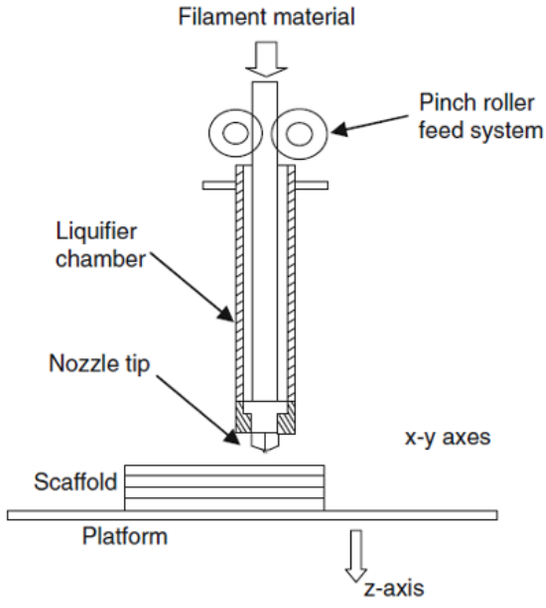

FIGURE 2.4: Example of the mechanism and parts of an FDM printing process [1].

• Loading of material

A chamber is preloaded with the solid material and can continuously supply material into the

chamber as needed by the process, meaning that a continuous filament can be pushed into the chamber generating an input pressure into the nozzle of the printing head.

• Liquification of the material
This principle works considering the melting point of the material that is being pushed into the nozzle by the mechanism that load the material. The material is liquified in a chamber that is heated to a set temperature that will melt the material and with the feeding mechanism it will push material through the nozzle. An example of this mechanism can be seen in Figure 2.4.

• Extrusion of the material
The amount, shape and size of the extrusion is determined by the feeding rate of material applied by the mechanism of the printer and the nozzle diameter and form. The nozzle diameter defines the minimum feature size that can be created by the printer, as well as the printing time, meaning that with a bigger nozzle leads to a faster printing process but less level of quality and level of detail. This also works all the way around, with a smaller nozzle smaller objects can be created and with a much better level of detail but with a longer printing time. The factors to consider in the extrusion of the material are:
o Feedrate
o Nozzle size
o Filament Diameter
o Filament properties like cooling, adhesion, melting temperature.
Two important factors to considerate are the flow rate of the material and the extrusion width.

• Solidification of the material
When the material is extruded it generates layers of materials that the first one is bonded to the printing bed and the consequent layer are bonded to each other. Once the material is extruded it should remain in the same shape and size in the process of being in liquified state to a solid state, this varies according to the cooling properties of the material used and can be minimized by controlling the environment temperature of the printing chamber or the printing bed, depending the capabilities of the printer.

• Moving the printing head and bed in a predefined path
The machine uses a printing head that represent the X and Y axis for the printing process, allowing to move in a 2D space thus creating the horizontal plane for each layer of the model. For the vertical axis it uses a printing bed that moves in the Z axis as we can see in Figure 2.5 allowing for the creation of each layer and giving the according height to each one of them. In both planes (horizontally and vertically) the extrusion rate and moving of the printing head must be considered to ensure a smooth and consistent extrusion and deposition of the material in each layer.

• Usage of supports structures

Due to the nature of how this process works, this process needs support structures to maintain the features of the object in in certain places like disconnected parts, or overhang parts of the model that have nothing below as can be seen in Figure 2.5 that need supports in order to for being built or infills that are other type of structures to give physical properties and strength to the object, being the last one the main purpose of this thesis. This support structures are no part of the original model and must be removed. There are some optimizations and techniques that are used in support structures and infills that will be discussed in chapter 3.

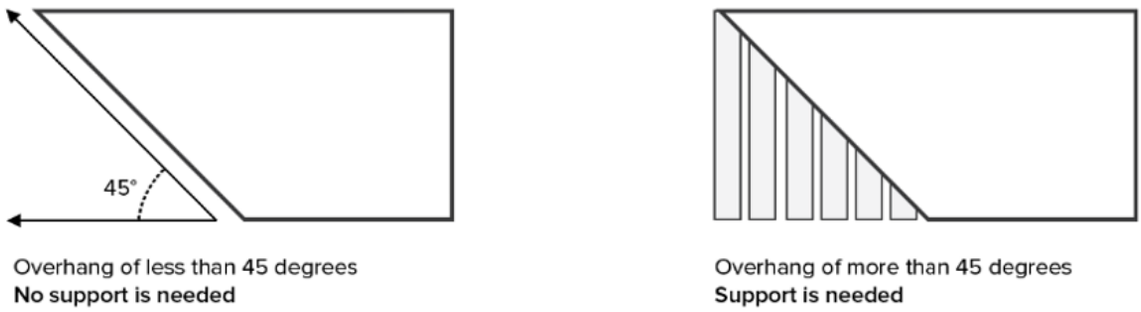

FIGURE 2.5: Example of a printed model that need structures in order to be printed [1].

2.3.1 Printers used in FDM

There are two main types in FDM 3D printers, industrial and desktop printers [7]. The industrial printers can produce objects with a higher accuracy due to the closer control of the processing parameters during printing, and other characteristics as a heated chamber that minimize the effects of the cooling of the materials, manage higher temperatures and support of dual extrusion, one for the object material that is resistant and the other for support structures. The desktop printers are evolving in a rapid way including most of the features that previously only industrial printers had like calibration algorithms, heated chambers, and dual extrusions. We can see in Table 2.1 a summary of the main differences between industrial and desktop 3D printers.

In the desktop FDM 3D printers we have some examples of various brands like Ultimaker^{TM}, Luzbot^{TM}, GoProto^{TM}, Prusa^{TM}, Creality^{TM} etc. Each one of this can have various models that provide better printing accuracy, finishes, support for different materials, and it is highly related to the price of the printer. The 3D printing machine in this work is the Ultimaker 2+^{TM} by Ultimaker^{TM}.

Comparison between Desktop FDM and Industrial FDM		
Properties	Desktop FDM	Industrial FDM
Accuracy	.15% (lower limit 0.2 mm)	1% (lower limit 1 mm)
Layer thickness	0.18 – 0.5 mm	0.1 – 0.25 mm
Minimum wall thickness	1 mm	0.8 – 1 mm
Common materials	ABS, PC	PLA, ABS
Support materials	Water-soluble	Same as printed part if single channel
Production capabilities	Low/medium	Low
Machine cost	+ $50,000 USD	$500 - $5000 USD

TABLE 2.1: Comparative between Desktop FDM and Industrial FDM

2.3.2 Materials used in FDM

In additive manufacturing there exist four main categories of materials that are used in the process such as polymers, metals, ceramics and organics. The available materials for FDM are Acrylonitrile butadiene styrene (ABS), Polylactide acid (PLA), Polycarbonate (PC), Polypropylene (PP), Nylon, elastomers, wax and ceramics [7]. The FDM process can print parts from the most used materials in Polymers that are ABS y PLA. These materials are commercialized as filaments with a predefined filament width and properties that must be considered in the extrusion process as well in the purpose of the model and its design. In Figure 2.6 we can see the materials used in additive manufacturing [8].

2.3.3 Advantages and disadvantages in FDM

Although this technique is very successful [1] and meet the demands of many industrial and commercial users due to the low cost of the machines and printing process compared to other techniques, it still has limitations as the operating principle, materials, geometry path, post production, accuracy and the most important in this technique the production speed or printing time [8]. In Table 2.2 we can see a comparison of the main advantages and disadvantages of the FDM process.

2.4 Gcode

Gcode is the name for a control language of most commercial 3D printers*. There are different types of gcode “flavors” that a specific 3D printer can understand like: Reprap, Marlin, and

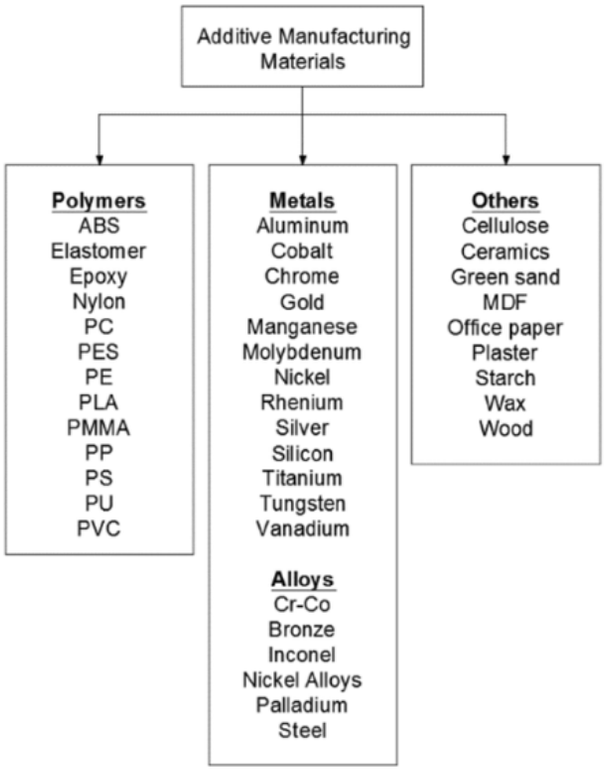

FIGURE 2.6: Materials used in Additive Manufacturing (AM) [8].

Advantages and Disadvantages in FDM	
Advantages	Disadvantages
Low cost of printing machine, materials, and maintenance	Low-speed printing
Easy to operate, compact design, the possibility of office/house/educational usage	Limited level of detail (resolution, finish, accuracy)
Low temperature operation	Requirement of support structures
Possibility of using different materials	Removal of support structures (post-processing)
Print the same object with different colors	The limited size of the printed object
Most used in the market	Calculate the printing time
	Calculate the extrusion of the material (over and underestimates)
	Correct calibration of the printer

TABLE 2.2: Advantages and Disadvantages in FDM

Ultigcode. Ultigcode is the one that the Ultimaker 2+ uses to print an object; it is based on the Marlin type so most of the commands are the same. Although there are different gcode types, they share most of the basic commands that a 3D printer understand.

This language is built with a set of commands that a 3D printer can understand with two main types of commands, G-commands and M-Commands. The Gcode in FDM tells the 3D printer to move to a specific point in space with a given speed and the amount of material extruded

Instruction	Explanation of Instruction
G0	Rapid move: Move from the current position to the specified coordinates without extruding material.
G1	Linear move: Move the nozzle at a specified speed from the current position to the specified coordinates while extruding a specified amount of material.
G2 and G3	Arc move: Move in a clockwise or counter-clockwise arc to a specified coordinate while extruding material.
G10 and G11	Retract and unretract filament according to the settings of the printer
G17/G18/G19	Set planes: In which plane should the nozzle move. G17 for X-Y plane (default), G18 for Z-X plane and G19 for Y-Z plane.
G20 and G21	Set units: G20 for inches and G21 for millimeters
G28	Homing: Move the nozzle and the bed to its home position
G90	Absolute mode: Interpret the coordinates as absolute coordinates
G91	Relative mode: Interpret the coordinates as relative coordinates

TABLE 2.3: Table of the most used instructions in 3D printing [1]

between each point, or to heat the nozzle of the printer to a specific temperature for example. In Table 2.3 we can observe a list of the most used commands in a Gcode file (extension: .gcode) which usually are stored in an SD card that is read by the printer in order to print an object, and example of a .gcode file can be seen in appendix 1. Each line of code is interpreted by the printer and executes them one by one. In appendix can be seen a Gcode instruction line that is an example of a G1 instruction, this type of instruction will constitute more than 90 percent of the file.

G1 F2280 X101.5 Y 112.3 E8.642

G0 F4000 Z0.27

In this instruction we can see a common instruction in a .gcode file, where the printer will move from the current position to the position X = 101.5mm and Y = 101.5mm with a speed of 2280 mm/min (38 mm/s) with an accumulative extrusion of 8.652 mm of material. The accumulative extrusion means that for example, if in a previous command the printer had extruded 4 mm of material, in this instruction only 4.642 mm of material would be extruded. In the second command we can see a common instruction of moving the printing bed in the Z axis (the vertical axis) to a height of 0.27 mm. In this axis while the number gets bigger, the distance between the nozzle and the printing bed increments.

2.5 Slicing a 3D Model

This procedure is done in all the techniques that work by generating layers and using them in order to print an object. The way it works is that given a 3D model a slicing software uses an algorithm to calculate an intersection between a curve, the model and a plane given a height that is constant for each layer of the model. An example of this procedure can be seen in Figure 2.7 where an object is sliced given the thickness for the layers [9]. When creating a slicer there are several factors to take into consideration as handling peaks, flat areas, staircase effects, postprocessing and the generation of all the support structures.

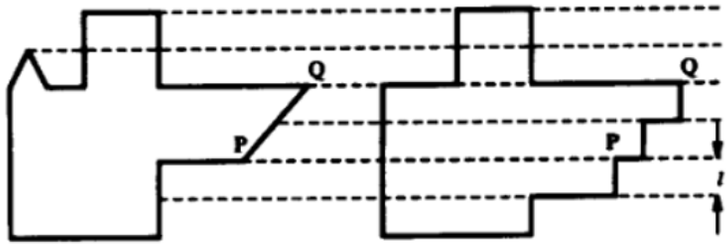

FIGURE 2.7: Uniform Slicing of an object [9].

The support structures can be generated by a slicing software or in a more optimal way they can be pre-generated in the .stl file of the model. These structures are the contours, shells, support structures and infills. The contours will be the outer walls that will have a specified wall thickness of the object and must be as smooth as possible because it is what we can see of the object. Another way to generate the contours is by making shells of an object as seen in the work of. The support structures are necessary to generate layers of the objects that have nothing below and need support in order to be printed. The support structures have to be removed in the post processing part of the printing. And the infills are the interior structure of the printed object that can't be seen in the printed object, but it influences significantly in the physical properties of the object. The infill is the one we are focused on this thesis.

2.6 Infills of the 3D Printed Objects

The term infill in additive manufacturing refers to the interior structure of an object that is printed [10]. It often has a regular pattern, which is selected by the user in the slicing software, along with a specific volume percentage.

One problems in additive manufacturing, that is continuously improved through research, is the planning of the infill of an object. The path planning searches to cover the whole surface of a layer by maintaining the physical properties of the object, and these infill areas need to be optimized in order to have better printing times and reducing the amount of material used in the

process. A slicing software usually has predefined patterns, from which the user can select, and this will affect printing times due to machine acceleration and deceleration of the nozzle, object accuracy, thermal-induced stresses and material used. While covering a layer surface, there are two distinct areas that need to be covered. The first one is the contour that refers to the line formed by the intersection of the layer plane with the outer surface of the part that can be seen in Figure 2.8 with the darker blue line. The contour must be dense in material to provide an object of good quality. In Figure 2.8 we can observe the infill of an object that consists of this area that is inside the contour. The infill pattern and percentage significantly influence the printing process, as well as physical properties of the printed object. In general, a higher volume percentage leads to a print that is more resistant to external loads, while consuming more material and prolonging the print time.

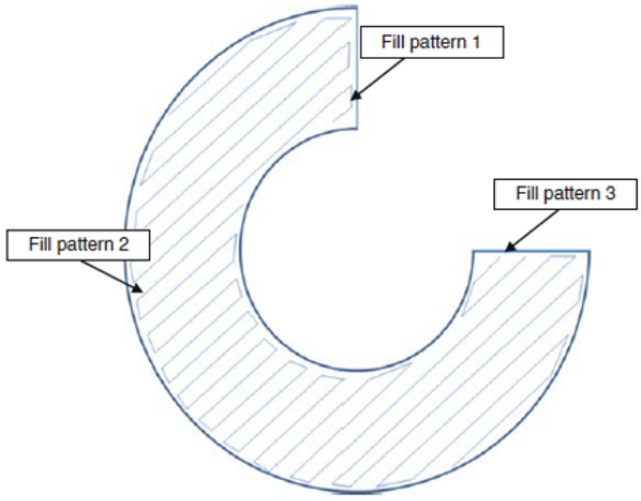

FIGURE 2.8: Non-continuous fill pattern with contour [10].

There are different problems in the generation and printing of infills in a 3D object, for example the way these patterns are formed, how they are printed, the order of the instructions and how they might not be continuous giving as result moving the nozzle to another position without printing increasing the overall printing time. A typical zig zag fill pattern can be seen in Figure 2.8 [1]. Since rapid changes in direction can make it difficult to control material flow and the total printing time, a common strategy would be to draw the outline of the part to be built. The internal fill pattern can be built more rapidly since the outline represents the external features of the part that corresponds to geometric precision. But as we can see in this example there are fill patterns that are disconnected increasing final the printing time.

An effective path planning strategy would automatically generate the infill geometry as well as create a set of movements in such a way that the total building time of each layer is minimized. This can be achieved by generating a path that covers the infill of a slice with the minimum percentage possible of covering area with a continuous path. Examples of current patterns used

in infills of 3D printing can be seen in Figure 2.9.

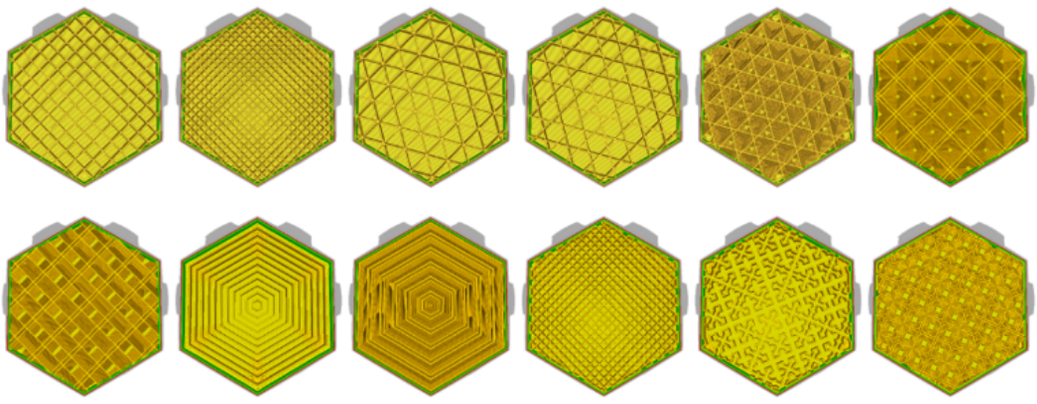

FIGURE 2.9: Infill patterns available in the Ultimaker 2+™. The infill patterns are displayed in the order as follows: Grid, Lines, Triangles, Tri-hexagon, Cubic, Cubic (subdivision), Octet, Quarter cubic, Concentric, Concentric 3D, Zig-Zag and Cross, Cross 3D [Ultimaker™].

2.7 Slicing

With slicing tools it is possible to convert digital 3D models into printing instructions for 3D printers. The general approach is cut the model into horizontal slices, which are used to create extrusion paths, and finally the slices are filled with material, according to a defined infill density. The goal of this section is to have a brief explanation of the different phases of a slicing tool in order to have the layers of an sliced object. Here, we focus on representing a model as the input 3D geometry for the slicer. The goal is to how to slice a model into a series of thin layers and produce a G-code file containing instructions meant to FDM printers.

Since we do not need to print objects that are 100 percent solid all of the time, creating a hollow model means that one needs to design the object with "walls". A cross-sectional layer is divided into a few different types of areas. Two of these are the shells and the infill, as shown in Figure 2.10. Shells refers to the number of times the outer walls of the design are traced by the 3D printer before starting the hollow inner sections of a user's design. Increasing this number will create thicker walls and improve the strength of the print. Each layer prints with at least one perimeter or shell (outer shell) to define the outline of that layer, and normally there are additional one or two inner shells printed before the infill. Support structures are commonly required in manufacturing technology. In this work, we focus on contour extraction (outer shell and inner shell) from a third party slicer, and the infill strategy with its generation.

A printing setting that we need to consider is the infill density, which defines the amount of material used inside the print. The calculation of percentage infill density is not based on the volume of the object but only a function of the extrusion width (nozzle diameter), so the user

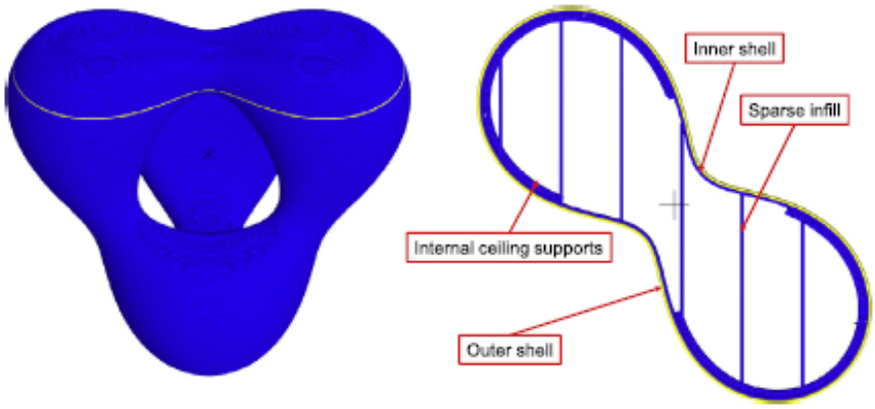

FIGURE 2.10: Cross sectional layer

will have to specify different percentage densities for different extrusion widths. It is important because the bigger the density, it takes more time to print an object, but is has a better resistance. So we need to find a balance in the infill density, so we can have a resistant object but that it doesn't take a lot of time to print.

An STL file consists of a list of triangle facet data which define the surface of a 3-dimensional object. Each facet is uniquely identified by a unit normal (i.e., a line perpendicular to the triangle and with a length of 1.0) and by three vertices. Orientation of a facet, shown in Figure 2.11, is determined by the direction of the unit normal and the order in which the vertices are listed, where p1, p2, p3 are three vertices of the triangle. First, the direction of the normal is outward. Second, the vertices are listed in counterclockwise order when looking at the object from the outside (right-hand rule).

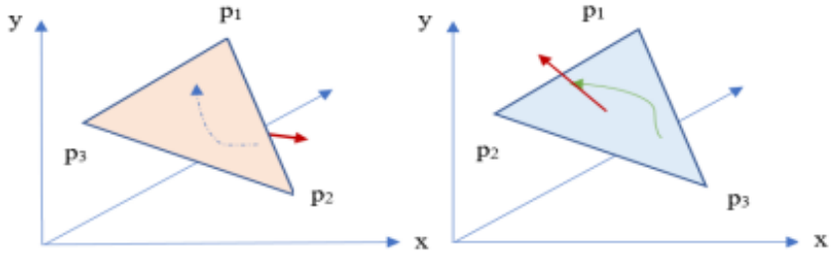

FIGURE 2.11: Direction of the face normal based on the right-hand rule for an STL being sliced

2.7.1 Preprocessing

The input is an unlabeled triangle model and the output is a printable configuration of the model using an infill algorithm. We define a metric to evaluate how well a set of properties for 3D printing is achieved; those properties include a volumetric approximation, generation of contours, support material, optimal infill generation, faster print time, and less angled paths

in the infill leading to minimizing the printing time. Then, we describe and demonstrate an optimization process that yields a careful balance of the desired properties.

2.7.2 Slicing Open Source

A slicer is a type of 3D printing software that converts digital 3D models into printing instructions so that a 3D printer can create an object. The slicer cuts a CAD model into horizontal layers based on the settings a user chooses, and calculates how much material the 3D printer will need to extrude and how long it will take to do. All of this information is then bundled up into a G-code file, which is sent to the 3D printer.

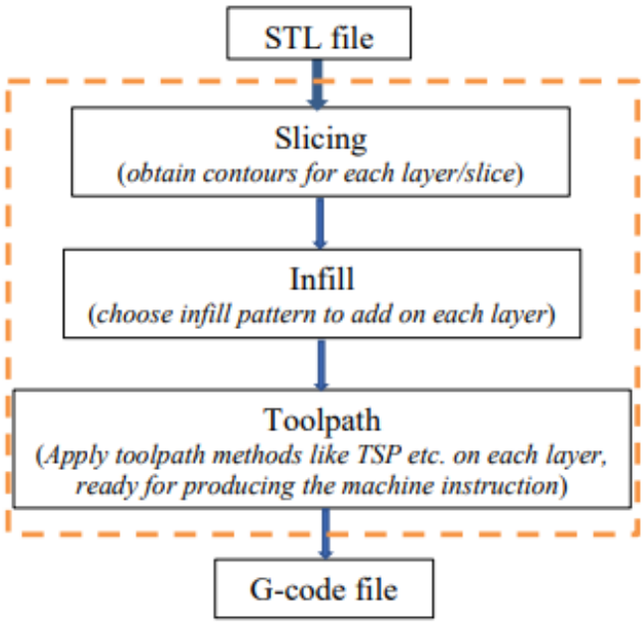

FIGURE 2.12: Software pipeline from input STL to output G-code

2.7.3 Slicing Pipeline

The Figure 2.12 shows the software pipeline between input STL file to output G-code file, i.e., convert a 3D model into printing instructions for the 3D printer. It cuts the model into horizontal layers, generates tool-paths to fill them and calculates the amount of material to be extruded. In current 3D printing practice, the most common technique for slicing is to produce contour data from STL files. The STL model is then sliced by intersecting it with horizontal slicing planes, each of which gives linear contours of a slice. There are two options for the STL-based slicing and direct slicing based on different 3D CAD systems with different data formats. In Figure 2.12 we can observe the pipeline of an STL file.

2.8 Particle Swarm Optimization (PSO)

The particle swarm algorithm consists of a number of simple particles that are placed in the search space of a problem or function, where each particle evaluates the objective function at its current location [11]. Then each particle determines where to move through the search space by different parameters like mixing the history of its own current and best evaluations. These are then compared with one or more members of the swarm, with some random mutations. The next iteration takes place after all the particles of the swarm have moved. Eventually the swarm as a whole moves closer to an optimum of the fitness function. The particle swarm is more than a collection of particles, because a particle by itself could not solve any problem, progress is made only when the particles interact with each other.

Each individual in the particle swarm is composed of three D-dimensional vectors, where D is the dimension of the search space. These are the current position x, the previous best position p, and the velocity v of the particle.

The current position x can be considered as a set of coordinates describing a point in space. In each iteration of the algorithm, the current position is evaluated as a problem solution. If that position is better than any that has been found so far, then the coordinates are stored in the second vector, p. The value of the best function result so far is stored in a variable that can be called p_{best} (for "previous best"), for comparison on later iterations. These values are stored in order to keep finding better positions and updating p and p_{best}. New points in the search space are chosen by adding v coordinates to x, and by adjusting v, which can be referred as the step size in the algorithm process. In this optimization process, in each iteration the velocity of each particle is adjusted so it can find an optimal or sub-optimal solution. The process of implementing PSO is as the following algorithm:

Algorithm PSO.

1. Initialize a population array of particles with random positions and velocities in the search space.
2. loop
   (a) 1: For each particle, evaluate the desired optimization fitness function.
   (b) 2: Compare particle's fitness evaluation with its p_{best}. If the current value is better than p_{best}, then set p_{best} equal to the current value, and p equal to the current location x in the search space.
   (c) 3: Identify the particle in the neighborhood with the best success so far, and assign its index to a variable g where you store the best one.

(d) 4: Change the velocity and position of the particle according to the following equation Eq. 2.1:

$$\begin{cases} \vec{v_i} \leftarrow \vec{v_i} + \vec{U}\,(0,\phi_1) \otimes (\vec{p_i} - \vec{x_i}\,) + \vec{U}\,(0,\phi_2) \otimes (\vec{p_g} - \vec{x_i}\,), \\ \vec{x_i} \leftarrow \vec{x_i} + \vec{v_i}\,. \end{cases} \tag{2.1}$$

where $\phi = \phi_1 + \phi_1 > 4$ and

$$x = \frac{1}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \tag{2.2}$$

(e) 5: If a good fitness is found or the maximum number of iterations its reached, exit loop.

3. end loop

The parameters of the algorithm is the size of the population. This is often set empirically on the basis of the dimension and perceived difficulty of a problem, in other word based on experience. Values in the range 20–50 are quite common. The parameters ϕ_1 and ϕ_2 in determine the magnitude of the random forces in the direction of personal best $\vec{p_i}$ and neighborhood best $\vec{p_g}$. These are often called acceleration coefficients. The behavior of the PSO changes radically with the value of ϕ_1 and ϕ_2. Other parameters that can be added to the algorithm to make it more complex but to also have theoretically better results are the inertia weight, the constriction of coefficients Eq. 2.2 and having a fully informed particle swarm.

2.8.1 Summary

Through this chapter we have discussed about the fundamentals of additive manufacturing regarding the different techniques that are used in the industry and focusing in the technique of Fused Deposition Modeling explaining how it works, the different key parts of a printed object, how the process is done and how from having a model it is transformed into a series of instructions that a 3D printer can understand in order to print an object. At the end of this chapter we focused on the slicing and infills of a model due to its essential role in the investigation, as we want to optimize the infill of a model by printing it as a continuous path, we will discuss more of path techniques in the next chapter, in the less amount of time possible and maintaining good physical properties of the object.

Chapter 3

State of Art

3.1 Related Work in Infill Optimization

In the industry exists the growing necessity to create rapidly an object to be referred as a process called additive manufacturing. Consequently, there have been an increasing demand and publishing of works in computer graphics in order to optimize the printing process and characteristics as: structural strength, appearance, stability, shapes of the objects, printing time, material employed in the process and printing volume. These characteristics can be influenced by different parts of the printing process like support structures, shells, contours and infills.
In here we will treat a tool planning problem that involves the infill of an object that is the interior structure of an object that is printed as can be seen in Figure 3.1. As we discussed in the last chapter the printing of an object, as the infill of such object, will be printed with the FDM technology that melts a filament that is extruded through the print nozzle of the 3D printer and it is done layer by layer. Because the extruded material is a viscoelastic material there are problems that need to be considered, as the extrusion control that affects the printing. These problems are such as uneven fills causing visual problems, or the generation of gaps between each path that can affect the structure strength that can be caused by the necessity of turning on and off the extrusion which is difficult to control and can cause problems that we can see in Figure 2.9. Thus there is the necessity of creating a path that minimizes these problems, specifically turning on and off the extrusion of material, and this can be achieved by creating a continuous path through all the layer or most of it, considering facts as the strength of the structure, the printing time, the computational process time to generate each layer, the amount of material used and the area covered.

Many tool path patterns have been developed for additive manufacturing process, such as raster, zig zag, contours, spirals, fractals, and continuous paths that have advantages and disadvantages

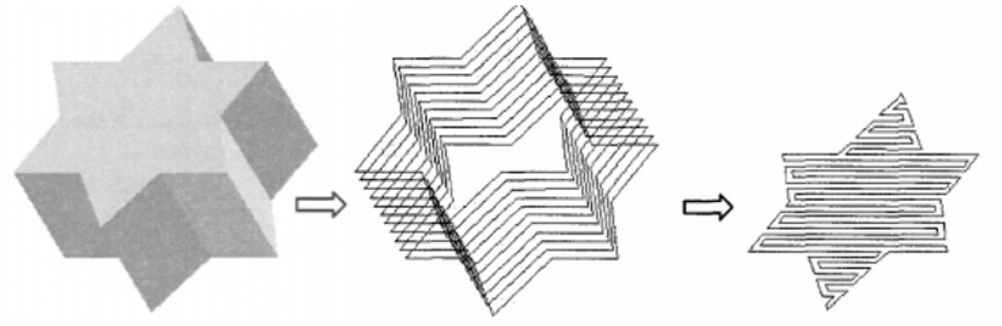

FIGURE 3.1: Tool for path planning [1].

like the implementation, the accuracy, tool-path passes and tool-path elements as seen in Table 3.1 and Table 3.2. In the next two tables we can see a comparison between these patterns used currently in additive manufacturing. In these tool paths the curvature of each path influences in a significant way the printing time and quality of the object. With sharp turns in the printing process there is the requirement of deaccelerating and accelerating causing a longer printing time [12], this will vary according to the angle of each turn that could also cause over-filling or under-filling in the printing.

There are two different classifications of parallel infills Figure 3.2, direction-parallel and contour parallel [13]. In the direction parallel there are two main tool paths raster scanning and zigzag. The raster technique is based on planar ray casting that goes in one direction where the 2D regions are filled with lines with a certain width [13]. From this technique the zigzag tool path was created, and it is widely used in AM figure 13-a. The zigzag works in a similar way

Advantages and Disadvantages of the reviewed tool path to generate the infill of an object		
Raster	+a	-b, -c, -d
Zigzag	+a	-b, -c, -d
Contour	+b	-a, -c
Spiral	+c	-a
Fractal curves	+c	-a, -d
Continuous	+c	-a, -d
Hilbert curves	+a, +c	-d
Fermat spirals	+c, +d	-a

TABLE 3.1: Advantages and Disadvantages of infill patterns

Indicators of performance of each reviewed tool path	
a	Easy implementation and simplicity of algorithms
b	Good accuracy of the model
c	Less tool path passes
d	Less tool path elements

TABLE 3.2: Indicators of performance for infills

to the raster technique where it creates lines that fill all the space of the object that also go in one direction and connects them with smaller segments on the ends to create a continuous path to minimize the path passes, thus reducing the printing time by eliminating the necessity of turning on and off the extrusion of material. RM There are two different classifications of parallel infills Figure 3.2, direction-parallel and contour parallel

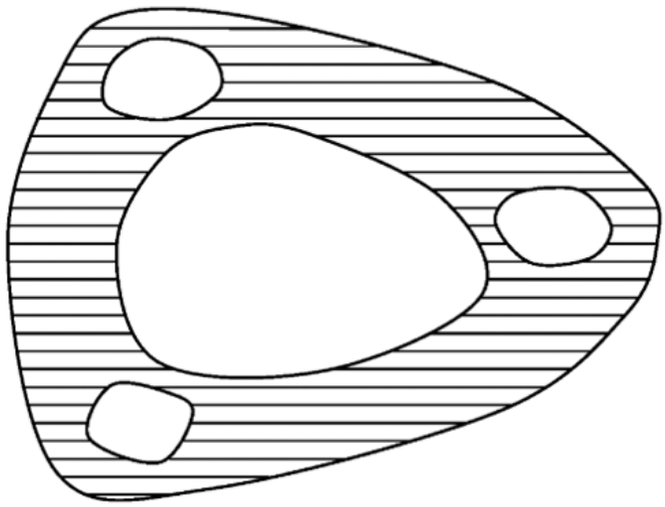

FIGURE 3.2: Example of the parallel infill method [13].

The zigzag pattern is the pattern most commonly used in 3D printing, specifically in FDM. This pattern consists in a path created from 90 degrees turns connected between them. Although this tool path is one of the simplest used in the industry, it has a high printing time due to the sharp turns, it doesn't provide a good accuracy of the model, and is not appropriate for models with complex contours. The contour technique focuses on the quality of the exterior of an object by generating contour patterns that go along the closed contours of the outline for each layer. There is also the possibility of creating a hybrid as seen on [14] that combines the two previous classifications by generating some contours and then it fills the space with a zigzag path. A combination of contour and zigzag pattern is commonly developed to meet both the geometrical accuracy and build efficiency requirements.

The contour technique focuses on the quality of the exterior of an object by generating contour patterns that go along the closed contours of the outline for each layer. There is also the possibility of creating a hybrid as seen on [14] that combines the two previous classifications by generating some contours and then it fills the space with a zig zag path. A combination of contour and zigzag pattern is commonly developed to meet both the geometrical accuracy and build efficiency requirements.

An example can be seen at [15] here an algorithm was designed for wilding-based applied to additive manufacturing where there is a path that fills the inner part of the object and a path that covers the outlines that provides a better surface quality for the final printing process. Giving as a result to fill the interior area of each layer to improve the efficiency in printing time,

and the contour tool-path is used to fabricate the area along the boundary of the contours to improve the geometrical quality of the model. This is widely used in most printing processes used in FDM as it can be seen in Figure 3.3-a, where we can see a zigzag pattern without a contour and in Figure 3.3-b there is a contour to give a better quality of the final printing process of the object in Figure 3.3-a with only the Zigzag and in Figure 3.3-b with zigzag and a contour.

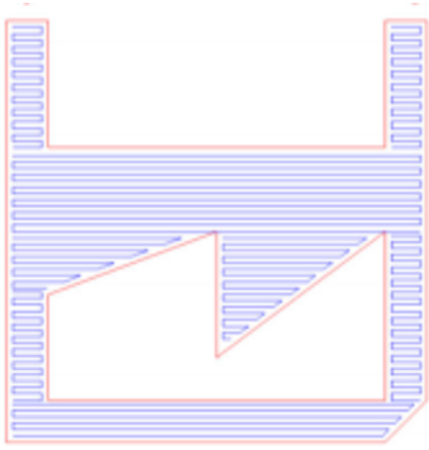

FIGURE 3.3: Example of the zigzag method, without a contour on the left a), and with a contour on the right b). [16].

The spiral tool path is widely applied on pocket machining [17], specially 2D pocket milling and uniform pocket cutting, where it generates spiral pattern as seen in Figure 3.4-a. This tool path can generate the problem that several spirals are generated in the layer and they are not connected between them. This can be addressed by generating a continuous path by combining it with a zigzag pattern like in Figure 3.4-b or it could be more complex objects that need to connect various continuous spirals paths like the one in Figure 3.4.

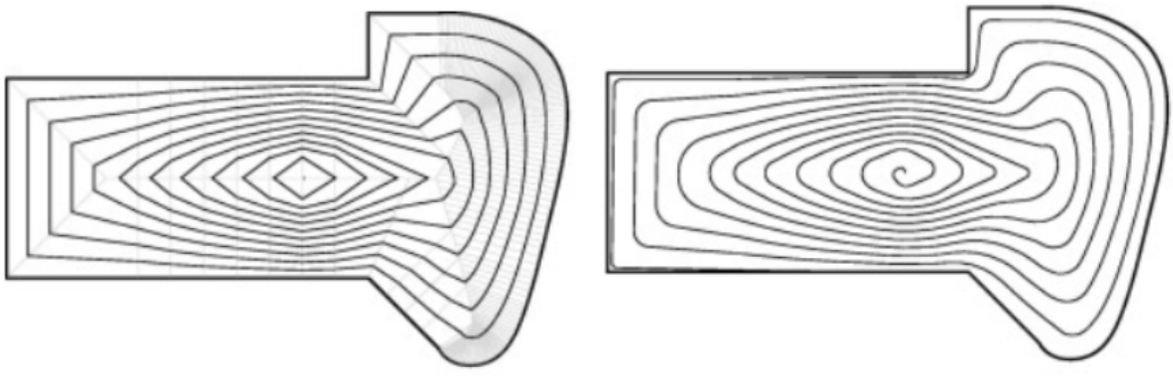

FIGURE 3.4: Example of the spiral method, without optimization on the left a), and with optimization on the right b). [17].

The space filling curves is a continuous tool path that is based on space curves to cover a layer that is represented by a 2D region. This space filling curves can be complex to generate depending on the pattern and model, and can contain a lot of sharp turns that can affect the printing time by increasing it. Continuous path planning can be considered as another tool-path generation method that can be included inside the space filling curves. In here we have different examples as Hilbert curves [18] and Fermat spirals [19].
The Hilbert curves are a continuous path planning that can be applied to generate the path of the infill of an object. This method creates a pattern for the infill that follows a path that in most of the times can be executed in a single continuous move of the printing head. As we have discussed this eliminates the necessity of turning off and on the extrusion of the nozzle and the rapid feed movement that as a result helps to save time in the printing process.
The way it works is that it generates recursively the pattern of the infill by having nodes at four points per circle or area defined, where each one of them is tangent to an adjacent one or it is tangent to the contour giving as a result a structure making the object stiffer and inherits isotropic properties to the object. An example of the Hilbert filling curve can be seen in [18] as a generated continuous path that covers all the region without any intersection as can be also seen in Figure 3.5.

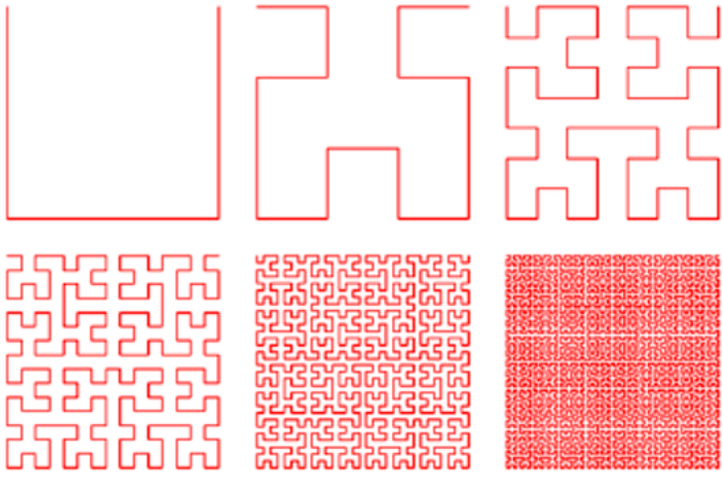

FIGURE 3.5: Example of Hilbert curve generation [18]

In the fractal world there is a difference in the creation of the path planning of each layer of a model. Fractals normally can be used to model natural objects that cannot be represented by Euclidean geometry, what they did [20] is to use a methodology to represent a fractal in a general layered manufacturing that is what is used in FDM. They do it by a grid that is generated in each layer of the object that consists of a number of pixels. With this the tool path can be extracted from the mathematical model of the fractal. This methodology only works for fractal models, it can be applied to iterative function systems (IFS) or complex fractals, however for multi-IFS fractals it is not possible.

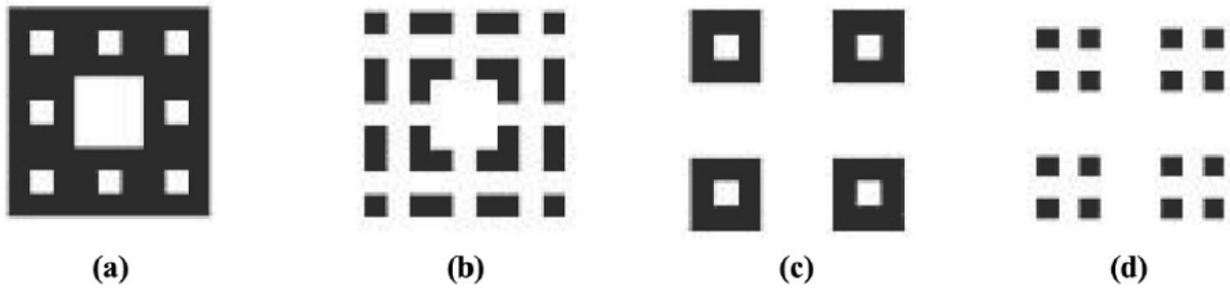

FIGURE 3.6: Example of fractal layers. [20].

Fermat spirals are another type of space filling curve created by [19]. This method started as a variant of the spiral method but with the singularity of creating a continuous path planning that can be applied to generate the path of the infill of an object giving as a result connected Fermat spirals. The main objective of Fermat spirals is to achieve global continuity and the generation of new space-filling patterns. This connected Fermat spirals are formed mostly by long paths that have low curvature in the path planning in order to reduce the acceleration and deacceleration of the printing head. The low curvature property working along with the continuity of the path influences the quality and efficiency of layered fabrication. The infill of each layer of the object is represented by a global continuous curve that always start and finish the Fermat spiral path at approximately the same location on the outer boundary of the filled region as can be seen in Figure 3.6; it provides a freedom in choosing start and end points along the boundary of a Fermat spiral that facilitates a scheme which systematically joins a set of Fermat spirals.

This is a curve optimization technique that gives as a result a lower printing time between each layer because it minimizes the movement of the printing head without extruding material and leads to an efficient and quality fabrication compared to conventional fill patterns.

In the research they show tool path generation results on different objects with different types of hollowness and concavity in a specific. They compared their method to traditional infill methods like zigzag and contour parallel to get a difference in different characteristics the path continuity, how many sharp turns are in each layer that in comparison to the zigzag pattern it generates less sharp turns by itself meaning they do not attempt to minimize them, the printing time, the visual quality of the interior infill and the contour of the object. More specific characteristic that they claim is that the use of Fermat spirals can prevent a curve from being locked in a pocket, and that this method is not recommended for shapes that have many holes and is not efficient in this way compared to other patterns.

Labyrinths are a tool used in the generation of patterns. This type of pattern, as in the Fermat spirals starts and finishes the path at approximately the same location. This pattern was created with the help of the Brownian motion to the production of organic labyrinths and mazes giving

FIGURE 3.7: Example of connected Fermat Spiral. [19].

control over both the path complexity and visual aesthetics as the curves evolve to generate the final result of a pattern to work as a space filling curve.

Domain decomposition [21] is another technique to obtain a continuous path that covers the infill area of a model by decomposing each layer of de model that is considered a 2D region into various subregions in the layer. Each subregion generated by this method contains a continuous fill. After each region has a continuous, each region path is connected to achieve global continuity as can be seen in Figure 3.7. An example can be seen in [21] where they decompose a polygon into a set of sub-polygons and generate a path that fills each sub-polygon using a closed zigzagging curve allowing to choose a start point arbitrarily as in Figure 3.8.

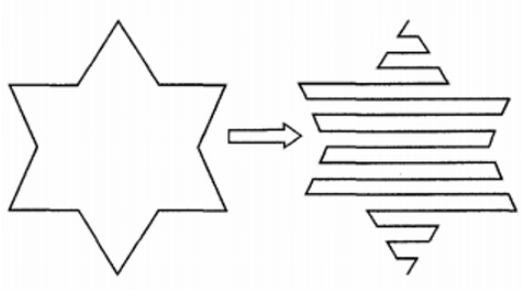

FIGURE 3.8: Example of domain decomposition. [21].

Comparison of Infill Optimization in FDM					
Author	Name Technique	Algorithm Complexity	Continuous Path	Sharp Turns	Task Performed
Ding [16], Misra [22]	Zig-Zag	1	No	Yes	Infill (all layers)
Yang [23]	Contours	1	No	Yes	Contours of infills
Held [24], Luo [25]	Spiral	2	Yes	No	Infill (all layers)
Papacharalampopoulos [18]	Hilbert Curve	2	Yes	Yes	Infill (all layers)
Chiu [20]	Fractal Curve	2	No	Yes	Infill of fractals
Zhao [19]	Fermat Spiral	3	Yes	No	Infill (one layer)
Pedersen [26]	Labyrinths	1	No	Yes	Infill (all layers)
Dwivedi [21], Ding [16]	Domain Decomposition	2	No	Yes	Subregions of infills

TABLE 3.3: Infill Optimization Comparison in FDM. The algorithm complexity is in a scale from on to three where three is the hardest.

3.2 Summary

In the Table 3.4 and Table 3.3 we observe a comparative table that demonstrates the advantages and disadvantages of each work listed by author and technique used in their work and the comparison of the infill optimization respectively.

RM Labyrinths are a tool used in the generation of patterns. This type of pattern, as in the Fermat spirals starts and finishes the path at approximately the same location. This pattern was created with the help of the Brownian motion to the production of organic labyrinths and mazes giving control over both the path complexity and visual aesthetics as the curves evolve to generate the final result of a pattern to work as a space filling curve.

With the review and analysis of these multiple papers focused in optimizing the infill of a 3D printed object shows the necessity of creating new algorithms that optimize the infill of an object. This should be done with the investigation to fill patterns with a continuous path and consecutive layers. One thing to consider is to have the consecutive layers fabricated differently, or at least not to similar in order to keep the strength in the objects. Finally, another factor we take in consideration is the start printing point and final printing point in each layer with the purpose of increasing the coherence between each layer. All of this with the combination of orientating properly the object and the decomposition of the object are also good opportunities for the optimization of the infill in an object.

Comparative table between each reviewed related work			
Dunlavey [13], 2003	Contour Parallel	Good accuracy of the model and the surface quality.	High printing time. No single continuous path
Jin [14], 2013	Zigzag	Easy implementation. Most common technique used. Fits most of the models.	High printing time. Sharp turns in each layer. No single continuous path
Abrahamsen [17], 2019	Spiral	Good accuracy of the model.	High printing time. No single continuous path.
Papacharalampopoulos [18], 2018	Hilbert curves	One continuous path. Low processing time for each layer. Easy implementation.	High amount of sharp turns in each layer.
Zhao [19], 2016	Fermat spirals	One continuous path. Minimal sharps turns. Improve surface quality. Low printing time.	High processing time for each layer. Not Suitable for models with holes inside the object

Table 3.4: Comparative between the reviewed Work

Chapter 4

Solution Proposal

4.1 Introduction

In this section we describe the problem, and how we plan to solve it by using a new generated pattern that is tested in different printers with different types of objects. The key part is to properly define the angles that we want to use in the experiments, the correct design and generation of the infill, having a continuous infill and the execution of each experiment. We can state that our solution is good and we can prove the hypothesis if our proposed algorithm is faster than the zig-zag pattern an we have a lower p-value than 0.05.

The printing problem we face in additive manufacturing is that it takes considerable amount of time, being primarily the infill of an object the factor of this thesis. This creates the necessity of having new technologies that minimize the printing time of an object, but also maintaining its physical properties. In this work we focus in the problem of the infills of the printed object.

The amount of time and material to print an object are the two of the most important factors in the printing process of an object that translates into the cost of a prototype in additive manufacturing, being the time factor very high in most of the processes in Additive Manufacturing. Therefore, there is the growing necessity of printing in a faster way in Additive Manufacturing to reduce the cost of a printed model in terms of material and time.

In the next subsections we will discuss about a new architecture proposal along a new infill algorithm generator. The architecture describes such algorithm in a high level approach, and in the algorithm we describe its baseline, the ideas from it becomes and why it works this way, and this will be demonstrate in Chapter 6.

4.2 FDM Infill Architecture using PSO

Before considering the architecture proposal and the need for this architectural, we provide a little background on software architecture and introduce some key aspects to it. The concept of software architecture as started to emerge a few decades ago with the description of programming languages, formalization, and classification of architectural styles. Here we are implementing a design pattern, a design pattern describes solutions to a recurring problem. We will use for this and all diagrams UML, which is often employed for architecture documentation. [27]

There are various elements on which an architecture can be helpful to describe the implementation of a new design to solve a problem. It will help us to make an analysis and evaluation to the proposed solution, also it supports an improved solution. Software architectures become amenable to analysis and evaluation. This helps to evaluate architectures and to guide in the selection of architectural variations as solutions to specific problems, in this case to the creation of a new algorithm proposal to create a new infill type for Additive Manufacturing.

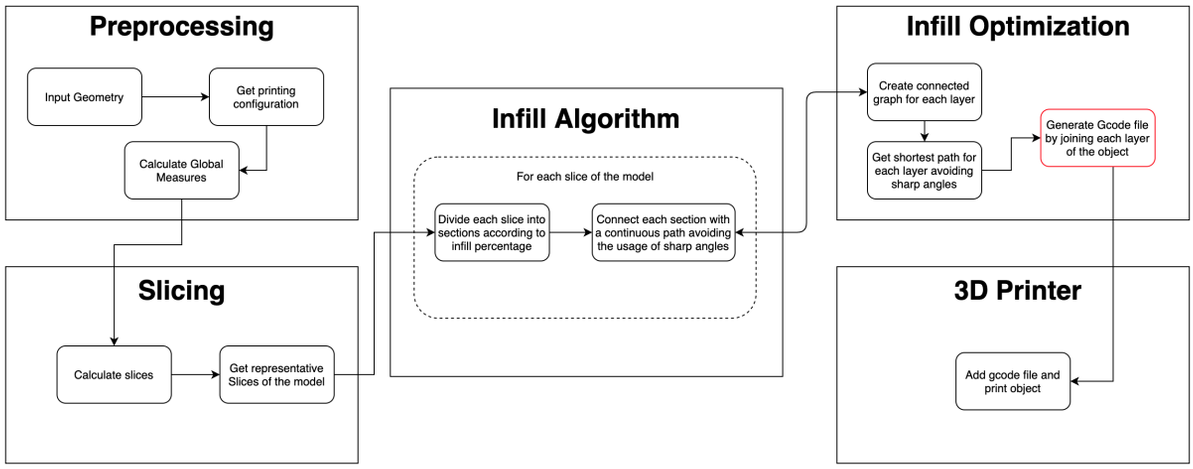

FIGURE 4.1: Proposal of the Architecture.

In Figure 4.1 we describe the architecture of the whole system that has as a result a gcode file that has the information of the object that will be printed with the new infill generation and optimization proposed in this work. There are four main parts that compose this process. First, from an input from the user that gives the mesh of a model with the extension .stl of the file that will be use in this preprocessing stage. We use a third party slicer that from the input geometry it calculates the global measures of the model and will give them as an output in a gcode file, in this step we are also able to do the next step of the process that is the slicing of the model. As we have discussed we have to make slices of an object with an exact measure that the user gives, that usually is 2 millimeters in height for a printed object with high resolution. In this second process slicing we find the representative slices of the model, this means all the slices that conform the complete model from the input height, following the calculation of the measures of each slice, that are the contour of the slice with its respective (x, y, z) position. All

of this information will be in the output gcode file, it will be read and parsed so we can store them and can work with them in order be able to do the next step of the process that is the slicing of the model.

The next step is the generation of the infill using our algorithm, in here we use the slices of the slicing process and for each layer we divide the layer in different sections according to the infill percentage, then we connect this sections with series of lines avoiding sharp angle creating a continuous path. The angles used for the generation of the infill are the best angles we got from the first experiments of this work that can be seen in the first part of Chapter 6 where we can see the design of experiments and in Chapter 7 where the results and discussions are described. We will use a type of spirals to connect each point and have a continuous path with the less amount of sharp turns possible, here is where the optimization algorithm will give us the best result from all the possible solutions, and will do for each slice of the model. The primary goal of optimization is having a continuous extrusion and avoiding sharp angles with the purpose of minimize the distance traveled along any curve where material is deposited and so minimizing the printing time. For this experiments we assume that the speed of the printing head and extrusion rate are constant. Finally in the fourth process we generate the gcode from the results of each slice that are a set of vectors that connect between each other and give as a result the gcode file containing all the instructions for printing the object using our infill algorithm.

In the following sections we will discuss in detail the steps Infill Algorithm and how we make this infill being optimal for the architecture given above Figure 4.1 and give some examples to illustrate how each step works on its own and as a complete system. It is important to mention that the preprocessing and slicing parts of the architecture model are not described here in detail, since we are using existing free source software to make this steps since it is not the principal objective of this thesis.

4.3 New Infill Algorithm Proposal

One of the main problems faced by the AM industry is the effective path planning used to build up an object. Path planning refers to defining the scanning strategy that will be used to cover the whole surface of a single layer (slice) of the object. AM software usually have predefined hatching patterns, from which the user can select the desired one. This can affect both build-up times and printing time due to machine acceleration/deceleration and dimensional accuracy of the object.

While covering a layer surface, there are two distinct areas that need to be printed. The first one, is called the contour and the second one that is what we are focusing on, is called the infill

that consists of the area enclosed by the contour. Depending on the use of the infill, it could be fully dense just like the contour, or have a user-defined infill percentage as it has in most infills commercially used. The infill percentage is defined as the ratio between the volume of material and the volume of free space/voids. In a fully dense build will have an infill percentage of 100% and an empty space will have an infill percentage of 0%, the usual percentage of infill used in practice is around 20%.

An effective path planning strategy would automatically generate the infill geometry as well as create a set of movements in such a way that the total build up time of the layer and so the object that is being printed is minimized. There are works that address the problem as the ones described in the Related Work in Chapter 3.

This work proposes an extended method to generate a continuous path that is appropriate for covering the infill of a slice, in three principal configurations, with high density, with lower density, and the average density of infill percentages. This new method for generating an infill algorithm adopts the idle times minimization methodology [28] which impacts in the printing time, and material consumption, more specifically by having a continuous path that uses the best curves possible, meaning the ones that take less time to print which are the ones closer to a 0 degree angle turn that will be explored and discussed in Chapter 6. This is due to the factor of proving that it reduces the printing time by having less sharp turns in the infill generation of the object.

An effective method to minimize the total build time and energy consumption in AM would be the reduction of sharp turns the printing nozzle performs in the printing process. Besides the time delays introduced by these moves, energy consumption during the acceleration/deceleration phases of the processing head is high, due to the high acceleration values. Also is important to notice that the changes of speed in the printing nozzle (acceleration and deceleration) could compromise the build quality as it provokes vibrations and other factors on the machine. This is why the minimization of this movements are an object of study and improvement. For this reasons and in this way, we are creating this new method of filling a certain space by using a single continuous curve with a minimization of sharp turns. It is necessary to construct an algorithm that generates this infill by maintaining the structural integrity and mechanical properties of the object. Therefore, a more robust way of generating the infill pattern and creating the path the nozzle will follow is needed.

The proposed method uses the idea of having a continuous path, minimizing the sharp turns the nozzle performs and converting the space into a geometric covering problem. For this we propose the following: to discretize each layer of the geometry in sections, and from this create a mesh of the layer object. Then a single continuous path would be created with the constraints: passing through all the sections of the layer and only using the curves that avoid the creation of sharp turns through all the layer. Therefore, an infill pattern with a single continuous path

that is tangent to all the sections of each layer of the object is created. Based on this pattern we can use an optimization algorithm that runs this several times, starting from different points or using different curves to create the best possible infill for a particular layer of a particular object. In Figure 4.2 we can see the algorithm for generating the infill by the method given above.

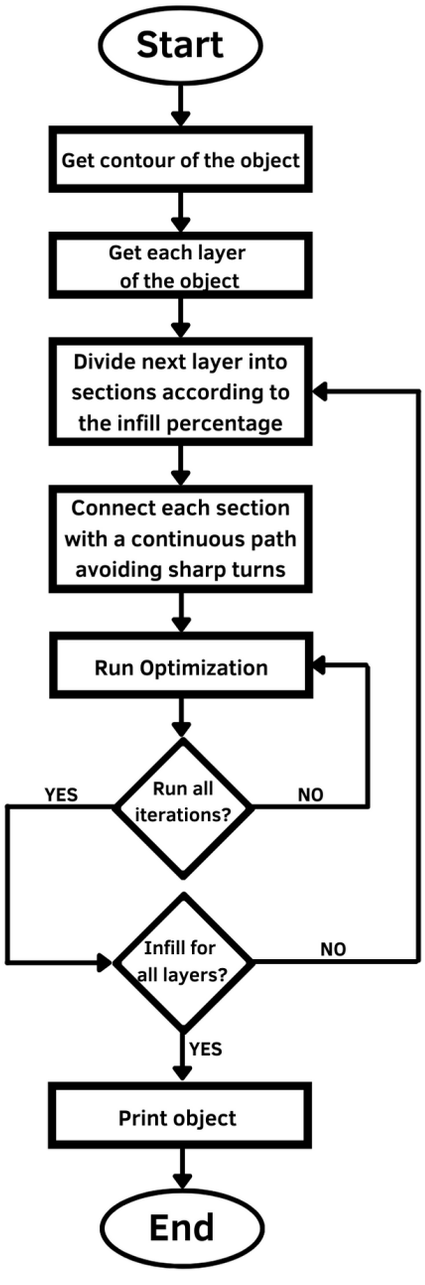

FIGURE 4.2: Description of new algorithm for infill generation with a continuous path.

In the infill generation we mentioned sections where the infill pattern will be generated, and these sections are defined for each object and each layer. The creation of the sections in a layer

depends accordingly to the infill density selected for the current object. The layer is divided into sections that visually can be seen as squares in a plane as in Figure 4.3, as bigger the infill density we will have more squares in the plane. Each section has a position that has to be covered when we are generating the infill for each layer, so when we have the infill generated for a single layer, the continuous path should pass through all the sections generated for that single layer.

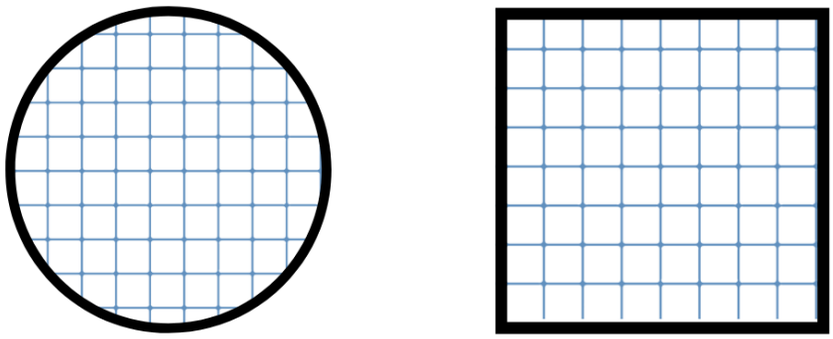

FIGURE 4.3: Visual representation of how a layer is divided into sections in (a) cylinder layer, and (b) cube layer.

After the generation of a layer of the object that contains the infill with a continuous path and avoid sharp turns, we use the heuristic PSO previously mentioned. This heuristic takes into consideration a function that estimates the printing time of the layer, and in the first iteration it will store it as the best solution. Then it will run all the next iterations with a predefined maximum of one hundred iterations where it will generate new infills where some aspects change, like the starting and ending points, the continuous path created, and some mutations applied to the continuous path. After each iteration it will use the fitness function to estimate the printing time of each infill generated and assign the best solution to the path that takes less amount of time. This will give us a result of the best continuous path found for the current layer that can be considered as a sub-optimal solution because at the moment due to the nature of the heuristic and the fitness function, we don't have a way to know that the solution found is the best of all the possible solutions that can be generated for a single layer.

This gives as a result a new infill pattern that can be printed in a single continuous path of the printing nozzle and with the sharp turns minimized and passes through all the sections of each layer of the object. By this we can save printing time, this method along with the experiment performed to prove that sharp turns increase the printing time and get the best curves for this algorithm, the experiments and results are presented in Chapter 6 and Chapter 7 respectively.

Chapter 5

Implementation

5.1 Introduction

In this section we show the implementation of the system, the design of the functionalities and how we plan to solve the problem through this functionalities. We do this by using different UML diagrams that are: Use case, activities, components and classes. Each diagram show different perspectives of the system that will allow to have a better understanding of the whole process.

5.2 Diagrams

UML, short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems. The UML uses mostly graphical notations to express the design of software projects. Using the UML will help us to project our ideas, communicate the implementation of the system, and validate the architectural design of the software.

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts, some of the most commonly used diagrams are: Use Case Diagram, Class Diagram, Component Diagram, Activity Diagram, Deployment Diagram, Sequence Diagram and Communication Diagram. For this section we will implement the first four diagrams that are: Use Case, Class, Component and Activity.

5.2.1 Use Case

A use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases.

In the use case seen in Figure 5.1 we can see an overall of the functionality of the system. There is a main user that could be a designer, a researcher, or any user that needs to print an object using a 3D printer with the FDM technique. Here we illustrate the actions the user can perform that are: Import an sliced object they want to print, it is a requirement that the file is in .gcode format to be processed. The user needs to fill the specifications of the printer used and the characteristics which the user wants the object to be printed. Next, so the process of the generation of the infill of the object can be performed where we use our algorithm to create the infill for the layers. Finally the user has the option to create the gcode from the sliced object. In this final part is where the system generates the infill of the object with the algorithm proposed and creates a gcode from this step.

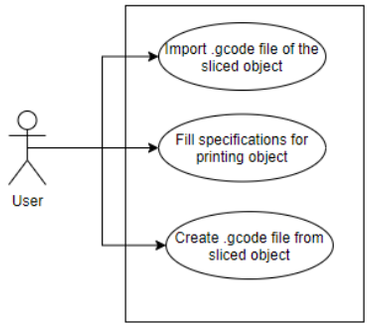

FIGURE 5.1: Use Case Diagram: Describes the functionality provided by the system.

5.2.2 Classes

A class diagram describes the structure of an object-oriented system by showing the classes in that system and the relationships between the classes. A class diagram also shows constraints, and attributes of classes. In the diagram of classes seen in Figure 5.2 we can observe the structure of the system in an object-oriented perspective. Here we represent the different relationships that exist between the objects of the systems, representing how they are build in object oriented programming and also showing their attributes and functions that can perform.

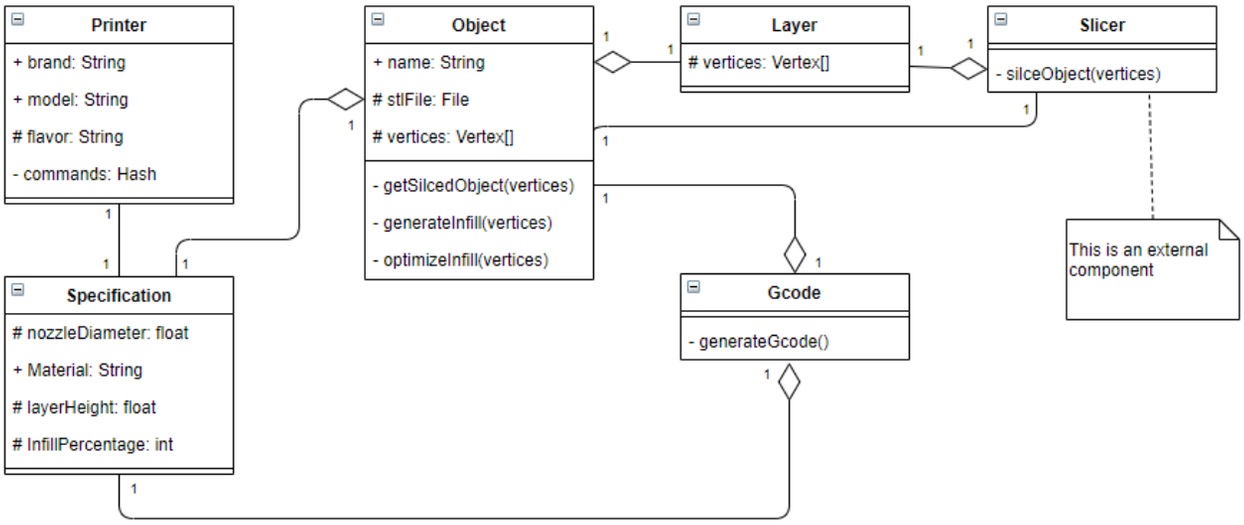

FIGURE 5.2: Class Diagram: Describes the structure of the system in an object-oriented perspective

5.2.3 Components

The main purpose of a component diagram is to show the structural relationships between the components of a system. Component diagrams provide a simplified, high-order view of a large system. Classifying groups of classes into components supports the interchangeability and reuse of code. This diagram documents how these components are composed and how they interact in a system. Here in Figure 5.3 we can observe the different main components of the system as well the relationships between them. Also we can see the interaction we have in the system between the components, as the main interactions are to Slice the object, visualize the object, create the gcode of the ready to print object, and finally the user being able to print the object.

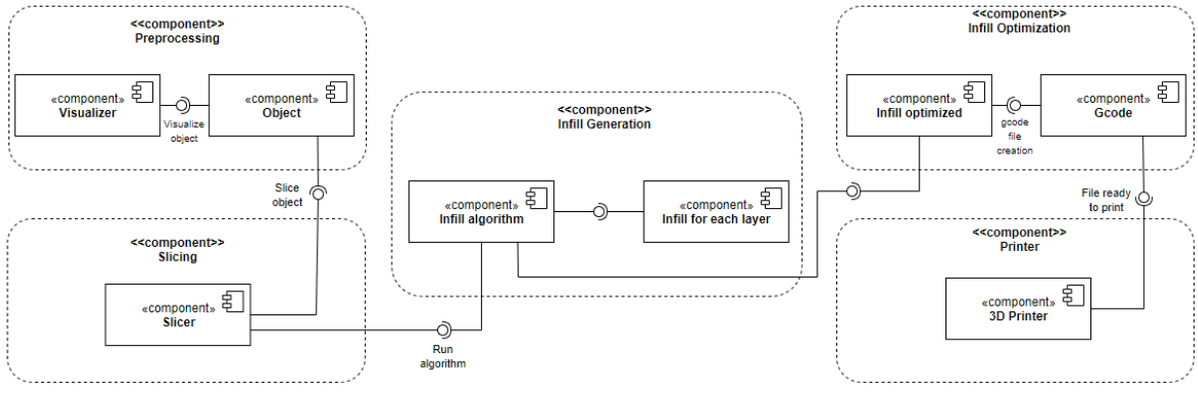

FIGURE 5.3: Component Diagram: Describes the structural relationships between the different components of the system

5.2.4 Activities

The Activity Diagram describes the system activities, or the person who does the activity, and the sequential flow of these activities. The activity diagram is one of the UML diagrams associated with object oriented approach, through it can be used in any other software development paradigm. Here in Figure 5.4 we can see the activity diagram where we describe the sequential flow of all the activities the user can perform in the system. From the beginning by importing the .stl file of the object to the system, and finally getting the gcode file that is ready for being printer in the chosen printer.

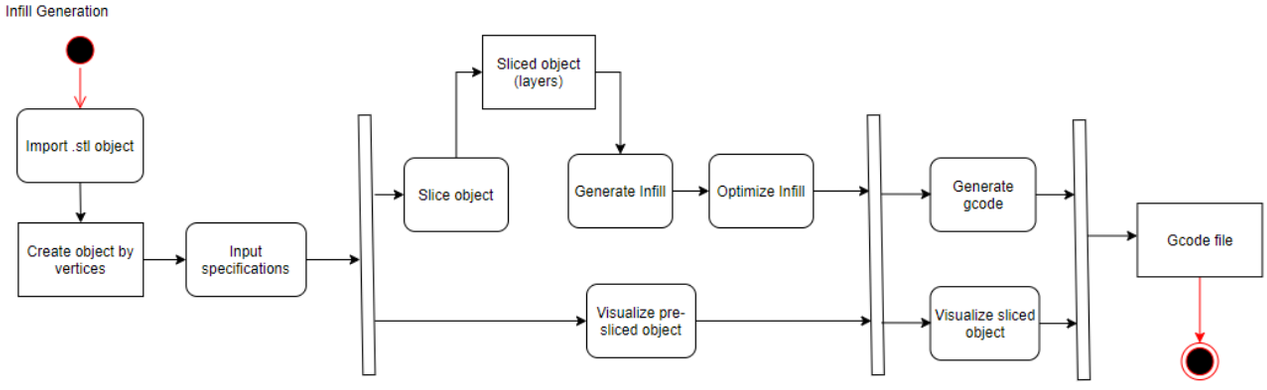

FIGURE 5.4: Activity Diagram: Describes the activities and the sequential flow of such activities that the user should perform to make use of the algorithm

In this section we have seen an explanation of how the system is intended to work, showing the design of the functionalities, the implementation of the system, the flow of the tasks that must be performed prior by the user outside the system, and the ones done inside the system, and how we plan to solve the problem by the creation a new algorithm that generates an alternative type of infill. We used four types of diagrams to show different perspectives of the system and this will allow the reader and the user to have a better understanding of the project.

Chapter 6

Experimental Design

6.1 Introduction

The project is about the optimization of the infills in the technology of Fused Deposition Modeling (FDM) technique in 3D printing. The term infill in additive manufacturing (or 3D printing) refers to the interior structure of an object that is printed. It often has a regular pattern, which is selected by the user in the slicing software, along with a specific volume percentage.

The main objective of the project is to reduce the printing time of a 3D model by creating and testing a new technique of the generation of the infill of a 3D object by creating continuous paths with predefined angles and maintaining the physical properties of the object compared to the techniques/patterns normally used to print.

The project was divided into two set of experiments in order to achieve the main objective of optimizing the printing time by minimizing the sharp turns and having a continuous path. The first set of experiments is to prove that the sharp angles in the infill increase the printing time of an object and to find the angles that affect less the printing time and that are going to be used in the generation of the infill. The second set of experiments is the creation, testing and implementation of an algorithm that creates the infill of an object by using the angles that are obtained from the first experiment and using a continuous path that passes through the areas of an object that was divided according to the infill percentage.

6.2 Angle Experiment

For the first set of experiments there are factors that need to be taken into account. I took into the account the following two factors:

1. The printer used for the tests of the angles. We have two levels, this is according to the number of printers we have access to and use on the experiments.

2. The angles used in the tests to get a range of angles that can be used in the second part of the project, knowing that the sharper the angle the more time it takes to print. The angle is taken into account from the position the printing head is coming, for example 0, 10 or 20 are angles in a straight line, and 70, 80 or 90 have sharp turns. Here we have five different levels chosen by experience of other experiments that are 0, 30, 45, 60 and 90.

There are some blocking factors that are predefined by experience and comparison with other infill techniques, this is the temperature of the printing head (200°C) and printing bed (60°C) of the printer, the speed at which the printing head moves (40mm/s) and the number of lines printed with the different angles (10 layers containing 10 lines each, giving a total of 100 lines). And the nuisance factors are a good calibration of the printer, the correct extrusion of the material and the material stuck in the extrusion chamber. We planned to have a complete factorial design of this experiment, we have the time and the resources (printing material and printers)

We want to find which factors (especially which levels of the factors) are the most significant for the project in order to complete the objective of reducing the printing time of a 3D object. Also we want to minimize the error of printing time (maybe also by giving a final estimation time of the process) in each experiment and each 3D model that is meant to be printed in the final product. This also applies to minimizing the error of the algorithm used to generate the infill of an object; and as a consequence, it gives another result of the reduction of material used in the process. By this we also need for every experiment to be reproducible and with an small standard deviation.

In this section we describe the problem, the design of the experiments and how we plan to solve it by using different patterns that are tested in different printers. The key part is to properly define the angles that we want to use in the experiments, the correct design and generation of the patterns, and the execution of each experiment.

In the design of the experiment for this paper we realized that we needed to design and test different patterns that have different angles. We are considering 13 different angles to be tested between 0 degrees and 90 degrees, being the 0 degrees one the base line for comparison for all the other patterns. The complete set of angles to create the patterns are, and can be seen graphically in figure 6.1:

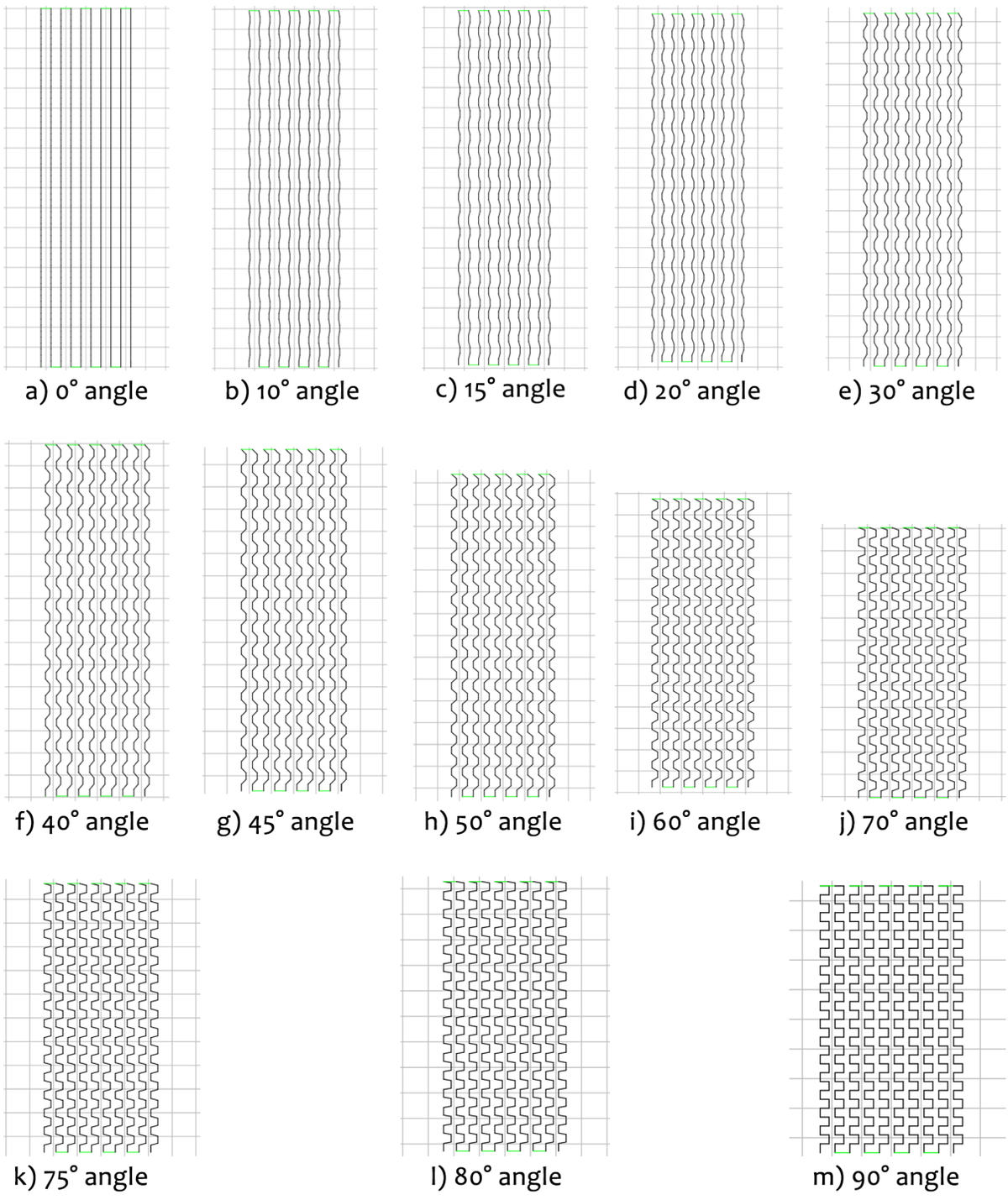

FIGURE 6.1: We show all the patterns created for the first set of experiments, the range of angles is from 0 degrees to 90 degrees, having in total 13 patterns and being the 0 degree the baseline of the experiments.

a Straight line (0 degrees, reference)

b 10 degrees

c 15 degrees

d 20 degrees

e 30 degrees

f 40 degrees

g 45 degrees

h 50 degrees

i 60 degrees

j 70 degrees

k 75 degrees

l 80 degrees

m 90 degrees

This zero degree pattern can be seen as a straight line and it is the fastest pattern that can be printed but sometimes it is not the best suitable for building a complete infill for an object. This is why we are considering this other 12 patterns with different angles. The way each pattern is built is the following: For each pattern we are considering ten parallel lines that are at a 4 millimeter of distance between each other. Then each line is 180 millimeters long and it is built by going 3 millimeters in straight line, the turning in the angle of the pattern, then another straight line and going back in the same angle until it reaches 180 millimeters in length. For each test there will be 10 layers, in each layer there will be the pattern that will be tested, giving as a result 1800 millimeters of material in each layer, and 18000 millimeters of material in each experiment. All of the tests were run at a 40 mm/s (or the instruction F2400 in gcode), this means that the nozzle will move in theory 40 millimeters each second and pouring the material into the printing bed or on top of another material according to the case, we say it is in theory because when it exists a turn, the nozzle will have to positive or negative accelerate but it will try to always be on the velocity specified. In figure 6.1 we can see the different patterns created for this tests, the representation of the results and discussion will be seen in Section 7.

6.3 Infill Generation and Object Printing

In this section, we present the materials and methods proposed to create a new concept of design for the manufacturing processes and giving as a result a new type of algorithm. Figure shows the concept of a tool-path generation approach based on a 3D model design. This approach considers a 3D model in order to generate a toll path using predefined slices and an algorithm chosen to generate a pattern for the infill of the object, having parameters considered for the printer as the nozzle diameter, size of the printing bed, type of material with its melting temperature and temperature of the bed, diameter of the material filament, layer height and infill percentage were the parameters considered for the experiment. For most of this parameters, they were set as blocking factors that are predefined by experience and by comparing them with other infill patterns, they were:

a The temperature of the printing head (200°C)

b The temperature of the printing bed (60°C)

c The speed at which the printing head moves (60mm/s)

d The size of the printing bed. For this experiments the Ender 3 Pro from Creality was used, having a size of the printing bed of 220mm x, 220mm y, 250mm z (x = width, y = depth, z = height)

e The printing nozzle (0.4mm)

f The diameter of the material filament (1.75mm)

g Layer height (0.2mm)

h Infill percentage. This varies from experiment to experiment:

   (a) 5 percentage of infill density

   (b) 10 percentage of infill density

   (c) 20 percentage of infill density

Next we describe the design of the experiments for this section by using different patterns of models that are tested in with different densities of infill. The key part is to properly the models that we want to use in the experiments by covering different possibilities of the contour of the object, the correct design and generation infills, the execution of each experiment, and the measurement of the time.

In the design of the experiment for this section we realized that we needed to design and test different models that have different contours. We are considering 6 different types of models to

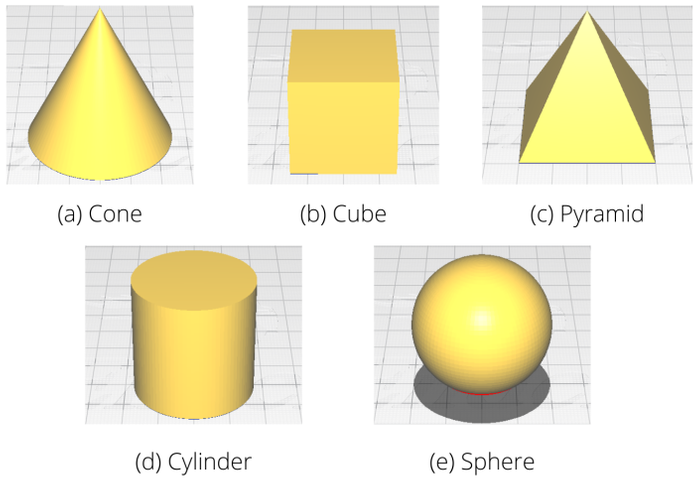

FIGURE 6.2: These are the models viewed in 3D that will be tested with the infill algorithm and measure the base and new printing times.

be tested with infills of 5, 10 and 20 percent, and also printing the same objects with a zig-zag pattern, being the zig-zag pattern the base line for comparison for all the other infills. The complete set of models that will be tested are a cube, pyramid, sphere, cylinder and cone and can be seen graphically in figure 6.2:

According to figure 6.3, we considered two main stages for the design and implementation of the generation of the proposed infill of this research. Geometric modeling deals with the design and usage of a 3D model in a file with the .stl format that the user chooses and the slicing of the 3D model into layers (with a 3rd party code) according to the user specification. And the analytical modeling is the section where the infill is created by processing the model layer by layer using the proposed algorithm, giving as a result a gcode file containing the instructions for the 3D printer to print the 3D model.

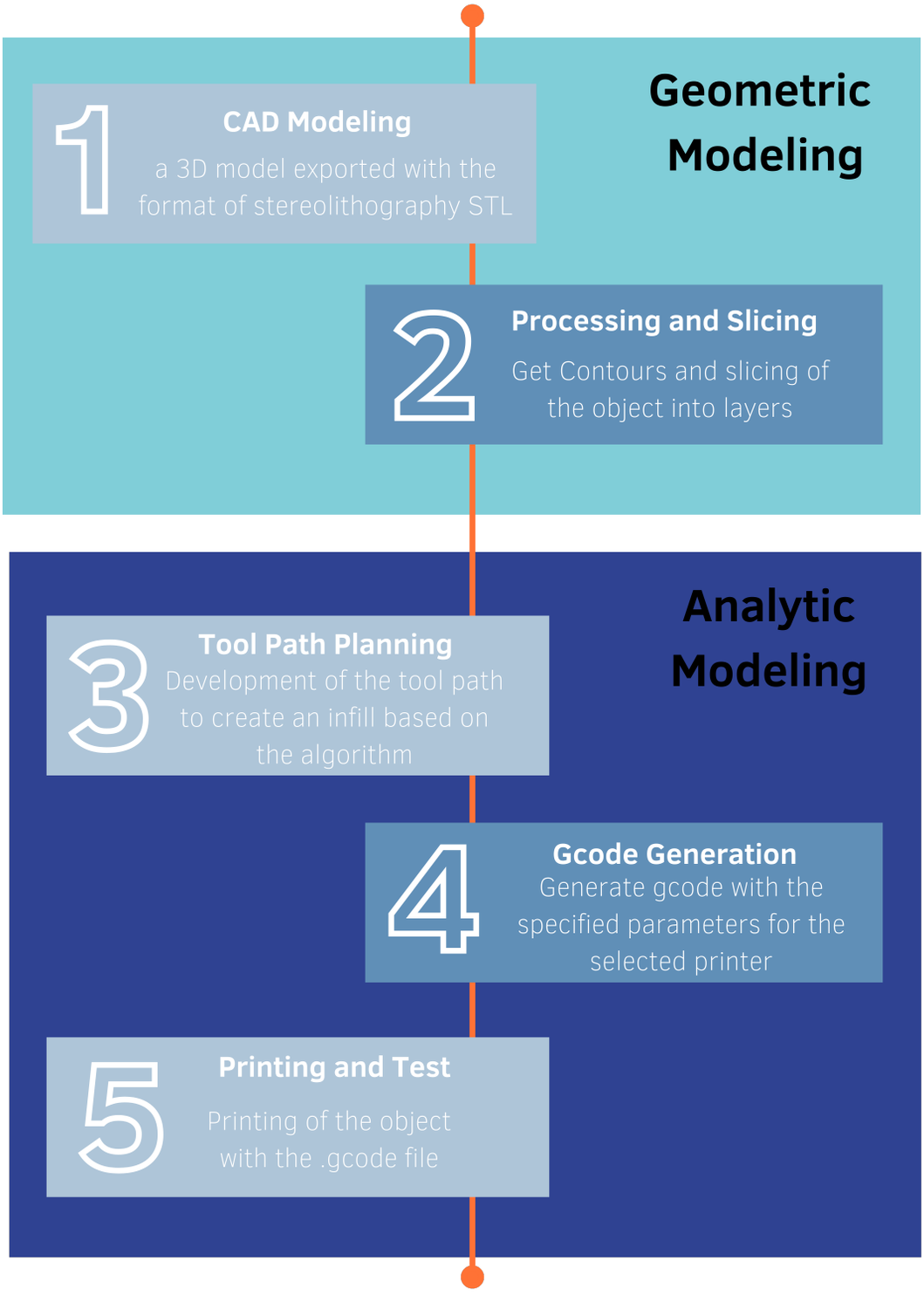

FIGURE 6.3: This is the workflow of the system where a geometric model is selected, sliced and parsed in order to generate an infill to the object with the proposed algorithm.

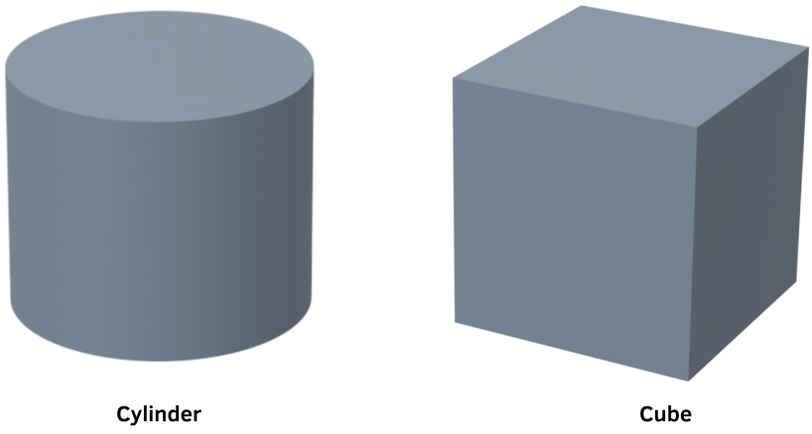

FIGURE 6.4: Here we have two models (cube and cylinder) created for this test, they have dimensions of 40mm x 40mm x 40mm (x,y,z).

6.3.1 CAD models for testing

Generating the tool path based on an STL file format is challenging as vertices are located in a disordered way and demand further processing in order to sort and rearrange and slice a 3D model; essential features such as three-dimensional sizes, segmentation size, and model logic are needed to be identified to create a slicing algorithm which slices the model into several layers, this is why we use a third party slicer like Cura or Slic3r to obtain the contours and slices of the whole 3D model so we can focus only in the generation of the infill inside the contour with the infill density specified by the user. In figure 6.4 we can see an example of two different objects that will be sliced using a third party slicer, creating a hollow object as seen in figure 6.5 (the top was removed for this image to show the inside of the object) that will be processed in order to get the contours of each layer and create an infill inside the object and finally be converted into gcode.

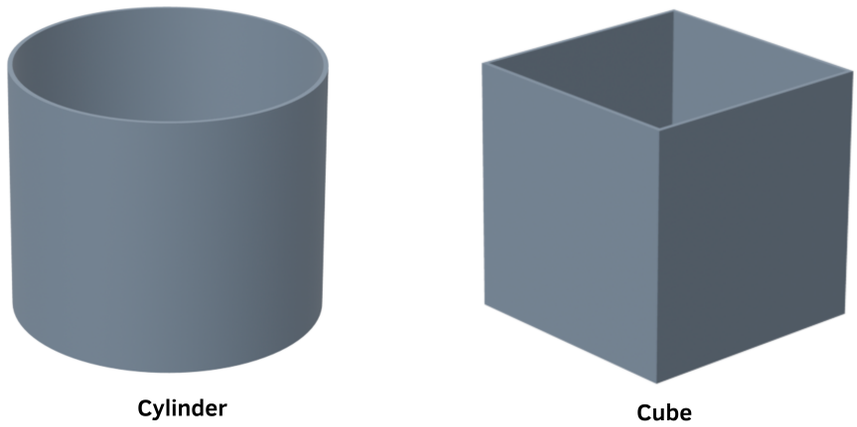

FIGURE 6.5: Here we have the sliced 3D models obtained for this tests, they have dimensions of 40mm x 40mm x 40mm (x,y,z). The top was removed for demonstrative purposes.

The search space is where the PSO algorithm will work to find the best solution using the defined parameters (covering the desired area avoiding or minimizing sharp turns in the infill), is viewed as a grid which can be described by the Cartesian plane. This search space contains small square-shaped cells where the reference for each cell is the center of the cell. Hence, the coordinate of each cell can be described with the x and y points on the Cartesian plane, with the possibility of passing through the center or passing through the cell with an offset from the center of a small percentage previously defined. In order to avoid ambiguous solutions, we assume that the printer moves along the mid-points of the cells from one cell to another in most of the cases.

The results for both sets of experiments are presented in the next chapter 7, in that chapter the results for the experiments are presented as well as the discussion for the results.

Chapter 7

Results and Discussions

7.1 Introduction

The project was divided into two set of experiments in order to achieve the main objective of optimizing the printing time by minimizing the sharp turns and having a continuous path. The first set of experiments is to prove that the sharp angles in the infill increase the printing time of an object and to find the angles that affect less the printing time and that are going to be used in the generation of the infill. The second set of experiments is the creation, testing and implementation of an algorithm that creates the infill of an object by using the angles that are obtained from the first experiment and using a continuous path that passes through the areas of an object that was divided according to the infill percentage.

7.2 Results Angle Experiment

The amount of time and material to print an object are the two of the most important factors in the printing process of a model that translates into the cost of a prototype in additive manufacturing, being the time factor very high in most of the processes in AM. Therefore, there is the growing necessity of printing in a faster way in additive manufacturing to reduce the cost of a printed model in terms of material and time.

The printing problem we face in additive manufacturing is that it takes a lot of time, primarily the infill of an object. This is why it exists the necessity of creating new technologies that minimize the printing time of an object by maintaining its physical properties and giving also as a result, the reduction of material used. In this paper we focused in the problem of the infills of the printed object. The amount of time and material to print an object are the two of the most important factors in the printing process of a model that translates into the cost of a prototype

Results of the Stress Tests for Ultimaker 2+ and Ultimaker 3				
Angle	Ultimaker 2+ [sec]	Ultimaker 3 [sec]	Ultimaker 2+ Δ [%]	Ultimaker 3 Δ [%]
0	466.0	465.247	0 (Reference)	0 (Reference)
10	467.0	465.573	0.2146	0.0701
15	467.0	465.879	0.2146	0.1358
20	467.0	465.956	0.2146	0.1523
30	467.0	466.371	0.2146	0.2415
40	472.0	469.959	1.2876	1.0127
45	474.0	473.267	1.7167	1.7238
50	477.0	476.445	2.3605	2.4068
60	482.0	482.293	3.4335	3.6638
70	488.0	487.477	4.7210	4.7781
75	488.0	488.535	4.7210	5.0055
80	489.0	489.321	4.9356	5.1744
90	491.0	491.787	5.3648	5.7044

TABLE 7.1: Results of first set of experiments in Ultimaker 2+ and Ultimaker 3

in additive manufacturing, being the time factor very high in most of the processes in AM. Therefore, there is the growing necessity of printing in a faster way in additive manufacturing to reduce the cost of a printed model in terms of material and time.

The printers used for testing are the Ultimaker 3, the Ultimaker 2+ and Ender 3 Pro, using the material PLA. We have 13 different tests that contain 10 layer each, and 10 lines of the pattern in each layer. Giving as a total 100 lines of the pattern in each layer. The results for the Ultimaker 3 for this tests are:

In the table 7.1 we can observe the results of running the experiments of the 13 patterns with the previously defined tests. Here in the columns two and three we have the record of the printing time in seconds with the Ultimaker 2+ and Ultimaker 3 respectively. In columns four and five we have the delta time between the fastest segment we can print in each printer (angle of zero degrees that is a straight line) and each angle experiment. We can notice that between the angles 0 and 40 there is a delta time lower than 1[%] compared to the sharper turns that doubles the percentage in the 50 degree angle and it is five times bigger in the 90 degree angle. In the table 7.2 it is identical the arrangement of the table and the results are very similar, with the difference that it is almost double between the 0 and 45 degree, and the 90 degree angle is 6 times bigger. These results are separated because the printers are different types of brands.

We use table 7.1 to draw the graphs in Figures 7.1, 7.2 and we use table 7.2 to draw the graphs in figure 7.3. In these three graphs we can observe a similar behaviour where the time starts to go up significantly in the 40-45 degrees angles. From these we can conclude that having sharp turns in the generation of the infill increases the printing time, thus for the generation of a new infill the algorithm will try to minimize the sharp turns and try to use only angles between

Results of the Stress Tests for Ender 3 Pro		
Angle	Ender 3 Pro [sec]	Ender 3 Pro Δ [%]
0	466.674	0 (Reference)
10	467.157	0.1035
15	467.468	0.1701
20	467.551	0.1879
30	467.963	0.2762
40	471.269	0.9846
45	474.753	1.7318
50	477.974	2.4214
60	484.026	3.7182
70	489.362	4.8416
75	490.223	5.0461
80	491.078	5.2293
90	494.563	5.9761

TABLE 7.2: Results of first set of experiments in Ender 3 Pro

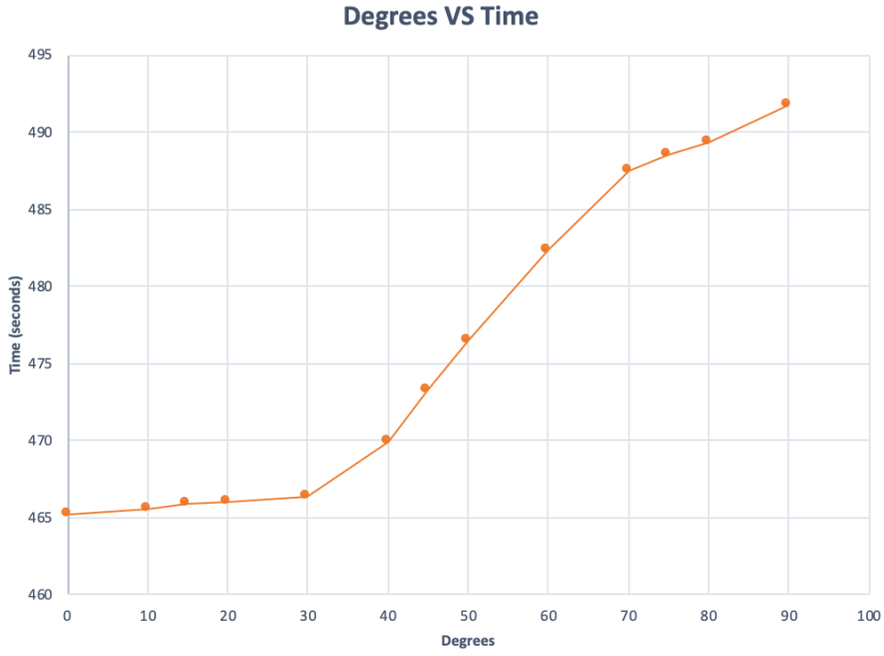

FIGURE 7.1: Comparison of times in the Ultimaker 3 experiment.

0 and 45 degrees, and also having a continuous path to cover the infill area. These tests and results will be shown in the next section 7.3.

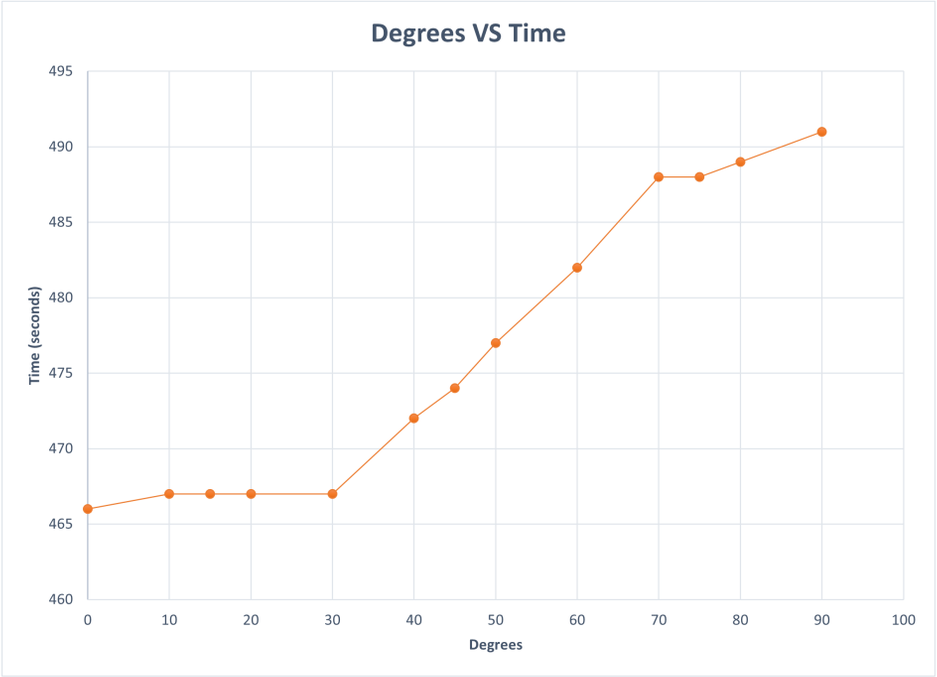

FIGURE 7.2: Comparison of times in the Ultimaker 2+ experiment.

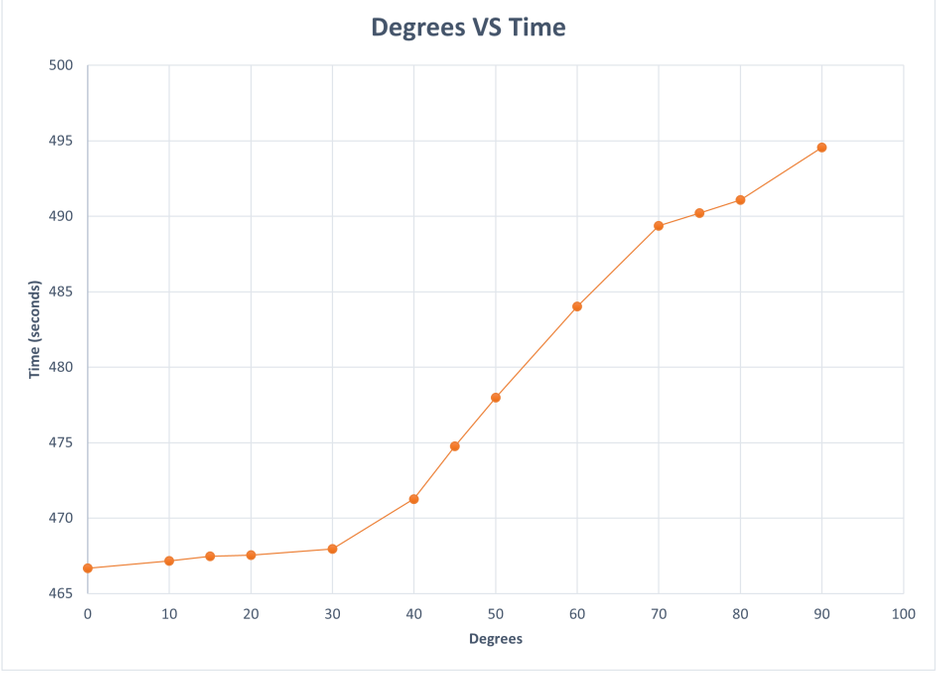

FIGURE 7.3: Comparison of times in the Ender 3 Pro experiment.

7.3 Results Infill Generation and Object Printing

The proposed method can create an infill pattern and a path plan that can be executed in a single continuous move of the processing head in order to print the 3D model. The main concept of the generation of this algorithm is to eliminate the rapid feed movements by minimizing the sharp turns, giving as a result the minimization of acceleration and deceleration, and printing time can be saved. In order to quantify the benefits arising from the proposed method, the results of the proposed case study is presented hereafter. We have five different test specimens that have been considered, this objects are a cube, a cone, a sphere, a pyramid and a cylinder; each specimen is the same size in volume through all the experiments while changing the infill percentage, and the deposition track width has been considered to be equal to 0.4mm. Since different infill geometries could result in different properties and results, we have different types and numbers of samples, giving a better evaluation of the experiments than comparing fewer samples with the same infill percentage or the same model. The first sample will be utilizing a infill of a zig-zag pattern (the most common in most infill generator solutions), while the second one will use the proposed method, all of these samples with different infill percentages. Finite Element Analyses have been used in order to quantify the properties of the samples in order to get the differences in time.

Based on the results of analysis regarding equal parameters with different types of objects and infill density, a comparison has been made between the building times base pattern (zig-zag) and the proposed method. The Infill Areas were computed to this end, and the nozzle diameter was considered to be the same in every case.

In order to simultaneously minimize process time of Additive Manufacturing processes, we proposed a new infill geometry generation and path planning method, able to create the infill layer by layer using a single continuous curve. This method needs to eliminate rapid feed movements and sharp turns made by the processing head during the build-up of every single layer. As another benefit of eliminating time delays occurring by rapid feed movements and sharp turn, the method offers indirectly the advantage of minimizing energy consumption and equipment premature wear that are side effects of the acceleration and deceleration movements associated with these problems.

We show tool path generation results on shapes with varying degrees of concavity. Comparisons are made to conventional zigzag infill pattern in terms of path continuity, amount of sharp turns, print time, as well as visual quality of the interior fill and fabricated surface exterior. The experiments have been conducted on a Ender 3D Pro FDM 3D printer with firmware Marlin. Printing results and analyses are based on the default printer setting, with tool path width set at 0.4mm, layer thickness at 0.2mm, and maximal nozzle speed at 80 mm per second. At the final phase of the generation of the path as seen in Figure 7.4, the G-code is used to

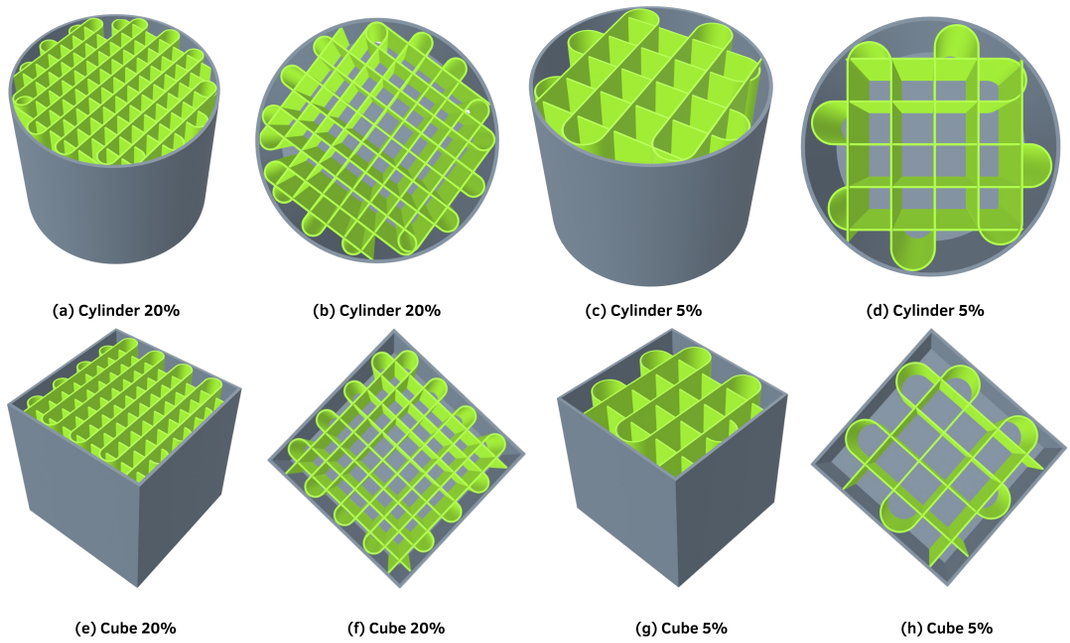

FIGURE 7.4: Visual representation of our infill generated by the proposed algorithm. From (a) to (d) we have the cylinder model in two different perspectives with an infill density of 20% and 5%. From (e) to (h) we have the cube model in two different perspectives with an infill density of 20% and 5%.

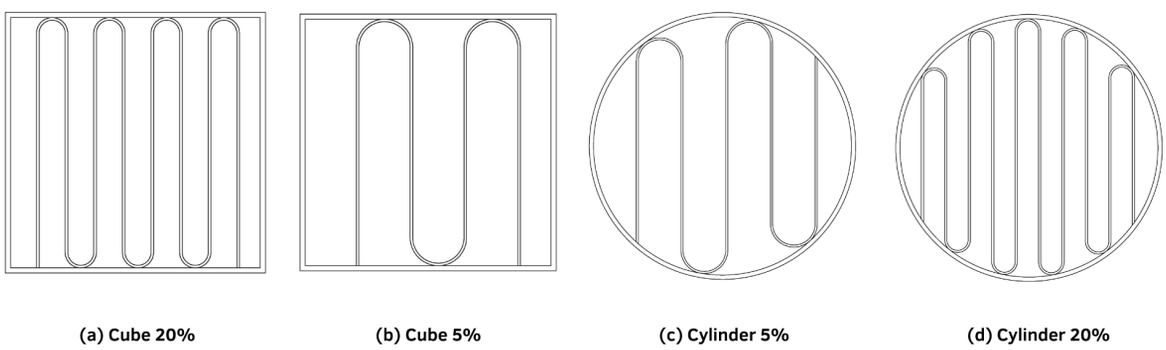

FIGURE 7.5: Visual representation of our infill generated by the proposed algorithm in one layer. In (a) and (b) we have the cube with an infill density of 20% and 5%. In (c) to (d) we have the cube model with an infill density of 20% and 5%.

transfer the tool paths to a file that can be printed in the 3D printer. Figure 7.4 shows the tool paths generated by our algorithm for the cube and cylinder models, having a different infill percentage in this figure, being 20% and 5% in both cases, thus creating a different inner structure. Note that some of the shapes are similar at the end, like cube and pyramid, and sphere, cone and cylinder, where all 2D slices of the 3D objects look similar but are changing in each layer according to the case. Each tool path is continuous and composed of connected paths. In Figure 7.5 we can see a single layer created for a layer with a shape of a square or a circle. All the results are produced with the default parameter settings described in Chapter 6.

There are no parameters that can be changed for the initial construction. Table 7.3, Table 7.4,

Results infill density of 5[%]					
Shape	Zig-Zag [min]	Our Method [min]	Delta Time Δ [%]	Zig-zag	Our Method
Cube	49.56	46.21	7.2495	470 cm	458 cm
Cylinder	37.69	34.87	8.0872	361 cm	352 cm
Pyramid	28.56	26.34	8.4282	205 cm	197 cm
Sphere	32.93	30.25	8.8595	239 cm	230 cm
Cone	26.37	24.43	7.9411	155 cm	146 cm

TABLE 7.3: Results of the infill generation with infill density of 5[%]

Results infill density of 10[%]					
Shape	Zig-Zag [min]	Our Method [min]	Delta Time Δ [%]	Zig-zag	Our Method
Cube	57.32	53.63	6.8805	585 cm	574 cm
Cylinder	45.54	42.32	7.6087	447 cm	439 cm
Pyramid	32.25	29.98	7.5717	244 cm	232 cm
Sphere	36.37	33.91	7.2545	302 cm	295 cm
Cone	27.94	26.23	6.5193	187 cm	174 cm

TABLE 7.4: Results of the infill generation with infill density of 10[%]

Results infill density of 20[%]					
Shape	Zig-Zag [min]	Our Method [min]	Delta Time Δ [%]	Zig-zag	Our Method
Cube	69.32	65.87	5.2376	822 cm	795 cm
Cylinder	54.96	51.87	5.9572	639 cm	615 cm
Pyramid	34.25	32.39	5.7425	318 cm	299 cm
Sphere	42.36	40.24	5.2684	427 cm	404 cm
Cone	29.68	28.38	4.5807	246 cm	221 cm

TABLE 7.5: Results of the infill generation with infill density of 20[%]

and Table 7.5 shows the percentage construction time for the two fill patterns: zigzag and our method. All the zigzag paths shown and fabricated in our experiments were generated with the Cura software with the default printer setting, with tool path width set at 0.4mm, layer thickness at 0.2mm, maximal nozzle speed at 50 mm per second, and the three selected infill densities.

The continuous path construction, connection and optimization algorithm is implemented in Python and the generation of the GCode file is also implemented in Python, and all of these solutions were made with a Intel CoreTM i5-5600K CPU 3.5GHz with 16GB RAM.

Another advantage occurs from infill of the proposed method, this is due to the presence of the path in all the critical points of the structure where each and every one is tangent and parallel to their adjacent neighbours, and so forming a solid structure as a result. This structure can provide the property of making the printed object stiffer depending the infill percentage. In the future, as a result of these tests, the velocity and acceleration should be included in the build time estimation of an object when it is sliced in a slicer software. In addition, the heuristic algorithm used here to optimize the path that will be created for each layer are expected to

Results of t-tests			
Infill Density	t-score	df [min]	p-value
20%	-5.9547	4	0.003992
10%	-7.5998	4	0.001608
5%	-10.635	4	0.0004427

TABLE 7.6: Results of the t-tests in the three different infill densities.

be improved in order to get a better identification of where the elements (cells that will be covered) and for determining the exact path of the head, thus creating a continuous path with minimized sharp turns. The usage of heuristics will also help in addressing a bigger catalog of 3D geometries and being able to achieve 3D infill for more complex objects.

In Table 7.3, Table 7.4, and Table 7.5, we report the percentages of difference in the printing time as a result of minimizing the sharp turns. Each row of the tables, the time for the zig-zag and our method, are the average time of running each test three times. We can observe that the number of sharp turns produced by our infill is lower than the zig-zag infill pattern because the printing time is better from 4.58 to 8.85 percent, giving as a result a lower printing time. We can observe also columns five and six that contain the total amount of material used in the experiments in the zig-zag infill and our proposed method, there is a small difference of material used in the infills from around 10 cm or 2% of difference in the 5% and 10% infill densities, to a difference of around 25 cm or 4%. From this we can conclude that printing the infill with our proposed method after making the comparison corrections of printing time with the material used, we still have a faster printing time between 4 and 8 percent.

In table 7.6 we can observe the t-score, the df and the p-value for each of the infill densities. From this we can observe that we got a maximum p-value of 0.003992, concluding that the hypothesis presented in 1 can be accepted. The calculations of the t-test were made in R, the code and results can be seen in the Appendix A.

Chapter 8

Conclusions and Future Work

We have discussed about the fundamentals of additive manufacturing regarding the different techniques that are used in the industry and focusing in the technique of Fused Deposition Modeling and explaining how it works. We the different key parts of a printed object and how the process is done. We focused on the slicing (from a third party software) and specifically in the infills of a model due to its essential role in Additive Manufacturing, as we want aimed to optimize the infill of a model by printing it as a continuous path and with a small amount of sharp turns.

We reviewed and analysed multiple papers focused in optimizing the infill of a 3D printed object, showing the necessity of creating new algorithms that optimize the infill of an object. This was done by investigating fill patterns with a continuous path, less sharp turns and having consecutive layers. Another factor considered is the start printing point and final printing point in each layer with the purpose of increasing the coherence between each layer, eliminating repetitive movements of the nozzle from one layer to its consecutive layer and that they can generate different infill patterns.

We explained how the system works, showing the design of the functionalities, the implementation of the system, the flow of the tasks performed by the user and the system. We described how we planned to solve the problem by the creation a new algorithm that generates an alternative type of infill, and several diagrams to explain the reader how everything is implemented.

Everything presented in this paper gave a result a new infill pattern that can be printed in a single continuous path of the printing nozzle and with the sharp turns minimized.

This research aimed to identify a problem in Additive Manufacturing that is the printing time, in order to find a possible solution that can help to print objects in a less amount of time. Based on a quantitative and qualitative analysis of the tests made with the algorithm created to generate a new type of infill, it can be concluded that indeed sharp turns and non continuous

paths affect the printing time due to the acceleration and deceleration of the printing head, and the non continuous path that can cause a retraction of the printing head, adding more printing time as a cost to the printing time; and the generation of a new infill that its purpose is to avoid sharp turns and avoid the retraction of the printing head, by not using certain angles and having a continuous path through all the infill. Also by adding an optimization process which its purpose is to find the best start point, the best end point, and the best path to follow, helps to get a better result in some of the cases. Therefore, sharp turns, continuous paths, an optimization process, and also the generation of the gcode with an appropriate flow and velocity, are important factors to consider when designing and testing new algorithms for infills. The results indicate the potential of this type of infill created by our algorithm that can improve the results of printing an object, by reducing the printing time it takes a model to be printed, that is one of the biggest problems in Additive Manufacturing.

Based on these conclusions, anyone who read this thesis should consider the types of objects this new infill was tested on. There there is still work that can be done to improve this part of the process, besides to make the algorithm able to create an infill of an object that has hollows inside the object. Further research is needed to determine the effects of adapting this proposed algorithm to generate n infill in more complex objects that are narrow on some parts, or as mentioned are hollowed inside the object. We also want to consider for future work some constrains or shapes that could affect the structural strength of a printed object, and measure the structural strength to make a comparison with our proposed methods and other infill methods.

The proposed methodology can be applied to create a new infill for objects that will be printed with the FDM technology. However, algorithm works for optimizing one layer at a time. In the future, we would like to investigate the results between fill patterns of consecutive layers, where consecutive layers can't be identical to provide good resiliency to the object.

Appendix A

Code and results for t-tests

A.1 Infill Density 20%

A.1.1 Code

```
datos.zigzag.proposal = data.frame(
Name= c('Cube', 'Cylinder', 'Pyramid', 'Sphere', 'Cone'),
Zigzag= c(69.32,54.96,34.25,42.36,29.68),
Proposal = c(65.87,51.87,32.39,40.24,28.38)
)
var.test(datos.zigzag.proposal$Zigzag,datos.zigzag.proposal$Proposal)
shapiro.test(datos.zigzag.proposal$Zigzag)
shapiro.test(datos.zigzag.proposal$Proposal)
t.test(datos.zigzag.proposal$Proposal,datos.zigzag.proposal$Zigzag, paired = TRUE)
```

A.1.2 Results

```
F test to compare two variances

data: datos.zigzag.proposal$Zigzag and datos.zigzag.proposal$Proposal
F = 1.1167, num df = 4, denom df = 4, p-value = 0.9174
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.1162676 10.7253397
sample estimates:
ratio of variances
1.116696
```

Shapiro-Wilk normality test

data: datos.zigzag.proposal$Zigzag
W = 0.94318, p-value = 0.6885

Shapiro-Wilk normality test

data: datos.zigzag.proposal$Proposal
W = 0.94045, p-value = 0.6691

Paired t-test

data: datos.zigzag.proposal$Proposal and datos.zigzag.proposal$Zigzag
t = -5.9547, df = 4, p-value = 0.003992
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.466238 -1.261762
sample estimates:
mean of the differences
-2.364

A.2 Infill Density 10%

A.2.1 Code

```
datos.zigzag.proposal = data.frame(
Name= c('Cube', 'Cylinder', 'Pyramid', 'Sphere', 'Cone'),
Zigzag= c(57.32,45.54,32.25,36.37,27.94),
Proposal = c(53.63,42.32,29.98,33.91,26.23)
)
var.test(datos.zigzag.proposal$Zigzag,datos.zigzag.proposal$Proposal)
shapiro.test(datos.zigzag.proposal$Zigzag)
shapiro.test(datos.zigzag.proposal$Proposal)
t.test(datos.zigzag.proposal$Proposal,datos.zigzag.proposal$Zigzag, paired = TRUE)
```

A.2.2 Results

F test to compare two variances

data: datos.zigzag.proposal$Zigzag and datos.zigzag.proposal$Proposal
F = 1.1457, num df = 4, denom df = 4, p-value = 0.8983

alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.1192835 11.0035470
sample estimates:
ratio of variances
1.145662

Shapiro-Wilk normality test

data: datos.zigzag.proposal$Zigzag
W = 0.94098, p-value = 0.6729

Shapiro-Wilk normality test

data: datos.zigzag.proposal$Proposal W = 0.93648, p-value = 0.6411

Paired t-test

data: datos.zigzag.proposal$Proposal and datos.zigzag.proposal$Zigzag
t = -7.5998, df = 4, p-value = 0.001608
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.645438 -1.694562
sample estimates:
mean of the differences
-2.67

A.3 Infill Density 5%

A.3.1 Code

```
datos.zigzag.proposal = data.frame(
Name= c('Cube', 'Cylinder', 'Pyramid', 'Sphere', 'Cone'),
Zigzag= c(49.56,37.69,28.56,32.93,26.37),
Proposal = c(46.21,34.87,26.34,30.25,24.43)
)
var.test(datos.zigzag.proposal$Zigzag,datos.zigzag.proposal$Proposal)
shapiro.test(datos.zigzag.proposal$Zigzag)
```

```
shapiro.test(datos.zigzag.proposal$Proposal)
t.test(datos.zigzag.proposal$Proposal,datos.zigzag.proposal$Zigzag, paired = TRUE)
```

A.3.2 Results

```
F test to compare two variances

data: datos.zigzag.proposal$Zigzag and datos.zigzag.proposal$Proposal
F = 1.1252, num df = 4, denom df = 4, p-value = 0.9117
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.1171565 10.8073384
sample estimates:
ratio of variances
1.125233


Shapiro-Wilk normality test

data: datos.zigzag.proposal$Zigzag
W = 0.91422, p-value = 0.4934

Shapiro-Wilk normality test

data: datos.zigzag.proposal$Proposal
W = 0.90623, p-value = 0.4453

Paired t-test

data: datos.zigzag.proposal$Proposal and datos.zigzag.proposal$Zigzag
t = -10.635, df = 4, p-value = 0.0004427
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.281316 -1.922684
sample estimates:
mean of the differences
-2.602
```

Bibliography

[1] Ian Gibson, David W Rosen, and Brent Stucker. Design for additive manufacturing. In *Additive manufacturing technologies*, pages 299–332. Springer, 2010.

[2] Marco Livesu, Stefano Ellero, Jonàs Martínez, Sylvain Lefebvre, and Marco Attene. From 3d models to 3d prints: an overview of the processing pipeline. In *Computer Graphics Forum*, volume 36, pages 537–564. Wiley Online Library, 2017.

[3] Aamir Khan Jadoon, Chenming Wu, Yong-Jin Liu, Ying He, and Charlie CL Wang. Interactive partitioning of 3d models into printable parts. *IEEE computer graphics and applications*, 38(4):38–53, 2018.

[4] Paulo Jorge Bártolo and Ian Gibson. History of stereolithographic processes. In *Stereolithography*, pages 37–56. Springer, 2011.

[5] Jyotirmoy Nandy, Hrushikesh Sarangi, and Seshadev Sahoo. A review on direct metal laser sintering: Process features and microstructure modeling. *Lasers in Manufacturing and Materials Processing*, 6(3):280–316, 2019.

[6] M Prechtl, A Otto, and M Geiger. Laminated object manufacturing of metal foil—process chain and system technology. In *AMST'05 Advanced Manufacturing Systems and Technology*, pages 597–606. Springer, 2005.

[7] Jaroslaw Kotlinski. Mechanical properties of commercial rapid prototyping materials. *Rapid Prototyping Journal*, 2014.

[8] Hugo I Medellin-Castillo and Jorge Zaragoza-Siqueiros. Design and manufacturing strategies for fused deposition modelling in additive manufacturing: A review. *Chinese Journal of Mechanical Engineering*, 32(1):53, 2019.

[9] André Dolenc and Ismo Mäkelä. Slicing procedures for layered manufacturing techniques. *Computer-Aided Design*, 26(2):119–126, 1994.

[10] Juraj Vanek, Jorge A Garcia Galicia, and Bedrich Benes. Clever support: Efficient support structure generation for digital fabrication. In *Computer graphics forum*, volume 33, pages 117–125. Wiley Online Library, 2014.

[11] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[12] Yu-an Jin, Yong He, Jian-zhong Fu, Wen-feng Gan, and Zhi-wei Lin. Optimization of tool-path generation for material extrusion-based additive manufacturing technology. *Additive manufacturing*, 1:32–47, 2014.

[13] Michael R Dunlavey. Efficient polygon-filling algorithms for raster displays. *ACM Transactions on Graphics (TOG)*, 2(4):264–273, 1983.

[14] GQ Jin, Weidong D Li, and L Gao. An adaptive process planning approach of rapid prototyping and manufacturing. *Robotics and Computer-Integrated Manufacturing*, 29(1):23–38, 2013.

[15] YuMing Zhang, Yiwei Chen, Pengjiu Li, and Alan T Male. Weld deposition-based rapid prototyping: a preliminary study. *Journal of Materials Processing Technology*, 135(2-3):347–357, 2003.

[16] Donghong Ding, Zengxi Stephen Pan, Dominic Cuiuri, and Huijun Li. A tool-path generation strategy for wire and arc additive manufacturing. *The international journal of advanced manufacturing technology*, 73(1-4):173–183, 2014.

[17] Mikkel Abrahamsen. Spiral tool paths for high-speed machining of 2d pockets with or without islands. *Journal of Computational Design and Engineering*, 6(1):105–117, 2019.

[18] Alexios Papacharalampopoulos, Harry Bikas, and Panagiotis Stavropoulos. Path planning for the infill of 3d printed parts utilizing hilbert curves. *Procedia Manufacturing*, 21:757–764, 2018.

[19] Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Connected fermat spirals for layered fabrication. *ACM Transactions on Graphics (TOG)*, 35(4):1–10, 2016.

[20] WK Chiu, YC Yeung, and KM Yu. Toolpath generation for layer manufacturing of fractal objects. *Rapid Prototyping Journal*, 2006.

[21] Rajeev Dwivedi and Radovan Kovacevic. Automated torch path planning using polygon subdivision for solid freeform fabrication based on welding. *Journal of Manufacturing Systems*, 23(4):278–291, 2004.

[22] Debananda Misra, V Sundararajan, and Paul K Wright. Zig-zag tool path generation for sculptured surface. *Geometric and Algorithmic Aspects of Computer-Aided Design and Manufacturing*, page 265, 2005.

[23] Y Yang, Han Tong Loh, JYH Fuh, and YG Wang. Equidistant path generation for improving scanning efficiency in layered manufacturing. *Rapid Prototyping Journal*, 2002.

[24] Martin Held and Christian Spielberger. Improved spiral high-speed machining of multiply-connected pockets. *Computer-Aided Design and Applications*, 11(3):346–357, 2014.

[25] M Luo, DH Zhang, BH Wu, and Y Zhang. Optimisation of spiral tool path for five-axis milling of freeform surface blade. *International Journal of Machining and Machinability of Materials*, 8(3-4):266–282, 2010.

[26] Hans Pedersen and Karan Singh. Organic labyrinths and mazes. In *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 79–86, 2006.

[27] Wilhelm Hasselbring. Software architecture: Past, present, future. In *The Essence of Software Engineering*, pages 169–184. Springer, Cham, 2018.

[28] Apostolos Fysikopoulos, Georgios Pastras, Aikaterini Vlachou, and George Chryssolouris. An approach to increase energy efficiency using shutdown and standby machine modes. In *IFIP International Conference on Advances in Production Management Systems*, pages 205–212. Springer, 2014.