



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΙΚΗΣ ΥΛΙΚΩΝ

Τριδιάστατη εκτύπωση υλικών διαβαθμισμένων ιδιοτήτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΦΙΛΙΠΠΙΔΟΥ ΚΩΝΣΤΑΝΤΙΝΑΣ

Επιβλέπων: Κ. Χαριτίδης

Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2019

Ευχαριστίες

Η παρούσα Διπλωματική Εργασία εκπονήθηκε κάτω από την επίβλεψη του Καθηγητή Κ. Χαριτίδη, στην ερευνητική μονάδα Προηγμένων, Σύνθετων, Νάνο Υλικών και Νανοτεχνολογίας (R-NanoLab) του Τομέα Επιστήμης και Τεχνικής των Υλικών της Σχολής Χημικών Μηχανικών του ΕΜΠ.

Θα ήθελα, για αρχή, να ευχαριστήσω θερμά τον Κ. Χαριτίδη για την εμπιστοσύνη που μου έδειξε και μπόρεσα να γίνω μέλος της ερευνητικής του ομάδας για το διάστημα εκπόνησης της Διπλωματικής μου Εργασίας. Η ευκαιρία που μου δόθηκε για να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, που είναι στην επιτομή της τεχνολογίας, είναι τεράστια και ανεκτίμητη. Οι γνώσεις που απέκτησα μέσω της σωστής καθοδήγησης είναι πολλές και δε θα μπορούσα να τις κερδίσω σε τέτοιο βαθμό σε άλλη περίπτωση.

Επίσης, ένα ακόμα άτομο που συνέβαλλε εξαιρετικά στην εκπόνηση της Διπλωματικής Εργασίας και θα ήθελα να την ευχαριστήσω θερμά είναι η Υποψήφια Διδάκτορας Ελένη Γκάρτζου. Η εμπειρία και οι γνώσεις που κατέχει πάνω σε θέματα της τριδιάστατης εκτύπωσης και υλικών απέδειξαν πως ήταν το καταλληλότερο άτομο για συνεργασία πάνω στο συγκεκριμένο θέμα, καθώς ήταν πάντοτε σε θέση να λύσει απορίες και να αντιμετωπίσουμε μαζί επιτυχώς όποια προβλήματα προέκυπταν κατά τη διαδικασία.

Ακόμη, ένα μεγάλο ευχαριστώ θα ήθελα να δώσω σε όλους τους επιστημονικούς συνεργάτες της ομάδας του R-NanoLab που διατηρούσαν πάντα ένα αρμονικό και φιλικό κλίμα συνεργασίας, θετικής ενέργειας που με βοήθησε να ενταχθώ γρήγορα στην ομάδα.

Τέλος, θέλω να ευχαριστήσω θερμά και να αφιερώσω την Διπλωματική μου Εργασία στους γονείς μου Μ. Φιλιππίδη, Φ. Μαλλά, την αδερφή μου Φ. Φιλιππίδου, τον Δ. Καββαθά και στο κοντινό μου φιλικό περιβάλλον και αγαπημένους δασκάλους και καθηγητές που όλα αυτά τα χρόνια μου προσφέρανε ο καθένας με το δικό του λιθαράκι τις γνώσεις και αμέριστη υποστήριξη για να φτάσω ως στο τέλος των σπουδών μου στο Ε.Μ.Π.

Περίληψη

Στη παρούσα ΔΕ έγινε η ανάπτυξη ενός κώδικα για την κατασκευή, μέσω 3D printing, πλεγματικών δομών διαβαθμισμένου πορώδους με προκαθορισμένη κατασκευαστική διαδρομή. Σκοπός της ΔΕ είναι η σύγκριση των πλεγματικών δομών που κατασκευάστηκαν μέσω του κώδικα που αναπτύχθηκε στο εργαστήριο, με αντίστοιχες δομές που κατασκευάζονται μέσω εμπορικών λογισμικών CAD/CAM, μέσω αλγόριθμων παραμετροποίησης και αυτόματου υπολογισμού της βέλτιστης κατασκευαστικής διαδρομής. Η αξιολόγηση έγινε μέσω της οπτικής παρατήρησης εκτυπωμένων δειγμάτων από τις διαφορετικές μεθόδους στο οπτικό μικροσκόπιο και σύγκρισης των γεωμετρικών τους χαρακτηριστικών με την ονομαστική γεωμετρία. Ο κώδικας που αναπτύχθηκε στο εργαστήριο (custom κώδικας) έγινε με χρήση γλώσσας προγραμματισμού Python και στη συνέχεια έγινε μετατροπή σε G-κώδικα, που είναι η γλώσσα λειτουργίας του τριδιάστατου εκτυπωτή. Το υλικό κατασκευής ήταν το θερμοπλαστικό πολυμερές πολυ(γαλακτικό οξύ) (poly-lactic acid - PLA) σε μορφή νήματος με σταθερή διάμετρο. Εκτυπώθηκαν δομές ορθογωνικές με τετραγωνικό πορώδες και δομές κυκλικές με σχηματισμό πορώδους από τα σημεία τομής ομόκεντρων κύκλων με τις διαμέτρους τους. Οι πλεγματικές αυτές δομές αποτελούνταν από μεμονωμένες ίνες, οι οποίες εναποτίθενται επιλεκτικά και συγκολλούνται κατά τη διάρκεια της εκτύπωσης. Οι παράμετροι που λήφθηκαν υπόψιν για τη κατασκευή του κώδικα είναι η ταχύτητα εκτύπωσης F (mm/min) και η ποσότητα υλικού που εξωθείται από τη κεφαλή E (mm), τα οποία ορίζουν την ογκομετρική παροχή. Για τον υπολογισμό της ποσότητας υλικού που πρέπει να εξωθηθεί σε κάθε βήμα, θεωρήθηκε σε πρώτη προσέγγιση ότι ο όγκος του υλικού που εισέρχεται ισούται με τον όγκο του υλικού που εξέρχεται σε κατάσταση τήγματος από το ακροφύσιο και η ροή είναι ομαλή και ισοθερμοκρασιακή χωρίς φαινόμενα διάγκωσης. Επίσης, ήταν απαραίτητο να καθοριστεί ένα μοντέλο για το σχήμα της διατομής της ίνας όταν εναποτίθεται πάνω στην επιφάνεια που εκτυπώνεται. Για αυτό το λόγο θεωρήθηκε πως το σχήμα της ίνας κατά την εξώθηση από την κεφαλή είναι το άθροισμα ενός ορθογωνίου και δύο ημικυκλίων. Με δεδομένες τιμές για το υψος και πάχος της ίνας και με σταθερή διάμετρο ακροφυσίου και διατομή του νήματος εκτύπωσης, έγιναν οι κατάλληλοι υπολογισμοί για τον υπολογισμό της παραμέτρου E σε κάθε βήμα. Από τις ορθογωνικές δομές κατασκευάστηκαν δύο τύποι με διαφοροποίηση στο μέγεθος του πορώδους. Στη κυκλική δομή κατασκευάστηκαν επίσης δύο τύποι δομών από custom κώδικα, με διαφοροποίηση στο είδος της διαδρομής και στο μοντέλο κατασκευής των κύκλων. Εφόσον οι αναγκαίες παράμετροι ορίστηκαν και εκτυπώθηκαν τα δείγματα, ακολούθησε η επεξεργασία τους. Όλες οι δομές παρατηρήθηκαν στο οπτικό μικροσκόπιο με σκοπό την αξιολόγηση της μορφολογίας των ινών, των σημείων σύνδεσης μεταξύ γειτονικών πόρων, αλλά και τη μέτρηση του πλάτους των ινών. Τα αποτελέσματα έδειξαν πως η ορθογωνική διατομή με το μεγαλύτερο μέγεθος στο πορώδες προερχόμενη από τον custom κώδικα είχε τη καλύτερη προσέγγιση στο επιθυμητό αποτέλεσμα. Τόσο η μορφή των ινών, τα σημεία σύνδεσης των πόρων όσο και το πλάτος τους δίνουν ικανοποιητικά αποτελέσματα. Τέλος, το πλάτος των ινών σε αυτή τη δομή βρέθηκε ότι έχει τη μικρότερη ποσοστιαία απόκλιση από την επιθυμητή τιμή.

Abstract

The report of this thesis is concerning the development of a coded program for the construction of mesh structures with graded porosity with pre-determined construction route via 3D printing technology. The main target of the thesis is the comparison of these mesh structures, which were constructed through the coded program that was developed in the laboratory, with the corresponding structures, which were constructed with commercial CAD/CAM software, through parameterization algorithms and automatic calculation of the optimal construction path. Evaluation of these structures was integrated through the visual observation of the printed structures coming from the different printed methods in the optical microscope and comparing their geometrical characteristics with the nominal geometry. The code that was developed in the laboratory (custom code) was made using Python programming language and later was converted to G-code, which is the 3D printer's operating language. The construction material was the thermoplastic polymer (lactic acid) or just poly-lactic acid (PLA) in the form of a fixed diameter filament. Rectangular structures with square shaped porosity and circular structures with the porous formation designed by the intersection of concentric circles with their diameters. These meshed structures consist of individual paths, which are selectively deposited and bonded during printing. The parameters taken into account for the construction of the code are the print speed F (mm/min) and the amount of material extruded from the printer's head E (mm), which determine the volumetric flow rate. In order to calculate the amount of extrusion of the material at each step, it was first assumed that the volume of material entering is equal to the volume of material exiting the melt nozzle and the flow is smooth and isothermal with no swelling effects. In addition, it was also necessary to define a pattern for the cross-section shape of the path when deposited on the printed surface. For that reason, it was assumed that the shape of the path during extrusion from the head is the sum of a rectangle and two semicircles. With given values for the path height, the thickness, the fixed nozzle diameter and the cross-section of the print thread, the appropriate calculations were made to calculate parameter E at each step. From the rectangular structures, two types were constructed, varying in the size of the porosity. As far as it concerns the circular structure, two types of custom code structures were constructed, using different type of the printer's head route and in the modelization of the circular construction. After the necessary parameters were set and the samples were printed, the process of their evaluation was followed. All structures were observed under the optical microscope to evaluate path morphology, adhesion / intersecting points between adjacent pores, and to measure the path width. The results showed that the rectangular structure with the largest group of porosity derived from the custom code, with no retraction, had the best approximation to the desired path width result. Both the shape of the paths, the bonding point of the pores and their width give good results. Finally, the path width in this structure was found to have the smallest percentage deviation from the desired value

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	6
1 ΤΡΙΔΙΑΣΤΑΤΗ ΕΚΤΥΠΩΣΗ – 3D PRINTING	8
1.1 ΟΡΙΣΜΟΣ ΚΑΙ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ	8
1.2.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΠΡΟΣΘΕΤΙΚΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	8
1.2.2 ΔΙΕΡΓΑΣΙΕΣ ΠΡΟΣΘΕΤΙΚΗΣ ΚΑΤΑΣΚΕΥΗΣ	8
1.2.3 ΕΦΑΡΜΟΓΕΣ & ΠΑΡΑΔΕΙΓΜΑΤΑ	11
2 ΥΛΙΚΑ ΔΙΑΒΑΘΜΙΣΜΕΝΩΝ ΙΔΙΟΤΗΤΩΝ (FGMs).....	14
2.1 ΟΡΙΣΜΟΣ.....	14
2.2 ΕΙΔΗ FGMs.....	14
2.2.1 ΧΗΜΙΚΗ ΣΥΣΤΑΣΗ ΤΩΝ FGMs.....	14
2.2.2 ΔΙΑΒΑΘΜΙΣΗ ΜΕ ΒΑΣΗ ΤΟ ΠΟΡΩΔΕΣ ΣΤΑ FGMs.....	15
2.2.3 ΔΙΑΒΑΘΜΙΣΗ ΜΕ ΒΑΣΗ ΤΗ ΜΙΚΡΟΔΟΜΗ ΣΤΑ FGMs.....	15
2.3 ΜΕΘΟΔΟΙ ΠΑΡΑΓΩΓΗΣ.....	15
2.3.1 ΠΑΡΑΓΩΓΗ ΜΕ ΦΥΣΙΚΗ ΕΝΑΠΟΘΕΣΗ ΑΤΜΩΝ (PVD).....	16
2.3.2 ΧΗΜΙΚΗ ΕΝΑΠΟΘΕΣΗ ΑΤΜΩΝ (CHEMICAL VAPOR DEPOSITION - CVD).....	17
2.3.3 ΜΕΤΑΛΛΟΥΡΓΙΑ ΣΚΟΝΗΣ (POWDER METALLURGY)	17
2.3.4 ΦΥΓΟΚΕΝΤΡΗΣΗ ΜΕ ΧΥΤΕΥΣΗ (CENTRIFUGAL CASTING METHOD).....	18
2.4 ΠΡΟΣΘΕΤΙΚΗ ΚΑΤΑΣΚΕΥΗ (ΠΚ) FGMs.....	18
2.4.1 ΟΜΟΓΕΝΗΣ ΣΥΝΘΕΣΗ	18
2.4.2 ΕΤΕΡΟΓΕΝΗΣ ΣΥΝΘΕΣΗ	19
2.4.3 ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΡΟΣΘΕΤΙΚΗΣ ΚΑΤΑΣΚΕΥΗΣ ΓΙΑ FGAM	19
2.4.4 ΕΝΕΡΓΕΙΑΚΑ ΚΑΤΕΥΘΥΝΟΜΕΝΗ ΕΝΑΠΟΘΕΣΗ (DIRECTED-ENERGY DEPOSITION -DED)	
21	
2.4.5 ΣΥΓΚΟΛΛΗΣΗ ΕΛΑΣΜΑΤΩΝ (SHEET LAMINATION)	21
2.4.6 ΨΕΚΑΣΜΟΣ ΥΛΙΚΟΥ (MATERIAL JETTING – MJ)	22
2.4.7 ΠΡΟΚΛΗΣΕΙΣ ΓΙΑ ΤΗ ΠΡΟΣΘΕΤΙΚΗ ΚΑΤΑΣΚΕΥΗ ΣΕ ΣΧΕΣΗ ΜΕ ΤΑ FGMs.....	22
3 ΚΑΤΑΣΚΕΥΗ FGM s ΜΕΣΩ ΕΠΙΛΕΚΤΙΚΗΣ ΕΝΑΠΟΘΕΣΗΣ ΥΛΙΚΟΥ	24
3.1 ΒΑΣΙΚΕΣ ΠΑΡΑΜΕΤΡΟΙ ΤΡΙΣΔΙΑΣΤΑΤΗΣ ΕΚΤΥΠΩΣΗΣ ΠΛΕΓΜΑΤΙΚΩΝ ΔΟΜΩΝ.....	24
3.2 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΤΕΧΝΙΚΗΣ ΕΝΑΠΟΘΕΣΗΣ ΥΛΙΚΟΥ- FUSED FILAMENT FABRICATION (FFF) .	25
3.2.1 SLICING.....	27
4 ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ	31
4.1 ΣΥΝΤΑΞΗ ΚΩΔΙΚΑ ΓΙΑ ΤΗΝ ΕΞΑΓΩΓΗ G-CODE	31
4.1.1 ΚΑΤΑΣΚΕΥΗ ΟΡΘΟΓΩΝΙΚΗΣ ΔΟΜΗΣ.....	31
4.1.2 ΚΑΤΑΣΚΕΥΗ ΚΥΚΛΙΚΗΣ ΔΟΜΗΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΞΩΝ	43

4.1.3	ΚΑΤΑΣΚΕΥΗ ΚΥΚΛΙΚΗΣ ΔΟΜΗΣ ΜΕ ΓΡΑΜΜΙΚΟΠΟΙΗΣΗ ΤΩΝ ΤΟΞΩΝ	50
4.2	ΣΧΕΔΙΑΣΜΟΣ ΜΕΣΩ CAD ΚΑΙ ΕΞΑΓΩΓΗ G-CODE ΜΕΣΩ SLICING	52
4.3	ΣΥΓΚΡΙΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΔΥΟ ΜΕΘΟΔΩΝ	53
4.4	ΠΕΙΡΑΜΑΤΙΚΟΣ ΕΞΟΠΛΙΣΜΟΣ.....	54
5	ΑΠΟΤΕΛΕΣΜΑΤΑ - ΣΧΟΛΙΑΣΜΟΣ	55
5.1	ΕΚΤΥΠΩΜΕΝΑ ΔΕΙΓΜΑΤΑ – ΠΑΡΑΤΗΡΗΣΗ ΜΕ ΤΟ ΟΠΤΙΚΟ ΜΙΚΡΟΣΚΟΠΙΟ	55
5.1.1	ΣΥΓΚΡΙΣΗ ΔΟΚΙΜΙΩΝ ΟΡΘΟΓΩΝΙΚΗΣ ΔΟΜΗΣ ΜΕΤΑΞΥ ΜΕΘΟΔΩΝ ΚΑΤΑΣΚΕΥΗΣ ΜΕ CUSTOM ΚΩΔΙΚΑ ΚΑΙ CAD ΜΟΝΤΕΛΟΥ	55
5.2	ΣΥΓΚΡΙΣΗ ΔΟΚΙΜΙΩΝ ΚΥΚΛΙΚΗΣ ΔΟΜΗΣ ΜΕΤΑΞΥ ΜΕΘΟΔΩΝ ΚΑΤΑΣΚΕΥΗΣ ΜΕ CUSTOM ΚΩΔΙΚΑ ΚΑΙ CAM ΜΟΝΤΕΛΟΥ.....	66
6	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	70
6.1	ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ ΚΑΙ ΠΕΡΑΙΤΕΡΩ ΔΙΕΡΕΥΝΗΣΗ	71
7	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	73
8	ΥΠΟΜΝΗΜΑ	77

1 ΤΡΙΔΙΑΣΤΑΤΗ ΕΚΤΥΠΩΣΗ – 3D PRINTING

1.1 ΟΡΙΣΜΟΣ ΚΑΙ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Η τριδιάστατη εκτύπωση (3D printing), είναι μια σειρά προηγμένων κατασκευαστικών τεχνολογιών που οδηγούν στη κατασκευή ενός φυσικού αντικειμένου απευθείας από ένα 3D ψηφιακό μοντέλο. Οι τεχνολογίες τριδιάστατης εκτύπωσης μπορεί να χρησιμοποιήσει ποικίλα υλικά, κατασκευάζοντας στρώμα-προς-στρώμα (layer-by-layer), για τη δημιουργία ενός ή πολλών αντικειμένων. Αυτή η πολυστρωματική εναπόθεση υλικού προσδίδει ένα μεγάλο αριθμό πλεονεκτημάτων, συμπεριλαμβανομένου και της ελευθερίας σχηματισμού πολύπλοκων δομών, τη χρήση σχεδόν 100% του υλικού και το μικρό χρόνο παραμονής [1]. Το 3D printing έχει μια ξεχωριστή 1.2 φιλοσοφία κατασκευής που επιτρέπει την ευελιξία κατασκευής περίπλοκων δομών που είναι δύσκολο να υλοποιηθούν χρησιμοποιώντας συμβατικές κατασκευαστικές μεθόδους, όπως η χύτευση και η μηχανική κατεργασία. Παράλληλα, επιτρέπει την παραγωγή αντικειμένων διαφορετικών χαρακτηριστικών/γεωμετρίας στην ίδια παρτίδα παραγωγής. Τα πλεονεκτήματα αυτά είχαν ως αποτέλεσμα, από τη δεκαετία του '80 έως σήμερα, η τριδιάστατη εκτύπωση να κεντρίσει τη προσοχή και το ενδιαφέρον της επιστημονικής και μηχανικής κοινότητας [2]. Η τριδιάστατη εκτύπωση είναι συνώνυμη με τον όρο Προσθετική Κατασκευή (ΠΚ) ή αλλιώς Additive Manufacturing (AM).

1.2.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΠΡΟΣΘΕΤΙΚΗΣ ΚΑΤΑΣΚΕΥΗΣ

Μερικά οφέλη της ΠΚ μπορούν να συνοψιστούν ως εξής [3]:

- Άμεση μετατροπή ενός σχεδίου σε φυσικό αντικείμενο
- Δημιουργία αντικειμένων με καλύτερη προσαρμογή χωρίς περαιτέρω επεξεργασία ή κατασκευαστικό κόστος
- Λειτουργικός σχεδιασμός που επιτρέπει τη κατασκευή πολύπλοκων εσωτερικών χαρακτηριστικών
- Ευέλικτα και ελαφριά αντικείμενα με κοίλες ή πλεγματικές δομές
- Δυνατότητα άμεσης κατασκευής ενός αντικειμένου στο τελικό ή σχεδόν τελικό σχήμα με ελάχιστη ή σχεδόν καθόλου επιπρόσθετη επεξεργασία
- Ελάχιστη έως και μηδενική απώλεια πρώτων υλών διότι γίνεται η μέγιστη αξιοποίηση του κατασκευαστικού υλικού
- Σημαντική μείωση του συνολικού χρόνου ανάπτυξης και κατασκευής με αποτέλεσμα να είναι διαθέσιμο συντομότερα στο εμπόριο
- Μικρότερο λειτουργικό foot-print για τη κατασκευή μιας μεγάλης ποικιλίας αντικειμένων
- Δυνατότητα παραγωγής κατά παραγγελία και προσαρμογές στις εκάστοτε συνθήκες

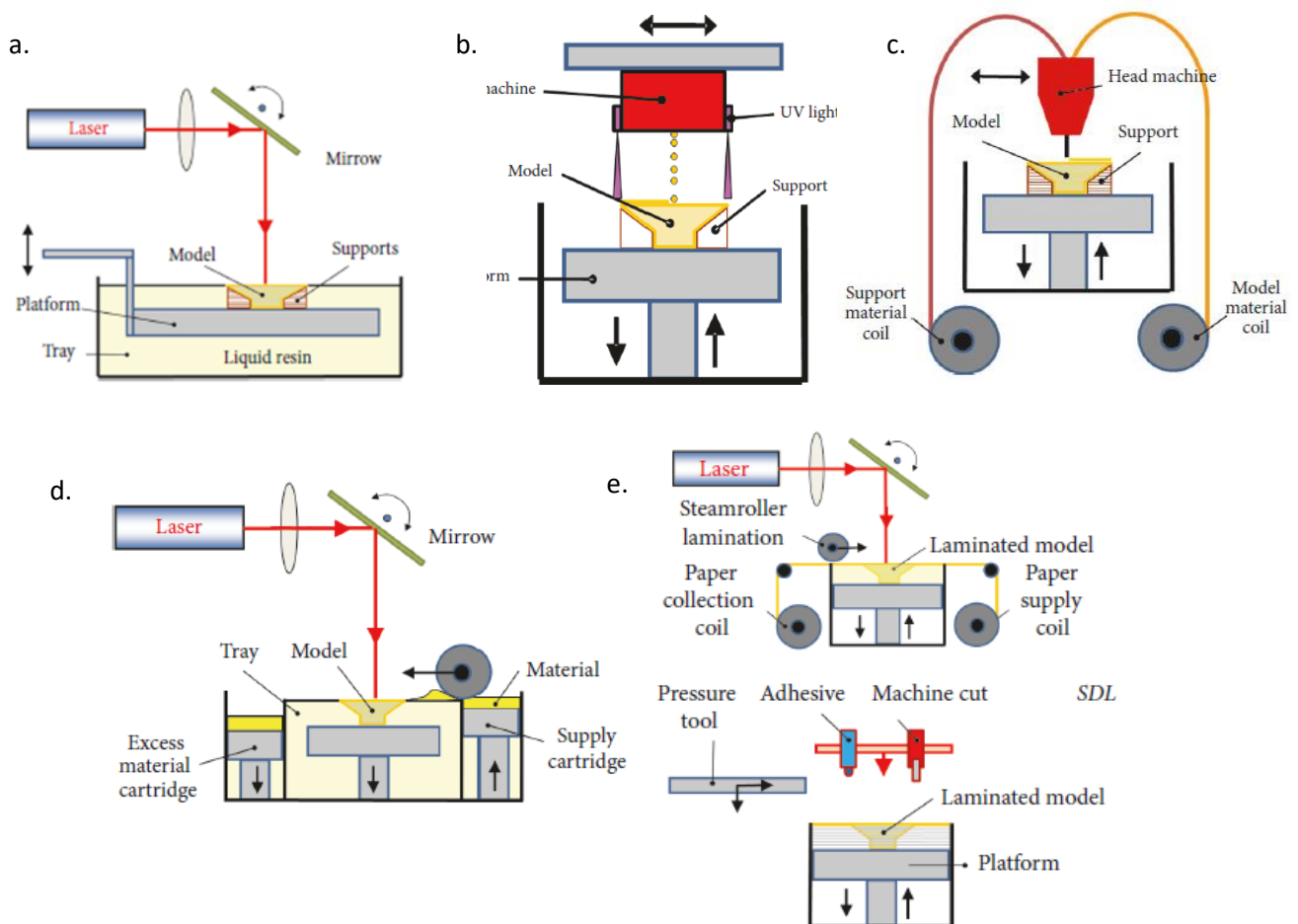
1.2.2 ΔΙΕΡΓΑΣΙΕΣ ΠΡΟΣΘΕΤΙΚΗΣ ΚΑΤΑΣΚΕΥΗΣ

Σύμφωνα με το Διεθνή Οργανισμό Τυποποίησης (International Organization for Standardization – ISO) και το ASTM 52900:2015 πρότυπο, οι AM διεργασίες διαχωρίζονται στις εξής κατηγορίες[3]:

1. Binder Jetting (BJ)
2. Directed Energy Deposition (DED)
3. Material Extrusion (ME)
4. Material Jetting (MJ)
5. Powder Bed Fusion (PBF)
6. Sheet Lamination (SL)
7. Vat Photopolymerization (VP)

Με τη διαδικασία της ΠΚ μπορούν να εκτυπωθούν υλικά από πολυμερή ή με πολυμερική βάση, μέταλλα, κράματα μετάλλων, σύνθετα υλικά, κεραμικά ακόμη και βιοϋλικά. Στη περίπτωση των θερμοπλαστικών υλικών, η εναπόθεση υλικού στο εκάστοτε στρώμα έχει ως βασική αρχή τη τήξη του υλικού. Με αυτό το τρόπο η εναπόθεση του τήγματος μπορεί να είναι κατευθυνόμενη και ελεγχόμενη, ταυτοχρόνως, η ήπια θέρμανση του υλικού δεν απαιτεί υψηλές απαιτήσεις συνθηκών λειτουργίας, δηλαδή την απαίτηση κενού ή ροής αδρανούς αερίου [3]. Αντίστοιχα, στα μέταλλα και στα κεραμικά είναι ιδιαίτερα συνήθης η χρήση κάποιου λέιζερ ή δέσμης ηλεκτρονίων για το σχηματισμό του τελικού αντικειμένου[4]–[6]. Επίσης, για κατασκευή μεταλλικών υλικών και κεραμικών είναι συνήθης και η χρήση συνδεδετικών υλικών και ρευστοποιητών σε πούδρα/σκόνη του υλικού αλλά και η χρήση πηγής διάδοσης μικροκυμάτων έχει αρχίσει να γίνεται τελευταία περισσότερο γνωστή και διαδομένη[3].

Από τη πρώτη πατέντα που ήρθε στην επιφάνεια το 1986 και τον πρώτο 3D printer (βασισμένο στη τεχνολογία της στερεολιθογραφίας) [7] να καταστρώνεται στα τέλη της δεκαετίας του 80, η πρόσβαση της βιομηχανίας της ΠΚ στο εμπόριο, έχει αυξηθεί σημαντικά τα τελευταία χρόνια (περίπου 1 δις στις ΗΠΑ έως το 1997). Παράλληλα, η ΠΚ έχει μετατραπεί από τη ταχεία προτυποποίηση στην παραγωγή λειτουργικών αντικειμένων. Τη σήμερα ημέρα, η ΠΚ χρησιμοποιείται σε όλους τους τομείς της βιομηχανίας από τη διαστημική έως τα παιχνίδια και τα τρόφιμα, και εκπροσωπεί μια βιομηχανία πολλών δισεκατομμυρίων. Πλέον υπάρχει η δυνατότητα αγοράς συστημάτων ΠΚ χαμηλού κόστους, τα οποία είναι διαθέσιμα πλέον σε μια τιμή της τάξης των 500\$ σε αντίθεση με τη δεκαετία του '90, όπου το κόστος ήταν πολύ υψηλότερο (περίπου 100,000\$)[3]. Στην Εικόνα 1 απεικονίζονται σχηματικά κάποιες διαδομένες τεχνολογίες ΠΚ [8]:



Εικόνα 1: Εκτύπωση με σκλήρυνση πολυμερούς μέσω πηγής ακτινοβολίας- α. Stereolithography, b. Resin Injection & Ultraviolet Light Photopolymerization, c. Εξώθηση τηκόμενου υλικού - Fused Deposition Modelling (FDM), Εκτύπωση με επιλεκτική εναπόθεση θερμικής ενέργειας σε κόκκους υλικού- d. Selective Laser Sintering, e. Selective Deposition Lamination

Πίνακας 1: Παρουσίαση ειδών προσθετικής κατασκευής με επεξηγήσεις για την αρχή λειτουργίας τους, διεθνείς ονομασίες, πλεονεκτήματα και μειονεκτήματα, υλικά επεξεργασίας, κατασκευαστικά όρια ως προς το μέγεθος αλλά και εμπορικά κατασκευαστικά μηχανήματα.

ASTM κατηγορία	Αρχή λειτουργίας	Παραδείγματα τεχνολογιών	Πλεονεκτήματα	Μειονεκτήματα	Υλικά	Κατασκευαστικός όγκος (mm)	Κατασκευαστικά μηχανήματα/Χώρες
Binder Jetting	Υγρό συνδετικό υλικό ψεκάζεται σε λεπτά στρώματα κόκκων. Το αντικείμενο κατασκευάζεται στρωματικά κολλώντας τα σωματίδια μεταξύ τους	3D inkjet τεχνολογία	-Δεν απαιτείται κάποιο υπόστρωμα ή βάση -Υπάρχει μεγάλη ευελιξία στο σχεδιασμό -Μεγάλος όγκος υλικού και υψηλή ταχύτητα εκτύπωσης -Σχετικά χαμηλό κόστος	-Εύθραυστα αποτελέσματα με περιορισμένες μηχανικές ιδιότητες -Πιθανή η μετεπεξεργασία	Πολυμερή, Κεραμικά, Σύνθετα, Μέταλλα, Υβρίδια	X <= 4000 Y <= 2000 Z <= 1000	ExOne, USA PolyPico, Ireland
Directed Energy Deposition	Εστιασμένη θερμική ενέργεια σε μεταλλικά υλικά κατά την εναπόθεση	Laser Deposition, Laser Engineered NetShaping, Electron beam, Plasma arc melting	-Υψηλόβαθμος έλεγχος της δομής που προέρχεται από τους κόκκους υλικού - Υψηλής ποιότητας κομμάτια -Εξαιρετη τεχνική για επισκευές	-Η ποιότητα της επιφάνειας και η ταχύτητα απαιτούν μια εξισορρόπηση -Περιορίζεται σε υβρίδια μετάλλων	Μέταλλα, Υβρίδια	X = 600-3000 Y = 500-3500 Z = 350 - 5000	Optomec, USA InssTek, USA Sciaky, USA Irepa Laser, France Trumpf, Germany
Material Extrusion	Το υλικό στρώχεται από ένα ακροφύσιο ή από ένα άνοιγμα	Fused Deposition Modelling/ Fused Filament Fabrication, Fused Layer Modelling	-Ευρεία και διαδεδομένη χρήση - Φθηνή τεχνική -Εφικτή η παραγωγή πλήρως λειτουργικών αντικειμένων	-Ανισοτροπία στη κάθετη διεύθυνση -Δεν ανέρχεται σε υψηλές λεπτομέρειες	Πολυμερή, Σύνθετα	X <=900 Y <= 600 Z <= 900	Stratasys, USA
Material Jetting	Σταγόνες του υλικού εναποτίθενται	3D inkjet technology, Direct Ink writing	-Υψηλή ακρίβεια εναπόθεσης σταγόνων -Χαμηλή παραγωγή παραπροϊόντων -Πολυχρωμία	-Συνήθως είναι απαραίτητη κάποια βάση για το υλικό -Χρησιμοποιούνται κυρίως φωτοπολυμερή και θερμοσκληρυνόμενες ρητίνες	Πολυμερή, Κεραμικά, Σύνθετα, Υβρίδια, Βιολογικά/ Βιούλικά	X <=300 Y <= 200 Z <= 200	Stratasys, USA 3D Systems, USA PolyPico, Ireland 3Dinks, USA WASP, Italy
Powder Bed Fusion	Θερμική ενέργεια συγκεντρώνεται σε μικρή περιοχή της πλατφόρμας που βρίσκονται οι κόκκοι του υλικού	Electron beam melting, Direct Metal Laser Sintering, Selective Laser Sintering/Melting	-Μικρό αποτύπωμα -Σχετικά φθηνή -Τα σφαιρίδια της πούδρας του υλικού λειτουργούν ως φυσικό υπόστρωμα -Μεγάλη γκάμα υλικών	-Σχετικά αργή -Απουσία ολοκληρωμένης δομής -Περιορισμός διαστάσεων -Απαιτήση υψηλής ισχύος -Η τελειοποίηση εξαρτάται από το μέγεθος των κόκκων	Μέταλλα, Κεραμικά, Πολυμερή, Σύνθετα, Υβρίδια	X = 200-300 Y = 200-300 Z = 200-350	ARCAM, Sweden;EOS, Germany;Concept Laser Cusing, Germany;MTT, Germany;Phoenix Systemy Group, France;Renishaw, UK;Realizer, Germany; Matsuura, Japan;3D Systems
Sheet Lamination	Φύλλα/ελάσματα υλικών συνδέονται	Laminated Object Manufacturing, Ultrasound consolidation/Ultrasound Additive Manufacturing	-Ταχεία τεχνική -Φτηνή -Εύκολος χειρισμός πρώτων υλών	-Η αντοχή και η ολοκλήρωση των αντικειμένων εξαρτάται από το συγκολλητικό μέσο που χρησιμοποιείται -Είναι πιθανό να απαιτείται κάποια μετεπεξεργασία -Περιορισμένη χρήση υλικών	Πολυμερή, Μέταλλα, Κεραμικά, Υβρίδια	X = 150-250 Y = 200 Z = 100-150	3D Systems, USA MCor, Ireland
Vat Photopolymerization	Υγρό πολυμερές σκληραίνεται μέσω ακτινοβολίας	Stereo Lithography, Digital Light Processing	-Εξαιρετη ακρίβεια -Μεγάλα κομμάτια -Εξαιρετη επιφανειακή τελειοποίηση και λεπτομέρειες	-Περιορίζεται στα φωτοπολυμερή -Χαμηλή διάρκεια ζωής, χαμηλές μηχανικές ιδιότητες -Ακριβή τεχνική και αργή η πορεία κατασκευής	Πολυμερή, Κεραμικά	X < 2100 Y <700 Z <800	Luthoz, Austria 3D Ceram, France

Όπως αναφέρθηκε, αυτές οι καινοτόμες μέθοδοι κατασκευής προσφέρουν πολλά πλεονεκτήματα, ωστόσο έχει να αντιμετωπίσει και τις αντίστοιχες προκλήσεις που αφορούν τη βελτιστοποίηση των μεθόδων επιφανειακής επεξεργασίας, κατανάλωση ενέργειας, την επαναληψιμότητα κ.α. Στον παρακάτω πίνακα αναγράφονται συνοπτικά οι προκλήσεις της ΠΚ για 4 βασικές κατηγορίες (4M - Materials, Making, Metrology & Market)[3].

ΠΙΝΑΚΑΣ 2: Τα «4M» της Προσθετικής Κατασκευής, οι προκλήσεις της.

Τομέας	Προκλήσεις
Υλοποίηση (Making)	<ul style="list-style-type: none"> ▪ Εξέλιξη της πολυστρωματικής τεχνολογίας με ευελιξία και προσαρμοστικότητα ▪ Ευρεία γκάμα υλικών συμπεριλαμβανομένων μετάλλων, κεραμικών, πολυμερών και των συνδυασμών τους με τη μορφή συνθέτων, υβριδίων και υλικών διαβαθμισμένων ιδιοτήτων(FGMs) ▪ Ανάπτυξη ενός αυτόνομου, φιλικού προς το χρήστη, ασφαλούς, σταθερού, ολοκληρωμένου συστήματος που θα καταφέρει να προσφέρει την απαιτούμενη ταχύτητα κατασκευής, υψηλή ανάλυση και ταυτόχρονα χαμηλή κατανάλωση ενέργειας για την κατασκευή και τον διαστασιακό έλεγχο ενός αντικειμένου.
Υλικά (Materials)	<ul style="list-style-type: none"> ▪ Κατασκευαστικές τεχνικές, ανταλλαγή δεδομένων και αλλαγή συστημάτων κατά τη 4^η Βιομηχανική Επανάσταση (Industry 4.0) ▪ Ομοιογένεια ▪ Βασικές ιδιότητες στην επιφάνεια ▪ Μέγεθος υλικού πρώτων υλικών από μερικά νανόμετρα έως μικρόμετρα
Μετρολογία (Metrology)	<ul style="list-style-type: none"> ▪ Διαχείριση και έλεγχος ως προς τη βελτιστοποίηση και διασφάλιση ▪ Υψηλό επίπεδο προσαρμοστικότητας των τεχνικών για επί τόπου μετρήσεις ▪ Ανάγκη για υψηλή ακρίβεια και ταχύτητα μετρήσεων
Αγορά (Market)	<ul style="list-style-type: none"> ▪ Factory 4.0 ή Manufacturing 4.0 ▪ Ευρός της ΠΚ από την αεροδιαστημική βιομηχανία στα παιχνίδια έως και τα τρόφιμα μπορούν να τη καθιστούν ως μιας κερδοφόρα βιομηχανία ▪ Φθηνότερα συστήματα μετατρέπουν περισσότερο προσβάσιμη τη ΠΚ

1.2.3 ΕΦΑΡΜΟΓΕΣ & ΠΑΡΑΔΕΙΓΜΑΤΑ

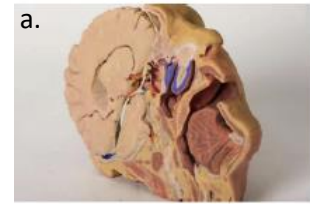
Η 3D εκτύπωση χρησιμοποιείται σε ένα μεγάλο πλήθος εφαρμογών κυρίως σε αεροναυτικές, ιατρικές, οδοντιατρικές και βιομηχανικές εφαρμογές και άλλες. Παρακάτω παρουσιάζονται μερικές ενδεικτικές εφαρμογές αναλυτικότερα:

❖ Ιατρικές εφαρμογές

Οι διάφορες βλάβες και τα διάφορα ελαττώματα που προκύπτουν σε ιστούς ή σε όργανα κατά τη μεταμόσχευσή τους θεωρούνται ίσως ένα από τα πιο άμεσα και μεγάλα προβλήματα στη κλινική ιατρική. Μια συχνή επίσης μέθοδος, που αποτελεί την εμφύτευση τεχνητών οργάνων είναι συνήθως επιτυχής και βελτιώνει τη ποιότητα ζωής το ασθενή. Η τριδιάστατη τεχνολογία, θεωρείται το κλειδί για να ξεπεραστούν οι παραπάνω περιορισμοί στην ιατρική, που κάνουν την εμφάνισή τους σε παραδοσιακές τεχνικές, μέσω της προσαρμογής ενός ανθρώπινου βιονικού ιστού ή οργάνου στις

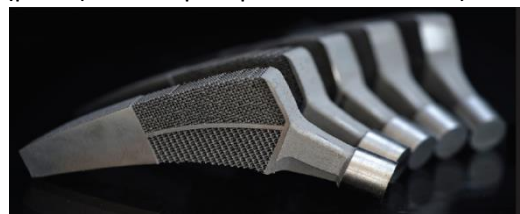
ανάγκες του ασθενή [9]. Υπάρχουν 4 διαφορετικοί τομείς στους οποίους ειδικεύεται η τριδιάστατη εκτύπωση:

1) Ο σχεδιασμός και η παραγωγή μοντέλων και οργάνων για σχεδιασμό και ανάλυση χειρουργικής διαδικασίας και φροντίδας. Παραδείγματος χάριν, μια ομάδα ερευνητών από την Κίνα και την Αμερική χρησιμοποίησαν HeLa κύτταρα – τα γνωστά αθάνατα κύτταρα από την Ενριέττα Λακς, και υδρογέλες από ζελατίνη/άλας αλγινικού οξέως/ινωδογόνο για να κατασκευάσουν μοντέλα *in vitro* καρκίνου της μήτρας [10]. Επίσης, στο πανεπιστήμιο Monash, στην Αυστραλία, αναπτύχθηκε για εκπαιδευτικούς σκοπούς ένα τριδιάστατο μοντέλο ανατομίας του ανθρώπινου οργανισμού χρησιμοποιώντας γύψο σε μορφή σκόνης και πλασικό, Εικόνα 2 [9].



Εικόνα 2: 3D εκτυπωμένα μοντέλα ανθρώπινης ανατομίας α. κεφάλι και β. χέρι από το Monash University.

2) Μόνιμα μη-βιοενεργά εμφυτεύματα. Αυτή η περίπτωση αντιστοιχεί σε μόνιμα ιατρικά εμφυτεύματα που χρησιμοποιούνται κατά κύριο λόγο στην οδοντιατρική και ορθοπεδική, όπου απαιτούνται μη αποικοδομήσιμα βιοϋλικά και καλή βιοσυμβατότητα μετά από τη χειρουργική διαδικασία. Η τριδιάστατη εκτύπωση έρχεται εδώ για να δώσει λύση με τη κατασκευή εμφυτευμάτων ελαφρών με προσαρμοσμένη ακαμψία [9], [11]. Συνήθη υλικά που χρησιμοποιούνται είναι το τιτάνιο (Ti) [12] και κράματά του, τα οποία μπορούν να είναι επικαλυμμένα με κάποια πρωτεΐνη, υδρογέλες, κολλαγόνο κ.α.[9] Άλλο είδος μη-βιοενεργών εμφυτευμάτων είναι η περίπτωση των meta-εμφυτευμάτων (meta-implants) που προέρχονται από τον ορθολογικό σχεδιασμό (rationally design) βάσει φυσικών μοντέλων που προσομοιώνουν τη συμπεριφορά τους και προβλέπουν το πως επηρεάζονται στο αντίστοιχο περιβάλλον. Υπάρχει μια σειρά τέτοιων υβριδικών meta-εμφυτευμάτων, σχεδιασμένα με τη παραπάνω μέθοδο, που μπορούν να παρουσιάσουν συμπεριφορές με τιμές αρνητικές και θετικές του λόγου Poisson. Σε αντίθεση με τα συμβατικά εμφυτεύματα, είναι σε θέση να φέρουν επαναλαμβανόμενα κυκλικά φορτία χωρίς συσσώρευση τάσεων και διατηρώντας ακέραιη τη διεπαφή με το οστό. Σκοπός είναι να γίνει συνδυασμός των μηχανικών ιδιοτήτων (π.χ. αρνητικός λόγος Poisson), με βιολειτουργικές ιδιότητες, όπως για παράδειγμα αναγέννηση ιστού. Μια αρκετά συνηθισμένη αυτών των υλικών είναι η ορθοπεδική διαδικασία της ολικής αντικατάστασης μηρού (Total hip replacement – THP). Τα συμβατικά εμφυτεύματα έχουν συνήθως περιορισμένο χρόνο ζωής ως προς τη μηχανική καταπόνηση, η οποία δε κατανέμεται ομοιόμορφα στο εμφύτευμα. Με τη μέθοδο επιλεκτικής τήξης υλικού με λέιζερ (Selective Laser Melting) έχει γίνει επιτυχώς η κατασκευή υβριδικού meta-εμφυτεύματος από Ti-6Al-4V όπως φαίνεται στη παρακάτω εικόνα, όπου έχει γίνει η κατασκευή δύο διαφορετικών πλεγμάτων. Ο λόγος που χρησιμοποιούνται δύο πλέγματα είναι γιατί οι τάσεις που εφαρμόζονται από τα κόκκαλα επαφής, κατανέμονται ομοιόμορφα [13].



Εικόνα 3: Meta-εμφύτευμα κατασκευασμένο με Ti-6Al-4V και χρήση SLM εκτύπωσης. Χρήση του εμφυτεύματος για ολική αντικατάσταση μηρού.

3) Κατασκευή τοπικών βιοενεργών και βιοαποικοδομήσιμων ικριωμάτων. Υπάρχουν δύο πιθανοί τρόποι για τη κατασκευή ιστών και οργάνων, οποίοι εξαρτώνται από το αν το κύτταρο θα επεξεργαστούν άμεσα κατά τη διάρκεια της διαδικασίας. Η περισσότερο έμμεση οδός κατασκευής ιστών βασίζεται στην ιστική μηχανική, κατά την οποία πρώτα σχηματίζεται το τριδιάστατο ικρίωμα και στη συνέχεια εμφυτεύονται τα κύτταρα [14]. Η άμεση οδός που είναι γνωστή ως κυτταρική ένωση (cell assembly), χρησιμοποιεί κύτταρα και γέλη για τη κατασκευή της δομής [9].

4) Είδη υποδημάτων

Οι ανατομικές σόλες και τα πρόσθετα έχουν διορθωτική δράση στο άτομο. Στο τομέα των υποδημάτων, η εσωτερική σόλα είναι ένα κομμάτι που βρίσκεται μεταξύ της πατούσας και της σόλας του παπουτσιού, κάτι το οποίο σημαίνει πως βρίσκεται σε άμεση επαφή με το πέλμα και λειτουργεί σα βοηθικό στήριξης του βάρους του σώματος [15]. Μια ανατομική σόλα μπορεί να βοηθήσει στη μείωση της πίεσης του πέλματος και στη σωστή στήριξη των σημείων, με αποτέλεσμα να μειωθεί η πίεση στην οποία υποβάλλεται το άκρο κατά τη διάρκεια σωματικής δραστηριότητας ή ακόμα και για απλή όρθια στάση κατά μεγάλη χρονική περίοδο [16].



Εικόνα 3. Αριστερά: Συνεργασία της Adidas με τη start-up Carbon για ένα προσαρμοσμένο στις ανάγκες του πελάτη παπούτσι. Δεξιά: Παραγωγή της Nike, το 2013, με τη SLS 3D printing τεχνική παπούτσια για επώνυμους ποδοσφαιριστές.

Οι παράγοντες που καθορίζουν τα χαρακτηριστικά που μπορεί να έχει ο σχεδιασμός μιας εσωτερικής σόλας αφορούν πρωτίστως τη μορφολογία της και το να εφαρμόζει σωστά με την αντίστοιχη ανατομία του ποδιού [17]. Αρκετά διαδεδομένα υλικά για τη ΠΚ αυτών των προϊόντων είναι τα ABS, TPU (thermoplastic polyurethane) αλλά ακόμα και πλαστικά ενισχυμένα με ίνες άνθρακα (CFRPs) [18], [19]. Μερικά οπτικά παραδείγματα παπουτσιών που κατασκευάζονται από διεθνείς εταιρείες μέσω 3D printing, φαίνονται στην Εικόνα 3 [20], [21].

❖ Αυτοκινητοβιομηχανία

Στο κόσμο της αυτοκινητοβιομηχανίας υπάρχει μεγάλο ενδιαφέρον για τη τρισδιάστατη εκτύπωση, με τους κατασκευαστές να χρησιμοποιούν τη τεχνολογία αυτή. Ο κατασκευαστής που έκανε ένα βήμα παρακάτω με τη τρισδιάστατη εκτύπωση είναι η Bugatti. Η εταιρεία αυτή κατασκεύασε μία ενιαία δαγκάνα φρένων οκτώ πιστονίων, η οποία αποτελεί τη μεγαλύτερη στην αυτοκινητοβιομηχανία. Επί προσθέτως, είναι και το μεγαλύτερο αντικείμενο τιτανίου που έχει ποτέ παραχθεί μέσω του 3D printing [22].



Εικόνα 4 : Δαγκάνα φρένων από τιτάνιο, κατασκευασμένη μέσω 3D printing τεχνολογίας από την εταιρεία Bugatti.

2 ΥΛΙΚΑ ΔΙΑΒΑΘΜΙΣΜΕΝΩΝ ΙΔΙΟΤΗΤΩΝ (FGMs)

2.1 ΟΡΙΣΜΟΣ

Τα υλικά που παρουσιάζουν αλλαγή στη σύνθεση, τη μικροδομή ή και το πορώδες στον κύριο όγκο του υλικού αναφέρονται ως υλικά διαβαθμισμένων ιδιοτήτων (Functionally Graded Materials – FGMs). Οι ιδιότητες των FGMs δεν είναι ομοιογενείς σε όλο τον όγκο του υλικού, αλλά εξαρτώνται από τη θέση στο χώρο και τη δομή του. Τα FGMs υλικά είναι σχεδιασμένα για συγκεκριμένη εφαρμογή ή λειτουργία, στην οποία η χωρική διαβάθμιση της δομής και/η σύνθεση δίνει από μόνη της προσαρμοσμένες ιδιότητες, όπως των χημικών ιδιοτήτων, μηχανικών, μαγνητικών, θερμικών και ηλεκτρικών ιδιοτήτων. Αυτό επιτυγχάνεται με την εις βάθος διαβάθμιση στη σύνθεση, στη μικροδομή και στις ιδιότητες. Υπάρχουν FGMs που είναι σχεδιασμένα έτσι ώστε να έχουν σταδιακά διαβαθμισμένες ιδιότητες (stepwise-graded structures) και άλλα να έχουν συνεχή διαβάθμιση στις ιδιότητες (continuous-graded structures) [23]. Τα διαβαθμισμένα υλικά δεν αποτελούν κάτι ξένο στη φύση. Ένα απλό παράδειγμα από τέτοια υλικά είναι τα φυτά μπαμπού (bamboo) [24], [25], που χρησιμοποιούνται εδώ και πολλά χρόνια για κατασκευή και διακόσμηση, αλλά ακόμα και τα δόντια και το ανθρώπινο δέρμα μπορούν να αποτελέσουν παραδείγματα διαβαθμισμένων υλικών.

2.2 ΕΙΔΗ FGMs

Κατά την έναρξη της ανάπτυξης των FGMs, η βασική ιδέα ήταν να αφαιρεθεί η αιχμηρή και απότομη διεπιφάνεια μεταβολής ιδιοτήτων που υπήρχε στα κλασικά υλικά, και να γίνει η αντικατάστασή της με μια βαθμωτά μεταβαλλόμενη διεπιφάνεια, η οποία είναι μεταφρασμένη σε μεταβαλλόμενη χημική σύσταση του σύνθετου υλικού σε αυτή τη περιοχή της διεπιφάνειας. Αυτά τα υλικά που έχουν διαβάθμιση στη σύνθεση, τη δομή και έχουν συγκεκριμένες ιδιότητες στην επιθυμητή διεύθυνση, μπορούν να θεωρηθούν ανώτερα από κάποια ομογενή υλικά με τα ίδια συστατικά. Επίσης κάποιες μηχανικές ιδιότητες όπως η σταθερά του Young, ο λόγος Poisson, το μέτρο διάτμησης, η πυκνότητα του υλικού και η σταθερά θερμικής διαστολής ποικίλουν ελαφρώς και συνεχώς ανάλογα με τη κατεύθυνση [23]. Το αυξανόμενο ενδιαφέρον για αυτό το είδος του υλικού έχει οδηγήσει στην ανάπτυξη διαφορετικών τύπων FGMs που αναλύονται στη συνέχεια [26]–[29].

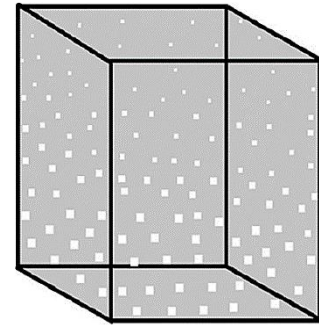
2.2.1 ΧΗΜΙΚΗ ΣΥΣΤΑΣΗ ΤΩΝ FGMs

Σε αυτόν τον τύπο FGMs, η χημική σύσταση μεταβάλλεται σύμφωνα με τη χωρική κατανομή στο υλικό. Αυτό μπορεί να σχηματιστεί σε μονοφασικό ή και πολυφασικό υλικό. Ένα μονοφασικό FGM παράγεται όταν το συστατικό παράγεται από μία μόνο φάση, με αποτέλεσμα της διαλυτότητας των χημικών στοιχείων από τη μία στην άλλη φάση. Αυτό συνήθως συμβαίνει κατά τη διάρκεια της τήξης [30]. Η βαθμωτή αλλαγή και η κατανομή των χημικών υλικών σε μία μόνο φάση έχει ως αποτέλεσμα το σχηματισμό των υλικών διαβαθμισμένων ιδιοτήτων. Σύμφωνα με διαγράμματα φάσης και τους θερμοδυναμικούς περιορισμούς, όταν ένα υλικό προστίθεται σε ένα άλλο, τα συστατικά θα είναι αναμίξιμα σε ένα συγκεκριμένο εύρος αναλογιών και συνθηκών ανάμειξης. Τέτοια υλικά μπορούν να ονομαστούν μονοφασικά υλικά – έχοντας μεταβαλλόμενη χημική σύσταση, εξαιτίας της διαλυτότητας. Αυτό το είδος των FGM είναι το λιγότερο διαδομένο. Τα πιο διαδομένα FGMs είναι πολυφασικά [31], [32]. Οι διαφορετικές φάσεις που προκύπτουν εξαρτώνται από τη ποσοτική σύσταση του ενισχυμένου υλικού και τις συνθήκες κατασκευής, όπως για παράδειγμα ο βαθμός ψύξης και η θερμική επεξεργασία. Για παράδειγμα στην κονιομεταλλουργία, η μέθοδος παραγωγής FGM πραγματοποιείται μεταβάλλοντας την απαιτούμενη αναλογία σωματιδίων από στρώμα σε στρώμα. Κατά τη τήξη, κάποια μεταλλικά σωματίδια θα αντιδράσουν με διαφορετικά

χημικά συστατικά και φάσεις. Έτσι δημιουργείται η ποικιλότητα σύμφωνα με τη χωρική κατανομή ενός FGM [23].

2.2.2 ΔΙΑΒΑΘΜΙΣΗ ΜΕ ΒΑΣΗ ΤΟ ΠΟΡΩΔΕΣ ΣΤΑ FGMs

Η διαβάθμιση του πορώδους στα FGMs αποτελεί μια ξεχωριστή κατηγορία, κατά την οποία το πορώδες του υλικού κατασκευάζεται για να αλλάζει με την μεταβολή της θέσης στον όγκο του υλικού. Το σχήμα και το μέγεθος του πόρου είναι σχεδιασμένο και ποικίλει, με βάσει τις απαιτούμενες ιδιότητες για ένα FGM (Εικόνα 5).

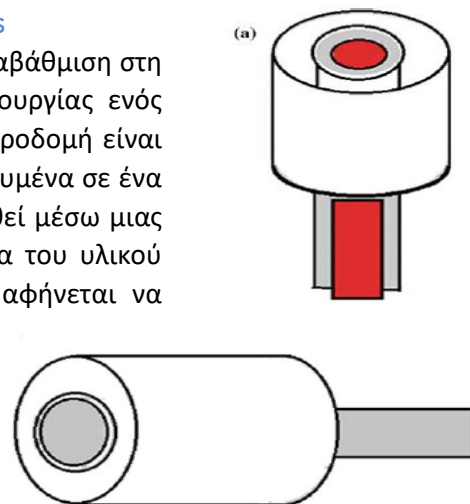


Εικόνα 5: Σχηματικό διάγραμμα διαβάθμισης πορώδους των FGMs

Αυτό το είδος υλικού διαβαθμισμένων ιδιοτήτων είναι πολύ σημαντικό στις βιοϊατρικές εφαρμογές, επειδή ο προς αποκατάσταση ιστός αποτελείται από λειτουργικά διαβαθμισμένο πορώδες, και η διαβάθμιση του πορώδους βοηθά στην ενσωμάτωση ενός εμφυτεύματος στους γειτονικούς ιστούς. Επίσης είναι σημαντική για τη διαδικασία επούλωσης, καλής κυκλοφορίας του αίματος, βοηθάει στη μείωση του συνολικού βάρους του εμφυτεύματος, βελτιώνει το μέτρο ελαστικότητας και ρυθμίζει τη πυκνότητά του, μια πολύ σημαντική παράμετρος που το προστατεύει από φαινόμενα συσσώρευσης τάσεων. Τα υλικά διαβαθμισμένου πορώδους μπορεί να είναι με διαβάθμιση πυκνότητας του πορώδους ή διαβάθμιση του μεγέθους του πορώδους. Η πυκνότητα του πορώδους μπορεί να παραχθεί με τη πυκνότητα του πορώδους να αλλάζει σύμφωνα με τη θέση στο χώρο. Το μέγεθος των πόρων σε ένα FGM μπορεί να παραχθεί με τη ποικιλία πόρων διαφορετικού μεγέθους ή σχήματος ή και τα δύο [23].

2.2.3 ΔΙΑΒΑΘΜΙΣΗ ΜΕ ΒΑΣΗ ΤΗ ΜΙΚΡΟΔΟΜΗ ΣΤΑ FGMs

Ένα άλλο είδος FGM είναι αυτό όπου υπάρχει διαβάθμιση στη μικροδομή πολυκρυσταλλικών υλικών. Ο σκοπός δημιουργίας ενός τέτοιου υλικού με διαφοροποιημένη διαβαθμισμένη μικροδομή είναι για να επιτυγχάνονται συγκεκριμένες και ιδιότητες στοχευμένα σε ένα υλικό. Η διαβάθμιση στη μικροδομή μπορεί να επιτευχθεί μέσω μιας διεργασίας στερεοποίησης, κατά την οποία η επιφάνεια του υλικού αφήνεται να κρυώσει. Ο πυρήνας του ίδιου υλικού αφήνεται να κρυώσει αργά, κάτι το οποίο βοηθάει στη παραγωγή διαφορετικών μικροδομών μεταξύ της επιφάνειας και των εσωτερικών τομών του υλικού. Παραδείγματος χάριν, μια μικροδομή που έχει ποικιλία ως προς τη μορφή της μπορεί να παραχθεί μέσω ελεγχόμενης θερμικής επεξεργασίας ενός κυλινδρικού κράματος τιτανίου, όπως φαίνεται στην Εικόνα 6 [23].



Εικόνα 6: Σχηματικό διάγραμμα ενός κυλινδρικού κομματιού εκτεθειμένο σε ροή (α) υγρού μετάλλου και (β) ψυκτικού νερού.

2.3 ΜΕΘΟΔΟΙ ΠΑΡΑΓΩΓΗΣ

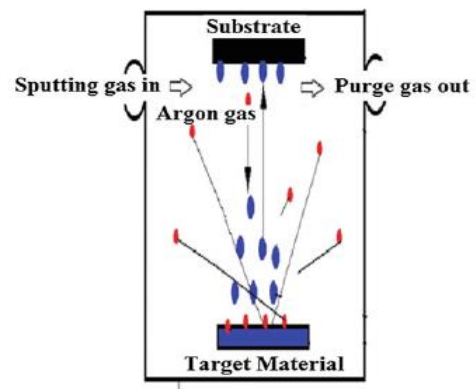
Τα υλικά διαβαθμισμένων ιδιοτήτων μπορούν να είναι σε μορφή λεπτών φιλμ που εφαρμόζονται σε μια επιφάνεια ενός υλικού με σκοπό να βελτιώσουν τις ιδιότητες της επιφάνειας, ή μπορούν να εμφανιστούν στον όγκο του υλικού, όπου οι ιδιότητες του υλικού μεταβάλλονται στο χώρο. Υπάρχουν διάφοροι τύποι διεργασιών εναπόθεσης που χρησιμοποιούνται για τη παραγωγή

λεπτών φιλμ με διαβαθμισμένες ιδιότητες ως επικαλυπτικά. Αυτές οι τεχνικές συμπεριλαμβάνουν τη φυσική εναπόθεση ατμών (Physical Vapor Deposition – PVD), τη χημική εναπόθεση ατμών (Chemical Vapor Deposition – CVD), την αυτοπροωθούμενη σύνθεση υψηλής θερμοκρασίας (Self-propagating high-temperature synthesis, SHS), ή ο συνδυασμός αυτών [23], [26], [33]. Οι περιπτώσεις που σχετίζονται με τα διαβαθμισμένα λεπτά φιλμ αντιστοιχούν σε εφαρμογές που πρέπει η επιφάνεια να έχει εντελώς διαφορετικές ιδιότητες από τον όγκο του υλικού. Σε μερικές εφαρμογές, είναι απαραίτητο ο όγκος του υλικού να έχει διαβαθμισμένες ιδιότητες εξαιτίας της φύσης των συνθηκών λειτουργίας του. Οι τρόποι παραγωγής αυτών των υλικών αφορούν τη μεταλλουργία σκόνης, τη φυγοκέντρηση και κάποιες μεθόδους χύτευσης (slip casting, tape casting) [27], [33], [34]. Σε όλες αυτές τις μεθόδους συμπεριλαμβάνονται δύο βασικά βήματα που είναι ο σχηματισμός μιας ομογενούς βαθμωτής δομής και την ενοποίηση (consolidation) του διαβαθμισμένου υλικού.

2.3.1 ΠΑΡΑΓΩΓΗ ΜΕ ΦΥΣΙΚΗ ΕΝΑΠΟΘΕΣΗ ΑΤΜΩΝ (PVD)

Η φυσική εναπόθεση ατμών είναι μια διεργασία που βασίζεται στην εφαρμογή ενός επικαλυπτικού μέσω ατμών και χρησιμοποιείται για τη παραγωγή λεπτών φιλμ και λεπτών λειτουργικά διαβαθμισμένων υλικών. Αυτή η μέθοδος επιτυγχάνεται με την εξάχνωση/εξάτμιση του υλικού που πρέπει να εναποτεθεί, και μεταφορά των ατμών στην επιφάνεια που θα επικαλυφθεί [35]. Υπάρχουν διάφοροι τρόποι ψεκασμού του στερεού υλικού όπως:

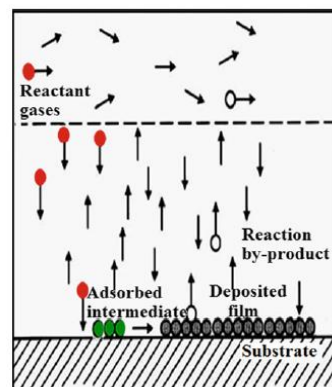
- PVD με βάση την εξάτμιση (Evaporation-Based PVD Process): Το υλικό θερμαίνεται από μια πηγή θερμότητας όπως ηλεκτρική αντίσταση, θερμό νήμα, ηλεκτρικό τόξο, δέσμη ηλεκτρονίων ή δέσμη λέιζερ. Οι πηγές θερμότητας βρίσκονται μακριά από το υπόστρωμα [36]. Τέσσερα βήματα ακολουθούν αυτή τη διεργασία που είναι α) εξάτμιση, β) μεταφορά στο υπόστρωμα, γ) αντίδραση με το υπόστρωμα – εάν είναι απαραίτητο, και, δ) εναπόθεση [23].
- Ιοντοβολή σε στόχο-υπόστρωμα PVD (Sputtering-Based PVD): Υπάρχουν διάφορα είδη μέσω για την εφαρμογή αυτής της μεθόδου όπως με τη χρήση συνεχούς ρεύματος (DC), συχνότητας ραδιοκυμάτων (RF), αντίδραση του στόχου με το επικαλυπτικό κ.α. Στη διαδικασία αυτή τα ιόντα αερίου διασκορπισμού, που είναι κάποιο αδρανές αέριο, τίθενται σε συνεχή σύγκρουση με την επιφάνεια του στόχου υλικού. Αυτό έχει ως αποτέλεσμα την εκτίναξη ατόμων από το υπόστρωμα του στόχου, έτσι ώστε τα άτομα του επικαλυπτικού να παίρνουν τη θέση των προηγούμενων και να εναποτίθενται στο υπόστρωμα, Εικόνα 7 [23].



Εικόνα 7: Σχηματική αναπαράσταση διεργασίας PVD με ιοντοβολή σε στόχο-υπόστρωμα.

2.3.2 ΧΗΜΙΚΗ ΕΝΑΠΟΘΕΣΗ ΑΤΜΩΝ (CHEMICAL VAPOR DEPOSITION - CVD)

Η χημική εναπόθεση ατμών (CVD) είναι μια διεργασία για την εναπόθεση λεπτών φιλμ και λειτουργικά διαβαθμισμένων επικαλυπτικών. Η CVD επιτυγχάνεται τοποθετώντας το προς επικάλυψη υλικό σε κενό, όπου και το επικαλυπτικό εξατμίζεται μέσω θέρμανσης ή μείωσης της πίεσης. Το πρόδρομο αέριο εισέρχεται στον χώρο που περιέχει το θερμό αντικείμενο που πρόκειται να επικαλυφθεί [37], [38]. Η χημική αντίδραση πραγματοποιείται κοντά στην επιφάνεια του υποστρώματος και ως αποτέλεσμα γίνεται η εναπόθεση του λεπτού φιλμ. Στην Εικόνα 8 απεικονίζεται η σχηματική αναπαράσταση της διεργασίας. Τα παραπροϊόντα από την αντίδραση εξέρχονται έξω από το θάλαμο. Η διεργασία αυτή μπορεί να λάβει μέρος από 200 έως 1600°C. Υπάρχουν διάφοροι τύποι CVD που αποτελούν διεργασίες χρήσης πλάσματος, ιόντων, φωτονίων, λέιζερ, θερμών νημάτων ή καύσης [20], [35].



Εικόνα 8: Σχηματικό διάγραμμα διεργασίας χημικής εναπόθεσης ατμών CVD.

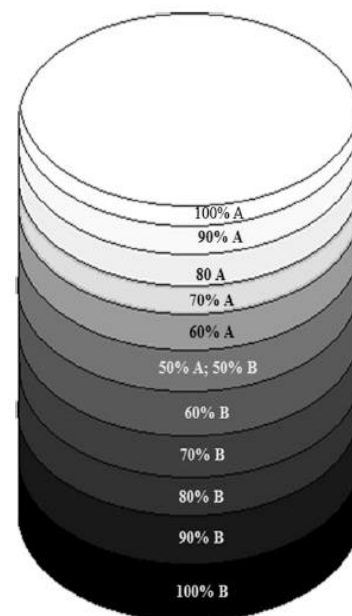
2.3.3 ΜΕΤΑΛΛΟΥΡΓΙΑ ΣΚΟΝΗΣ (POWDER METALLURGY)

Η μεταλλουργία σκόνης είναι μια παλιά κατασκευαστική διαδικασία, η οποία χρησιμοποιείται πλέον και για τη παραγωγή λειτουργικά διαβαθμισμένων υλικών. Τα βήματα που ακολουθούνται για τη παραγωγή των FGMs είναι:

- Προετοιμασία της σκόνης των υλικών
- Επεξεργασία της σκόνης
- Διαδικασίες διαμόρφωσης
- Τήξη ή πυροσυσσωμάτωση σε υψηλή θερμοκρασία και πίεση

Τα διαβαθμισμένα υλικά κατασκευάζονται με τη χρήση σωματιδίων σε μορφή σκόνης από τα υλικά που απαιτούνται για τη σύνθεσή τους. Η σκόνη των σωματιδίων χρησιμοποιείται για τη δημιουργία του αντικειμένου από μείγματα της σκόνης με μεταβολή του μέσου όρου του μεγέθους των σωματιδίων ή τη σύσταση. Με τη στοίβαξη της σκόνης βαθμωτά (Εικόνα 9), μια σταδιακή μεταβολή στη σύνθεση επιτυγχάνεται για το «πράσινο αντικείμενο» (green part) που παράγεται, το οποίο είναι εύθραυστο και είναι απαραίτητο να υποβληθεί σε θερμική κατεργασία και πυροσυσσωμάτωση.

Η διαδικασία τήξης απαιτεί υψηλές θερμοκρασίες που μπορούν να προωθήσουν τη συμπύκνωση και τις διαδικασίες διάχυσης και χημικών αντιδράσεων. Επίσης κάποιες θερμοδυναμικές παράμετροι, κατά τη διαδικασία τήξης, μπορούν να παίξουν σημαντικό ρόλο για το σχεδιασμό των λειτουργικά διαβαθμισμένων υλικών. Παραδείγματος χάριν, κατά την υγρή φάση της διαδικασίας τήξης, είναι πιθανή η παραγωγή FGM με διαβαθμισμένο πορώδες. Ακόμη, διαβάθμιση στη χημική σύνθεση ή στη μικροδομή μπορεί να επιτευχθεί κατά τη διαδικασία τήξης [23].



Εικόνα 9: Βαθμωτή στοίβαξη διαφορετικών περιεκτικότητας πούδρας.

2.3.4 ΦΥΓΟΚΕΝΤΡΗΣΗ ΜΕ ΧΥΤΕΥΣΗ (CENTRIFUGAL CASTING METHOD)

Η μέθοδος της φυγόκεντρησης σε συνδυασμό με χύτευση αφορά την έγχυση τηγμένου μετάλλου το οποίο περιέχει κάποιο άλλο ενισχυμένο υλικό – είτε τηγμένο είτε σε στερεά φάση, σε ένα καλούπι/μήτρα μορφοποίησης το οποίο περιστρέφεται για τη παραγωγή FGM. Με περιστροφή της μήτρας δημιουργείται μια φυγόκεντρος δύναμη που βοηθά στην ώθηση του τηγμένου υλικού προς τα τοιχώματα του καλουπιού. Έτσι επέρχεται ο διαχωρισμός του αιωρούμενου στερεού υλικού, και του τήγματος των δύο υλικών, εξαιτίας των διαφορετικών πυκνοτήτων τους, οδηγώντας στη δημιουργία ενός FGM. Η κατανομή της διαβάθμισης των σχηματιζόμενων FGM από αυτή τη διαδικασία μπορεί να επηρεαστεί σημαντικά από τη διαφορετική πυκνότητα μεταξύ των σωματιδίων του ενισχυμένου υλικού και του τηγμένου υλικού, το μέγεθος των σωματιδίων και τη κατανομή του μεγέθους, το ιξώδες του τηγμένου υλικού, και τον χρόνο στερεοποίησης. Η CCM είναι από τις πιο αποτελεσματικές μεθόδους για την επεξεργασία του κύριου όγκου των FGMs εξαιτίας του μεγάλου ελέγχου στη σύνθεση και τη μικροδομή [23].

2.4 ΠΡΟΣΘΕΤΙΚΗ ΚΑΤΑΣΚΕΥΗ (ΠΚ) FGMs

Οι τεχνικές με τις οποίες υλικά διαβαθμισμένων ιδιοτήτων παράγονται μέσω ΠΚ περιγράφονται με τον όρο Functionally Graded Additive Manufacturing (FGMA). Μέσω της ΠΚ είναι δυνατός ο έλεγχος της πυκνότητας και τη διεύθυνσης της εξώθησης του υλικού μέσα σε μια πολύπλοκη τριδιάστατη κατανομή ή ακόμη και το συνδυασμό διαφόρων υλικών για τη παραγωγή μιας δομής μεταβάλλοντας τις δύο προαναφερθείσες παραμέτρους κατά την εξώθηση [39]. Η πιθανή διαβάθμιση της μικροδομής που μπορεί να ληφθεί μέσω της FGAM μπορεί να χαρακτηριστεί σε 3 τύπους:

- Μεταβαλλόμενη πυκνότητα με ομογενή σύνθεση
- Ετερογενής σύνθεση με ταυτόχρονο συνδυασμό 2 ή παραπάνω υλικών κατά τη μετάβαση στη διαφορετική βαθμίδα
- Συνδυασμός της μεταβολής πυκνότητας με την ετερογενή σύνθεση

2.4.1 ΟΜΟΓΕΝΗΣ ΣΥΝΘΕΣΗ

Η ομογενής σύνθεση με την FGAM δημιουργεί διαβάθμιση πορώδους ή πυκνότητας μέσω στρατηγικής διαμόρφωσης τη μικροδομής στο χώρο ή της μορφολογίας των πλεγματικών δομών στον όγκο του υλικού μέσω την προσέγγισης με voxel [40], [41]. Για τον σχεδιασμό ενός FGAM υλικού είναι απαραίτητη η υποδιαίρεση του χώρου στον όγκο του υλικού, και ένας τρόπος για αυτό είναι η χρήση των voxels. Η μέθοδος των voxel αποτελεί ένα είδος ψηφιοποίησης ενός δεδομένου σχήματος. Τα voxels ορίζονται ως η αντιστοίχιση του όγκου του υλικού σε στοιχειώδεις όγκους, κατά αντιστοιχία με τη μέθοδο των pixel σε μία εικόνα. Ο χωρισμός ενός σχήματος γίνεται σε στοιχειώδεις μονάδες, οι οποίες ψηφιακά αντιστοιχίζονται σε τιμές μηδέν 0 (deactivated region/false) και ένα 1 (activated region/true). Επομένως με 0 γίνεται αντιστοίχιση στο κενό χώρο και με το 1 γίνεται αντιστοίχιση στον χώρο όπου βρίσκεται το υλικό. Ένα voxel μοντέλο μπορεί να οριστεί ως ένας 3D-πίνακας όπου κάθε στοιχείο αυτού του πίνακα, αντιπροσωπεύει ένα voxel [41]. Η έμπνευση πίσω από τη τεχνολογία των FGAMs προέρχεται από πλεγματικές δομές υλικών που έχουν βρεθεί στη φύση, όπως για παράδειγμα οι διαβαθμίσεις στη πυκνότητα της δομής στα φοινικόδεντρα, στις αφρώδεις δομές των οστών ή ενός μυ. Η μεταβολή της πυκνότητας σε ένα υλικό με τη χρήση της ΠΚ έχει προσδώσει στα υλικά αυτά τη δυνατότητα να διαμορφωθούν ως ελαφριές δομές με προσαρμοσμένη διάταξη πλέγματός και με ποικιλομορφία στη γεωμετρία, έτσι ώστε να επιτευχθεί υψηλή αντοχή με μικρότερο βάρος [40].

2.4.2 ΕΤΕΡΟΓΕΝΗΣ ΣΥΝΘΕΣΗ

Η FGAM εκφράζει την ιδέα της χρήσης πολλών υλικών μέσα από μια δυναμική σύνθεση βαθμιδώσεων ή πολύπλοκης μορφολογίας. Η γεωμετρική και υλική διαμόρφωση των φάσεων ελέγχει πάνω απ' όλα τις λειτουργίες και τις ιδιότητες ενός FGAM προϊόντος. Η πολλαπλή χρήση υλικών της FGAM επιδιώκει να βελτιώσει τον ενδοεπιφανειακό δεσμό μεταξύ ανόμοιων και ασύμβατων υλικών. Συνήθεις αστοχίες όπως αποκόλληση ή ρωγμές προκαλούνται από τις επιφανειακές τάσεις που υφίστανται κατά τη συμβατική ΠΚ με πολλαπλά υλικά εξαιτίας της διακριτής αλλαγής των ιδιοτήτων των υλικών, που με αυτό το τρόπο μπορούν να αποφευχθούν [40], [42]. Οι τάσεις σε παράλληλη και εγκάρσια διεύθυνση που προκαλούνται από τη διαφορετική σταθερά θερμικής διαστολής σε κρίσιμα σημεία μπορούν να μειωθούν σημαντικά [43] καθώς η διαφοροποίηση των ιδιοτήτων του υλικού μπορεί να βελτιώσει τις επιμέρους τάσεις [40], [44].



Εικόνα 10: Σχήμα στα αριστερά, αναπαράσταση συμβατικής μεθόδου ΠΚ με πολλαπλά υλικά. Σχήμα στα δεξιά, αναπαράσταση εκτύπωσης με πολλαπλά υλικά με την FGAM.

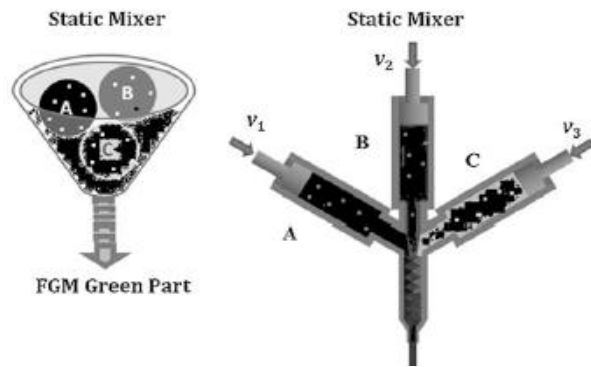
Με τη τήξη και ανάμειξη ενός υλικού στο άλλο κατά την εκτύπωση χρησιμοποιώντας μία δυναμική βαθμιδωση, τα προς εκτύπωση στοιχεία μπορούν να αποκτήσουν τις βέλτιστες ιδιότητες και των δύο υλικών. Αυτές μεταδίδονται σε όλο το βάρος του υλικού και ταυτόχρονα διατηρούν την ανθεκτικότητα, την αντίσταση στη φθορά, την αντοχή στην κρούση ή τις φυσικές, χημικές ή βιοχημικές ή μηχανικές ιδιότητες [35], [45]. Πλέον τα ετερογενή μείγματα υλικών δε χρειάζεται να υπονομεύσουν τις ιδιότητές τους για να επιτευχθούν οι επιθυμητές ιδιότητες. Η κατασκευή με πολλά υλικά ταυτόχρονα μπορεί επίσης να προσδώσει σε συγκεκριμένο σημείο στο χώρο, ιδιότητες προσαρμοσμένες στις εκάστοτε ανάγκες [46].

2.4.3 ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΡΟΣΘΕΤΙΚΗΣ ΚΑΤΑΣΚΕΥΗΣ ΓΙΑ FGAM

2.4.3.1 ΕΞΩΘΗΣΗ ΥΛΙΚΟΥ (MATERIAL EXTRUSSION)

Η εξώθηση υλικού χρησιμοποιείται για τη δημιουργία πρωτοτύπων, λειτουργικών αντικειμένων και FGMs. Οι πιο σημαντικές κατασκευαστικές παράμετροι είναι το πάχος στρώσης και τα χαρακτηριστικά του πλέγματος πλήρωσης (Κεφάλαιο 3). Αυτές οι παράμετροι παίζουν σημαντικό ρόλο στις τελικές ιδιότητες του εκτυπωμένου υλικού. Ακόμη, η πίεση κατά την εξώθηση πρέπει να έχει μια σταθερή τιμή, έτσι ώστε να υπάρχει καλή εναπόθεση [23].

Τα συστήματα εξώθησης τηκόμενου υλικού (Fused Filament Fabrication) είναι ικανά να χειρίζονται πολλαπλούς εξωθητές υλικού με το καθένα να έχει διαφορετικό υλικό κατεργασίας. Μια άλλη τεχνική ΠΚ για εξώθηση υλικού ονομάζεται Freeze-form Extrusion Fabrication (FEF) [47]. Σε αυτή τη διαδικασία κατασκευάζεται ένα 3D αντικείμενο πολυστρωματικά με χρήση υλικού σε μορφή πάστας. Η κατασκευή του αντικειμένου γίνεται σε θερμοκρασία χαμηλότερη από το σημείο πήξης του νερού, έτσι ώστε να στερεοποιηθεί η πάστα ενώ εναποτίθεται, παράγοντας το αντικείμενο, το οποίο, έπειτα, υπόκειται σε μετεπεξεργασία. Η διεργασία έχει χρησιμοποιηθεί για τη παραγωγή κεραμικών, όπως οξειδίο του αλουμινίου, του ψευδαργύρου, διβορίου και βιοενεργά ικρίσματα. Για την παραγωγή των FGMs χρησιμοποιείται συνήθως ένας τριπλός εξωθητήρας, όπου ο καθένας περιέχει μια πάστα υλικού. Σε ένα στατικό μίξερ, γίνεται ανάμειξη των υλικών σε μια ομογενή πάστα, όπως φαίνεται στην Εικόνα 11 [23], [40]. Το υλικό που σχηματίζεται παγώνει και στεγνώνει σε υψηλή πίεση και χαμηλή θερμοκρασία για αρκετές ώρες. Σε αυτό το στάδιο, το κομμάτι συντήκεται σε υψηλή θερμοκρασία που ταυτόχρονα είναι χαμηλότερη από τη θερμοκρασία τήξης των υλικών παρασκευής. Βασικά πλεονεκτήματα αυτής της μεθόδου είναι πως αυτή η τεχνική είναι άμεση και φθηνή. Ωστόσο υπάρχουν και κάποιοι περιορισμοί που αφορούν τη περιορισμένη ακρίβεια και ταχύτητα κατασκευής σε σχέση με άλλες μεθόδους ΠΚ [23].

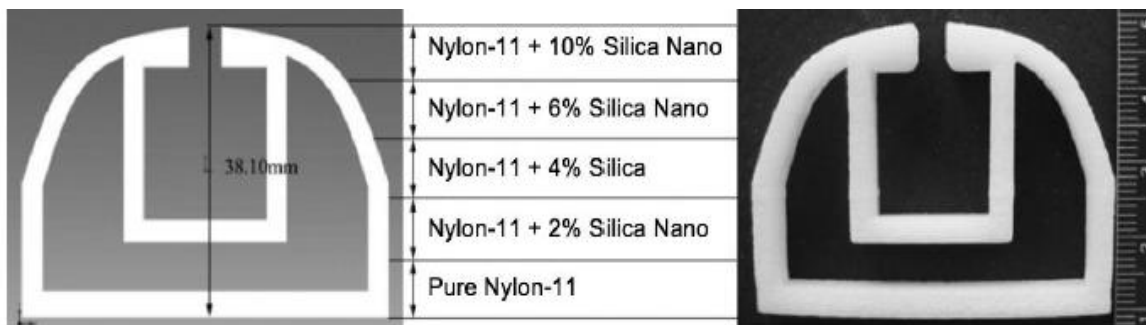


Εικόνα 11: Σχηματικό διάγραμμα ενός στατικού μίξερ και ενός τριπλού εξωθητή.

2.4.3.2 ΤΗΞΗ ΚΟΚΚΩΝ (POWDER BED FUSION)

Η χρήση μεθόδων που βασίζονται στη τήξη κόκκων υλικού μπορούν να παράγουν αντικείμενα με σύνθετη γεωμετρία. Μια αρκετά διαδεδομένη μέθοδος είναι η πυροσυσσωμάτωση με λέιζερ (Selective Laser Sintering – SLS). Η SLS έχει χρησιμοποιηθεί για τη κατασκευή λειτουργικά διαβαθμισμένου νανοσύνθετου πολυμερούς Nylon-11 με διαφορετικά κλάσματα όγκου νανοσωματιδίων διοξειδίου του πυριτίου (0-30%) μέσου μεγέθους 15 nm, με τα αποτελέσματα να φαίνονται στην Εικόνα 12 [48].

Στην SLS οι κατασκευαστικές παράμετροι για διαφορετικές συνθέσεις επιλέχθηκαν μέσω πειραματικού σχεδιασμού, για το προσδιορισμό της σχέσης μεταξύ παραγόντων που έχουν επίδραση στη διαδικασία και το αποτέλεσμά της. Οι πυκνότητες και οι μικροδομές από τα νανοσύνθετα εξετάστηκαν με οπτική μικροσκοπία και ηλεκτρονικό μικροσκόπιο διέλευσης (TEM). Οι μηχανικές ιδιότητες των διαφορετικών συστάσεων εξετάστηκαν και βρέθηκε ότι παρουσιάζουν μη γραμμική

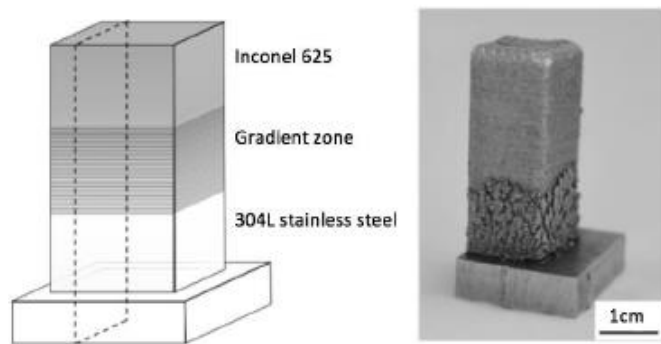


Εικόνα 12: Λαβή κατασκευασμένη με διαφορετικά κλάσματα όγκου των κύριων υλικών κατασκευής και χρήση της τεχνικής εκτύπωσης με πυροσυσσωμάτωση με λέιζερ (Selective Laser Sintering). Κάθε στρώμα έχει ύψος 7.62mm.

μεταβολή ως συνάρτηση του κλάσματος πλήρωσης. Επίσης η Τήξη με Λέιζερ (Selective Laser Melting – SLM) είναι μια τεχνική που μπορεί να χρησιμοποιηθεί για τη παρασκευή λειτουργικά διαβαθμισμένων μεταλλικών κομματιών. Με αυτή τη τεχνική μπορεί να εφαρμοστεί η ετερογενής σύνθεση μετάλλων χρησιμοποιώντας πολλαπλά συστήματα τροφοδοσίας. Τέλος η τεχνική με την οποία γίνεται τήξη με δέσμη ηλεκτρονίων μπορεί να χρησιμοποιηθεί για τη παρασκευή FGMs [40].

2.4.4 ΕΝΕΡΓΕΙΑΚΑ ΚΑΤΕΥΘΥΝΟΜΕΝΗ ΕΝΑΠΟΘΕΣΗ (DIRECTED-ENERGY DEPOSITION -DED)

Η εναπόθεση μεταλλικού υλικού με λέιζερ (Laser Metal Deposition – LMD) είναι μια σημαντική τεχνική σε αυτή τη κατηγορία ΠΚ και χρησιμοποιείται συχνά για την κατασκευή/επιδιόρθωση ενός προϊόντος. Η DED έχει τη δυνατότητα να επεξεργάζεται, διορθώνει, ενισχύει κομμάτια ή να προσθέσει υλικά σε μια υπάρχουσα δομή σε μία ενιαία διεργασία, η οποία δεν είναι επιτεύξιμη με άλλες μεθόδους ΠΚ. Σε αυτή τη τεχνική μια δέσμη από λέιζερ τήκει τη μεταλλική σκόνη που βρίσκεται μέσα στη κεφαλή εκτύπωσης και αυτά εναποτίθενται πολυστρωματικά από τη κεφαλή. Σε αυτή τη μέθοδο είτε η κεφαλή κινείται και το εκτυπωμένο αντικείμενο μένει στάσιμο στη πλατφόρμα εκτύπωσης και χτίζεται είτε αντίστροφα, η πλατφόρμα εκτύπωσης είναι αυτή που κινείται στο χώρο. Η DED μπορεί να χρησιμοποιηθεί για τη κατασκευή μεταλλικών αντικειμένων με διαβάθμιση στη σύνθεση μέσω προσαρμογής της σύστασης της μεταλλικής πούδρας που διοχετεύεται από την κεφαλή. Ένα παράδειγμα λειτουργικά διαβαθμισμένων υλικών, είναι η κατασκευή ανοξείδωτου ατσάλιου 304L με Inconel 625 με τη συγκεκριμένη μέθοδο εκτύπωσης [49].

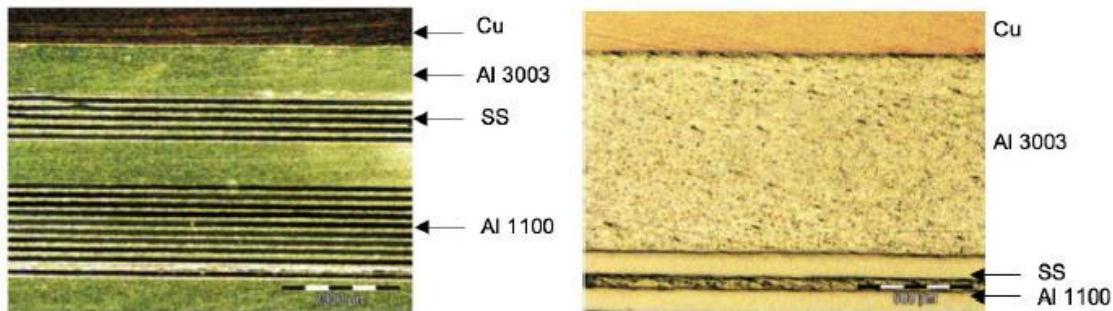


Εικόνα 13: Σχηματικό διάγραμμα και φωτογραφία από τη διαβάθμιση του εκτυπωμένου κράματος.

2.4.5 ΣΥΓΚΟΛΛΗΣΗ ΕΛΑΣΜΑΤΩΝ (SHEET LAMINATION)

Η συγκόλληση ελασμάτων είναι μια διεργασία προσθετικής κατασκευής με τη χρήση υλικών σε στέρεα μορφή. Η περισσότερο συνηθισμένη τεχνική αυτής της διεργασίας είναι η προσθετική κατασκευή με υπερήχους (ultrasonic additive manufacturing - UAM) ή αλλιώς ενοποίηση μέσω υπερήχων (ultrasonic consolidation - UC). Κατά αυτή τη τεχνική χρησιμοποιούνται ελάσματα μετάλλων που συγκολλούνται με υπερήχους. Συνήθη μέταλλα είναι ανοξείδωτο αλουμίνιο, χαλκός, τιτάνιο, αλουμίνιο. Συνήθως απαιτείται και επιπρόσθετη επεξεργασία με CNC εργαλειομηχανές για την απομόνωση αχρησιμοποίητου υλικού κατά τη συγκόλληση. Έχει γίνει επιτυχής καινοτόμα ανάπτυξη μεταλλικών FGM χρησιμοποιώντας ανοξείδωτο αλουμίνιο και φύλλα αλουμινίου και

χαλκού [23]. Τα φύλλα ενοποιήθηκαν μέσω συγκόλλησης με υπερήχους. Χρησιμοποιώντας το μηχάνημα για τη UC έγινε παραγωγή μηχανικών δονήσεων της τάξεως των 20kHz.



Εικόνα 14: Εικόνες οπτικής μικροσκοπίας FGM δοκμίου. Αριστερά, περιοχή εκτυπωμένου υλικού με τη μέθοδο ενοποίησης μέσω υπερήχων, με εμφανείς τις περιοχές διαφορετικών υλικών που χρησιμοποιήθηκαν. Δεξιά, εικόνα της μικροδομής του ίδιου υλικού.

2.4.6 ΨΕΚΑΣΜΟΣ ΥΛΙΚΟΥ (MATERIAL JETTING – MJ)

Η τεχνολογία ψεκασμού υλικού έχει τη δυνατότητα να παράγει κομμάτια από μεγάλη γκάμα πολυμερών, με μεγάλη χρωματική ποικιλία, με μακροσκοπικά λείες επιφάνειες, διατηρώντας ακρίβεια στις διαστάσεις, της τάξεως του 0.1mm, και όλα αυτά σε μία μόνο εκτύπωση [50]. Αυτή η διεργασία δίνει τη δυνατότητα διαμόρφωση λεπτών τοιχωμάτων και πολύπλοκων γεωμετριών. Σε αυτή τη μέθοδο εκτύπωσης (PolyJet), η κεφαλή εκτύπωσης εξωθεί φωτοπολυμερή ως υγρά σταγονίδια παράλληλα με το υλικό στήριξης, και μια υπεριώδης πηγή ακτινοβολίας, συνήθως UV λάμπα, βοηθάει στη σκλήρυνση του υλικού [51]. Συνήθως στη Polyjet τεχνολογία, ακολουθείται και ένα επιπρόσθετο βήμα σκλήρυνσης για τη ρύθμιση της ποιότητας της επιφάνειας [50], [52], [53]. Σε αυτό το είδος κατασκευαστικής μεθόδου υπάρχει μεγάλη ευελιξία όσον αφορά το τύπο και τον αριθμό ψεκαζόμενων υλικών. Έτσι μπορεί να θεωρηθεί κατάλληλη μέθοδος για τη κατασκευή FGM με πολύπλοκη γεωμετρία και σύνθεση της διαβάθμισης σε διαφορετικές κατευθύνσεις. Ωστόσο, υπάρχει μεγάλο ποσοστό υλικού που δε χρησιμοποιείται και οδηγείται στα απορρίμματα, όπως τα υλικά που χρησιμοποιούνται για τη δημιουργία του υποστρώματος [52].



Εικόνα 25: Δοκίμια εφελκυσμού κατασκευασμένα από διαφορετικά είδη εμπορικών υλικών (TangoBlackPlus και VeroCyan) για ψεκασμό υλικού με διαβάθμιση στη σύσταση.

2.4.7 ΠΡΟΚΛΗΣΕΙΣ ΓΙΑ ΤΗ ΠΡΟΣΘΕΤΙΚΗ ΚΑΤΑΣΚΕΥΗ ΣΕ ΣΧΕΣΗ ΜΕ ΤΑ FGMs

Οι κατασκευές που προέρχονται από τη ΠΚ είναι προς το παρόν επιρρεπείς σε εσωτερικά και εξωτερικά ελαττώματα, εξαιτίας των πολυάριθμων παραμέτρων και δυσκολία προσδιορισμού της συνδυαστικής επίδρασης που έχουν. Συνεπώς, τα ποιοτικά χαρακτηριστικά μπορούν να διαφοροποιούνται σε μεγάλο βαθμό μεταξύ παρτίδων ή και τύπου μηχανημάτων [3]. Η κατασκευή των FGAM με περίπλοκες εσωτερικές δομές και ακριβή κατανομή των φάσεων των διαφόρων

συστατικών στη μικροδομή σημαίνει ότι η ταχύτητα κατασκευής, ακρίβεια και αποτελεσματικότητα κατά την αλλαγή υλικών μεταξύ στρωμάτων πρέπει να βελτιωθεί για τα υπάρχοντα συστήματα ΠΚ [46]. Οι εμπορικά διαθέσιμες ΠΚ τεχνολογίες είναι ακόμα λειτουργικές σε ομογενή υλικά, ενώ εστιάζουν σε μια βασική γεωμετρική περιγραφή του αντικειμένου, όπου γίνεται ανάθεση μεμονωμένων υλικών σε συγκεκριμένες περιοχές. Επιπλέον, ο χαρακτηρισμός των ιδιοτήτων είναι η μεγαλύτερη πρόκληση για την FGAM και απαιτεί υψηλού επιπέδου και ακριβείας μετρήσεις *in situ* [3]. Παρόλο που υπάρχει ένα καθιερωμένο πλαίσιο μοντελοποίησης για να αποδειχθεί η προσέγγιση της μεταβλητής ιδιότητας διαβαθμισμένης κατασκευής, παραμένει η ανάγκη να εξεταστούν οι διαδικασίες και τα πρωτόκολλα που μπορούν να εγγυηθούν ένα πιο αξιόπιστο και προβλέψιμο αποτέλεσμα, ειδικότερα όταν εμπλέκεται η κατανομή των υλικών στις διάφορες φάσεις και η μετάβαση στις ιδιότητες μέσα στη δομή [44], όπως και ζητήματα σχετικά με την επιλογή των υλικών, το είδος της δομής και η ταχύτητα κατασκευής, για την παραγωγή FGAM με χαμηλότερο κόστος [54]. Για να γίνει μεταφορά στα λειτουργικά FGAM προϊόντα, καινοτόμα συστήματα τροφοδοσίας πρέπει να αναπτυχθούν.

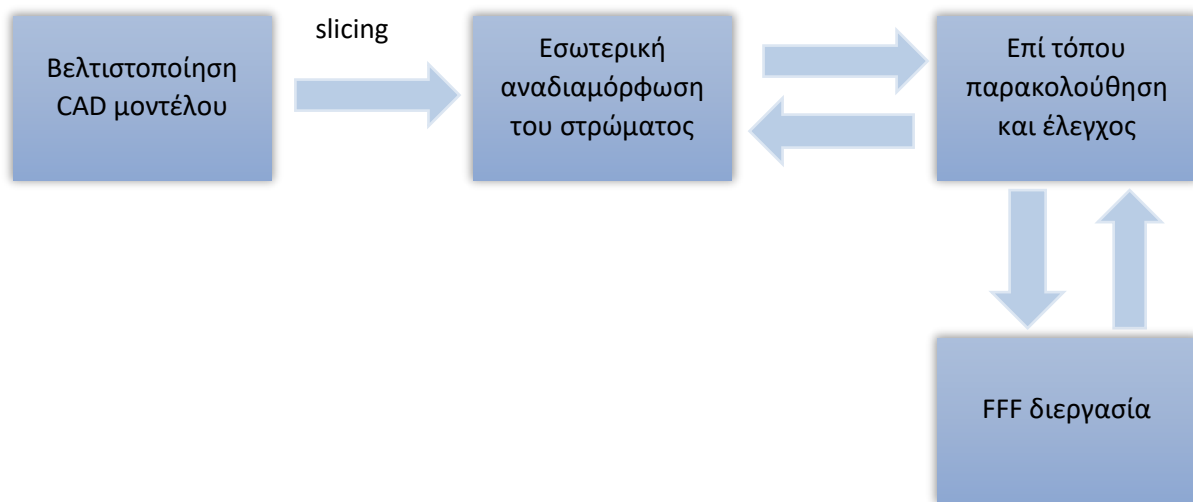
3 ΚΑΤΑΣΚΕΥΗ FGMs ΜΕΣΩ ΕΠΙΛΕΚΤΙΚΗΣ ΕΝΑΠΟΘΕΣΗΣ ΥΛΙΚΟΥ

3.1 ΒΑΣΙΚΕΣ ΠΑΡΑΜΕΤΡΟΙ ΤΡΙΣΔΙΑΣΤΑΤΗΣ ΕΚΤΥΠΩΣΗΣ ΠΛΕΓΜΑΤΙΚΩΝ ΔΟΜΩΝ

Ένα πολύ σημαντικό πλεονέκτημα της ΠΚ τεχνολογίας είναι η κατασκευή όχι μόνο περίπλοκων δομών αλλά και ιδιαίτερα ελαφρών δομών, χωρίς παράλληλα υποβάθμιση των μηχανικών τους ιδιοτήτων. Ειδικότερα, σε εφαρμογές της αυτοκινητοβιομηχανίας και αεροδιαστημικής, αυτός είναι ένας πολύ σημαντικός παράγοντας από όπου μπορεί να εξοικονομηθούν χρήματα λόγω της μείωσης κατανάλωσης καυσίμων και του συνολικού αποτυπώματος άνθρακα στο περιβάλλον [1], [3]. Για αυτό το σκοπό είναι αρκετά διαδεδομένες οι πλεγματικές δομές. Για το σχεδιασμό πλεγματικών δομών είναι πάντα απαραίτητο να γίνει η επιλογή και η βελτιστοποίηση της γεωμετρικής δομής που απαιτείται. Τα περισσότερα εμπορικά λογισμικά ψηφιακής σχεδίασης (CAD) έχουν περιορισμένες δυνατότητες και εργαλεία σχεδιασμού και προσομοίωσης πλεγματικών δομών, επομένως συχνά εφαρμόζεται η χρήση αλγορίθμων και υπολογιστικών μεθόδων.

Στη παρούσα ΔΕ, η τεχνολογία τριδιάστατης εκτύπωσης που χρησιμοποιήθηκε για την κατασκευή πλεγματικών δομών ήταν η Fused Filament Fabrication (FFF), η οποία ανήκει στην ευρύτερη κατηγορία των μεθόδων ΠΚ εξώθησης υλικού (Material Extrusion). Τα στοιχεία που παίζουν σημαντικό ποιοτικό ρόλο στο τελικό αποτέλεσμα αυτής της διαδικασίας είναι η διαστατική ακρίβεια των μεμονωμένων ινών πλαστικού που δημιουργούν την κάθε στρώση και το μοτίβο τοποθέτησής τους στον χώρο για την κατασκευή μιας πλεγματικής δομής. Η ποιότητα της κάθε στρώσης υλικού και η διαστατική ακρίβεια εξαρτώνται από το πάχος της στρώσης, τις ιδιότητες του υλικού, τη θερμοκρασία εκτύπωσης και κάποιες παραμέτρους της διεργασίας όπως τη διάμετρο του ακροφυσίου, την ακρίβεια του συστήματος ΠΚ και τον ρυθμό εναπόθεσης υλικού. Ο ρυθμός εναπόθεσης υλικού μελετήθηκε εκτενώς στα πειράματα που έγιναν, καθώς οι πλεγματικές δομές που παρήχθησαν αποτελούνταν από μεμονωμένες ίνες καθορισμένου ύψους/πλάτους, για την εξώθηση των οποίων έπρεπε να γίνει προσεκτική ρύθμιση της ταχύτητας εκτύπωσης σε συνδυασμό με την ογκομετρική παροχή του υλικού. Η απόκλιση της πλεγματικής δομής από την ονομαστική γεωμετρία είναι περισσότερο ορατή όταν στο προς κατασκευή αντικείμενο, το υλικό που εναποτίθεται από τη κεφαλή είναι παραπάνω από το ρυθμισμένο (overfill) με αποτέλεσμα να δημιουργούνται ανωμαλίες στις ίνες/εξογκώματα ή αντίστοιχα σε σημεία που είναι μικρότερη η εναπόθεση (underfills), δημιουργώντας ελλείψεις και κενά. Αυτά τα χαρακτηριστικά αποδίδονται ως ομαλότητα του στρώματος (layer evenness), και καθορίζονται επίσης από τις παραμέτρους τροχιάς της κεφαλής [55]. Η μεθοδολογία βελτιστοποίησης των παραπάνω παραμέτρων που ακολουθήθηκε περιλαμβάνει τα παρακάτω στάδια [55]–[57]:

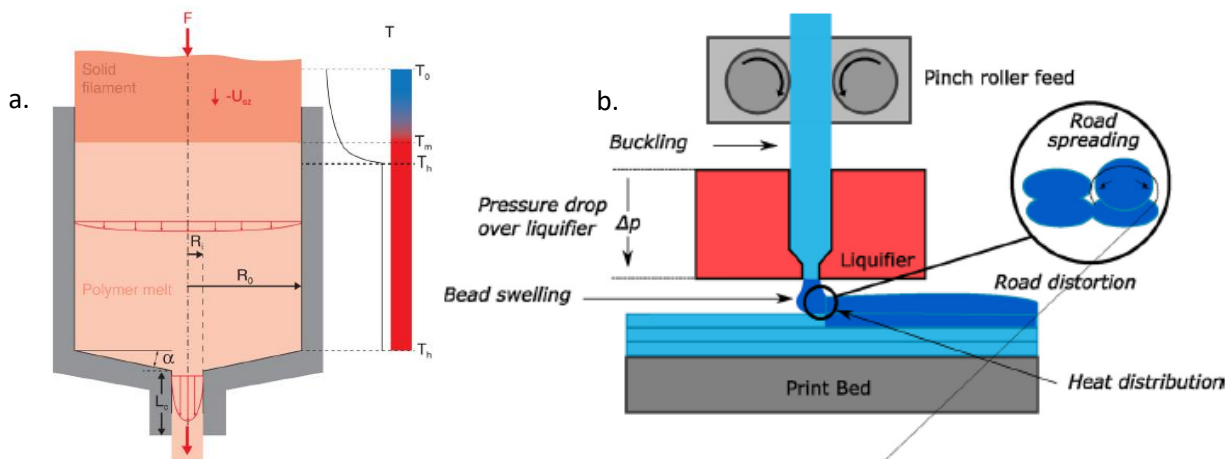
1. Βελτιστοποίηση των μοντέλων CAD, ανάλογα με το τελικό σκοπό, μεταβολή του προσανατολισμού του αντικειμένου, της βάσης και του διαχωρισμού του σχήματος σε επιμέρους σημεία (slicing).
2. Εσωτερικός σχεδιασμός του στρώματος. Αυτό περιλαμβάνει τη δημιουργία / βελτιστοποίηση της τροχιάς της κεφαλής, τον προγραμματισμό της εναπόθεσης και την επιλογή παραμέτρων διεργασίας.
3. Ανάπτυξη εξειδικευμένων αλγορίθμων δημιουργίας των τεχνολογικών προγραμμάτων κατασκευής και σύγκριση εκτυπωμένων δοκιμών με κώδικα που έχει παραχθεί μέσω CAM.
4. Ο επιτόπου έλεγχος και παρακολούθηση της FFF διεργασίας. Αυτό αφορά την ακριβή τοποθέτηση στους άξονες x, y, z , έλεγχος της τροφοδοσίας, και ανίχνευση των ελαττωμάτων κατά την κατασκευή.



Εικόνα 16: Σχηματικό διάγραμμα της πορείας βελτιστοποίησης παραμέτρων μιας εκτυπωτικής διαδικασίας.

3.2 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΤΕΧΝΙΚΗΣ ΕΝΑΠΟΘΕΣΗΣ ΥΛΙΚΟΥ- FUSED FILAMENT FABRICATION (FFF)

Στη τεχνολογία FFF, το υλικό τροφοδοσίας είναι θερμοπλαστικό με τη μορφή νήματος, το οποίο καταλήγει σε έναν θερμαινόμενο εξωθητή [58], [59]. Μέσω αυτής της τεχνικής το υλικό εξωθείται προς τα κάτω από ένα ακροφύσιο, καθώς η θερμοκρασία αυξάνεται και ξεπερνάει τη θερμοκρασία τήξης του υλικού, Εικόνα 17α [60]. Η αύξηση της θερμοκρασίας το καθιστά κατάλληλα ιξώδες για να γίνει η εξώθηση και η χρήση του ακροφυσίου είναι απαραίτητη για τον έλεγχο της διαμέτρου της ίνας που θα εξωθηθεί [60]. Το τήγμα εναποτίθεται σε ένα στρώμα ήδη εκτυπωμένου στερεοποιημένου υλικού, με αποτέλεσμα οι θερμές ίνες του υλικού να συγκολλούνται, Εικόνα 17b [59]. Η κατασκευή του αντικειμένου προκύπτει από αριθμητικό έλεγχο βασισμένο σε κάποιο κώδικα που ελέγχει τη κίνηση της κεφαλής σε όλη τη πλατφόρμα εναπόθεσης, σε όλα τα στρώματα εκτύπωσης, καθώς και τις διάφορες παραμέτρους της διαδικασίας.



Εικόνα 17: α. Μοντέλο τήξης και ροής του πολυμερούς μέσα στη κεφαλή του εκτυπωτή κατά την FFF, β. Βασικά στοιχεία της FDM/FFF διεργασίας.

Συνήθη υλικά που χρησιμοποιούνται σε αυτή τη διεργασία είναι τα θερμοπλαστικά, όπως τα πολυμερή ακρυλονουτριλο-βουταδιεν-στερένιο (ABS), Νύλον και πολυ(γαλακτικό οξύ) (PLA), με σύνηθες μέτρο ελαστικότητας μεταξύ 1.3-3.6 GPa και αντοχή 30-100MPa [61]. Φυσικά, οι μηχανικές αντοχές των τριδιάστατα εκτυπωμένων αντικειμένων μπορεί να διαφέρουν σημαντικά από τις

ιδιότητες των συμπαγών αντικειμένων που προκύπτουν μέσω χύτευσης και αυτό οφείλεται στον τρόπο που μια δομή μπορεί να σχηματιστεί. Επομένως, για να βελτιστοποιηθούν οι μηχανικές αντοχές των τελικών FFF αντικειμένων, πρέπει πρωτίστως να κατανοηθεί το πως η εκτυπωτική διαδικασία μπορεί να έχει αντίκτυπο στη τελική ποιότητα. Αυτό, όμως, δεν είναι εύκολα υλοποιήσιμο, καθώς ένας μεγάλος αριθμός κατασκευαστικών παραμέτρων υπεισέρχεται στην εκτυπωτική διαδικασία [59].

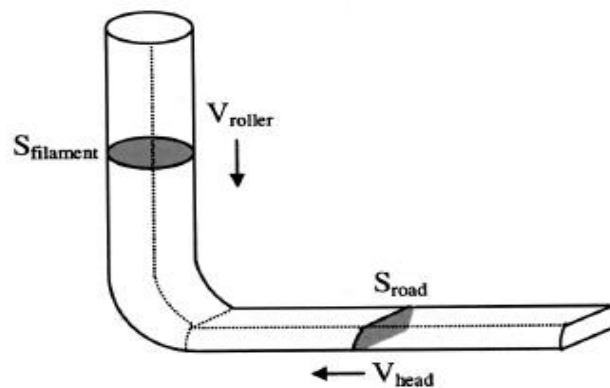
Ένα από τα μεγαλύτερα κατασκευαστικά προβλήματα είναι όταν δεν υπάρχει ομαλότητα σε ένα στρώμα. Αυτή η επιφανειακή ανωμαλία (overfills και underfills) θα συνεχίσει να υπάρχει στα επόμενα στρώματα και επηρεάζει τη ποιότητα του εκτυπωμένου αντικειμένου μετά το πέρας της εκτύπωσης. Τα overfills προκαλούνται όταν γίνεται εξώθηση υλικού παραπάνω από το απαιτούμενο και τα underfills προκαλούνται όταν γίνεται εξώθηση υλικού λιγότερο από το απαιτούμενο [55]. Προβλήματα επίσης με την τροφοδοσία παρουσιάζονται και όταν πρόκειται να στρίψει η κεφαλή, και όσο αυξάνεται η πολυπλοκότητα ενός σχήματος τόσο πιο συχνά εμφανίζεται αυτό το φαινόμενο. Εάν θεωρηθεί ότι έχουμε ισορροπία στο σύστημα και σταθερές συνθήκες ισχύει:

$$m_{in} = m_{out} \quad (1)$$

Όπου m_{in} (kg/s) και m_{out} (kg/s) αντιπροσωπεύουν τις μάζες του νήματος που οδηγείται στον εξωθητή και του ημι-στερεού υλικού ή αλλιώς ίνας που ρέει από το ακροφύσιο, αντιστοίχως. Ακόμη:

$$S_{road} = \frac{\rho_{in} V_{roller} S_{filament}}{\rho_{out} V_{head}} \quad (2)$$

Όπου ρ_{in} και ρ_{out} είναι οι πυκνότητες του νήματος και της ίνας, V_{roller} είναι η ταχύτητα εισροής του νήματος σε (m/s), V_{head} είναι η ταχύτητα κίνησης της κεφαλής σε (m/s) και $S_{filament}$ και S_{road} (m²) είναι οι διατομές του νήματος και της ίνας αντίστοιχα. Για να αποφευχθούν τα overfills και underfills εξαιτίας του μη επαρκούς ελέγχου της FFF εκτύπωσης σε κάθε στρώμα, πρέπει να γίνουν αλλαγές στη γεωμετρία της διατομής της ίνας και προσαρμογή με τις σχεδιασμένες τιμές, εννοώντας πως το S_{road} πρέπει να μείνει σταθερό [55].



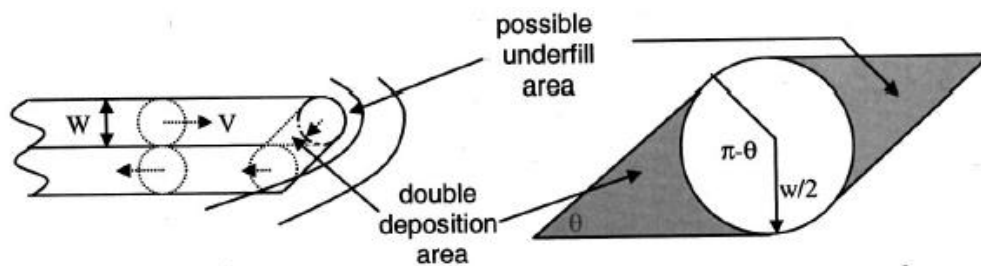
Εικόνα 18: Σχηματική αναπαράσταση σταθερής κατάστασης εναπόθεσης

Παρακάτω παρουσιάζονται εν συντομία οι βασικοί λόγοι δημιουργίας των overfills και underfills καθώς και η αντιμετώπισή τους:

- Σχηματισμός overfill εξαιτίας της μείωσης της ταχύτητας της κεφαλής στα σημεία στροφής. Σε αυτές τις περιοχές υπάρχει πολλή έντονη και συνεχής επιτάχυνση και επιβράδυνση. Για τη βελτίωση της κατάστασης μπορούν να εφαρμοστούν δύο συστήματα εκ των οποίων το ένα περιλαμβάνει την επιβράδυνση στην πλατφόρμα ως προς X-Y άξονες σε αυτό το σημείο και επιτάχυνση σε μια προκαθορισμένη ταχύτητα σε ένα συγκεκριμένο εναρκτήριο σημείο. Το άλλο σύστημα αφορά μια ενδιάμεση ταχύτητα μεταξύ της πρώτης προς τη δεύτερη κίνηση

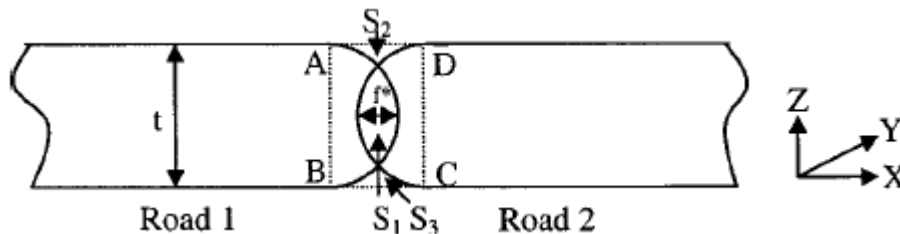
σύμφωνα με τη προκαθορισμένη επιβράδυνση/επιτάχυνση. Τα συστήματα αυτά ονομάζονται stop-at-turn και blended-move αντίστοιχα.

- Σχηματισμός overfill και underfill εξαιτίας της γεωμετρίας στις γωνίες. Υπάρχουν κάποια σημεία στροφής όπου το υλικό εναποτίθεται διπλά. Αυτές οι περιοχές ονομάζονται Double Deposition area (DDA). Αυτό οφείλεται στη γωνία θ κατά την οποία στρίβει η κεφαλή. Όσο μεγαλύτερη η γωνία, τόσο πιο πιθανό είναι να υπάρξει αυτή η διπλή εναπόθεση. Η περιοχή αυτή για να έχει τη μικρότερη πιθανότητα να εμφανιστεί θα πρέπει $\theta = \pi$. Επίσης, η DDA είναι ανάλογη του πλάτους της ίνας. Τα overfills αυξάνονται/μειώνονται όσο το πάχος της είναι αυξάνεται/μειώνεται. Τα underfills μπορούν να σχηματιστούν σε αυτές τις περιοχές σε μικρότερο βαθμό, ωστόσο αυτές οι περιοχές αντισταθμίζονται μέσω της αρνητικής μετατόπισης λόγω του γεμίσματος του περιγράμματος. Με λίγα λόγια, τα underfills μπορούν να εξαλειφθούν μέσω προσεκτικού σχεδιασμού διαδρομής του συστήματος, αλλά τα overfills στις στροφές είναι ουσιαστικά αναπόφευκτα.



Εικόνα 19: Εξογκώματα και εσοχές που δημιουργούνται κατά την εκτυπωτική διαδικασία

- Σχηματισμός overfill και underfill εξαιτίας της μη κατάλληλου πάχους ίνας ή μετατόπισης. Όταν ολοκληρώνεται η εναπόθεση ενός στρώματος, μεταξύ δύο γειτονικών οδών μπορεί να παρατηρηθούν είτε underfills είτε overfills. Για να υπάρξει μια ισότιμη επιφάνεια, δύο γειτονικές ίνες πρέπει να επικαλύπτονται. Στην Εικόνα 20 φαίνεται ότι το πλεόνασμα υλικού στην κοινή περιοχή εναπόθεσης (S_1) θα αντισταθμίσει το κενό στους χώρους πάνω και κάτω από τα ημικύκλια (S_2, S_3) και αν θεωρηθεί πως τα ρευστά είναι ασυμπίεστα τότε το άθροισμα των ημικυκλίων θα μας δώσει την S_1 επιφάνεια. Για να αποκτηθεί η ισότιμη επιφάνεια πρέπει να ακολουθηθούν δύο βήματα για τη προσαρμογή αυτών των δύο παραμέτρων.



Εικόνα 20: Επιθυμητή εναπόθεση των ινών

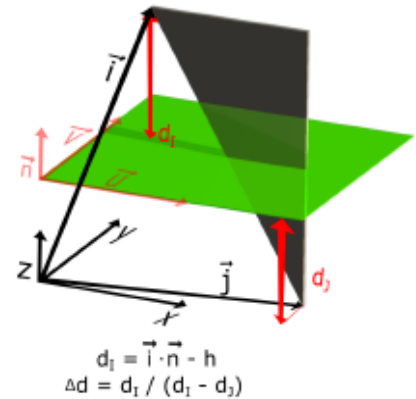
Αρχικά, όταν υπάρχει κάποιο κενό μεταξύ δύο γειτονικών ινών, είναι αρκετό είτε να αυξηθεί το πλάτος της ίνας είτε να μειωθεί η απόσταση μεταξύ των ινών. Αντίθετα, εάν κάποιο εξογκωμα εμφανίζεται μεταξύ τους, πρέπει να μειωθεί το πάχος της ίνας ή να αυξηθεί η απόστασή τους.

3.2.1 SLICING

Το αν θα είναι η εκτύπωση επιτυχής ή όχι εξαρτάται σημαντικά από τον σωστό τεμαχισμό του τριδιάστατου σχεδίου (3D slicer). Οι 3D slicers ανήκουν στην ευρύτερη κατηγορία των CAM (Computer-Aided-Manufacturing) λογισμικών που αφορούν τη 3D εκτύπωση και μετατρέπουν τα 3D ψηφιακά μοντέλα σε οδηγίες εκτύπωσης για τον εκτυπωτή. Ο 3D slicer ουσιαστικά «κόβει»/τεμαχίζει

το CAD μοντέλο σε οριζόντια στρώματα με βάση κάποιες ρυθμίσεις. Ταυτόχρονα, υπολογίζει πόσο υλικό είναι απαραίτητο για το χτίσιμο του αντικειμένου, αλλά και ο χρόνος που απαιτείται για την εκτύπωση. Όλες αυτές οι πληροφορίες οδηγούν στη δημιουργία ενός GCode αρχείου, το οποίο στέλνεται σα πληροφορία στο printer [62]. Πριν το τεμαχισμό του μοντέλου πρέπει να έχει γίνει προηγουμένως σχεδιασμός ή 3D σάρωση ενός φυσικού μοντέλου και η μετατροπή του σε .STL αρχείο. Η ονομασία των αρχείων STL προέρχονται από τη λέξη STereoLithography και μπορούν να εξαχθούν από τα περισσότερα CAD λογισμικά. Για αυτό το λόγο πλέον, η ονομασία STL μεταφράζεται ως «Standard Tessellation Language» και ο σκοπός αυτών των αρχείων είναι μόνο η περιγραφή της γεωμετρίας της επιφάνειας.

Οι αλγόριθμοι που ακολουθούνται από τα λογισμικά slicing βασίζονται στο χωρισμό ενός επιπέδου σε μικρά τρίγωνα. Ουσιαστικά ένα STL αρχείο δίνει τα απαιτούμενα τρίγωνα που αναπαριστούν την εξωτερική επιφάνεια του 3D μοντέλου, έτσι ώστε να ξεκινήσει η διαδικασία του τεμαχισμού. Ο τρόπος τεμαχισμού βασίζεται στη κοπή επιπέδων που είναι παράλληλα με τη βάση του επιπέδου, παρέχοντας ποικιλία στις τιμές κατά τον άξονα z, που αντικατοπτρίζει το ύψος [63]. Όπως φαίνεται στην Εικόνα 21, κάθε επίπεδο τομής διασταυρώνεται με τα διανύσματα u και v και κόβει τα επιμέρους τρίγωνα. Επί προσθέτα, το ύψος h του επιπέδου από τη βάση και το μοναδιαίο διάνυσμα n είναι γνωστά. Με τη βοήθεια των διαφόρων κορυφών των τριγώνων είναι εφικτό να υπολογισθεί η απόσταση από τις κορυφές στο επίπεδο όπου γίνεται η τομή [63]. Για να είναι εφικτή η δημιουργία πεπερασμένων τριγώνων για το χωρισμό της επιφάνειας πρέπει να ισχύουν οι ακόλουθες προϋποθέσεις [64]:



Εικόνα 21: Τρίγωνο κομμένο από ένα επίπεδο τομής (πράσινο) με τα διανύσματα χώρου i και j.

1. Κάθε άκρο πρέπει να μοιράζεται με άλλα δύο τρίγωνα
2. Ένα διάνυσμα να μοιράζεται με έναν αριθμό τριγώνων
3. Κάθε τρίγωνο πρέπει να έχει τουλάχιστον μια κορυφή κοινή με κάποιο άλλο τρίγωνο-συνδεσιμότητα
4. Εάν ένα διάνυσμα μοιράζεται με ένα δεύτερο τρίγωνο, τότε αυτό είναι επίσης διάνυσμα του δεύτερου τριγώνου
5. Κανένα τρίγωνο δεν έχει κάποιο σημείο τομής με το εσωτερικό άλλων τριγώνων.

Πρακτικά είναι πολύ σημαντικό να γνωρίζει κάποιος το πως λειτουργούν τα λογισμικά αυτά, ως προς τις παραμέτρους που εισάγει ο χρήστης. Η γνώση αυτών των παραμέτρων μπορεί να δώσει το σωστό υπόβαθρο για να μπορεί να διαχειριστεί αυτού του είδους τα προγράμματα και παρουσιάζονται παρακάτω [62]:

1. Ύψος στρώματος (Layer Height)

Το ύψος του στρώματος αντικατοπτρίζει τη ποιότητα της εκτύπωσης. Αυτή η παράμετρος ελέγχει το ύψος/πάχος του κάθε στρώματος. Τα εκτυπωμένα υλικά που έχουν μικρό πάχος στρώματος καταφέρουν να παρουσιάσουν περισσότερες λεπτομέρειες και πιο λείες επιφάνειες επειδή είναι δυσκολότερο να διακριθούν τα επιμέρους στρώματα. Με λίγα λόγια, όσο πιο μικρό το πάχος στρώματος τόσο καλύτερη η απόδοση του μοντέλου, με επίπτωση την αύξηση του χρόνου εκτύπωσης. Ως η βέλτιστη τιμή πάχους στρώματος θεωρείται τα 0.05 mm, κάτι το οποίο δεν είναι ρεαλιστικό σενάριο για τους περισσότερους FFF εκτυπωτές για λειτουργικούς λόγους. Σε πολλές περιπτώσεις, πάχος 0.1 mm δίνει ικανοποιητικά αποτελέσματα όσον αφορά τις λεπτομέρειες. Για

περιπτώσεις που δεν απαιτείται υψηλή λεπτομέρεια, χρησιμοποιείται πάχος της τάξεως 0.2 mm ή περισσότερο.

2. Πάχος φλοιού (Shell thickness)

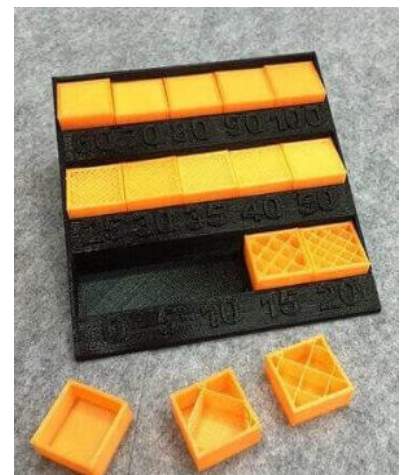
Για την εκτύπωση ενός αντικειμένου και πριν την εναπόθεση των επιμέρους στρωμάτων, είναι απαραίτητο ένα εξωτερικό κούφιο τοίχωμα/κέλυφος που να κρατάει σταθερή τη δομή σε κάθε αλλαγή στρώματος. Αυτό ουσιαστικά αποτελεί ένας από τους πιο σημαντικούς παράγοντες με τους οποίους η δομή διατηρείται κατά την εκτύπωση.

3. Αναρρόφηση υλικού (Retraction)

Αυτή η ρύθμιση δίνει την εντολή στον εκτυπωτή να «τραβήξει» προς τα πίσω το υλικό τροφοδοσίας, διακόπτοντας τη ροή υλικού από το ακροφύσιο. Αυτό συνήθως χρησιμοποιείται όταν υπάρχουν ασυνέχειες στις προς εκτύπωση επιφάνειες, όπου δεν πρέπει να υπάρχει υλικό. Η αναρρόφηση είναι συνήθως ενεργοποιημένη εκτός κι αν δεν υπάρχουν ασυνέχειες στο αρχικό προς εκτύπωση μοντέλο. Αυτή η ρύθμιση προκαλεί ορισμένες φορές και κάποια προβλήματα, καθώς το νήμα μπορεί να φράξει το ακροφύσιο.

4. Πυκνότητα του εσωτερικού (Fill Density)

Το πλέγμα πλήρωσης ή αλλιώς infill αναφέρεται στη πυκνότητα του χώρου μέσα από τον εξωτερικό φλοιό του αντικειμένου που γεμίζεται με ένα προκαθορισμένο πλεγματικό μοτίβο, η πυκνότητα του οποίου καθορίζεται σε ποσοστό επί τοις εκατό %. Εάν ένα αντικείμενο έχει εκτυπωθεί με 100% infill, σημαίνει πως εσωτερικά είναι συμπαγές στερεό. Όσο μεγαλύτερο είναι αυτό το ποσοστό τόσο πιο δυνατό και βαρύ θα είναι το αντικείμενο αλλά απαιτείται μεγαλύτερη διάρκεια στην εκτύπωση αλλά και ποσότητα υλικού. Όπως μπορεί να συμπεράνει κανείς, αυτό μπορεί να αποδειχθεί αρκετά ακριβό και χρονοβόρο εάν κάθε φορά χρησιμοποιείται 100% infill. Για λειτουργικά αντικείμενα, συνήθως ένα 75-100% infill είναι απαραίτητο. Τα μοτίβα πλήρωσης που υποστηρίζονται από τα συνήθη slicing λογισμικά είναι 'ομογενή' ως προς τις ιδιότητές τους σε επίπεδο στρώσης, διατηρώντας ίδιο προσανατολισμό, πυκνότητα και απόσταση μεταξύ των μεμονωμένων ινών, συνεπώς είναι αρκετά περιορισμένα ως προς τις γεωμετρικές παραμέτρους που μπορούν να οριστούν από τον χρήστη και δεν υποστηρίζουν προς το παρόν δυνατότητα διαβάθμισης της πυκνότητας σε 2 ή 3 διαστάσεις.



Εικόνα 22: Παρουσίαση διαφορετικού ποσοστού infill σε δοκίμια.

5. Ταχύτητα εκτύπωσης (Print Speed)

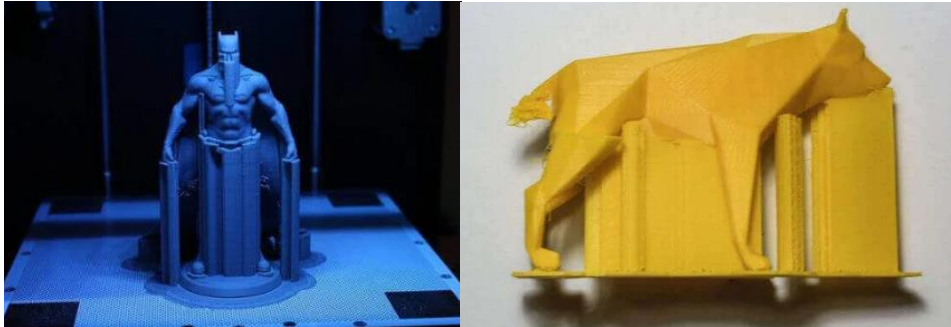
Η ταχύτητα με την οποία κινείται η κεφαλή καθώς εναποθέτει το νήμα, ονομάζεται ταχύτητα εκτύπωσης. Η ρύθμιση αυτής της παραμέτρου εξαρτάται συνήθως από τον σχεδιασμό του αντικειμένου, το νήμα που χρησιμοποιείται και το ύψος του στρώματος. Εάν μια δομή είναι πολύπλοκη, χαμηλή ταχύτητα εκτύπωσης προτιμάται, γύρω στα 20-50 mm/s, για να προσδώσει υψηλής ποιότητας χαρακτηριστικά. Η χρήση πολύ υψηλής ταχύτητας με σκοπό τη ταχεία εκτύπωση, μπορεί να δημιουργήσει επιπλοκές και απόκλιση από την ονομαστική γεωμετρία.

6. Χρήση στηριγμάτων (Supports)

Τα στηρίγματα βοηθούν συγκεκριμένες δομές ενός 3D αντικειμένου να χτιστούν πάνω σε κάποια βάση διότι είναι δύσκολο ή και ανέφικτο να χτιστούν διαφορετικά. Εφόσον όλα τα προς εκτύπωση αντικείμενα εκτυπώνονται σε στρώματα, μέρη των οποίων η διεύθυνση εκτύπωσης ξεπερνά τη γωνία 45 μοιρών κατά τον άξονα Z, είναι απαραίτητο να υπάρχει κάποιο υπόβαρθο για την εφαρμογή του

πρώτου στρώματος. Τα κριτήρια για να αποφασίσει κάποιος εάν πρέπει να χρησιμοποιήσει στήριγμα κατά την εκτύπωση είναι:

- Οτιδήποτε που μοιάζει με το σχήμα Y μπορεί να εκτυπωθεί χωρίς στήριγμα διότι υπάρχει μια βαθμιδωτή ράμπα που διαθέτει αρκετό υλικό από κάτω για να διατηρηθεί στη σωστή θέση και να μην υπάρχει «κρέμασμα» του υλικού.
- Δομές με σχήμα παρόμοιο με το H, είναι πιθανό να παρουσιάσουν κάποιο «κρέμασμα» υλικού λόγω βαρύτητας. Σε αυτές τις περιπτώσεις, το στήριγμα είναι προερατικό.
- Οποιοδήποτε σχήμα παρόμοιο με το T μπορεί να παρουσιάσει «κρέμασμα» και η παρουσία του στηρίγματος είναι απαραίτητη.



Εικόνα 23: Στην εικόνα από τα αριστερά παρουσιάζονται στηρίγματα για το δείγμα σε όλο το αντικείμενο, ενώ στην εικόνα δεξιά έχουν κατασκευαστεί στηρίγματα σε συγκεκριμένα σημεία.

7. Είδος πρόσφυσης πλατφόρμας (Platform Adhesion Type)

Αυτού του είδους η παράμετρος επηρεάζει ιδιαίτερα το πως το εκτυπώμενο στοιχείο θα κολλήσει στη πλατφόρμα εκτύπωσης. Στο κάτω μέρος του αντικειμένου, μπορεί να υπάρξει κάποιου είδους κάμψη κατά την εκτύπωση και να είναι η κύρια αιτία που δεν γίνεται καλή προσαρμογή στη πλατφόρμα εκτύπωσης. Ωστόσο υπάρχουν δύο τρόποι για να γίνει καλύτερη προσαρμογή και να μη παρουσιάζονται τέτοιου είδους φαινόμενα:

- Raft: Ένα υπόστρωμα με οριζόντιο πλέγμα το οποίο δρα ως μια ενδιάμεση πλατφόρμα για να γίνει η σωστή προσαρμογή με τη πλατφόρμα εκτύπωσης. Το raft μπορεί να αφαιρεθεί από το αντικείμενο μετά το τέλος της εκτύπωσης.
- Brim: Το υπόστρωμα αυτό έχει τη μορφή περιγράμματος στη βάση του προς εκτύπωση αντικειμένου, το οποίο κρατάει τις άκρες του μοντέλου κάτω, χωρίς να αφήνει σημάδια στη βάση του. Τα brims είναι απαραίτητα για να σταθεροποιηθούν ευαίσθητα σημεία τα οποία είναι περισσότερο απονωμένα από το υπόλοιπο σχήμα.



Εικόνα 24: Κατασκευασμένο δείγμα με brim για υπόστρωμα

8. Αρχικό πάχος στρώματος (Initial Layer Thickness)

Η διαφορά αυτής της παραμέτρου με το πάχος κάθε στρώματος είναι ότι το αρχικό στρώμα αναφέρεται στα mm υλικού που θα χρησιμοποιηθούν στη πρώτη στρώση στη πλατφόρμα εκτύπωσης πριν γίνει η εφαρμογή του infill. Αυτό βοηθάει στο να είναι πιο άθραυστη και σταθερή η βάση.

Τα επικρατέστερα λογισμικά τα οποία χρησιμοποιούνται για την εκτυπωτική διαδικασία είναι τα Cura, Slic3r, Simplify3D, Markerbot Desktop.

4 ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Όπως αναφέρθηκε, τα εμπορικά διαθέσιμα λογισμικά slicing έχουν ιδιαίτερα περιορισμένες δυνατότητες υπολογισμού τροχιάς και κατασκευής FGM δομών. Στην παρούσα ΔΕ επιλέχθηκαν 2 τρόποι κατασκευής δομών με διαβαθμισμένο πορώδες. Στην πρώτη μέθοδο, έγινε σχεδιασμός του πλέγματος μέσω CAD και στη συνέχεια το 3D μοντέλο εισάχθηκε σε slicing λογισμικό για την αυτόματη εξαγωγή του G-κώδικα. Στην δεύτερη μέθοδο, αναπτύχθηκαν εργαλεία υπολογισμού της τροχιάς και εξαγωγής G-κώδικα σε προγραμματιστικό περιβάλλον, με δυνατότητα καθορισμού όλων των γεωμετρικών χαρακτηριστικών της πλεγματικής δομής. Αυτός ο τρόπος, βασίζεται στην ανάλυση της λειτουργίας ενός slicing λογισμικού με σκοπό την εξαγωγή πληροφοριών σχετικά με τον τρόπο λειτουργίας του. Με τον τρόπο αυτό, δίνεται στον χρήστη τη δυνατότητα να αποφασίσει εκείνος ποιες παραμέτρους και πως θα τις ορίσει στο σύστημα αλλά και τη διαδρομή που θα ακολουθήσει η κεφαλή εκτύπωσης. Αυτό μπορεί να επιτευχθεί με την ανάπτυξη κώδικα από τον χρήστη και τη μετατροπή αυτού του κώδικα σε G-code (Customized toolpath generation). Μέσω αυτής της προσέγγισης υπάρχει η δυνατότητα άμεσης παρέμβασης στο σύστημα και δεν περιορίζεται από τον 'κλειστό' τρόπο υπολογισμού της τροχιάς που γίνεται στα εμπορικά slicing λογισμικά, ο οποίος μπορεί να εισάγει σημαντικά κατασκευαστικά σφάλματα σε FGM δομές. Συνεπώς, σκοπός αυτής της ΔΕ είναι η σύνταξη ρουτινών για την εξαγωγή G-code και η σύγκριση των αποτελεσμάτων με τις δομές που προκύπτουν από την αυτόματη εξαγωγή G-code μέσω slicing.

4.1 ΣΥΝΤΑΞΗ ΚΩΔΙΚΑ ΓΙΑ ΤΗΝ ΕΞΑΓΩΓΗ G-CODE

4.1.1 ΚΑΤΑΣΚΕΥΗ ΟΡΘΟΓΩΝΙΚΗΣ ΔΟΜΗΣ

4.1.1.1 ΔΙΑΒΑΘΜΙΣΗ ΠΟΡΟΔΟΥΣ 2.5X2.5, 5X5 ΕΩΣ 10X10 MM

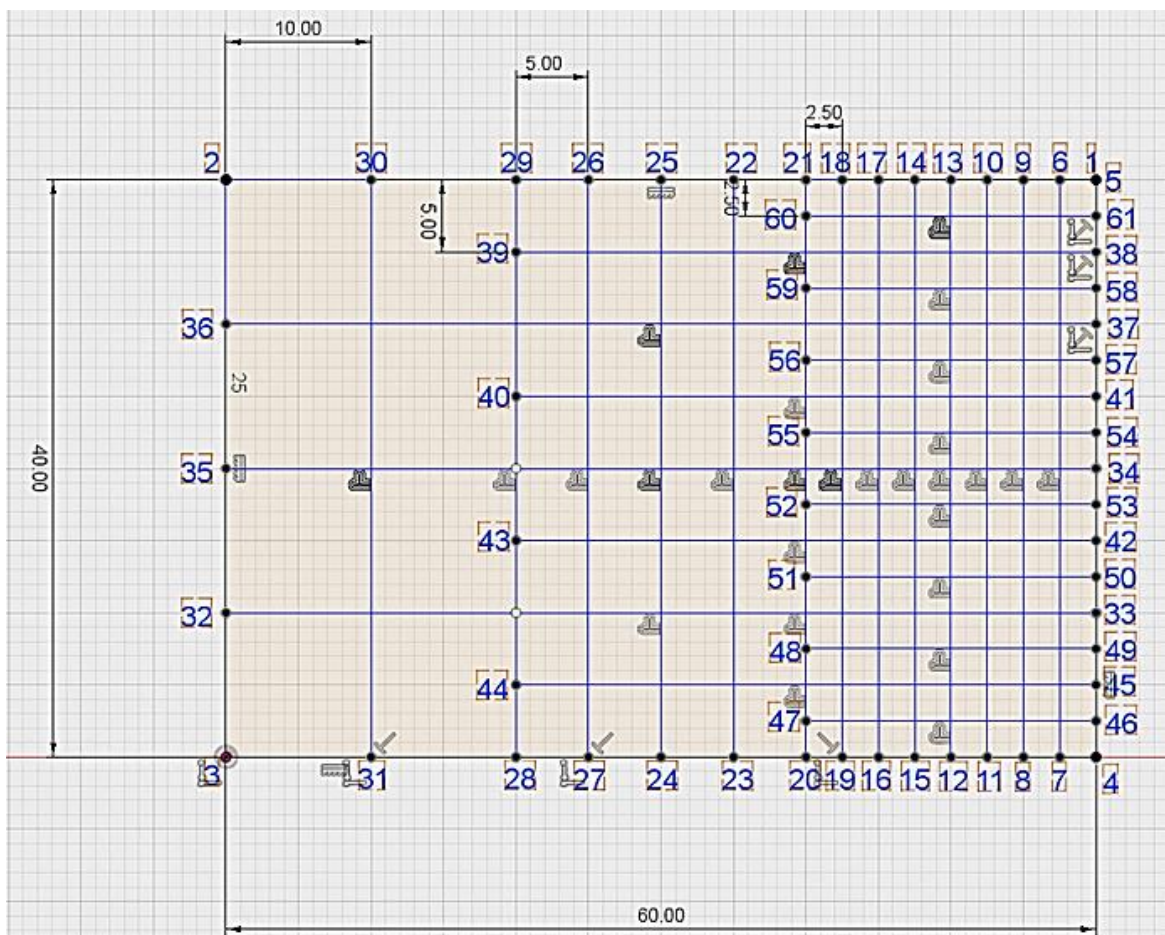
Η πρώτη φάση της πειραματικής διαδικασίας περιλαμβάνει το σχηματισμό μιας απλής πλεγματικής δομής, διαβαθμισμένου πορώδους με σχήμα πόρων τετράγωνο σε 2D, όπως φαίνεται στην Εικόνα 23. Το μέγεθος του δοκιμίου είναι 60 x 40 mm, σχήματος ορθογωνίου παραλληλογράμμου και γίνεται σχηματισμός τριών περιοχών τετραγωνικού σχήματος πόρων, τριών περιοχών διαβάθμισης, με μέγιστο μέγεθος πόρου 10 x 10 mm, μεσαίο μέγεθος πόρου 5 x 5 mm και μικρότερο μέγεθος πόρου 2.5 x 2.5 mm. Είναι εμφανές ότι η αύξηση ή αντίστροφα μείωση του μεγέθους των πόρων γίνεται πολλαπλασιάζοντας ή διαιρώντας τον αρχικό πόρο με τον αριθμό 2. Για παράδειγμα εάν θεωρήσουμε για αρχικό πόρο τον 2.5 x 2.5 mm αν $2.5 \times 2 = 5 \text{ mm}$ και $5 \times 2 = 10 \text{ mm}$, και αντίστροφα εάν θεωρήσουμε για αρχικό μέγεθος πόρου το 10 x 10 mm. Στην Εικόνα 23 γίνεται αναπαράσταση των ευθύγραμμων τμημάτων που αποτελούν την εξωτερική ορθογώνια δομή καθώς και το ζητούμενο πλέγμα. Τα αριθμητικά σημεία που αναπαρίστανται στο σχήμα αποτελούν τη διαδρομή που επιθυμούμε να ακολουθήσει η κεφαλή/ακροφύσιο (nozzle) του 3D printer.

Ανάλυση της μεθόδου επεξεργασίας του προβλήματος

Σε ένα ορθοκανονικό σύστημα αξόνων κάθε ευθύγραμμο τμήμα που σχηματίζεται αποτελείται από ένα αρχικό και ένα τελικό σημείο, δηλαδή από ένα αρχικό και τελικό ζεύγος συντεταγμένων (X, Y). Στη δομή αυτή υπάρχουν δύο είδη ευθύγραμμων τμημάτων: αυτά που κατά μήκος τους η κεφαλή εξωθεί υλικό (extrusion - L_{ei} όπου $i = 1, \dots, 32$) και αυτά στα οποία η κεφαλή κατά μήκος τους κάνει απλή μετάβαση χωρίς να εξωθεί υλικό (transition - L_{ti} όπου $i = 1, \dots, 28$), δηλαδή υπάρχουν 32 ευθύγραμμα τμήματα που γίνεται εξώθηση υλικού και 28 όπου γίνεται απλή μετάβαση χωρίς εξώθηση. Μετά από αυτή τη θεώρηση έγινε συγγραφή κώδικα σε γλώσσα προγραμματισμού Python 2.7 και εξαγωγή αυτών των δυάδων σε μορφή λίστας. Η μορφή των

δεδομένων έγινε σε δυάδες σημείων που αντιστοιχούν στην αρχή και στο τέλος κάθε κατηγορίας L_i , όπως π.χ. για το L_{e1} {S1,S2}, όπου S1 είναι το σημείο με τον αριθμό 1 που αντιστοιχεί στα (X1, Y1) και αντίστοιχα για το L_{t1} {S5,S6}. Ο δείκτης i που χρησιμοποιείται για την αρίθμηση των L_i αλλά υποδηλώνει και ποιο κατασκευάζεται πρώτο, δεύτερο κ.ο.κ. Στο κώδικα της Python ορίστηκαν σε κλάσεις οι συντεταγμένες (X_i, Y_i) , ο υπολογισμός της απόστασης μεταξύ 2 σημείων και τα ζεύγη των S_i για να υπολογιστούν τα L_i .

Στη συνέχεια έγινε συγκέντρωση του ζεύγους των συντεταγμένων (X, Y) , που σχηματίζουν την εξωτερική περίμετρο του δοκιμίου αλλά και το πλέγμα. Θεωρούμε ως αρχικό σημείο που βρίσκεται η κεφαλή το S3 με συντεταγμένες $(X3, Y3) = (110.000, 120.000)$. Για να βρεθούν τα ζεύγη $[X, Y]$ βάσει της διαδρομής που απαιτείται και των λόγων της κατασκευής γραμμικών ευθύγραμμων τμημάτων, γίνεται η δημιουργία γραμμικών συναρτήσεων, δηλαδή της μορφής $y = a*x + b$. Η λίστα που κατασκευάστηκε επιτρέπει την εξαγωγή των δεδομένων με τα ζεύγη των συντεταγμένων σε μορφή πίνακα. Να σημειωθεί ότι οι συντεταγμένες κάθε σημείου είναι σε mm, με ακρίβεια 3^{ου} δεκαδικού ψηφίου.

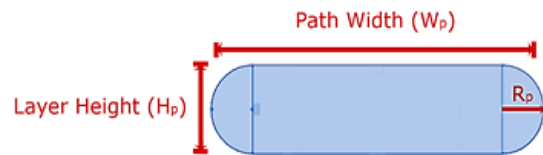


Εικόνα 25: Αναπαράσταση 2D της πλεγματικής δομής που σχεδιάζεται με customized κώδικα, με αναφορά στα αρχικά και τελικά σημεία κάθε θήματος, των διαστάσεων και της διαβάθμισης του πορώδους.

Έπειτα έγινε συγκέντρωση των τμημάτων που γίνεται εξώθηση υλικού, δηλαδή των L_{ei} με την απόλυτη τιμή του μήκους των ευθύγραμμων τμημάτων που σχηματίζονται, D_{kj} όπου k το αρχικό σημείο και j το τελικό σημείο ενός ευθύγραμμου τμήματος. Η λίστα αυτή κατασκευάστηκε με συγγραφή κώδικα και η εξαγωγή της έγινε σε αρχείο text με τα ζεύγη L_{ei} , D_{kj} σε μορφή πίνακα, διατηρώντας ακρίβεια 3^{ου} δεκαδικού ψηφίου.

Το επόμενο βήμα για την εξέλιξη του G κώδικα απαιτεί τον υπολογισμό της παραμέτρου E, δηλαδή το ποσό του υλικού που πρέπει να εξωθήσει το ακροφύσιο ή αλλιώς Extrusion parameter. Ο υπολογισμός της παραμέτρου E αφορά κάθε τμήμα που γίνεται εξώθηση υλικού, δηλαδή κάθε τμήμα L_{ei} , με το αντίστοιχο D_{ij} . Για τον υπολογισμό της παραμέτρου E κάνουμε τη παρακάτω θεώρηση:

Το υλικό που χρησιμοποιείται είναι ένα θερμοπλαστικό που όταν λιώνει εναποτίθεται σαν ίνα κατά μήκος των L_{ei} τμημάτων. Τα επιθυμητά γεωμετρικά χαρακτηριστικά των ινών αυτών είναι να έχουν μήκος D_{ij} , **layer height $H_p = 200 \mu\text{m}$** και πλάτος **$W_p = 200 \mu\text{m}$** (Σχ.26). Για την κατασκευή θα χρησιμοποιηθεί μεταλλικό ακροφύσιο κυκλικής διατομής **nozzle_diameter = 200 μm , ίσο με το W_p** . Ο υπολογισμός του όγκου κάθε ίνας μπορεί να γίνει κάνοντας διαφορετικές παραδοχές για τη διατομή της (ελλειψοειδής, ορθογώνια ή σύνθετη) και σύμφωνα με τις τιμές που έχουν δοθεί για το ύψος και το πλάτος της [65]. Για την περίπτωση που μελετάται θα θεωρήσουμε ότι η διατομή της ίνας (A_p) προκύπτει από το άθροισμα ενός ορθογωνίου και δύο ημικυκλίων (διαμέτρου ίσης με το ύψος της στρώσης). Χρησιμοποιώντας το ισοζύγιο μάζας, χωρίς να ληφθεί υπόψη η διαστολή του νήματος (σταθερή πυκνότητα), μπορούμε να γράψουμε τη σχέση ισότητας μεταξύ του όγκου του νήματος που εισέρχεται και εκείνου της ίνας που εξέρχεται, συναρτήσει των διατομών και του μήκους τους A_f , l_f (f: filament/νήμα) και A_p , l_p (p: path/ίνα) αντίστοιχα, όπου **$A_f = 2,74889 \text{ mm}^2$** το εμβαδόν της διατομής του νήματος τριδιάστατης εκτύπωσης. Έχοντας το εμβαδό της διατομής της ίνας, του νήματος και το μήκος που πρέπει να έχει η ίνα, μπορεί να γίνει ο υπολογισμός του μήκος νήματος l_f που πρέπει να ωθήσει το σύστημα τροφοδοσίας προς τον θάλαμο τήξης.



Εικόνα 26: Σχηματική αναπαράσταση της διατομής A_p των ινών θερμοπλαστικού που εναποτίθενται από την κεφαλή εκτύπωσης.

$$A_p l_p = A_f l_f \quad (3)$$

$$A_p = W_p H_p - \left(1 - \frac{\pi}{4}\right) H_p^2 \quad (4)$$

Για παράδειγμα, ίνα μήκους 10 mm, πλάτους 0.4 mm και ύψους 0.2 mm, έχει όγκο ίσο περίπου με 0.7 mm^3 και χρειάζεται 0.29 mm νήματος διαμέτρου 1.75 mm για να σχηματιστεί.

Δύο βασικές παράμετροι του G κώδικα που επηρεάζουν άμεσα το υλικό κατεργασίας είναι η παράμετρος E, που μετρείται σε mm και ορίζει την ονομαστική απόσταση κίνησης του θερμοπλαστικού νήματος για να εισέλθει στην κεφαλή εκτύπωσης (ίση με l_f) και η παράμετρος F (σε mm/min), που ορίζει την ταχύτητα πρόωσης των κινούμενων μερών του εκτυπωτή (ταχύτητα κίνησης της κεφαλής κατά τον X και κατά τον Y άξονα), καθώς και του βηματικού κινητήρα που ρυθμίζει τον ρυθμό τροφοδοσίας του υλικού προς την κεφαλή και κατ' επέκταση την ογκομετρική παροχή του υλικού (ίνας θερμοπλαστικού) που εξέρχεται από το ακροφύσιο. Οι σχέσεις που συνδέουν τα E, F με τον όγκο υλικού που εξωθείται V_p , της ογκομετρικής παροχής Q και της ταχύτητας εξώθησης από το ακροφύσιο u_{ex} είναι:

$$V_p = A_f E, \text{ όπου } E = l_f \text{ (mm)} \quad (5)$$

$$Q = \dot{V}_p = A_f F \text{ (mm/sec)} \quad (6)$$

$$A_p u_{ex} = F A_f \rightarrow u_{ex} = \frac{A_f}{A_p} F \quad (7)$$

Η παράμετρος F (Feed rate: ταχύτητα εκτύπωσης) θεωρείται ίση με 1200 mm/min. Σύμφωνα με τις παραπάνω σχέσεις γίνεται συγγραφή κώδικα για τον υπολογισμό της παραμέτρου E σε κάθε τμήμα και εξάγεται λίστα η οποία περιλαμβάνει μια τριπλέτα στοιχείων, δηλαδή $[X_j Y_j E_i]$, όπου $X_j Y_j$ οι συντεταγμένες στο άκρο του L_{ei} . Τα E_i να είναι σε mm και με ακρίβεια 3^{ου} δεκαδικού.

Για να λειτουργήσει σωστά ο G κώδικας απαιτείται η μετατροπή της προηγούμενης συνάρτησης ο υπολογισμός νέας παραμέτρου E σε κάθε βήμα που χρειάζεται να γίνεται η εξώθηση υλικού. Αυτό γίνεται με τον υπολογισμό αθροιστικών τιμών των E_i τα νέα E'_i θα υπολογιστούν για κάθε L_{ei} σύμφωνα με τη σχέση:

$$E'_i = E_i + \sum_{m=1}^{m=i-1} E_m \text{ για } i \geq 2 \quad (8)$$

Δηλαδή στην εκάστοτε τιμή του E_i θα προστίθενται όλες οι προηγούμενες τιμές των E_i πάντα σύμφωνα με τη σειρά κατασκευής.

Τελευταίο κομμάτι του προβλήματος, αποτελεί η έκφραση των δεδομένων από τον κώδικα της Pythοn σε G-κώδικα. Οι βασικές εντολές που χρησιμοποιήθηκαν είναι οι G0 και G1. Οι εντολές G εξαρτώνται κυρίως από τις παραμέτρους θέσης, δηλαδή τις συντεταγμένες ενός σημείου X, Y, Z σε mm, που στη προκειμένη περίπτωση γίνεται εστίαση σε δύο από τις τρεις συντεταγμένες, δηλαδή X,Y. Η διαφορά της G0 με τη G1 έγκειται στο γεγονός, ότι η G1 αφορά την ελεγχόμενη κίνηση της κεφαλής και του συστήματος τροφοδοσίας, ενώ η G0 αφορά τη γρήγορη κίνηση της κεφαλής σε μια καθορισμένη θέση χωρίς τη λειτουργία του συστήματος τροφοδοσίας. Επομένως, απαραίτητο για την εντολή G1 είναι και η παράμετρος E καθώς και η F.

Παρακάτω παρατίθεται ο κώδικας, σε μορφή ψευδοκώδικα, που αναπτύχθηκε σε Pythοn, όπου περιεγράφηκε στα προηγούμενα βήματα.

Κλάση Point: Αντιπροσωπεύει ένα σημείο στο καρτεσιανό επίπεδο.

x, y: Αρχικοποίηση συντεταγμένων στο (0,0).

n: Αρχικοποίηση αριθμού σημείου στο 0.

Συνάρτηση to_string():

επέστρεψε το Point σε μορφή "[X0.000,Y0.000]"

Συνάρτηση distance(Point p):

επέστρεψε την απόσταση του σημείου μας (x,y) από το p(p.x, p.y).

Κλάση Line: Αντιπροσωπεύει μια ευθεία στο καρτεσιανό επίπεδο.

Περιέχει 2 Point που είναι οι 2 άκρες τις ευθείας.

a = Point()

b = Point()

isLe: Flag που δηλώνει αν η ευθεία πρέπει να εκτυπωθεί από τον printer. Αρχικοποιείται ως ψευδές.

dist: Το μήκος της ευθείας. Αρχικοποιείται ως 0.

E: Extrusion αν η ευθεία πρέπει να εκτυπωθεί (isLe αληθές)

```

Συνάρτηση to_string():
    επέστρεψε τη Line σε μορφή "[S1, S2]"
Συνάρτηση line_distance():
    υπολόγισε και επέστρεψε το μήκος της ευθείας.
Κλάση Grid: Αντιπροσωπεύει το Grid το οποίο θα εκτυπώσει ο printer.
    lines[]: Πίνακας με τις ευθείες του Grid.
    points[]: Πίνακας με τα σημεία του Grid.
    active_point: Το σημείο που βρίσκεται η κεφαλή.
    F: Ταχύτητα εκτύπωσης. Αρχική τιμή 1200.
    Wp: Διάμετρος ακροφυσίου. Αρχική τιμή 0.200.
    Hp: Ύψος στρώματος. Αρχική τιμή 0.200.
    Af: Διατομή νήματος. Αρχική τιμή 2.74899
    Ap: Διατομή ίνας. Αρχική τιμή 0.
    lp: Μήκος ίνας. Αρχική τιμή 0.
    E: Extrusion. Αρχική τιμή 0.
Συνάρτηση save_active_point():
    αποθήκευση του active_point στον πίνακα points.
Συνάρτηση initialize(starting_x, starting_y):
    # αρχικοποίηση και αποθήκευση του active_point στα σημεία starting_x και
starting_y.
    active_point.x = starting_x
    active_point.y = starting_y
    save_active_point()
Συνάρτηση up(distance):
    # μετακίνηση και αποθήκευση του active_point κατά απόσταση distance προς τα
πάνω.
    active_point.y = active_point.y + distance
    save_active_point()
Συνάρτηση down(distance):
    # μετακίνηση και αποθήκευση του active_point κατά απόσταση distance προς τα
κάτω.
    active_point.y = active_point.y - distance
    save_active_point()
Συνάρτηση right(distance):
    # μετακίνηση και αποθήκευση του active_point κατά απόσταση distance προς τα
δεξιά.
    active_point.x = active_point.x + distance
    save_active_point()

```

```

Συνάρτηση left(distance):

    # μετακίνηση και αποθήκευση του active_point κατά απόσταση distance προς τα
αριστερά.

    active_point.X = active_point.X - distance

    save_active_point()

Συνάρτηση up_left(distance):

    # μετακίνηση και αποθήκευση του active_point κατά απόσταση distance διαγώνια
προς τα πάνω και αριστερά.

    active_point.y = active_point.y + distance

    active_point.x = active_point.x - distance

    save_active_point()

Συνάρτηση calculateLines():

    # υπολογισμός του flag isLe ευθειών του σχήματος Grid.

    Για i από 1 έως 4:

        αποθήκευσε στον πίνακα lines[] το Line(i, i+1) με isLe αληθές.

    Για i από 5 έως 60:

        αποθήκευσε στον πίνακα lines[] το Line(i, i+1)

        με isLe αληθές αν το i είναι άρτιος.

        με isLe ψευδές αν το i είναι περιττός.

Συνάρτηση save_lines_to_file():

    αποθήκευση των ευθειών του πίνακα lines[] στο αρχείο
"cs1_line_extrusion_transition_list.txt"

    διαχωρισμένες σε isLe αληθές και ψευδές,

    στη μορφή "Le = { [S1, S2], ...} Lt = { [S5, S6], ...}"

Συνάρτηση calculate_points(X_dist, Y_dist, starting_x, starting_y):

    # υπολογισμός σημείων του σχήματος Grid.

    X_dist: μήκος του Grid.

    Y_dist: ύψος του Grid.

    gradeLength: μήκος δοκιμίου. X_dist / 3

    first_gradeLength: απόσταση του πρώτου τμήματος πυκνότητας του Grid.
gradeLength / 8

    second_gradeLength: απόσταση του δεύτερου τμήματος πυκνότητας του Grid.
gradeLength / 4

    third_gradeLength: απόσταση του τρίτου τμήματος πυκνότητας του Grid.
gradeLength / 2

    initialize(starting_x, starting_y)

    # περίμετρος

    left(X_dist)

    down(Y_dist)

```

```

right(X_dist)
up(Y_dist)
# σημεία σχήματος
Για i από 1 ως 5:
    left(first_gradeLength)
    down(Y_dist)
    left(first_gradeLength)
    up(Y_dist)
Για i από 1 ως 3:
    left(second_gradeLength)
    down(Y_dist)
    left(second_gradeLength)
    up(Y_dist)
left(third_gradeLength)
down(Y_dist)
up_left(third_gradeLength)
right(X_dist)
up(third_gradeLength)
left(X_dist)
up(third_gradeLength)
right(X_dist)
up(second_gradeLength)
left(2 * gradeLength)
Για i από 1 ως 2:
    down(third_gradeLength)
    right(Y_dist)
    down(third_gradeLength)
    left(Y_dist)
down(third_gradeLength)
right(Y_dist)
down(first_gradeLength)
left(gradeLength)
up(second_gradeLength)
right(gradeLength)
Για i από 1 ως 4:
    up(second_gradeLength)
    left(gradeLength)

```

```

        up(second_gradeLength)
        right(gradeLength)
    Συνάρτηση save_points_to_file():
        αποθήκευση των σημείων του πίνακα points στο αρχείο "cs1_point_XY_list.txt"
        στη μορφή "point_XY_list = { [X170.000,Y160.000], ...}"
    Συνάρτηση pass_points_to_lines():
        αντιστοίχιση και μεταφορά των σημείων του πίνακα points[] στις ευθείες του
        πίνακα lines[]
    Συνάρτηση save_line_distances_to_file():
        αποθήκευση του μήκους των γραμμών στο αρχείο "cs1_path_length_list.txt"
        στη μορφή "path_length_list = {[Lel, 120.000], ...}"
    Συνάρτηση calculate_extrusion():
        # υπολογισμός extrusion για κάθε ευθεία στον πίνακα lines όπου isLe αληθές
        Για κάθε l στον πίνακα lines[]:
            Αν l.isLe:
                lp = l.line_distance()
                l.lf = (Ap * lp) / Af
                E = l.lf + E
                l.E = E
    Συνάρτηση save_gcode_extrusion_list_to_file():
        αποθήκευση των ευθειών που έχουν extrusion στο αρχείο
        "cs1_extrusion_parameter_list.txt "
        στη μορφή "G1 X50.00000 Y160.00000 E3.11748 F1200 ..."
    Συνάρτηση save_gcode_transition_list():
        αποθήκευση των ευθειών που δεν έχουν extrusion στο αρχείο
        "cs1_transition_parameter_list.txt"
        στη μορφή "G X165.00000 Y160.00000 E0 F1200 ..."
    Συνάρτηση save_gcode_extrusion_list_to_file():
        αποθήκευση των εντολών gcode για τον σχεδιασμό του Grid στο αρχείο
        "case_study_1.gcode"
    # Κώδικας
    grid = Grid()
    # Task 1
    grid.calculate_lines()
    grid.save_data_lines_to_file()
    # Task 2
    grid.calculate_points(60, 40, 170, 160)
    grid.save_points_to_file()

```

```

# Task 3
grid.pass_points_to_lines()
grid.save_line_distances_to_file()

# Task 5
grid.calculate_extrusion()
grid.save_gcode_extrusion_list_to_file()

# Task 6
grid.save_gcode_transition_list()
grid.save_gcode_to_file()

```

Αφού έγινε η εξαγωγή του τελικού κώδικα γίνεται προσθήκη μιας σειράς εντολών G κώδικα για τη κατασκευή ενός substrate/raft όπου κρατάει σταθερή τη δομή που εκτυπώθηκε αλλά και των διαστάσεων Z.

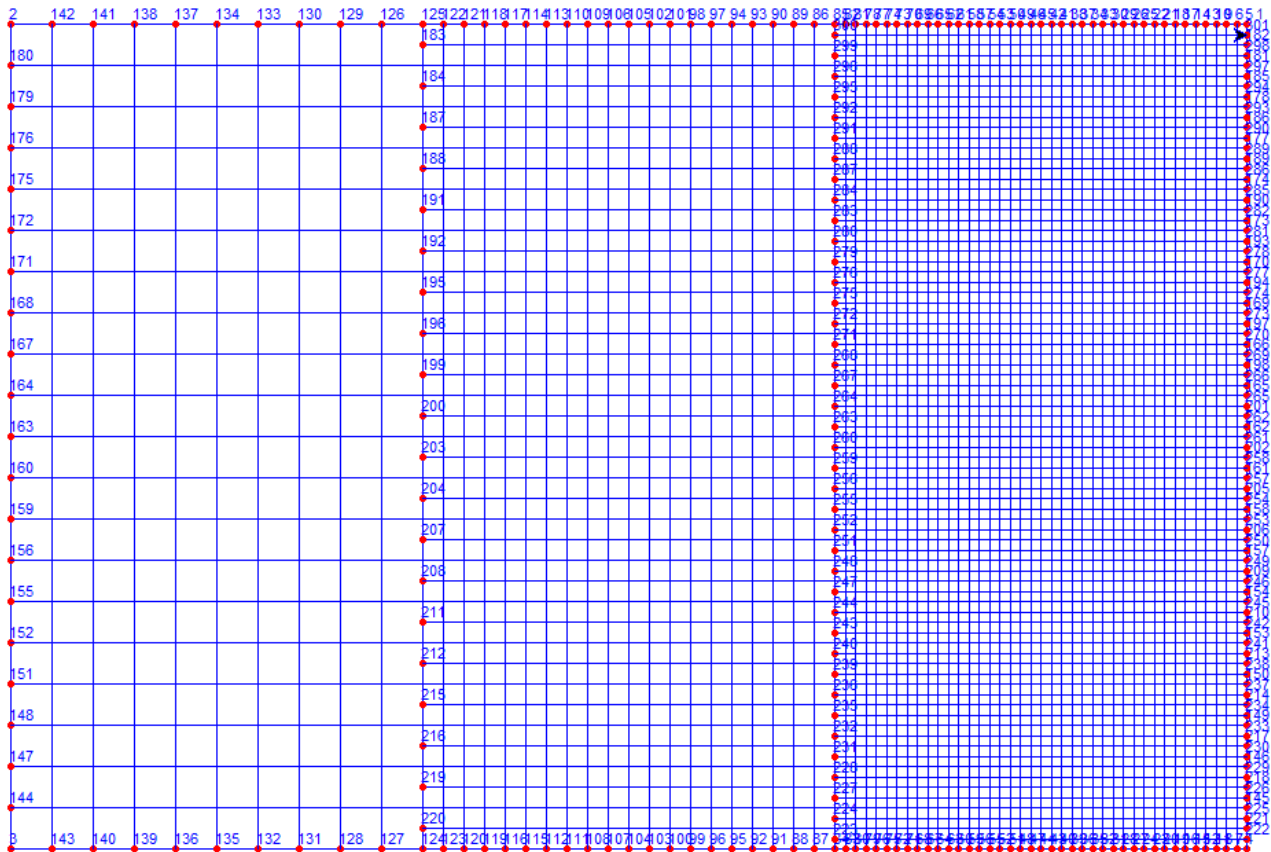
Στη συνέχεια εκτυπώθηκαν δύο διαφορετικές παραλλαγές της δομής. Στη μία περίπτωση κατά τη μετάβαση σε άλλο σημείο χωρίς εξώθηση υλικού, προστέθηκε ένα βήμα όπου γίνεται αναρρόφηση υλικού (retraction), το οποίο ρυθμίζεται δηλώνοντας στον G κώδικα αρνητική τιμή στη παράμετρο E και στην άλλη περίπτωση χρησιμοποιήθηκε χωρίς κάποια περαιτέρω αλλαγή ο κώδικας που αναπτύχθηκε, δηλαδή κατά την απλή μετάβαση σε άλλο σημείο χωρίς εξώθηση υλικού δεν έγινε αναρρόφηση. Παρακάτω παρουσιάζεται ένας πίνακας με τις βασικές εντολές του G κώδικα που χρησιμοποιήθηκαν:

Πίνακας 3: Βασικές εντολές G-κώδικα, με τις παραμέτρους που απαιτείται να οριστούν μαζί με τις εντολές, περιγραφή των εντολών και παραδείγματα σύνταξής τους.

Εντολή	Παράμετροι	Περιγραφή	Παράδειγμα
G0	Άξονας [X,Y,Z], θέση mm	Γρήγορη κίνηση της κεφαλής στην καθορισμένη θέση, χωρίς την λειτουργία του συστήματος τροφοδοσίας	G0 X50.600
G1	Άξονας [X,Y,Z,E], θέση mm, ταχύτητα πρόωσης [F] σε (mm/min),	Ελεγχόμενη κίνηση της κεφαλής και του συστήματος τροφοδοσίας	G1 F600 X10.000 Y20.000 E0.12399 F1200
G21	-	Μέτρηση αποστάσεων σε mm	G21
G90/91	-	Χρήση απόλυτων/σχετικών συντεταγμένων	G90
G92	Άξονας [X,Y,Z,E]	Επανακαθορισμός των συντεταγμένων της τρέχουσας θέσης (ορισμός σημείου αναφοράς σχετικών συντεταγμένων)	G92 X10.000

4.1.1.2 ΔΙΑΒΑΘΜΙΣΗ ΠΟΡΩΔΟΥΣ 0.5X0.5, 1X1 ΕΩΣ 2X2 MM

Η δεύτερη φάση της πειραματικής διαδικασίας περιλαμβάνει το σχηματισμό μιας απλής ορθογωνικής πλεγματοειδούς δομής, διαβαθμισμένου τετραγωνικού πορώδους με σχήμα πόρων τετράγωνο σε 2D, όπως φαίνεται στην Εικόνα 27.



Εικόνα 27: Αναπαράσταση 2D της πλεγματοειδούς δομής που σχεδιάζεται με customized κώδικα, με αναφορά στα αρχικά και τελικά σημεία κάθε βήματος, των διαστάσεων και της διαβάθμισης του πορώδους.

Το μέγεθος του δοκιμίου είναι 60 x 40 mm, σχήματος ορθογωνίου παραλληλογράμμου και γίνεται σχηματισμός τριών περιοχών πόρων, τριών περιοχών διαβάθμισης, με μέγιστο μέγεθος πόρου 2 x 2 mm, μεσαίο μέγεθος πόρου 1 x 1 mm και μικρότερο μέγεθος πόρου 0.5 x 0.5 mm. Είναι εμφανές ότι η αύξηση ή αντίστροφα μείωση του μεγέθους των πόρων γίνεται πολλαπλασιάζοντας ή διαιρώντας τον αρχικό πόρο με τον αριθμό 2. Για παράδειγμα εάν θεωρήσουμε για αρχικό πόρο τον 0.5 x 0.5 mm αν $0.5 \times 2 = 1$ mm και $1 \times 2 = 2$ mm, και αντίστροφα εάν θεωρήσουμε για αρχικό μέγεθος πόρου το 2 x 2 mm.

Για το 2ο case study χρησιμοποιήθηκαν σε μεγάλο βαθμό οι κλάσεις και οι συναρτήσεις του 1ου case study μιας και το ζητούμενο σχήμα Grid ήταν το ίδιο με περισσότερα σημεία. Οι διαφορές βρίσκονται στις συναρτήσεις calculate_lines και calculate_points. Ακολουθήθηκε η ίδια συλλογιστική πορεία που αναπτύχθηκε στη προηγούμενη δομή με τη διαφορά πως τα σημεία που διαχειρίστηκαν ήταν συνολικά 301. Θεωρούμε ότι ισχύουν ακόμα η παραδοχή και οι εξισώσεις που πάρθηκαν προηγουμένως στη Παρ. 4.1.1.1 για το σχήμα και τον υπολογισμό το ποσού εξώθησης της ίνας και της κατανάλωσης νήματος. Έγινε πάλι εξαγωγή των ζητούμενων στοιχείων και η τελική ανάπτυξη G κώδικα.

Συνάρτηση calculateLines():

```
# υπολογισμός του flag isLe ευθειών του σχήματος Grid.  
Για i από 1 έως 4:  
    αποθήκευσε στον πίνακα lines[] το Line(i, i+1) με isLe αληθές.  
Για i από 5 έως 300:  
    αποθήκευσε στον πίνακα lines[] το Line(i, i+1)  
    με isLe αληθές αν το i είναι άρτιος.  
    με isLe ψευδές αν το i είναι περιττός.
```

Συνάρτηση calculate_points(X_dist, Y_dist, starting_x, starting_y):

```
# υπολογισμός σημείων του σχήματος Grid.  
X_dist: μήκος του Grid.  
Y_dist: ύψος του Grid.  
gradeLength: απόσταση των τριών τμημάτων πυκνότητας του Grid. X_dist/ 3  
first_gradeLength: απόσταση του πρώτου τμήματος πυκνότητας του Grid.  
gradeLength / 8  
second_gradeLength: απόσταση του δεύτερου τμήματος πυκνότητας του Grid.  
gradeLength / 4  
third_gradeLength: απόσταση του τρίτου τμήματος πυκνότητας του Grid.  
gradeLength / 2  
initialize(starting_x, starting_y)  
# περίμετρος  
left(X_dist)  
down(Y_dist)  
right(X_dist)  
up(Y_dist)  
# σημεία σχήματος  
Για i από 1 ως 21:  
    left(first_gradeLength)  
    down(Y_dist)  
    left(first_gradeLength)  
    up(Y_dist)  
Για i από 1 ως 11:  
    left(second_gradeLength)  
    down(Y_dist)  
    left(second_gradeLength)  
    up(Y_dist)
```

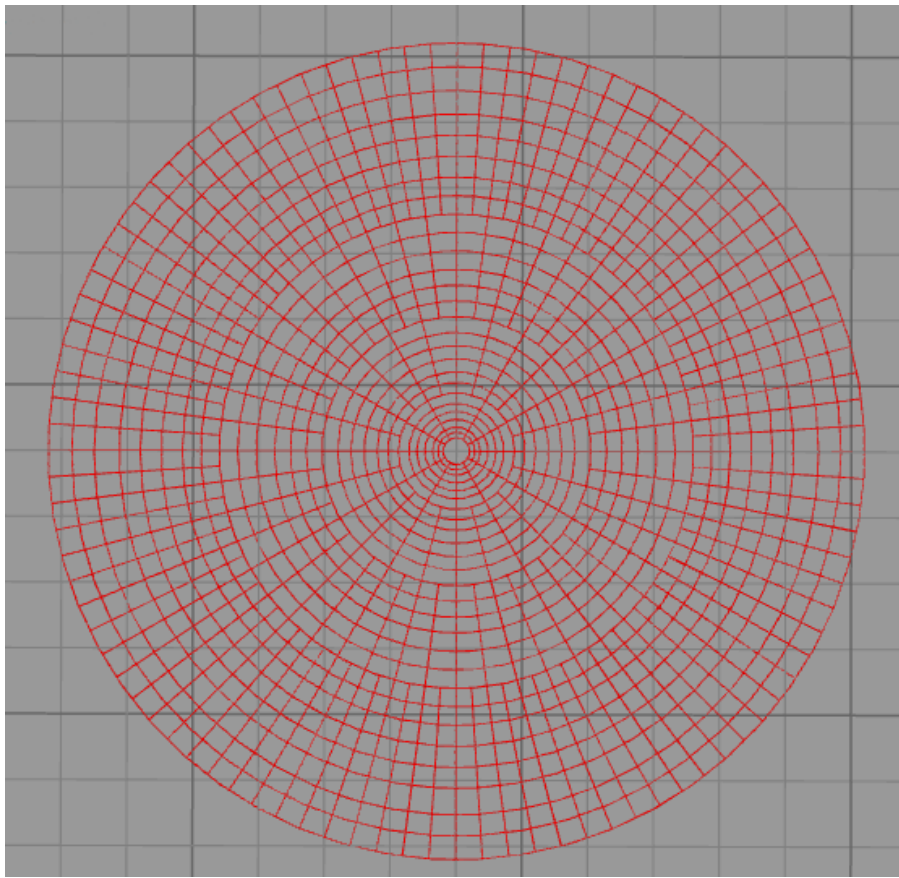
```

Γιὰ i ἀπό 1 ὡς 5:
    left (third_gradeLength)
    down (Y_dist)
    left (third_gradeLength)
    up (Y_dist)
left (third_gradeLength)
down (Y_dist)
upLeft (third_gradeLength)
Γιὰ i ἀπό 1 ὡς 10:
    right (X_dist)
    up (third_gradeLength)
    left (X_dist)
    up (third_gradeLength)
right (X_dist)
up (second_gradeLength)
left (2 * gradeLength)
Γιὰ i ἀπό 1 ὡς 9:
    down (third_gradeLength)
    right (Y_dist)
    down (third_gradeLength)
    left (Y_dist)
down (third_gradeLength)
right (Y_dist)
down (third_gradeLength)
left (Y_dist)
down (third_gradeLength)
right (Y_dist)
down (first_gradeLength)
left (gradeLength)
up (second_gradeLength)
right (gradeLength)
Γιὰ i ἀπό 1 ὡς 20:
    up (second_gradeLength)
    left (gradeLength)
    up (second_gradeLength)
        right (gradeLength)

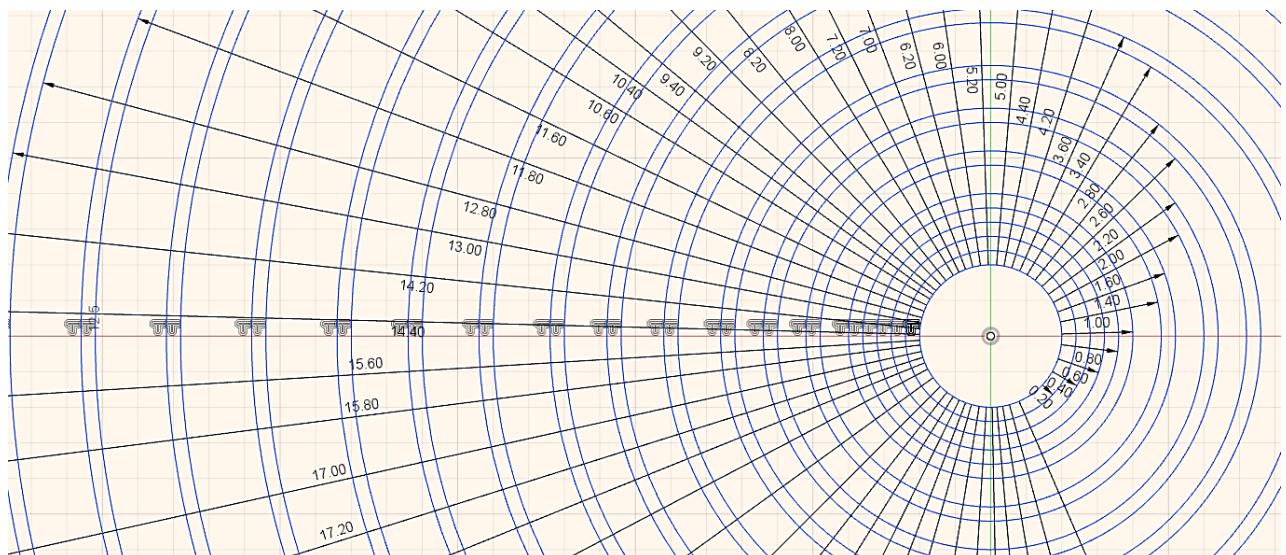
```

4.1.2 ΚΑΤΑΣΚΕΥΗ ΚΥΚΛΙΚΗΣ ΔΟΜΗΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΞΩΝ

Η τρίτη φάση της πειραματικής διαδικασίας περιλαμβάνει το σχηματισμό μιας λιγότερο απλής πλεγματοειδούς δομής, διαβαθμισμένου πορώδους σε 2D, όπως φαίνεται στις Εικόνες 28-29.



Εικόνα 28: Αναπαράσταση 2D της κυκλικής πλεγματοειδούς δομής που θα εκτυπωθεί, εικόνα από Simplify 3D λογισμικό



Εικόνα 29: Αναπαράσταση 2D της κυκλικής πλεγματοειδούς δομής με τις διαστάσεις των εσωτερικών διαμέτρων κάθε κύκλου

Το σχήμα αυτό αποτελείται από μια σειρά ομόκεντρων κύκλων με κέντρο το $(X, Y) = (95.000, 95.000)$. Ο αρχικός και μικρότερος σε μέγεθος κύκλος έχει ακτίνα 1 mm και γίνεται αύξηση της ακτίνας κατά 0.4, 0.6, 0.8, 1.00, 1.20, 1.40, 1.60 και 1.80 mm σταδιακά, καταλήγοντας σε κύκλο μέγιστης ακτίνας

31.20 mm. Οι κάθετες γραμμές χωρίστηκαν σε 4 διαφορετικές περιοχές ανάλογα με το μήκος τους από τον τελευταίο κύκλο με τη μέγιστη ακτίνα. Όλες οι ακτίνες μεταξύ τους έχουν απόσταση ίση με $\theta = 360/n$ όπου n ο αριθμός των σημείων που τέμνουν τον εξωτερικό κύκλο. Μετρήθηκε ο αριθμός των γραμμών που τέμνουν τον εξωτερικό κύκλο και βρέθηκε πως $n = 96$ άρα η γωνία θ μεταξύ των γραμμών είναι 3.75° . Τα σημεία τομής των ευθύγραμμων τμημάτων με τους κύκλους βρέθηκαν από τις παρακάτω παραμετρικές εξισώσεις για το κύκλο:

$$x = x_0 + r \times \cos\theta \quad (9)$$

$$y = y_0 + r \times \sin\theta \quad (10)$$

Όπου

x_0 και y_0 : το κέντρο του κύκλου,

r : η ακτίνα του κύκλου, και,

x και y : τα σημεία τομής με τον εκάστοτε κύκλο

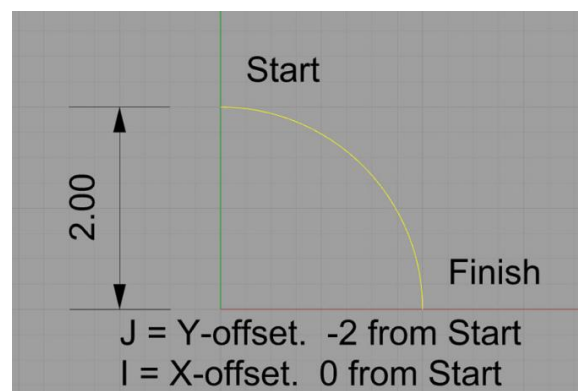
θ : η κλίση της κάθε γραμμής.

Για τη συγγραφή του κώδικα σε γλώσσα προγραμματισμού Python χρησιμοποιήθηκαν κάποια κοινά κομμάτια που χρησιμοποιήθηκαν στο σχηματισμό των απλών δομών που σχηματίστηκαν προηγουμένως. Ωστόσο, δε χρησιμοποιήθηκε η ίδια προσέγγιση με τις λίστες όπως προηγουμένως, δηλαδή δε καταφύγαμε στη δημιουργία πολλών επιμέρους λιστών, αλλά μιας τελικής με όλο το G κώδικα. Επίσης, για τον σχηματισμό των ευθύγραμμων τμημάτων, δηλαδή για τον υπολογισμό της παραμέτρου E, θεωρούμε ότι ισχύουν οι παραδοχές και οι εξισώσεις που ορίστηκαν προηγουμένως κατά το σχηματισμό της πρώτης διαβαθμισμένης δομής.

Για τον υπολογισμό της εξώθησης υλικού για τους κύκλους θεωρήσαμε πως το μήκος του τμήματος l_p , δηλαδή το μήκος της ίνας του κύκλου που σχηματίζεται, είναι ίσο με τη περίμετρο του εκάστοτε κύκλου που σχηματίζεται. Άρα:

$$l_p = 2\pi r \quad (11)$$

Τέλος για την αποτύπωση του G κώδικα ισχύουν οι εντολές που χρησιμοποιήθηκαν προηγουμένως και οι ίδιες λειτουργίες τους τόσο για τους κύκλους όσο και για τα ευθύγραμμα τμήματα, με τη διαφορά ότι για τη δημιουργία των κύκλων έγινε πρόσθεση δύο ακόμα παραμέτρων, των I και J που αφορούν τη δημιουργία τόξων και κύκλων σε αυτό το κώδικα αλλά και η εντολή G2 που δίνει τη δυνατότητα δεξιόστροφης (clockwise) κίνησης της κεφαλής/δεξιόστροφου σχηματισμού ενός τόξου ή κύκλου. Οι εντολές αυτές λειτουργούν συνδυαστικά και εφαρμόζονται όπως θα εξηγήσουμε στο παρακάτω παράδειγμα [66]:



Εικόνα 29: Σχηματική αναπαράσταση μεθόδου σχεδιασμού τόξων σε γλώσσα G-code

Έστω ότι θέλουμε να σχεδιάσουμε ένα τόξο με κέντρο το σημείο O (0,0) και η ακτίνα που θέτουμε είναι $R = 2$. Το σημείο από όπου ξεκινάει η κεφαλή είναι το (0,2) και θέλουμε να φτάσουμε στο (2,0).

$J = Y - \text{offset} = 0 - 2 = -2$ μονάδες κάτω από το αρχικό σημείο, και,

$$I = X - \text{offset} = 2 - 0 = 2$$

Άρα στο συγκεκριμένο παράδειγμα ο G κώδικας που φτιάχνεται είναι ο παρακάτω:

G2 X2 Y0 I0 J-2

Παρακάτω δίνεται ο κώδικας της Python που αναπτύχθηκε, για το χτίσιμο της δομής, σε μορφή ψευδοκώδικα, όπως προηγουμένως.

Κλάση Point: Η ίδια κλάση που χρησιμοποιήθηκε και στα 2 προηγούμενα Case studies.

Κλάση Circle: Αντιπροσωπεύει έναν κύκλο στο επίπεδο.

r: Ακτίνα του κύκλου.

center_X, center_Y: οι συντεταγμένες του κέντρου.

F: η ταχύτητα εκτύπωσης.

E: Extrusion.

perimeter: Η περίμετρος.

points[]: Σημεία του κύκλου από τα οποία περνάνε οι ευθείες προς το κέντρο.

Συνάρτηση gcode_draw_circle():

Επιστρέφει την εντολή G-code για τον σχεδιασμό του κύκλου σε μορφή

```
"G1 X95.000 Y96.000 F1200"
```

```
"G2 X95.000 Y96.000 I0.000 J-1.000 E0.072"
```

Συνάρτηση calculate_circle_points(n):

Υπολογισμός και αποθήκευση των n σημείων του κύκλου από τα οποία περνάνε οι ευθείες που πάνε στο κέντρο

Για i από 0 ως n:

angle = i * (360 / n) # Υπολογισμός της γωνίας που σχηματίζουν οι ευθείες κάθε διαδοχικού σημείου

```
points.append(Point(center_X + r * συνημίτονο(math.radians(angle) + (π / 2)),
```

```
center_Y + r * ημίτονο(math.radians(angle) + (π / 2))))
```

Κλάση ConcentricCircles: Αντιπροσωπεύει τους ομόκεντρους κύκλους του ζητούμενου σχήματος.

circles[]: Πίνακας των κύκλων του σχήματος.

current_point: Το σημείο που βρισκόμαστε κάθε χρονική στιγμή.

F = 1200

Wp = 0.200

Hp = 0.200

Af = 2.74899

Ap = 0

lp = 0

E = 0

Συνάρτηση gcode_move_write(x, y):

Επιστρέφει εντολή G-code για μετακίνηση με extrusion από το current_point στις συντεταγμένες x, y

```

στη μορφή "G1 X95.000 Y96.000 E25.172 F1200"

Συνάρτηση gcode_move_no_write(x, y):
    Επιστρέφει εντολή G-code για μετακίνηση χωρίς extrusion από το
current_point στις συντεταγμένες x, y
    στη μορφή "G0 X94.500 Y95.866 F1200"

Συνάρτηση calculate_circle_extrusion(r) και calculate_extrusion(distance)
    Υπολογισμός του extrusion για τους κύκλους και τις ευθείες με τον ίδια
λογική που χρησιμοποιήθηκε
    και στα 2 προηγούμενα case studies.

Συνάρτηση calculate_all_points():
    # Υπολογισμός των σημείων των κύκλων από τα οποία περνάνε οι ευθείες προς
το κέντρο.
    # Πρώτη τάξη κύκλων με 12 σημεία
    Για i από 0 ως 6:
        circles[i].calculate_circle_points(12)
    # Δεύτερη τάξη κύκλων με 24 σημεία
    Για i από 6 ως 12:
        circles[i].calculate_circle_points(24)
    # Τρίτη τάξη κύκλων με 48 σημεία
    Για i από 12 ως 18:
        circles[i].calculate_circle_points(48)
    # Τέταρτη τάξη κύκλων με 96 σημεία
    Για i από 18 ως 27:
        circles[i].calculate_circle_points(96)

Συνάρτηση save_Gcode():
    Αποθήκευση G-code εντολών για τον σχεδιασμό των ομόκεντρων κύκλων και των
ευθειών προς το κέντρο
    στο αρχείο "case_study_3.gcode".
    current_R = initial_R
    # Μετακίνηση στο κέντρο
    f.write(gcode_move_no_write(center_X, center_Y))
    # Πρώτο σει κύκλων
    circles.append(
        Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
    )
    f.write(circles[len(circles) - 1].gcode_draw_circle())
    # Πρώτη τάξη κύκλων με κάθε σει να έχει απόσταση 0.40
    grade = 0.40
    Για c από 1 ως 2:
        current_R += grade

```

```

        circles.append(
            Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
            f.write(circles[len(circles) - 1].gcode_draw_circle())
# Δεύτερη τάξη κύκλων με κάθε σει να έχει απόσταση 0.60
grade = 0.60
Για c από 1 ως 3:
    current_R += grade
    circles.append(
        Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
        f.write(circles[len(circles) - 1].gcode_draw_circle())
# Τρίτη τάξη κύκλων με κάθε σει να έχει απόσταση 0.80
grade = 0.80
Για c από 1 ως 3:
    current_R += grade
    circles.append(
        Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
        f.write(circles[len(circles) - 1].gcode_draw_circle())
# Τέταρτη τάξη κύκλων με κάθε σει να έχει απόσταση 1.00
grade = 1.00
Για c από 1 ως 3:
    current_R += grade
    circles.append(
        Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
        f.write(circles[len(circles) - 1].gcode_draw_circle())
# Πέμπτη τάξη κύκλων με κάθε σει να έχει απόσταση 1.20
grade = 1.20
Για c από 1 ως 4:
    current_R += grade
    circles.append(
        Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
        f.write(circles[len(circles) - 1].gcode_draw_circle())
# Έκτη τάξη κύκλων με κάθε σει να έχει απόσταση 1.40
grade = 1.40
Για c από 1 ως 5:
    current_R += grade

```

```

        circles.append(
            Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
            f.write(circles[len(circles) - 1].gcode_draw_circle())
            # Εβδομη τάξη κύκλων με κάθε σει να έχει απόσταση 1.60
            grade = 1.60
            Για c από 1 ως 3:
                current_R += grade
                circles.append(
                    Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
                    f.write(circles[len(circles) - 1].gcode_draw_circle())
                    # Ογδοη τάξη κύκλων με κάθε σει να έχει απόσταση 1.80
                    grade = 1.80
                    Για c από 1 ως 3:
                        current_R += grade
                        circles.append(
                            Circle(current_R, center_X, center_Y, F,
calculate_circle_extrusion(current_R))
                            f.write(circles[len(circles) - 1].gcode_draw_circle())
                            calculate_all_points()
                            # Ευθείες από τον 27ο στον 1ο κύκλο
                            inner_circle = circles[0]
                            outer_circle = circles[26]
                            step = 8
                            j = step
                            Για i από 0 ως 10 με βήμα 2:
                                f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))
                                f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))
                                f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))
                                f.write(gcode_move_no_write(outer_circle.points[j + step].x,
outer_circle.points[j + step].y))
                                j += step * 2
                            i = 10
                                f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))
                                f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))
                                f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))

```



```

# Ευθείες από τον 27ο στον 6ο κύκλο

f.write(gcode_move_no_write(outer_circle.points[0].x,
outer_circle.points[0].y))

inner_circle = circles[6]

outer_circle = circles[26]

step = 4

j = step

Για i από 0 ως 22 με βήμα 2:

    f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))

    f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))

    f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))

    f.write(gcode_move_no_write(outer_circle.points[j + step].x,
outer_circle.points[j + step].y))

    j += step * 2

i = 22

    f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))

    f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))

    f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))

# Ευθείες από τον 27ο στον 12ο κύκλο

f.write(gcode_move_no_write(outer_circle.points[0].x,
outer_circle.points[0].y))

inner_circle = circles[12]

outer_circle = circles[26]

step = 2

j = step

Για i από 0 ως 46 με βήμα 2:

    f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))

    f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))

    f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))

    f.write(gcode_move_no_write(outer_circle.points[j + step].x,
outer_circle.points[j + step].y))

    j += step * 2

i = 46

    f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))

```

```

        f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))

        f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))

        # Ευθείες από τον 27ο στον 18ο κύκλο

        f.write(gcode_move_no_write(outer_circle.points[0].x,
outer_circle.points[0].y))

        inner_circle = circles[18]
        outer_circle = circles[26]

        step = 1
        j = step

        Για i από 0 ως 94 με βήμα 2:

        f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))

        f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))

        f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))

        f.write(gcode_move_no_write(outer_circle.points[j + step].x,
outer_circle.points[j + step].y))

        j += step * 2

        i = 94

        f.write(gcode_move_write(inner_circle.points[i].x,
inner_circle.points[i].y))

        f.write(gcode_move_no_write(inner_circle.points[i + 1].x,
inner_circle.points[i + 1].y))

        f.write(gcode_move_write(outer_circle.points[j].x,
outer_circle.points[j].y))

# Κώδικας
circles = ConcentricCircles(1, 95, 95, 1200)

circles.save_Gcode()

```

Για μία εναλλακτική σχεδίαση των ομόκεντρων κύκλων χρησιμοποιήθηκε και η βιβλιοθήκη της ργthon `pygcode` με την οποία επιτεύχθηκε η γραμμικοποίηση των κύκλων. Επιπλέον, χρησιμοποιήθηκαν `pygthon scripts` για την μορφοποίηση των παραγόμενων εντολών από τη βιβλιοθήκη `pygcode`, όπως ο υπολογισμός του `Extrusion` με τη λογική των προηγούμενων `case studies`. Ωστόσο αν και η λογική του κώδικα είναι σωστή, τα αποτελέσματα δεν είναι τόσο ενθαρρυντικά. Παρατηρήθηκε ότι υπήρχε πρόβλημα στο σχηματισμό των ομόκεντρων κύκλων, διότι ουσιαστικά φαίνεται σα να μη γίνεται εξώθηση υλικού, επομένως στο τυπωμένο σχήμα, δεν αποτυπώνονται καλά οι κύκλοι. Για αυτό το λόγο προχωρήσαμε σε μια τέταρτη φάση που θα αναλυθεί ακριβώς από κάτω.

4.1.3 ΚΑΤΑΣΚΕΥΗ ΚΥΚΛΙΚΗΣ ΔΟΜΗΣ ΜΕ ΓΡΑΜΜΙΚΟΠΟΙΗΣΗ ΤΩΝ ΤΟΞΩΝ

Η τέταρτη φάση της πειραματικής διαδικασίας αφορά την ίδια δομή με τη προηγούμενη, δηλαδή εκτυπώνεται πάλι δομή με ομόκεντρους κύκλους και γραμμές, με τη διαφορά ότι ακολουθήθηκε λίγο διαφορετική προσέγγιση για το σχηματισμό το κύκλων. Όπως προαναφέρθηκε ο κώδικας `G` που αναπτύχθηκε για το σχηματισμό των κύκλων έγινε με την εφαρμογή εντολών `I` και `J`.

Αυτό δημιούργησε πρόβλημα στο πρόγραμμα που συνδέεται με τον εκτυπωτή, διότι δουλεύει κυρίως θεωρώντας πως κάθε τόξο αποτελείται από στοιχειώδη ευθύγραμμα τμήματα. Έτσι αποφασίστηκε να ακολουθηθεί αυτή η προσέγγιση και εφαρμόστηκε με τη βοήθεια μιας ειδικής βιβλιοθήκης της Python, της *pygcode* που μία από τις δυνατότητές της είναι η μετατροπή τόξων σε στοιχειώδη ευθύγραμμα τμήματα με την εντολή *arclinearization*. Όπως αναφέρθηκε αυτή η εντολή αφορά τη γραμμικοποίηση τόξων και όχι ολόκληρων κύκλων. Για αυτό το λόγο η πρώτη και βασική ενέργεια ήταν να μετατραπεί ο κάθε κύκλος σε δύο ημικύκλια ή αλλιώς σε δύο τόξα.

Για να γίνει αυτό, θεωρήθηκε πάλι πως πρέπει να γίνει μετάβαση της κεφαλής στο κέντρο των κύκλων (X, Y) = (95.000, 95.000) και στη συνέχεια μετάβαση στο άνω ακραίο σημείο του εκάστοτε κύκλου και δεξιόστροφος σχηματισμός του πρώτου ημικυκλίου – χρήση της εντολής G2. Το επόμενο βήμα που ακολουθεί είναι πάλι η μετάβαση στο άνω άκρο του κύκλου και ο αριστερόστροφος σχηματισμός του άλλου ημικυκλίου – χρήση της εντολής G3. Αφού έχει αναπτυχθεί κώδικας Python για το σχηματισμό των κύκλων με αυτή τη μέθοδο όπως φαίνεται και παρακάτω, γίνεται και μετατροπή και αποθήκευση σε G κώδικα. Από τις εντολές που χρησιμοποιήθηκαν, είναι εφικτό να ελέγξει κανείς την ακρίβεια με την οποία θα γίνει η γραμμικοποίηση των τόξων. Επιλέχτηκε στη συγκεκριμένη περίπτωση να γίνει με ακρίβεια 0.05 mm, δηλαδή να σχηματιστούν τμήματα 0.05 mm, διότι δεν είναι στις προδιαγραφές του εκτυπωτή η κατασκευή τμημάτων μικρότερου μεγέθους. Η μετατροπή και σύγκριση των γραμμών κώδικα φαίνεται ενδεικτικά για ακτίνα R = 1 mm παρακάτω:

;χωρισμός του κυκλου σε 2 τόξα

```
G1 X95.000 Y96.000 F1200
G2 X95.000 Y94.000 I0.000 J-1.000
G1 X95.000 Y96.000 F1200
G3 X95.000 Y94.000 I0.000 J-1.000
```

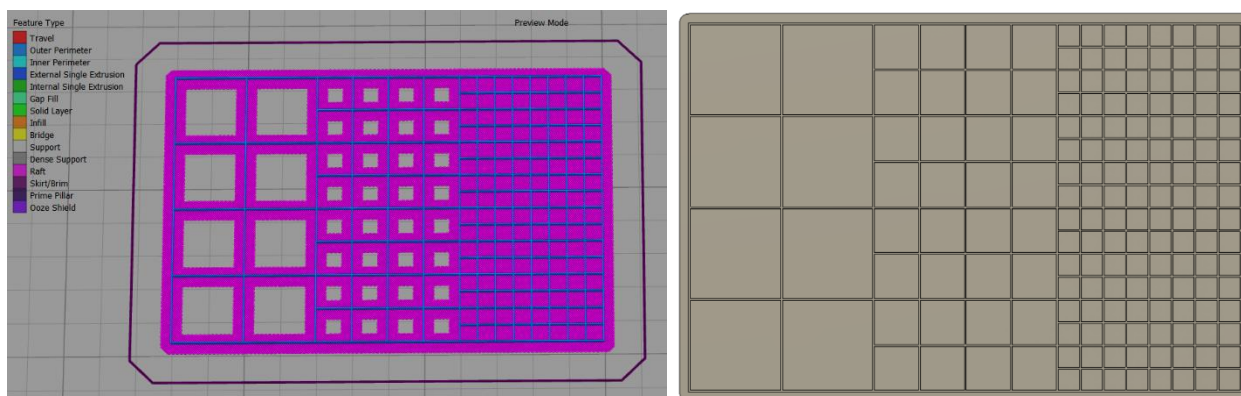
;μορφή κώδικα των τόξων σε ευθύγραμμα τμήματα μήκους 0.05mm

```
G1 X95.263 Y95.983 E0.001 F1200
G1 X95.719 Y95.719 E0.001 F1200
G1 X95.983 Y95.263 E0.002 F1200
G1 X95.983 Y94.737 E0.002 F1200
G1 X95.719 Y94.281 E0.003 F1200
G1 X95.263 Y94.017 E0.003 F1200
G1 X95 Y94 E0.004 F1200
G1 X95 Y96 F1200
G1 X94.737 Y95.983 E0.005 F1200
G1 X94.281 Y95.719 E0.005 F1200
G1 X94.017 Y95.263 E0.006 F1200
G1 X94.017 Y94.737 E0.006 F1200
G1 X94.281 Y94.281 E0.007 F1200
G1 X94.737 Y94.017 E0.007 F1200
G1 X95 Y94 E0.008 F1200
```

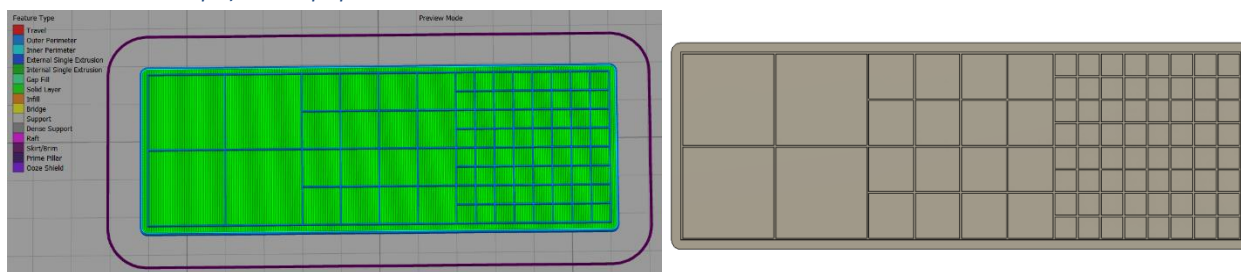
4.2 ΣΧΕΔΙΑΣΜΟΣ ΜΕΣΩ CAD ΚΑΙ ΕΞΑΓΩΓΗ G-CODE ΜΕΣΩ SLICING

Στη συνέχεια έγινε σύγκριση των δομών που παράχθηκαν με αντίστοιχες δομές και κώδικα που συντάσσεται αυτόματα από slicing λογισμικό. Για το σκοπό, έπρεπε να γίνει σχεδιασμός των δομών αυτών σε CAD λογισμικό (Fusion 360 Autodesk) και μετατροπή τους σε STL αρχεία, που είναι συμβατά με slicing λογισμικό. Έπειτα έγινε επεξεργασία των αρχείων STL σε slicing λογισμικό (Simplify3D), για την εξαγωγή του G-code. Οι πλεγματικές δομές σχεδιάστηκαν με πλάτος 0.2 mm ίσο με το πλάτος των μεμονωμένων ινών που εναποθέτει η κεφαλή εκτύπωσης. Μεταβάλλοντας το σημείο έναρξης της εναπόθεσης υλικού, προσδιορίστηκε ότι η βέλτιστη τροχιά προκύπτει ξεκινώντας από πόρους μεγαλύτερης διαμέτρου και προχωρώντας σταδιακά πόρους μικρότερης διαμέτρου.

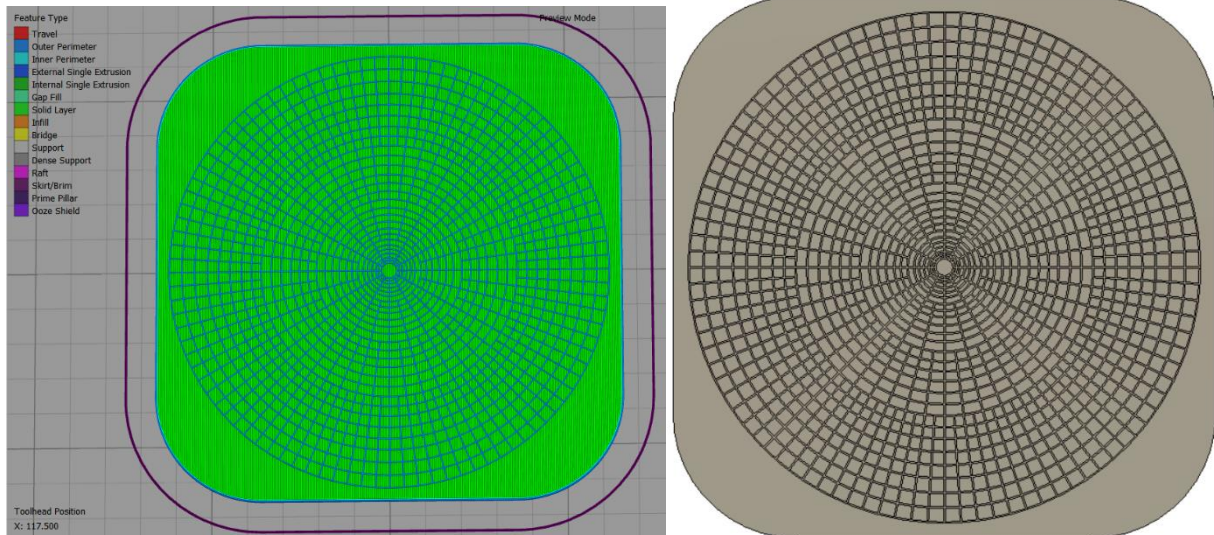
Η δομή στην Εικόνα 30 διαμορφώθηκε μέσω CAM λογισμικό και η διαδρομή που ακολουθήθηκε για τη κατασκευή του πλέγματος είναι διαφορετική από του customized κώδικα, και έχει οριστεί από το λογισμικό εκτύπωσης. Στην Εικόνα 31 χρησιμοποιείται διαφορετικό υπόστρωμα καθώς το υπόστρωμα της Εικόνας 30 αποδείχθηκε αργότερα αρκετά λεπτό και εύθραυστο. Επίσης στην Εικόνα 30 τα δείγματα που εκτυπώθηκαν με βάση τη παραπάνω δομή ακολουθούν δύο διαφορετικούς τρόπους κατασκευής της πλεγματικής δομής. Ο ένας αφορά τη περιμετρική κατασκευή των τετραγώνων του πλέγματος και ο άλλος δημιουργεί κάποιου είδους «σκαλάκια» από τις πλευρές των τετραγώνων, σε διάφορα σημεία, μέχρι να σχηματίσει τη τελική δομή. Στην Εικόνα 32, όπου μπορεί κανείς να δει τη κυκλική δομή μαζί με το υπόστρωμα, πρέπει να αναφέρουμε ότι η διαδρομή της κατασκευής του πλέγματος που έγινε μέσω του λογισμικού του εκτυπωτή είναι παρόμοια με τη διαδρομή «σκαλάκια» που ακολουθήθηκε στην ορθογωνική δομή.



Εικόνα 30: Ορθογωνική δομή με πορώδες 2.5x2.5, 5x5 έως 10x10 mm και διαστάσεις 40x60mm, με περιμετρικό στο πορώδες υπόστρωμα και δομή πλέγματος κατασκευασμένη με βάση τη βέλτιστη διαδρομή του λογισμικού του εκτυπωτή – κατασκευή «σκαλιών» από πλευρές των τετραγώνων.



Εικόνα 31: Ορθογωνική δομή με πορώδες 2.5x2.5, 5x5 έως 10x10 mm και διαστάσεις 20x60mm, με υπόστρωμα μεγαλύτερης πλήρωσης. Υπάρχουν δύο είδη κατασκευής του πλέγματος όπου το ένα βασίζεται στη περιμετρική κατασκευή των τετραγώνων και το άλλο κατασκευάζεται με βάση την ίδια διαδρομή όπως στην Εικόνα 30.



Εικόνα 32: Αναπαράσταση 2D της κυκλικής δομής μαζί με το υπόστρωμα.

4.3 ΣΥΓΚΡΙΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΔΥΟ ΜΕΘΟΔΩΝ

Το επόμενο βήμα είναι η σύγκριση και η αξιολόγηση των διαφορετικών μεθόδων εκτύπωσης. Αυτό έγινε με την επαναλαμβανόμενη παρατήρηση και μέτρηση των πλάτους των ινών στα δοκίμια σε διαφορετικά σημεία με σκοπό να υπολογιστεί η μέση τιμή και η απόκλιση από την επιθυμητή τιμή για να αξιολογηθεί εάν κατά μέσο όρο έχει εκτυπωθεί το επιθυμητό πλάτος. Τα διαφορετικά αυτά σημεία αφορούν τα διαφορετικά πορώδη που υπάρχουν μέσα στη δομή. Άλλα στοιχεία που παρατηρήθηκαν είναι η οπτική παρατήρηση των δοκιμίων, η ομοιομορφία της ίνας αλλά και σημεία τομής των ινών, που μπορεί να προέρχονται είτε από συγκόλληση που προέρχεται από το τρόπο εκτύπωσης είτε σημεία επικάλυψης. Αυτά τα στοιχεία είναι πολύ σημαντικά γιατί έτσι αποδεικνύεται η ποιότητα της εκτυπωτικής διαδικασίας και, επίσης είναι σημαντικά για τη διασφάλιση των μηχανικών ιδιοτήτων. Ακόμη, είναι σημαντικό να προσδιοριστεί εάν η διαδρομή της κεφαλής μπορεί να επηρεάσει την ποιότητα την ίνας και το πως διαμορφώνεται ανάλογα με το διαφορετικό πορώδες.

Για την οπτική παρατήρηση και τη μέτρηση των ινών έγινε χρήση του οπτικού μικροσκοπίου. Η μέτρηση των ινών έγιναν σε κλίμακα 200 μm. Για τον υπολογισμό της μέσης τιμής του πλάτους των ινών χρησιμοποιείται ο τύπος:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (12)$$

Για τον υπολογισμό της διασποράς χρησιμοποιείται ο τύπος:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (13)$$

Η απόκλιση υπολογίζεται ως εξής:

$$\text{απόκλιση} = \sum_{i=1}^n |x_i - x_{width}| \quad (14)$$

4.4 ΠΕΙΡΑΜΑΤΙΚΟΣ ΕΞΟΠΛΙΣΜΟΣ

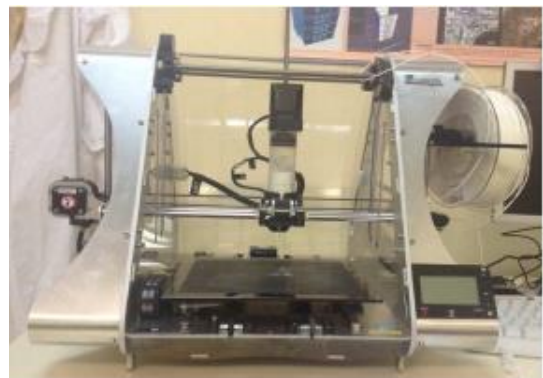
Το υλικό που χρησιμοποιήθηκε ήταν το πολυγαλακτικό οξύ, σε μορφή νήματος και η μέθοδος εκτύπωσης ήταν με εξώθηση υλικού σε κατάσταση τήγματος (FDM). Η διάμετρος του νήματος PLA μετρήθηκε σε όλο το μήκος του με ψηφιακό παχύμετρο και βρέθηκε ότι η μέση τιμή της διαμέτρου είναι 1.74 mm.

Ο εκτυπωτής που χρησιμοποιήθηκε για τη κατασκευή των δειγμάτων είναι της εταιρείας ZMorph 2.0S και τα δείγματα παρατηρήθηκαν στο οπτικό μικροσκόπιο. Και οι δύο εξοπλισμοί ανήκουν το εργαστήριο Νανοτεχνολογίας της Σχολής Χημικών Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου. Στον εκτυπωτή Zmorph από τη φωτογραφία είναι εμφανή τα σημεία της κεφαλής, της πλατφόρμας εκτύπωσης και του καρουλιού που αποτελεί πηγή τροφοδοσίας. Στις Εικόνες , φαίνεται η κεφαλή, το ακροφύσιο και η πλατφόρμα εκτύπωσης με την ειδική επίστρωση που βοηθάει στη καλύτερη πρόσφυση του υλικού.

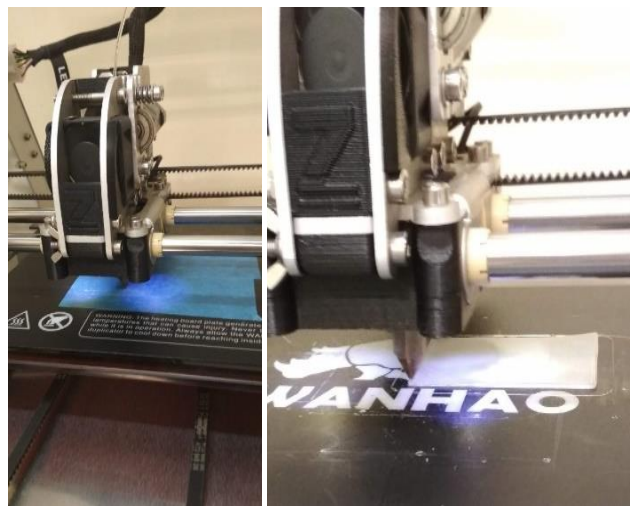
Η διαδικασία που ακολουθήθηκε για να υλοποιηθεί η πειραματική διαδικασία περιγράφεται στα βήματα παρακάτω:

1. Αφού ελέγξουμε ότι ο εκτυπωτής είναι συνδεδεμένος με τον Η/Υ, τον ανοίγουμε από το ειδικό κουμπί.
2. Ανοίγουμε το λογισμικό του Η/Υ που συνδέεται με τον εκτυπωτή, Voxelizer.
3. Επιλέγουμε στο λογισμικό το υλικό που θα χρησιμοποιηθεί για την εκτύπωση (PLA).
4. Βάζουμε το νήμα του PLA στη κεφαλή έτσι ώστε να αρχίσει να θερμαίνεται.
5. Φορτώνουμε τον G κώδικα στο λογισμικό.
6. Πατάμε το Start για να ξεκινήσει η εκτύπωση.

Αφού τελειώσει η εκτύπωση, αφήνουμε για λίγα λεπτά το δείγμα να σιγουρευτούμε ότι έχει κρυώσει και έχει γίνει σωστή σκλήρυνση του πολυμερούς, και προσεκτικά, τόσο για να μη χαλάσει το δείγμα όσο και να μη φθαρεί η επιφάνεια της πλατφόρμας εκτύπωσης, με μια σπάτουλα το αφαιρούμε από τη πλατφόρμα.



Εικόνα 33: Στην πάνω εικόνα φαίνεται το οπτικό μικροσκόπιο του εργαστηρίου και στην κάτω εικόνα ο 3D εκτυπωτής του εργαστηρίου.



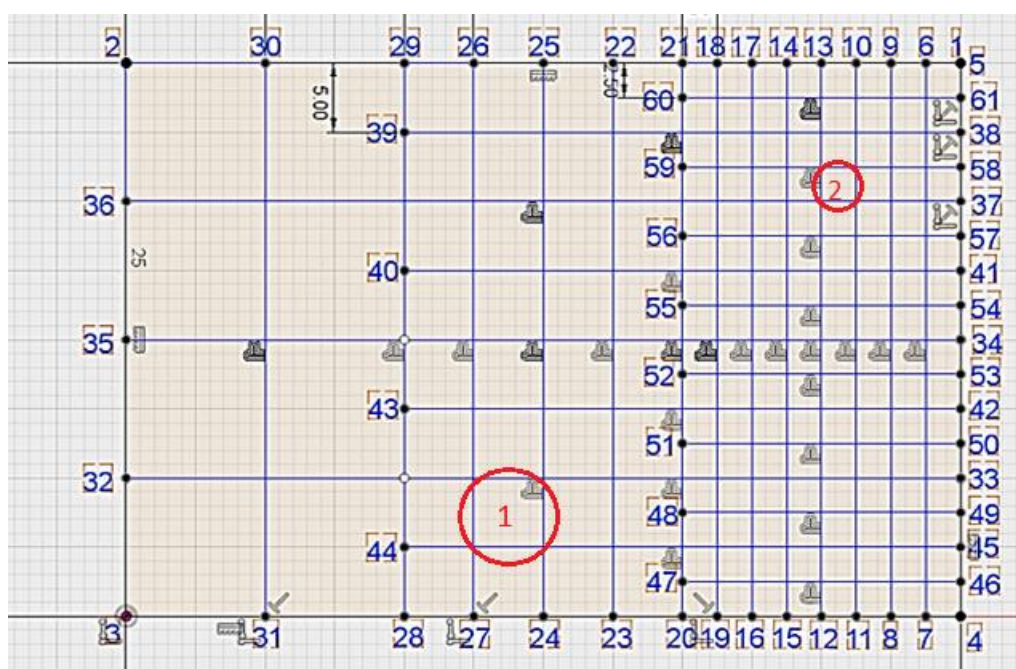
Εικόνα 34: Στην εικόνα φαίνεται η κεφαλή του εκτυπωτή (αριστερά), ενώ η δεξιά εικόνα αποτελεί στιγμιότυπο κατά τη διαδικασία εκτύπωσης ενός από τα δείγματα της διπλωματικής εργασίας.

5 ΑΠΟΤΕΛΕΣΜΑΤΑ - ΣΧΟΛΙΑΣΜΟΣ

5.1 ΕΚΤΥΠΩΜΕΝΑ ΔΕΙΓΜΑΤΑ – ΠΑΡΑΤΗΡΗΣΗ ΜΕ ΤΟ ΟΠΤΙΚΟ ΜΙΚΡΟΣΚΟΠΙΟ

5.1.1 ΣΥΓΚΡΙΣΗ ΔΟΚΙΜΙΩΝ ΟΡΘΟΓΩΝΙΚΗΣ ΔΟΜΗΣ ΜΕΤΑΞΥ ΜΕΘΟΔΩΝ ΚΑΤΑΣΚΕΥΗΣ ΜΕ CUSTOM ΚΩΔΙΚΑ ΚΑΙ CAD ΜΟΝΤΕΛΟΥ

Η παρατήρηση και οι μετρήσεις στο μικροσκόπιο, έγιναν σε σημεία μακριά από την εξωτερική περίμετρο του σχήματος και σε τυχαία σημεία στη δομή που να μπορούν να συγκριθούν. Επίσης σε κάθε τετράγωνο που έγινε η παρατήρηση, πάρθηκαν 4 μετρήσεις της ίνας σε 9 διαφορετικά σημεία. Στη κυκλική δομή έγιναν παρατηρήσεις και μετρήσεις σε πολλά σημεία της, ωστόσο λόγω μη διακριτού προσανατολισμού, όλες οι μετρήσεις ήταν σε τυχαία και διαφορετικά σημεία.

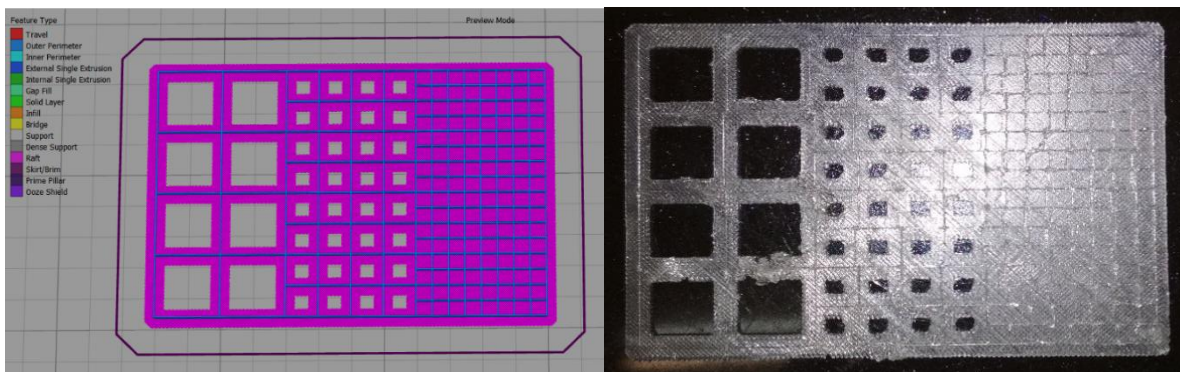


Εικόνα 35: Σχήμα ορθογωνικής δομής με πορώδες 10x10, 5x5 έως 2.5x2.5 mm. Τα κυκλωμένα σημεία 1 και 2 είναι αυτά που παρατηρήθηκαν με το οπτικό μικροσκόπιο και πάρθηκαν μετρήσεις για το προσδιορισμό του πλάτους της ίνας (μm).

5.1.1.1 ΟΡΘΟΓΩΝΙΚΗ ΔΟΜΗ ΜΕ ΔΙΑΒΑΘΜΙΣΜΕΝΟ ΠΟΡΩΔΕΣ 10x10, 5x5 ΚΑΙ 2.5x2.5 mm.

5.1.1.1.1 1° ΔΕΙΓΜΑ CAM 40x60

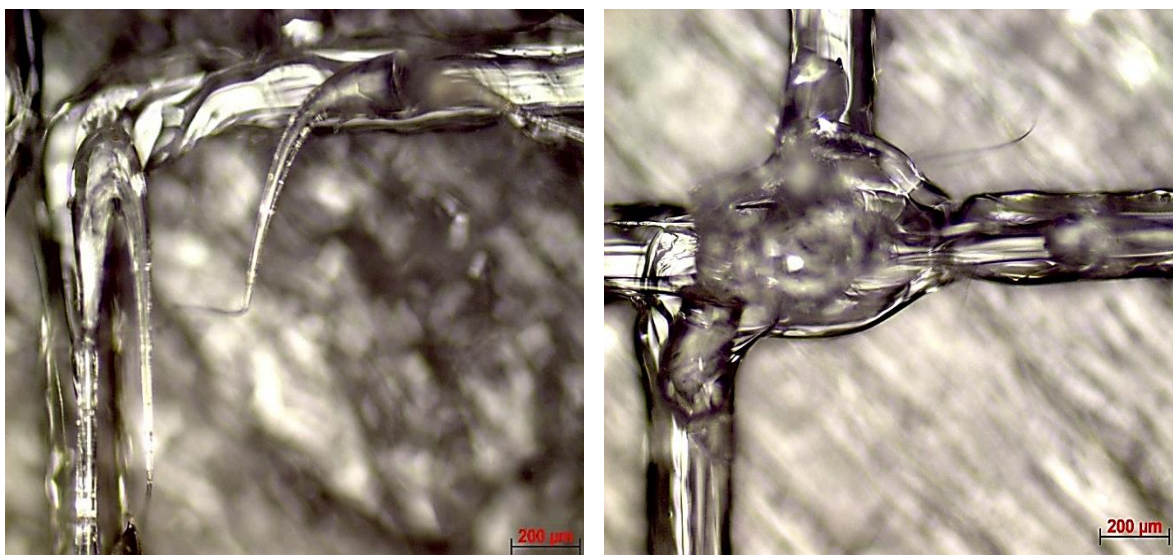
Το δείγμα αυτό προέρχεται από G-code που συντάχτηκε από το Simplify3D, όπου ακολουθήθηκε η βέλτιστη διαδρομή. Οι διαστάσεις του είναι 40 x 60 mm και ονομάστηκε CAM 40x60. Η δημιουργία του είναι περισσότερο δοκιμαστική. Στα επόμενα δείγματα έγινε μείωση των διαστάσεων των δοκιμών για να μειωθεί ο χρόνος κατασκευής τους.



Εικόνα 36: Δείγμα CAM 40x60 δοκιμαστικό που ακολουθεί τη βέλτιστη διαδρομή που συνιστά το λογισμικό εκτύπωσης, στα αριστερά η προσομοιωμένη δομή από το λογισμικό εκτύπωσης και στα δεξιά το εκτυπωμένο δείγμα.

- **Παρατηρήσεις:**

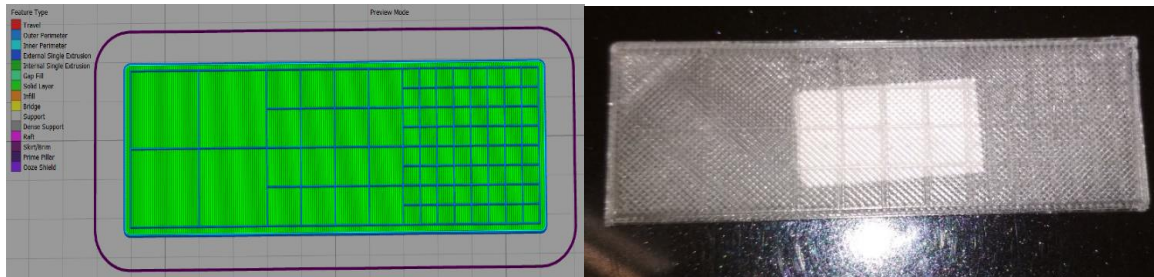
Οι ίνες δε φαίνονται να διατηρούν καλή ομοιομορφία και στα σημεία συγκόλλησης παρουσιάζουν κάποιο είδος μετατόπισης. Αυτό ίσως να οφείλεται στο γεγονός ότι η διαδρομή που ακολουθείται με βάση το λογισμικό εκτύπωσης, να μην έχει μεγάλη ακρίβεια ως προς τη θέση εναπόθεσης της ίνας. Η μέση τιμή του πλάτους των ινών είναι στα $222 \pm 16 \mu\text{m}$, μεγαλύτερη από την επιθυμητή μέση τιμή που είναι στα $200 \mu\text{m}$.



Εικόνα 37: Παρατήρηση ινών στο οπτικό μικροσκόπιο του CAM40X60. Έγινε παρατήρηση της μορφής των ινών, των σημείων συγκόλλησής τους, αλλά και μέτρηση του πλάτους της ίνας.

5.1.1.1.2 2^ο ΔΕΙΓΜΑ SAMPLE 1 20x60

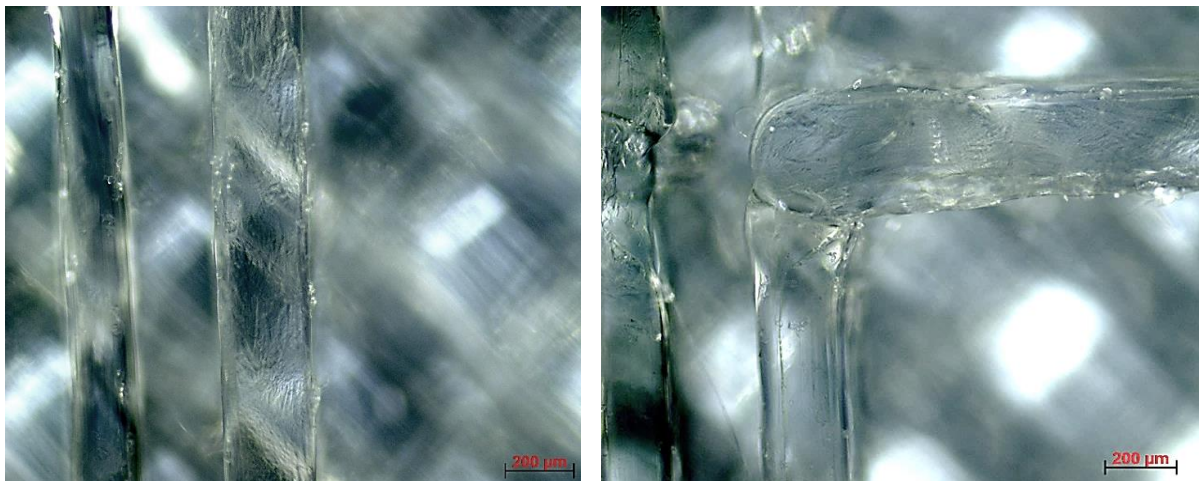
Το δείγμα αυτό προέρχεται από τη χρήση ενός CAM μοντέλου όπου ακολουθήθηκε περιμετρική πορεία για το σχηματισμό των τετραγώνων και έχει διαφορετικό υπόστρωμα. Οι διαστάσεις του είναι 20x60 mm και ονομάστηκε SAMPLE 1 20x60.



Εικόνα 38: Δείγμα SAMPLE 2 20x60 από CAM αλγόριθμο που ακολουθεί τη περιμετρική διαδρομή, στα αριστερά η προσομοιωμένη δομή από το λογισμικό εκτύπωσης και στα δεξιά το εκτυπωμένο δείγμα.

- Παρατηρήσεις:

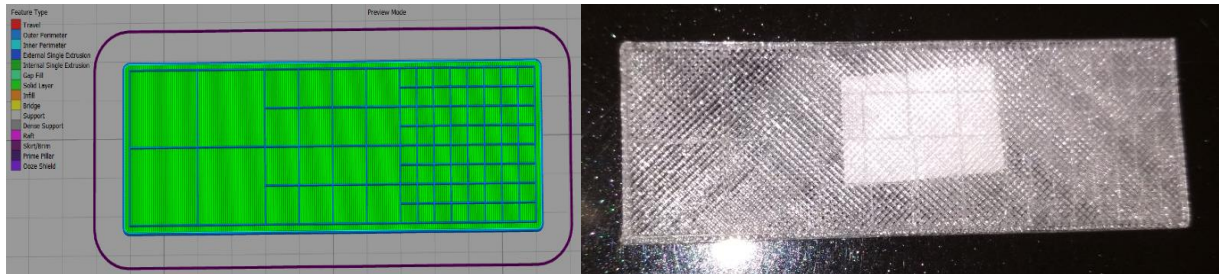
Οι ίνες φαίνονται να έχουν ομοιόμορφο σχήμα και τα σημεία συγκόλλησης δεν παρουσιάζουν κάποια μετατόπιση ή ανομοιομορφία. Ωστόσο η μέση τιμή του πλάτους των ινών είναι στα $285 \pm 13 \mu\text{m}$, αρκετά μεγαλύτερο από τα $200 \mu\text{m}$ που είναι η επιθυμητή τιμή.



Εικόνα 39: Παρατήρηση ινών στο οπτικό μικροσκόπιο του SAMPLE 1 20x60. Έγινε παρατήρηση της μορφής των ινών, των σημείων συγκόλλησής τους, αλλά και μέτρηση του πλάτους της ίνας.

5.1.1.1.3 3^ο ΔΕΙΓΜΑ SAMPLE 2 20x60

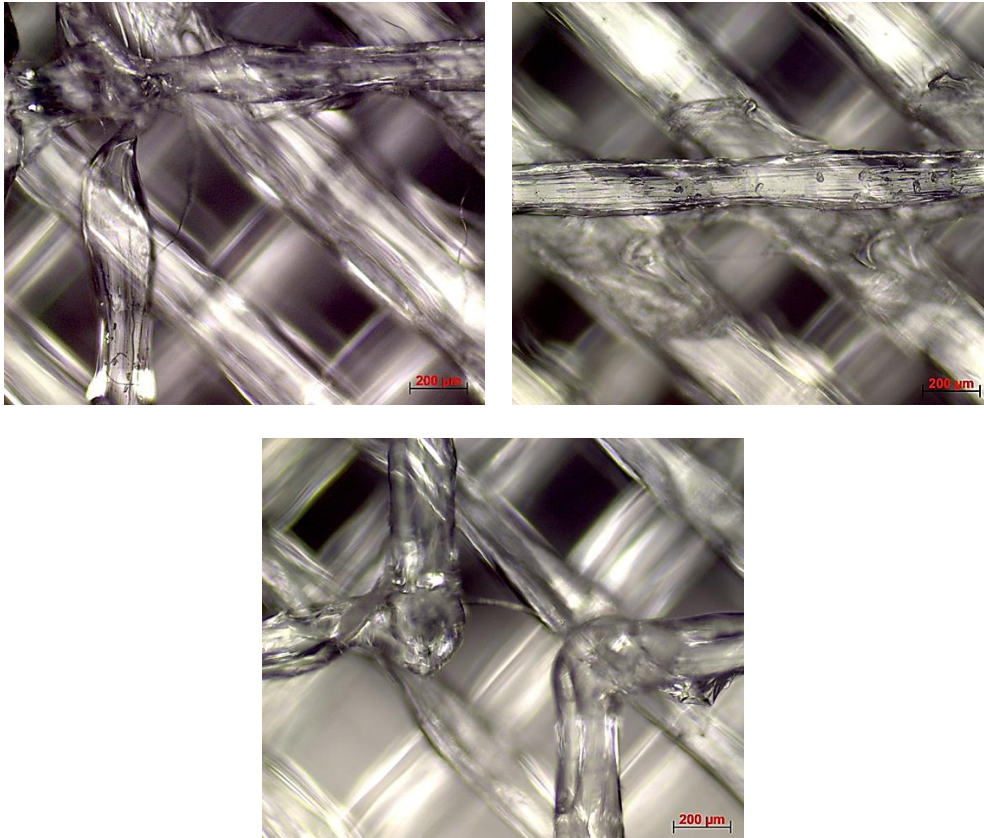
Το δείγμα αυτό προέρχεται από τη χρήση ενός CAM μοντέλου όπου ακολουθήθηκε η βέλτιστη διαδρομή, με τη κατασκευή «σκαλιών» για τη σύνθεση των τετραγώνων του πλέγματος. Οι διαστάσεις του είναι 20x60 mm και ονομάστηκε SAMPLE 2 20x60.



Εικόνα 40: Δείγμα SAMPLE 2 20x60, όπου εκτυπώθηκε από έτοιμο CAM αλγόριθμο, ακολουθώντας τη δημιουργία «σκαλιών» για τη σύνθεση του πλέγματος, στα αριστερά η προσομοιωμένη δομή από το λογισμικό εκτύπωσης και στα δεξιά το εκτυπωμένο δείγμα.

- **Παρατηρήσεις:**

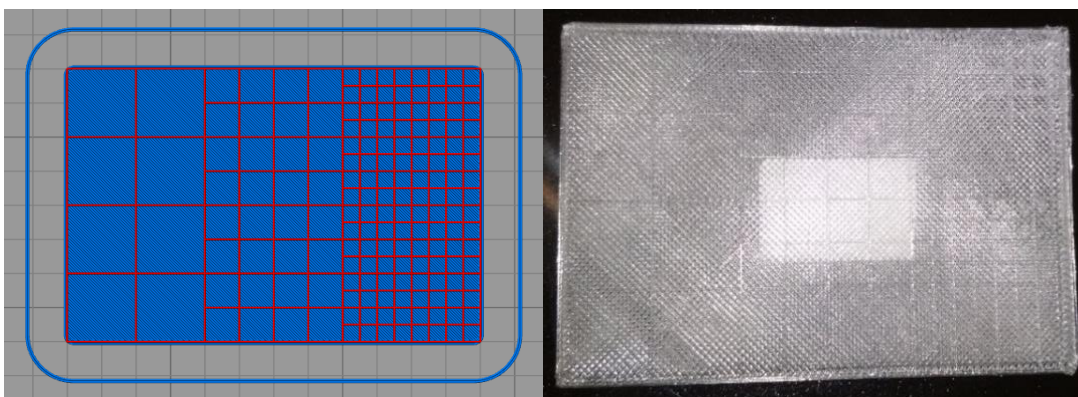
Οι ίνες δεν παρουσιάζουν ομοιομορφία και στα σημεία συγκολλήσεων είτε δεν υπάρχει καθόλου συγκόλληση είτε δεν είναι καθόλου καλή. Η μέση τιμή του πλάτους των ινών είναι 191 ± 12 μm , η οποία είναι κοντά στην επιθυμητή τιμή, 200 μm . Ωστόσο τα προβλήματα που παρουσιάζει το πλέγμα στις συγκολλήσεις, οδηγούν το δείγμα στο να μη μας προσφέρει το επιθυμητό αποτέλεσμα.



Εικόνα 41: Παρατήρηση ινών στο οπτικό μικροσκόπιο του SAMPLE 2 20x60. Έγινε παρατήρηση της μορφής των ινών, των σημείων συγκόλλησής τους, αλλά και μέτρηση του πλάτους της ίνας.

5.1.1.1.4 4ο ΔΕΙΓΜΑ SAMPLE 3 CUSTOM NO RETRACTION 40x60

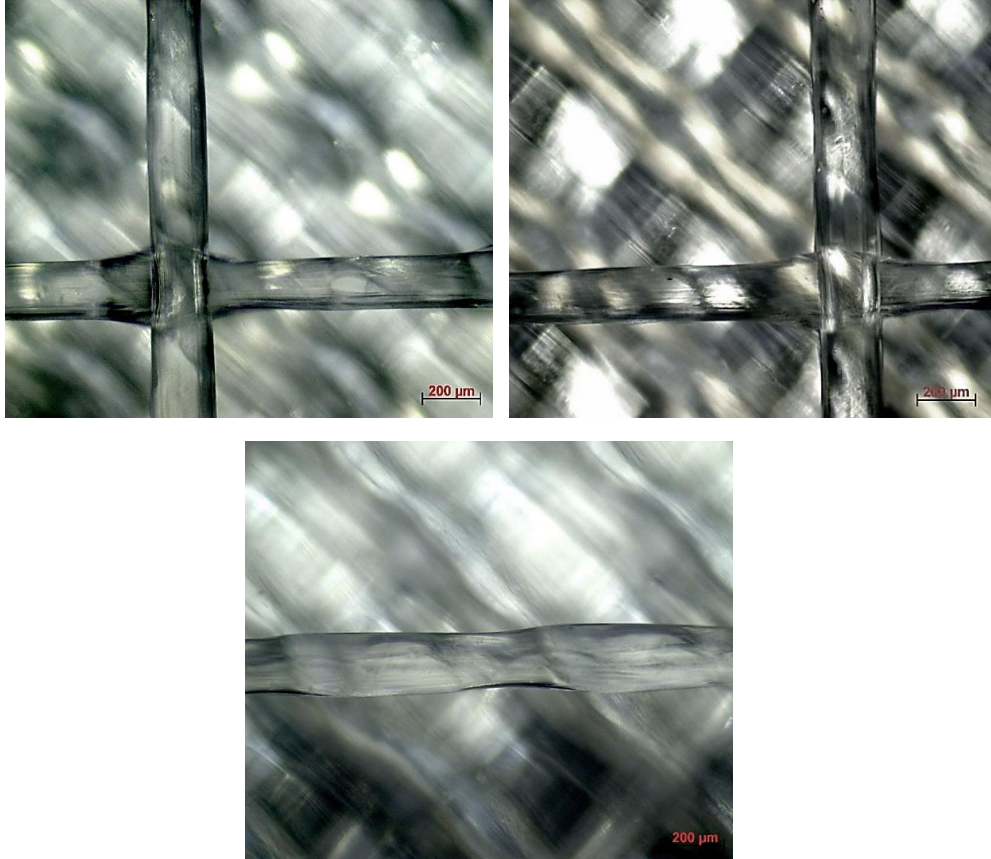
Το δείγμα αυτό προέρχεται από την υλοποίηση του κώδικα που αναπτύχθηκε στη ΔΕ. Φέρει διαστάσεις 40 x 60 mm και ονομάστηκε SAMPLE 3 custom No retraction 40x60. Σε αυτό το δείγμα, δε γίνεται αναρρόφηση υλικού κατά τη μεταφορά της κεφαλής στα σημεία όπου δε γίνεται εναπόθεση υλικού.



Εικόνα 42: Δείγμα SAMPLE 3 40x60 No Retraction, όπου εκτυπώθηκε από custom κώδικα, στα αριστερά η προσομοιωμένη δομή από το λογισμικό εκτύπωσης και στα δεξιά το εκτυπωμένο δείγμα.

- **Παρατηρήσεις:**

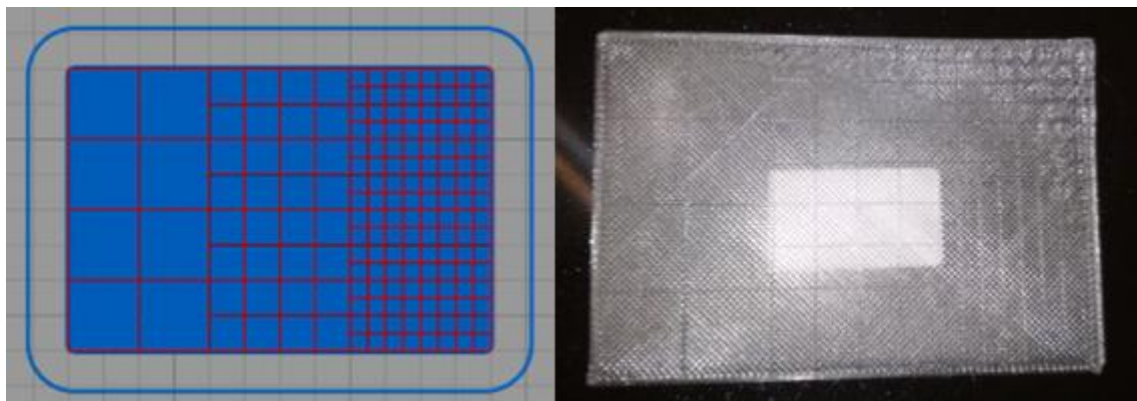
Οι ίνες φαίνονται ομοιόμορφες με σταθερό πλάτος και τα σημεία που γίνεται επικάλυψη των ινών φαίνεται πως οι ίνες πλαταίνουν, πράγμα αναμενόμενο. Επίσης ο μέσος όρος του πλάτους των ινών είναι $202 \pm 8 \mu\text{m}$, που είναι πολύ κοντά στην επιθυμητή τιμή των $200 \mu\text{m}$.



Εικόνα 43: Παρατήρηση ινών στο οπτικό μικροσκόπιο του SAMPLE 3 custom No Retraction 40x60. Έγινε παρατήρηση της μορφής των ινών, των σημείων επικάλυψής τους, αλλά και μέτρηση του πλάτους της ίνας.

5.1.1.1.5 5^ο ΔΕΙΓΜΑ SAMPLE 4 CUSTOM WITH RETRACTION 40x60

Το δείγμα αυτό προέρχεται από την υλοποίηση του κώδικα που αναπτύχθηκε στη διπλωματική εργασία με τη διαφορά από το προηγούμενο στο γεγονός ότι γίνεται αναρρόφηση υλικού, κατά τη

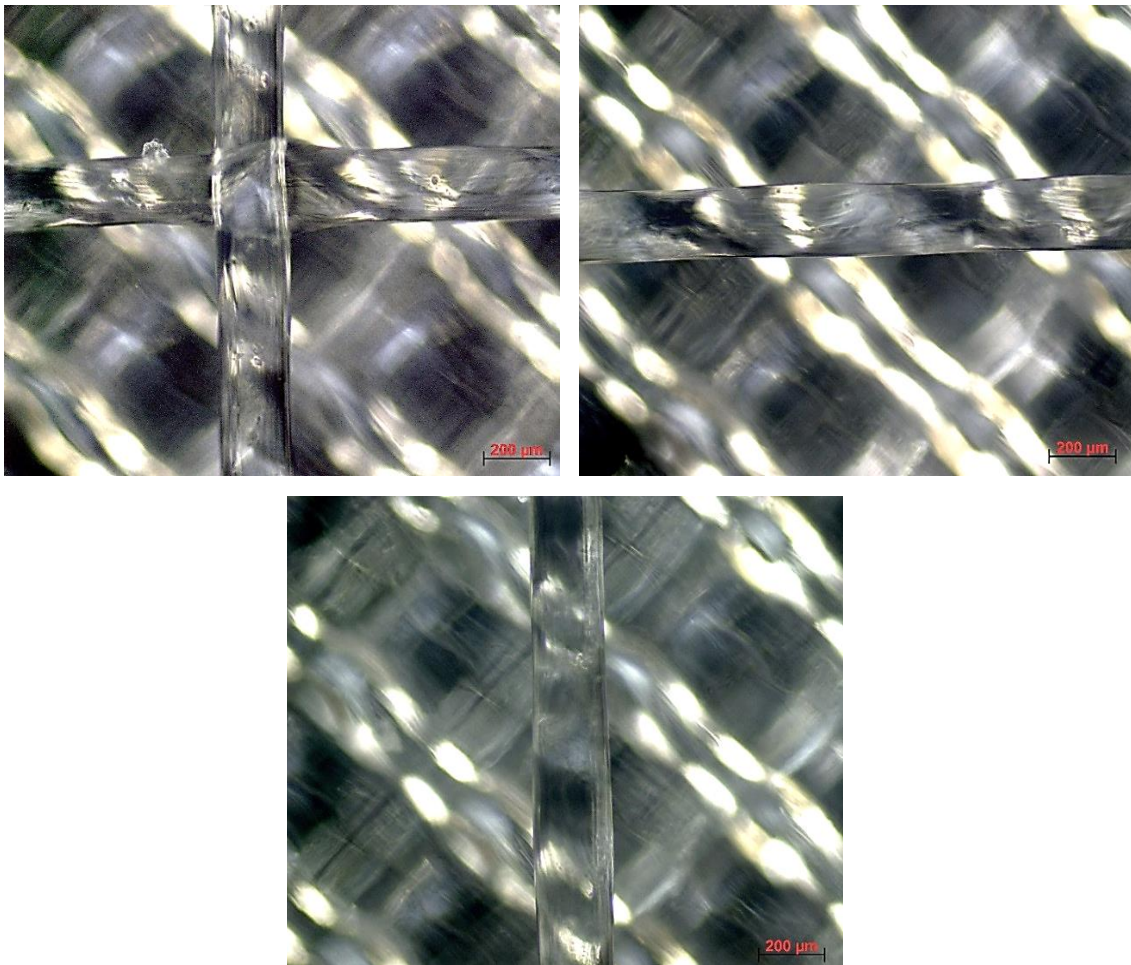


Εικόνα 44: Δείγμα SAMPLE440x60WithRetraction, όπου εκτυπώθηκε από custom κώδικα, στα αριστερά η προσομοιωμένη δομή από το λογισμικό εκτύπωσης και στα δεξιά το εκτυπωμένο δείγμα.

μεταφορά της κεφαλής, στα σημεία όπου δε γίνεται εναπόθεση υλικού. Η ονομασία του δείγματος είναι SAMPLE 4 Custom With Retraction 40x60.

- **Παρατηρήσεις:**

Οι ίνες φαίνονται ομοιόμορφες με σταθερό πλάτος. Στο σημείο επικάλυψης των ινών υπάρχει η φυσική πλάτυνσή τους, και φαίνεται να υπάρχει καλύτερη ένωση στα σημεία αυτά απ' ότι στα δείγματα που εκτυπώθηκαν μέσω CAM. Η μέση τιμή του πλάτους των ινών, ωστόσο, είναι $230 \pm 8 \mu\text{m}$, αρκετά μεγαλύτερη τιμή από την επιθυμητή, $200 \mu\text{m}$.



Εικόνα 45: Παρατήρηση ινών στο οπτικό μικροσκόπιο του SAMPLE 4 custom With Retraction 40x60. Έγινε παρατήρηση της μορφής των ινών, των σημείων επικάλυψής τους, αλλά και μέτρηση του πλάτους της ίνας.

5.1.1.1.6 Γενικές παρατηρήσεις:

Από τα δείγματα που εκτυπώθηκαν, αυτό που μπορεί να γίνει αποδεκτό και παρουσίασε τα καλύτερα αποτελέσματα όσον αφορά την ομοιομορφία στο πλάτος των ινών, καλύτερη απόδοση στα σημεία συνάντησης των ινών και μέση τιμή κοντινότερη στην επιθυμητή είναι το δείγμα SAMPLE 3 custom 40x60 No Retraction. Στη συνέχεια, παρουσιάζεται ένας συγκριτικός πίνακας (Πίνακας 4) με τα δείγματα και τις παραμέτρους που ελέγχθηκαν.

Πίνακας 4: Συγκεντρωτικός και συγκριτικός πίνακας των διαφόρων ορθογωνικών δειγμάτων με πορώδες 10x10, 5x5 έως 2.5x2.5 mm που εκτυπώθηκαν, των μεθόδων που χρησιμοποιήθηκαν, τα κριτήρια που χρησιμοποιήθηκαν για την αξιολόγησή τους και τα συμπεράσματα

Δείγμα	Είδος εκτύπωσης	Ομοιομορφία ίνας	Συγκολλήσεις/Επικαλύψεις	Μέσο Πλάτος (μm)	Αποδοχή
CAM 40x60	CAM μοντέλο/ «σκαλάκια»	X	X	222 ± 16	X
SAMPLE 1 20x60	CAM μοντέλο/περιμετρικά	✓	✓	285 ± 13	X
SAMPLE 2 20x60	CAM μοντέλο/ «σκαλάκια»	X	X	191 ± 12	X
SAMPLE 3 40x60 custom No Retraction	Custom κώδικας	✓	✓	202 ± 8	✓
SAMPLE 4 custom With Retraction	Custom κώδικας	✓	✓	230 ± 8	X

Στον Πίνακα 5 παρουσιάζεται η στατιστική επεξεργασία των αποτελεσμάτων από τις μετρήσεις του πλάτους των ινών με το οπτικό μικροσκόπιο, καθώς και ένα συγκριτικό διάγραμμα από τη μέση τιμή του πλάτους των ινών (Εικόνα 46).

Πίνακας 5: Στατιστική επεξεργασία αποτελεσμάτων μετρήσεων ινών ορθογωνικών δειγμάτων με πορώδες 10x10, 5x5 έως 2.5x2.5 mm.

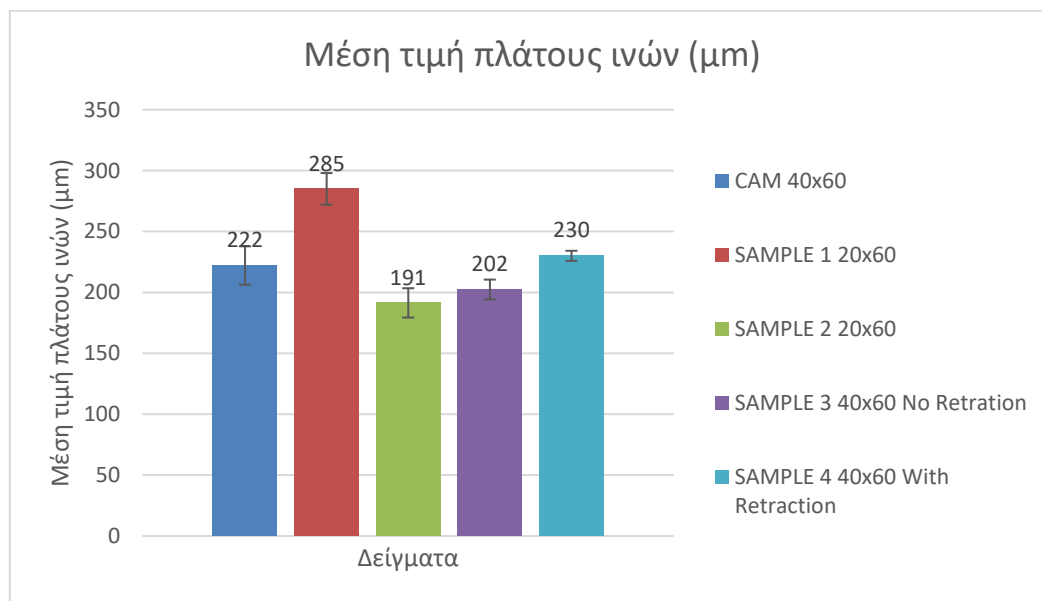
Δείγμα	\bar{x} (μm)	% απόκλιση	Τυπική απόκλιση σ (μm)	Τυπικό σφάλμα (μm)	95% CI ($\pm\mu\text{m}$)
CAM 40x60	222	11.0	48.20	8.03	16
SAMPLE 1 20x60	285	42.5	39.70	6.62	13
SAMPLE 2 20x60	191	4.3	36.90	6.15	12
SAMPLE 3 40x60 No Retraction	202	1.2	24.78	4.13	8
SAMPLE 4 With Retraction	230	15.0	24.91	4.15	8

Το τυπικό σφάλμα (SE) και τα άνω/κάτω άκρα του 95% του διαστήματος εμπιστοσύνης (95% CI) υπολογίζονται από τις σχέσεις:

$$SE = \frac{\sigma}{\sqrt{n}} \quad (15)$$

άνω/κάτω άκρο = $\bar{x} \pm 1.96 * SE$ με στρογγυλοποίηση δεκαδικών ψηφίων (16)

όπου n ο αριθμός των μετρήσεων σε κάθε σημείο, \bar{x} η μέση τιμή και σ η τυπική απόκλιση.



Εικόνα 46: Συγκριτικό διάγραμμα μέσης τιμής πλάτους ινών ορθογωνικών δομών με τετραγωνικό πορώδες μεγεθους 10x10, 5x5 έως 2.5x2.5 mm

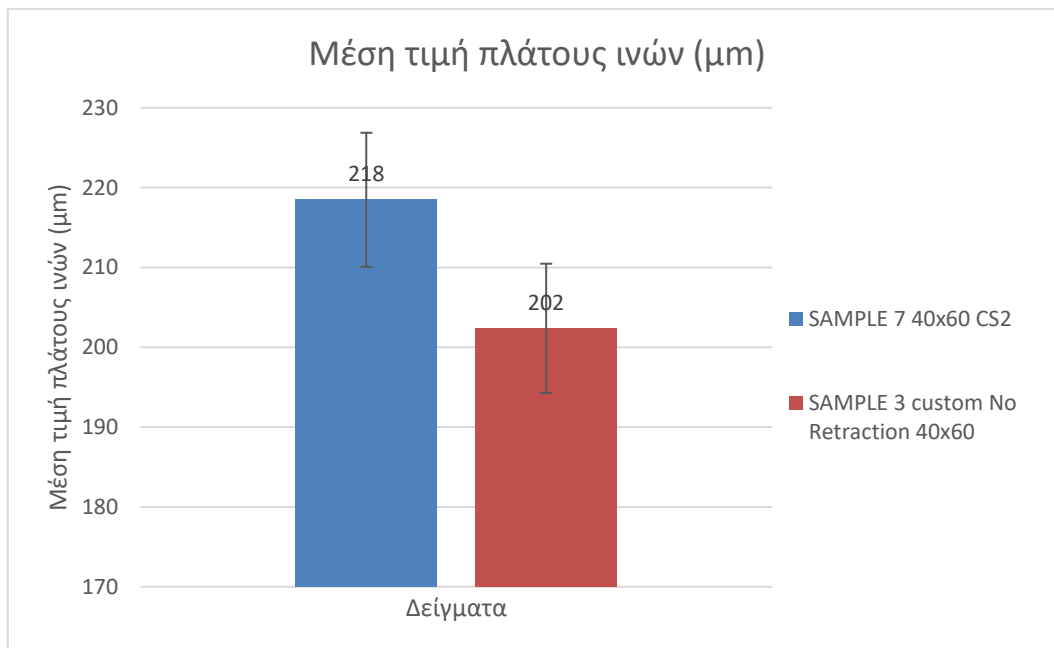
περιοχή αυτή. Τέλος, η δομή και το πορώδες έχουν ένα ικανοποιητικό αποτέλεσμα, που δείχνει, μακροσκοπικά, ότι είναι εφικτή η κατασκευή μικρότερου πορώδους. Όσον αφορά το πλάτος των ινών, η μέση τιμή κυμαίνεται στα $218 \pm 8.40 \mu\text{m}$, τιμή που είναι κοντά στην επιθυμητή $200 \mu\text{m}$.

5.1.1.2.2 Γενικές παρατηρήσεις:

Σε σύγκριση με την αποδεκτή ορθογωνική δομή με μεγαλύτερο πορώδες, η μέση τιμή του πλάτους είναι μεγαλύτερη. Αυτό μπορεί να οφείλεται στο γεγονός ότι ο πληθυσμός των ινών που συναντιούνται και επικαλύπτονται, είναι πιο συχνός, σε μικρότερα διαστήματα, οπότε η πλάτυνση των ινών είναι πιο συχνό φαινόμενο. Στη συνέχεια, παρουσιάζεται ένας συγκριτικός πίνακας μεταξύ των δύο διαφορετικών δειγμάτων και το σχετικό διάγραμμα σύγκρισης του μέσου πλάτους των ινών.

ΠΙΝΑΚΑΣ 6: Στατιστική επεξεργασία και σύγκριση της αποδεκτής ορθογωνικής δομής με το μεγάλο πορώδες SAMPLE 3 40x60 No Retraction με την ορθογωνική δομή μικρού πορώδους 2x2, 1x1 έως 0.5x0.5 mm.

Δείγμα	\bar{x} (μm)	% απόκλιση	Τυπική απόκλιση σ	Τυπικό σφάλμα	95% CI
SAMPLE 3 40x60 No Retraction	202	1.2	24.78	4.13	8
SAMPLE 7 CS2	218	9.2	25.70	4.28	8



Εικόνα 49: Διάγραμμα σύγκρισης ορθογωνικής δομής με πορώδες 2x2, 1x1 έως 0.5x0.5 mm και αποδεκτής ορθογωνικής δομής με πορώδες 10x10, 5x5 έως 2.5x2.5 mm.

5.2 ΣΥΓΚΡΙΣΗ ΔΟΚΙΜΙΩΝ ΚΥΚΛΙΚΗΣ ΔΟΜΗΣ ΜΕΤΑΞΥ ΜΕΘΟΔΩΝ ΚΑΤΑΣΚΕΥΗΣ ΜΕ CUSTOM ΚΩΔΙΚΑ ΚΑΙ CAM ΜΟΝΤΕΛΟΥ

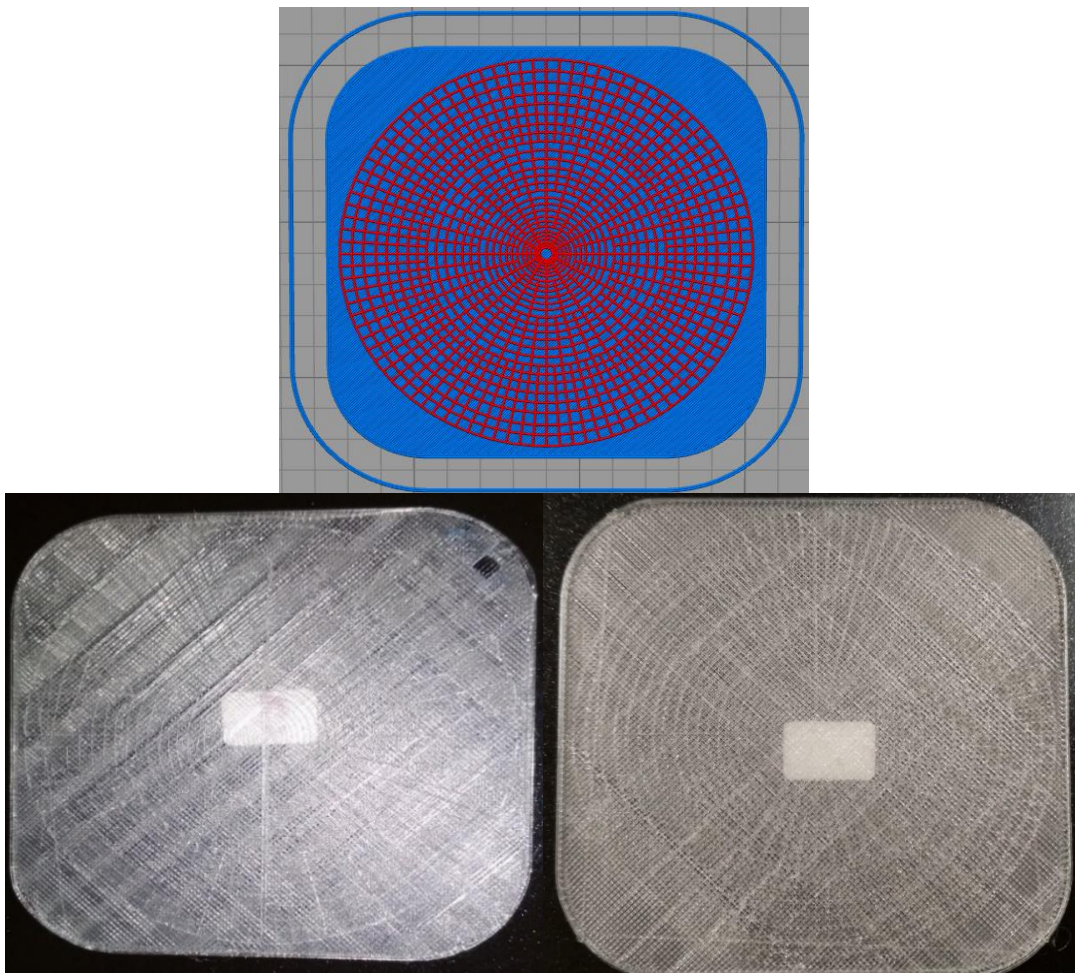
Τα δείγματα που κατασκευάστηκαν προήλθαν από τη χρήση έτοιμου CAM μοντέλου αλλά και από τον κώδικα που αναπτύχθηκε στην διπλωματική.

5.2.1.1.1 Δείγμα 1°

Το δείγμα αυτό προήλθε από τη χρήση CAM μοντέλου που ακολουθεί τη βέλτιστη διαδρομή. Το δείγμα αυτό έχει διαστάσεις 30x70 mm και ονομάζεται SAMPLE 1 CS3.

5.2.1.1.2 Δείγμα 2°

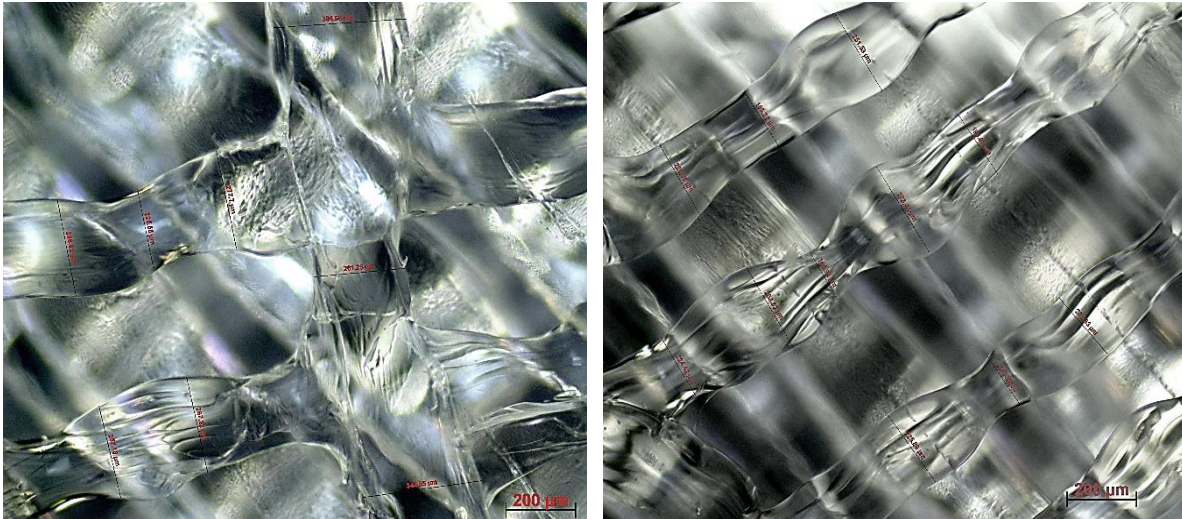
Το δείγμα αυτό προήλθε από τη χρήση του κώδικα που υλοποιήθηκε σε αυτή τη διπλωματική εργασία. Πρέπει πρώτα να σημειωθεί πως αντιμετωπίστηκαν αρκετά προβλήματα με τη συγκεκριμένη δομή, καθώς, αρχικά δεν μπορούσαν να εκτυπωθούν όλες οι γραμμές – ακτίνες λόγω της πολύ μικρής ροής. Τα καλύτερα αποτελέσματα παρουσιάστηκαν από τη μέθοδο γραμμικοποίησης των τόξων των κύκλων όπως περιγράφεται στη Παράγραφο 4.1.3. Το δείγμα ονομάστηκε ARC LIN 6.



Εικόνα 50: Προσομοίωση κυκλικής δομής (πάνω) από το λογισμικό εκτύπωσης του 3D printer. Κάτω αριστερά παρουσιάζεται η κυκλική δομή SAMPLE 1 CS3 που προέρχεται από την εκτύπωση CAM μοντέλου. Κάτω δεξιά παρουσιάζεται η κυκλική δομή που προήλθε από τη χρήση custom κώδικα ARC LIN 6.

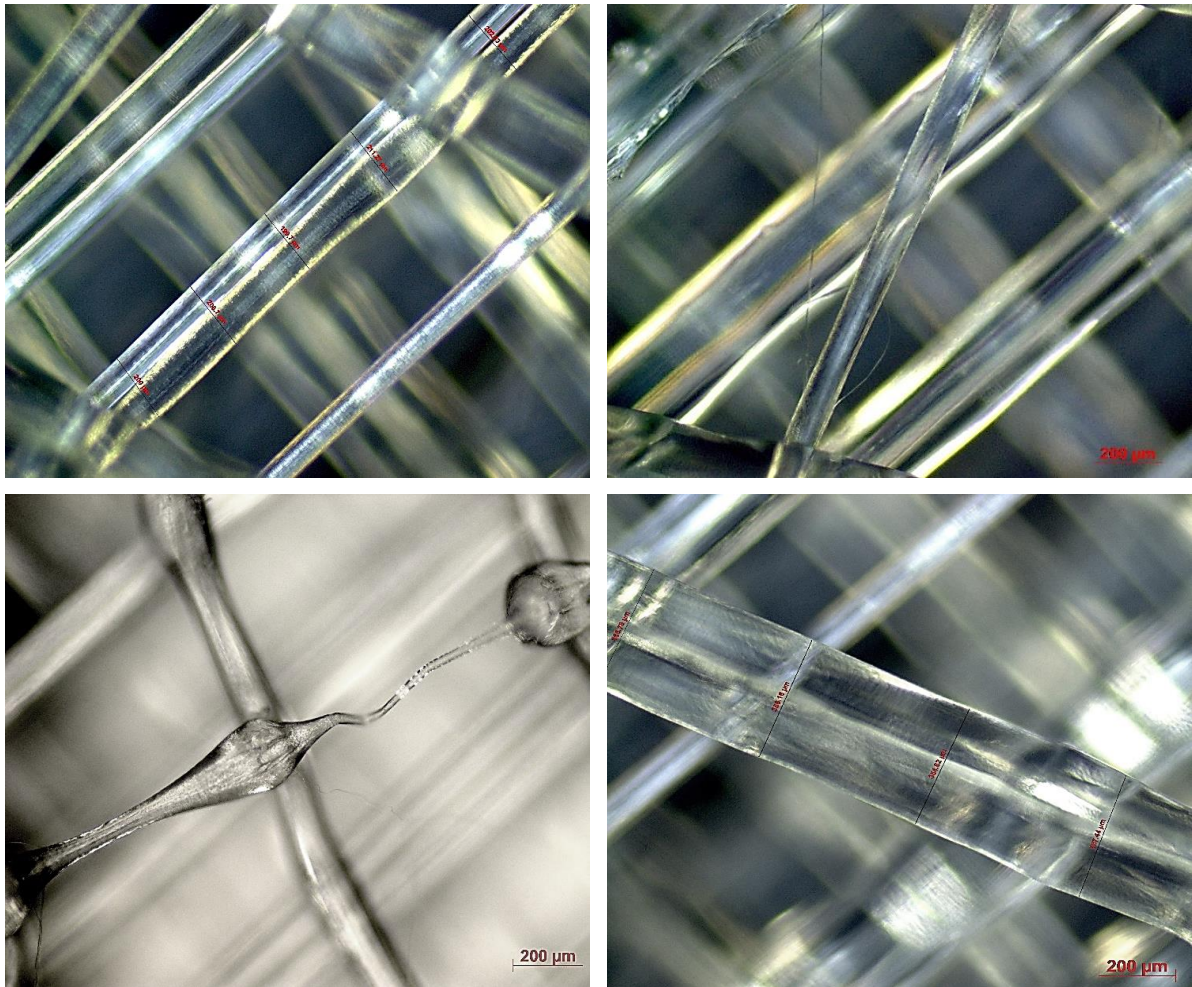
- **Παρατηρήσεις:**

Σε αυτό το σημείο να σημειωθεί ότι σε αυτή τη δομή, με τη χρήση του κώδικα, αντιμετωπίστηκαν κάποια προβλήματα κατά την εκτύπωσή της, που αφορούν τη πολύ μικρή εξώθηση και εναπόθεση υλικού κατά την εκτύπωση. Έγιναν διάφορες αλλαγές και προσπάθειες προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα. Αρχικά, έγινε προσπάθεια αύξησης της εξώθησης υλικού, πολλαπλασιάζοντας με έναν αριθμητικό συντελεστή τον όρο E_i στην εξίσωση (6) χωρίς μεγάλη επιτυχία. Στη συνέχεια, έγινε αλλαγή σε κάποιες εντολές του κώδικα για να ταιριάζουν με το τρόπο παρουσίασης των εντολών του λογισμικού, παραδείγματος χάριν οι εντολές G1 έγιναν G01 κ.ο.κ. Η αλλαγή αυτή, δε συνεπάγεται ότι είναι λάθος ο τρόπος γραφής στις προηγούμενες περιπτώσεις καθώς η εντολή είναι η ίδια. Ο λόγος που έγινε αυτή η αλλαγή, ήταν απλώς για να εξεταστεί το ενδεχόμενο αυτό, και να απορριφθεί, προκειμένου να βγει ένα πόρισμα. Ένας άλλος παράγοντας που μεταβλήθηκε είναι η ταχύτητα εκτύπωσης F που από 1200 που είχε ρυθμιστεί αρχικά, έγινε 900. Ωστόσο δεν υπήρχε πάλι κάποια αλλαγή. Σε αυτή τη φάση, έγινε η σκέψη για τη γραμμικοποίηση των κύκλων και η αλλαγή της διαδρομής εκτύπωσής τους σε δύο τόξα. Αυτό φαίνεται πως βελτίωσε τη ποιότητα εκτύπωσης και έδωσε το παραπάνω αποτέλεσμα στο δείγμα ARC LIN 6, Εικόνα 50.



Εικόνα 51: Παρατήρηση ινών στο οπτικό μικροσκόπιο του SAMPLE 1 CS3. Έγινε παρατήρηση της μορφής των ινών, των σημείων συγκολλήσεων τους, αλλά και μέτρηση του πλάτους της ίνας.

Παρατηρείται πως στο κέντρο του ARC LIN 6, εμφανίζεται μια έντονη διάμετρος. Αυτή δημιουργήθηκε ουσιαστικά από τη κίνηση της κεφαλής που περνάει από το κέντρο κάθε φορά που τελειώνει τη δημιουργία ενός ημικυκλίου και προχωράει στον επόμενο κύκλο. Ακόμη είναι φανερό ότι οι ίνες είναι αρκετά λεπτές και εύθραυστες και σε κάποια σημεία δε φαίνεται να έχει γίνει καλή εκτύπωση. Τέλος, πάνω δεξιά υπάρχει εμφανής απουσία υλικού. Το raft σε αυτή τη δομή ήταν αρκετά λεπτό και εύθραυστο και κατά την αποκόλληση από την πλατφόρμα φαίνεται να υπήρξε αυτή η αστοχία. Παρατηρείται πως στο δείγμα SAMPLE 1 CS3, που δημιουργήθηκε μέσω CAM, δεν υπάρχει ομοιομορφία στο πλάτος των ινών. Ωστόσο οι συγκολλήσεις φαίνεται να είναι καλά εφαρμοσμένες. Η μέση τιμή του πλάτους των ινών κυμαίνεται στα $259 \pm 26 \mu\text{m}$, αρκετά μεγαλύτερη από την επιθυμητή, $200 \mu\text{m}$. Για το δείγμα ARC LIN 6 είναι εμφανές πως υπάρχει δεν υπάρχει ομοιομορφία στο πλάτος μεταξύ των ινών ούτε στα τόξα ούτε στις διαμέτρους. Οι ίνες εμφανίζονται σε κάποια σημεία πολύ λεπτές, σε κάποια άλλα κοντά στην επιθυμητή τιμή και σε κάποια άλλα μεγαλύτερες. Ωστόσο, αυτό ήταν το καλύτερο αποτέλεσμα που εξάχθηκε από τη συγγραφή κώδικα. Η μέση τιμή



Εικόνα 52: Παρατήρηση ινών στο οπτικό μικροσκόπιο του ARC LIN 6. Έγινε παρατήρηση της μορφής των ινών, των σημείων συγκολλήσεων τους, αλλά και μέτρηση του πλάτους της ίνας. Στις εικόνες πάνω παρουσιάζονται ίνες από τους κύκλους. Πάνω αριστερά είναι ίνα από τον 3^ο κύκλο (από μέσα προς τα έξω) και πάνω δεξιά είναι ίνα από τον 13^ο κύκλο. Στις εικόνες κάτω παρουσιάζονται ίνες από τις διαμέτρους των κύκλων.

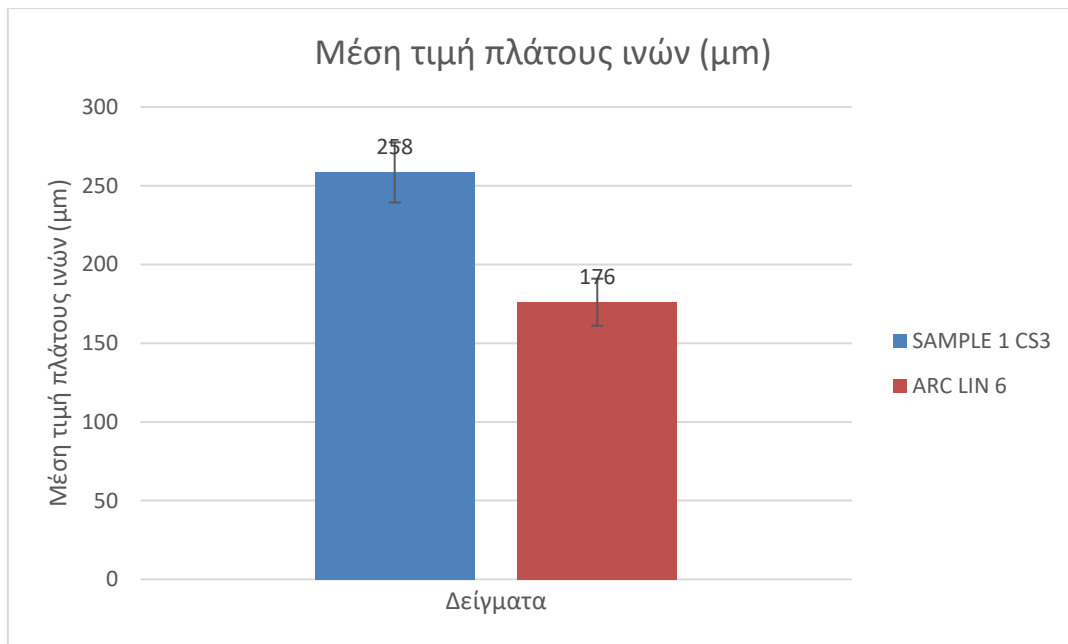
του πλάτους των ινών κυμαίνεται στα $176 \pm 21 \mu\text{m}$, αρκετά χαμηλότερη από το επιθυμητό πλάτος που είναι ίσο με $200 \mu\text{m}$.

5.2.1.1.3 Γενικές παρατηρήσεις

Στη κυκλική δομή και οι δύο τρόποι, CAM και custom κώδικας, δεν επέφεραν καλά αποτελέσματα καθώς και στις δύο περιπτώσεις υπάρχει απόκλιση από την επιθυμητή τιμή του πλάτους των ινών αλλά και μορφολογικά, δεν παρουσιάστηκε κάποιο αξιολογικό αποτέλεσμα. Παρακάτω φαίνεται ένας συγκριτικός πίνακας των δειγμάτων και το αντίστοιχο διάγραμμα:

Πίνακας 7: Στατιστική επεξεργασία δειγμάτων κυκλικής δομής προερχόμενα από CAM μοντέλο και custom κώδικα.

Δείγμα	\bar{x} (μm)	% απόκλιση	Τυπική απόκλιση σ	Τυπικό σφάλμα	95% CI
SAMPLE 1 CS3	259	29.2	81.03	9.83	19
ARC LIN 6	176	12.0	63.39	7.58	15



Εικόνα 53: Συγκριτικό διάγραμμα της μέσης τιμής του πλάτους των ινών μεταξύ των δειγμάτων SAMPLE 1 CS3, που προέκυψε από CAM μοντέλο, και του ARC LIN 6, που προέκυψε από τη χρήση του custom κώδικα με γραμμικοποίηση των τόξων.

6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη παρούσα διπλωματική εργασία μελετήθηκε η ανάπτυξη δομών υλικών με διαβαθμισμένες ιδιότητες μέσω κώδικα που αναπτύχθηκε στο εργαστήριο, σε γλώσσα προγραμματισμού Python και η μετατροπή σε εντολές G-code, της γλώσσας του τριδιάστατου εκτυπωτή. Η μέθοδος εκτύπωσης που ακολουθήθηκε είναι Fused Filament Fabrication – FFF και ως υλικό τροφοδοσίας χρησιμοποιήθηκε το πολυμερές PLA (Polylactic Acid). Η σύνθεση του κώδικα βασίστηκε στον σχεδιασμό της τροχιάς της κεφαλής και οι παράμετροι που λήφθηκαν υπόψη για τη συγγραφή του κώδικα είναι το πλάτος της ίνας W_p (μm), η διάμετρος του ακροφυσίου $nozzle_diameter$ (μm), η οποία είναι ίση με το πλάτος της ίνας, η ταχύτητα εκτύπωσης F (mm/min) και η εξώθηση υλικού E extrusion (mm). Για να υπολογιστεί η εξώθηση υλικού E ήταν απαραίτητη η γνώση κάποιων γεωμετρικών χαρακτηριστικών και διαστάσεων που αφορούν την ίνα που εναποτίθεται καθώς και του νήματος που εισάγεται στην κεφαλή από την τροφοδοσία.

Τα λογισμικά εκτύπωσης αναπτύσσουν και ακολουθούν αλγορίθμους για την εκτύπωση βάσει των παραμέτρων που ορίζει ο χρήστης για τα χαρακτηριστικά του προς εκτύπωση μοντέλου. Αυτοί οι αλγόριθμοι έχουν ως σκοπό να προσδίδουν τη βέλτιστη διαδρομή. Επειδή, όμως, αυτό το σύστημα είναι αρκετά γενικευμένο, εξετάζεται το κατά πόσον ένας κώδικας επικεντρωμένος σε μια δομή, μπορεί να ξεπεράσει και να δώσει συγκρίσιμα αποτελέσματα σε σχέση με τα λογισμικά αυτά. Ο στόχος στη συγκεκριμένη περίπτωση είναι η κατασκευή των δομών, με επιθυμητή τιμή πλάτους ινών 200 μm . Πρακτικά αυτό είναι αδύνατο να επιτευχθεί επακριβώς διότι πάντα θα υπάρχει κάποιο επί τοις εκατό σφάλμα στο ποσοστό του υλικού που ρέει από τη κεφαλή και πολλές φορές επηρεάζεται και από την συνεχή αναρρόφηση και εκρόφηση υλικού που μπορεί να μην είναι αρκετά αποδοτική, καθώς η βαρύτητα και φαινόμενα μεταφοράς ρευστών επηρεάζουν το σύστημα.

Τα δείγματα που εκτυπώθηκαν με βάση τη παραπάνω διαδικασία συγκρίθηκαν με δείγματα που κατασκευάστηκαν με βάση μοντέλα CAM, όπου είχε επιλεγεί η βέλτιστη διαδρομή εκτύπωσης σύμφωνα με το λογισμικό. Επίσης, τα δείγματα χωρίζονται σε ειδών σχήματα α) ορθογώνιο σχήμα με τετραγωνικό διαβαθμισμένο πορώδες, και, β) κυκλική δομή με πορώδες που σχηματίζεται από τη τομή ομόκεντρων κύκλων με μια σειρά διαμέτρων. Για τη σύγκριση των δειγμάτων, έγινε οπτική παρατήρηση στο οπτικό μικροσκόπιο και μέτρηση του πλάτους των ινών σε διάφορες τοποθεσίες. Στη συνέχεια, έγινε καταγραφή των τιμών αυτών, υπολογισμός της μέσης τιμής του πλάτους των ινών, υπολογισμός της επί τοις εκατό απόκλισης από την επιθυμητή τιμή και υπολογισμός μεγεθών όπως τυπική απόκλιση και η διακύμανση. Ένα άλλο κομμάτι της αξιολόγησης των δειγμάτων, συμπεριλαμβάνει την οπτική παρατήρηση ως προς την ομοιομορφία του πλάτους των ινών και τα σημεία συγκολλήσεων ή επικαλύψεων των ινών. Όσον αφορά την ορθογωνική δομή, τα δείγματα που εκτυπώθηκαν με βάση το CAM λογισμικό ακολούθησαν μια διαδρομή που για να καταλήξει στη κατασκευή των τετραγώνων του πλέγματος κατασκεύαζε σκαλάκια από τις πλευρές τους, ενώ ο άλλος τρόπος κατασκεύαζε τα τετράγωνα περιμετρικά. Όσον αφορά την κυκλική δομή, τα δείγματα που εκτυπώθηκαν με βάση το CAM λογισμικό ακολούθησαν μια διαδρομή παρόμοια με αυτή από τα σκαλάκια της ορθογωνικής διατομής.

Αρχικά, συγκρίθηκαν τα δείγματα που εκτυπώθηκαν από CAM λογισμικό με τα δείγματα από τον custom κώδικα. Να σημειωθεί επίσης πως στον custom κώδικα αποφασίστηκε να εκτυπωθεί μία εκδοχή όπου κατά τη μεταφορά της κεφαλής από κάποιες θέσεις που δεν απαιτείται η εξώθηση υλικού, θα γίνεται αναρρόφηση της ίνας και μία εκδοχή που δε θα γίνεται αναρρόφηση της ίνας και απλώς θα σταματάει η τροφοδοσία υλικού.

Τα δείγματα που εκτυπώθηκαν από τον custom κώδικα και τον CAM ήταν δείγματα ορθογωνικής δομής με τετραγωνικό πορώδες 10x10, 5x5 έως 2.5x2.5 mm. Συγκρίθηκαν με το οπτικό μικροσκόπιο μεταξύ τους τα ίδια σημεία και λήφθηκαν μετρήσεις για τη μέση τιμή του πλάτους των ινών. Μετά από οπτική παρατήρηση και υπολογισμούς, θεωρήθηκε ως αποδεκτή εκτυπωμένη δομή αυτή που προήλθε από τον custom κώδικα και χωρίς αναρρόφηση υλικού SAMPLE 3 Custom No Retraction, (βλ. Πίνακα 4, Παράγραφος 5.1.1.1.5). Η δομή από τον custom κώδικα χωρίς αναρρόφηση έδωσε ικανοποιητικά αποτελέσματα τόσο οπτικά όσο και στην επίτευξη μέσης τιμής πλάτους ινών στα $202 \pm 8 \mu\text{m}$ (1.19% απόκλιση από την επιθυμητή τιμή) και την ταυτόχρονη διατήρηση της μικρότερης διακύμανσης τιμών του πλάτους των ινών στη δομή.

Εν συνεχεία, έγινε κατασκευή νέας ορθογωνικής δομής με μικρότερο πορώδες 2x2, 1x1, 0.5x0.5mm. Αυτή η ορθογωνική δομή κατασκευάστηκε με βάση τον custom κώδικα, καθώς τα αποτελέσματα από τη προηγούμενη διαδικασία έχουν ήδη αποδείξει ότι ο custom κώδικας δίνει αποδεκτά αποτελέσματα. Μετά τη κατασκευή αυτής της δομής έγινε οπτική παρατήρηση και λήψη μετρήσεων του πλάτους των ινών. Λόγω του μικρού πορώδους της περιοχής κυρίως 0.5x0.5 mm, εμφανίζεται συχνά πλάτυνση των ινών στα σημεία συνάντησής του λόγω επικάλυψης. Σε σύγκριση με το SAMPLE 3 Custom No Retraction, ο μέσος όρος του πλάτους των ινών είναι $218 \pm 8 \mu\text{m}$ (απόκλιση 9.24%) και μπορεί να δικαιολογηθεί από τη θεώρηση που μόλις προαναφέρθηκε.

Στην επόμενη φάση εκτυπώθηκαν οι κυκλικές δομές από CAM λογισμικό και custom κώδικα. Καμία από τις δύο περιπτώσεις δεν επέφεραν τα επιθυμητά αποτελέσματα (βλ. Πίνακα 7 Παρ. 5.2). Τα προβλήματα που αντιμετωπίστηκαν επίσης στον custom κώδικα επιλύθηκαν ως ένα μέρος με τη γραμμικοποίηση των τόξων, αλλά και πάλι τα αποτελέσματα δεν είναι ικανοποιητικά, καθώς εμφανίζεται μεγάλη ανομοιογένεια στις ίνες, δε διατηρείται ένα σταθερό μέγεθος μέσα σε κάποια όρια και η μέση τιμή είναι μικρότερη από την επιθυμητή και ίση με $176 \pm 15 \mu\text{m}$ (απόκλιση 12%). Το δείγμα από CAM μοντέλο, επίσης παρουσιάζει μεγάλη απόκλιση ίση με 29.2% από την επιθυμητή τιμή πλάτους ίνας. Επομένως, τα δοκίμια κυκλικής γεωμετρίας χρειάζονται περαιτέρω βελτίωση.

6.1 ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ ΚΑΙ ΠΕΡΑΙΤΕΡΩ ΔΙΕΡΕΥΝΗΣΗ

Από τις προηγούμενες παρατηρήσεις, έγινε ξεκάθαρο πως η μέθοδος με τον custom κώδικα για τη κατασκευή υλικών με διαβαθμισμένες ιδιότητες, είναι πολλά υποσχόμενη καθώς ο χρήστης έχει τη δυνατότητα να καθορίσει άμεσα κάποιες σημαντικές κατασκευαστικές παραμέτρους, να καθορίσει τη διαδρομή εκτύπωσης αλλά και να ορίσει τα υπολογιστικά μοντέλα για τη μορφή των ινών όταν εναποτίθενται. Ωστόσο υπάρχουν περιθώρια για βελτίωση και περαιτέρω μελέτη για το μέλλον:

1) Όσον αφορά τις κυκλικές δομές, το μεγαλύτερο πρόβλημα παρουσιάστηκε στο σχηματισμό των κύκλων και χρησιμοποιήθηκε η σύνθεση 2 τόξων/ημικυκλίων για το σχηματισμό τους. Για αυτό το λόγο, μπορεί μελλοντικά να μελετηθεί ο σχηματισμός των κύκλων από τη σύνθεση παραπάνω τόξων, ή αλλιώς να «τεμαχιστεί» ο κύκλος σε περισσότερα τόξα, όπως, για παράδειγμα, σύνθεση 4 τόξων ή και παραπάνω, ειδικότερα σε κύκλους με μεγαλύτερες διαμέτρους.

2) Μελέτη του μικρότερου πορώδους που μπορεί να εκτυπωθεί στην ορθογωνική δομή για τη διατήρηση μέσης τιμής των ινών κοντά την επιθυμητή. Ο λόγος είναι επειδή τα σημεία συνάντησης των ινών στο πορώδες 0.5x0.5 mm υπάρχει συχνή επικάλυψη των ινών και άρα πλάτυνση με αποτέλεσμα να αυξάνεται η μέση τιμή.

3) Εκτύπωση και μελέτη ορθογωνικής δομής με custom κώδικα με περισσότερα στρώματα, ανάπτυξη κατά άξονα Z.

4) Τριδιάστατη εκτύπωση και μελέτη των παραπάνω δομών με άλλα υλικά τροφοδοσίας που όπως ABS, Carbon Fiber filament και μελέτη των μηχανικών ιδιοτήτων.

7 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] G. Esteban Muñiz, "Additive Manufacturing in FP7 and Horizon 2020," no. June, p. 69, 2014.
- [2] Z. Chen *et al.*, "3D printing of ceramics: A review," *J. Eur. Ceram. Soc.*, vol. 39, no. 4, pp. 661–687, 2019.
- [3] S. A. M. Tofail, E. P. Koumoulos, A. Bandyopadhyay, S. Bose, L. O'Donoghue, and C. Charitidis, "Additive manufacturing: scientific and technological challenges, market uptake and opportunities," *Mater. Today*, vol. 21, no. 1, pp. 22–37, 2018.
- [4] V. K. Balla, S. Bose, and A. Bandyopadhyay, "Processing of bulk alumina ceramics using laser engineered net shaping," *Int. J. Appl. Ceram. Technol.*, vol. 5, no. 3, pp. 234–242, 2008.
- [5] M. W. Khaing, J. Y. H. Fuh, and L. Lu, "Direct metal laser sintering for rapid tooling: processing and characterisation of EOS parts," *Mater. Process. Technol.*, vol. 113, pp. 269–272, 2001.
- [6] L. S. Bertol, W. K. Júnior, F. P. da Silva, and C. Aumund-Kopp, "Medical design: Direct metal laser sintering of Ti-6Al-4V," *Mater. Des.*, vol. 31, no. 8, pp. 3982–3988, 2010.
- [7] S. Gabriel and C. W. Hull, "United States Patent (19)," no. 19, 1984.
- [8] M. Jiménez, L. Romero, I. A. Domínguez, M. D. M. Espinosa, and M. Domínguez, "Additive Manufacturing Technologies: An Overview about 3D Printing Methods and Future Prospects," *Complexity*, vol. 2019, 2019.
- [9] Q. Yan *et al.*, "A Review of 3D Printing Technology for Medical Applications," *Engineering*, vol. 4, no. 5, pp. 729–742, 2018.
- [10] Y. Zhao *et al.*, "Three-dimensional printing of Hela cells for cervical tumor model in vitro," *Biofabrication*, vol. 6, no. 3, 2014.
- [11] A. A. Al-Tamimi, P. R. A. Fernandes, C. Peach, G. Cooper, C. Diver, and P. J. Bartolo, "Metallic bone fixation implants: a novel design approach for reducing the stress shielding phenomenon," *Virtual Phys. Prototyp.*, vol. 12, no. 2, pp. 141–151, 2017.
- [12] J. Wind, R. S. Cooke, J. Gray, T. Fannin, and T. Fegan, "Medical rapid prototyping and 3D CT in the manufacture of custom made cranial titanium plates," *Med. Eng. Technol.*, vol. 23, pp. 26–28, 1999.
- [13] H. M. A. Kolken, S. Janbaz, S. M. A. Leeflang, K. Lietaert, H. H. Weinans, and A. A. Zadpoor, "Rationally designed meta-implants: A combination of auxetic and conventional meta-biomaterials," *Mater. Horizons*, vol. 5, no. 1, pp. 28–35, 2018.
- [14] R. Langner and A. R. Vaccaro, "Tissue Engineering Articles," *Science (80-.)*, vol. 260, pp. 920–926, 1993.
- [15] C. L. Riegger, "Anatomy of the Ankle and Foot: General Osteology and foot shape," *Phys. Ther. J.*, vol. 68, 1988.
- [16] A. K. Buldt, P. Levinger, G. S. Murley, H. B. Menz, C. J. Nester, and K. B. Landorf, "Foot posture is associated with kinematics of the foot during gait: A comparison of normal, planus and cavus feet," *Gait Posture*, vol. 42, no. 1, pp. 42–48, 2015.
- [17] M. P. Schweltnus, G. Jordaan, and T. D. Noakes, "Prevention of common overuse injuries by the use of shock absorbing insoles: A prospective study," *Am. J. Sports Med.*, vol. 18, no. 6, pp. 636–641, 1990.

- [18] P. Crabtree, V. G. Dhokia, S. T. Newman, and M. P. Ansell, "Manufacturing methodology for personalised symptom-specific sports insoles," *Robot. Comput. Integr. Manuf.*, vol. 25, no. 6, pp. 972–979, 2009.
- [19] M. Davia-Aracil, J. J. Hinojo-Pérez, A. Jimeno-Morenilla, and H. Mora-Mora, "3D printing of functional anatomical insoles," *Comput. Ind.*, vol. 95, pp. 38–53, 2018.
- [20] Nike, "Nike debuts first-ever football cleat built using 3D printing technology," *Nikeinc.Com*, 2013. [Online]. Available: <http://nikeinc.com/news/nike-debuts-first-ever-football-cleat-built-using-3d-printing-technology#/inline/17741>.
- [21] B. I. SA, "Adidas' new 3D-printed shoe can be shaped according to your feet," *10 September*, 2018. [Online]. Available: <https://www.businessinsider.co.za/how-adidas-and-carbon-made-a-3d-printed-shoe-2018-9>.
- [22] C. Scott, "Bugatti Reveals World's Largest 3D Printed Functional Titanium Component in New Brake Caliper," *January 22*, 2018. [Online]. Available: <https://3dprint.com/201088/bugatti-3d-printed-brake-caliper/>.
- [23] L. Chen and T. Goto, *Functionally Graded Materials*, vol. 2–2. 2003.
- [24] J. Gottron, K. A. Harries, and Q. Xu, "Creep behaviour of bamboo," *Constr. Build. Mater.*, vol. 66, pp. 79–88, 2014.
- [25] M. Naebe and K. Shirvanimoghaddam, "Functionally graded materials: A review of fabrication and properties," *Appl. Mater. Today*, vol. 5, pp. 223–245, 2016.
- [26] A. L. Dumont, J. P. Bonnet, T. Chartier, and J. M. F. Ferreira, "MoSi₂/Al₂O₃ FGM: Elaboration by tape casting and SHS," *J. Eur. Ceram. Soc.*, vol. 21, no. 13, pp. 2353–2360, 2001.
- [27] A. Saiyathibrahim, S. S. Mohamed Nazirudeen, and P. Dhanapal, "Processing Techniques of Functionally Graded Materials - A Review," *Proc. Int. Conf. Syst. Sci. Control. Eng. Technol.*, vol. 01, pp. 98–105, 2015.
- [28] M. M. Nemat-Alla, M. H. Ata, M. R. Bayoumi, and W. Khair-Eldeen, "Powder Metallurgical Fabrication and Microstructural Investigations of Aluminum/Steel Functionally Graded Material," *Mater. Sci. Appl.*, vol. 02, no. 12, pp. 1708–1718, 2011.
- [29] X. Jin, L. Wu, Y. Sun, and L. Guo, "Microstructure and mechanical properties of ZrO₂/NiCr functionally graded materials," *Mater. Sci. Eng. A*, vol. 509, no. 1–2, pp. 63–68, 2009.
- [30] I. Bharti, N. Gupta, and K. M. Gupta, "Novel Applications of Functionally Graded Nano, Optoelectronic and Thermoelectric Materials," *Int. J. Mater. Mech. Manuf.*, vol. 1, no. 3, pp. 221–224, 2013.
- [31] R. M. Mahamood and E. T. Akinlabi, "Laser metal deposition of functionally graded Ti₆Al₄V/TiC," *Mater. Des.*, vol. 84, pp. 402–410, 2015.
- [32] S. I. Ao, E. International Association of Engineers., M. Shukla, and S. Pityana, "World Congress on Engineering : WCE 2012 : 4-6 July, 2012, Imperial College London, London, U.K.," vol. III, p. 2065, 2012.
- [33] R. M. Mahamood, E. T. Akinlabi, M. Shukla, and S. Pityana, "Functionally graded material: An overview," *Lect. Notes Eng. Comput. Sci.*, vol. 3, no. January, pp. 1593–1597, 2012.
- [34] D. Nuruzzaman, J. Siti Nur Sakinah, M. Faizal, and B. Shah Nor, "A Review on the Fabrication Techniques of Functionally Graded Ceramic-Metallic Materials in Advanced Composites," *Sci. Res. Essays*, vol. 8, no. 21, pp. 828–840, 2013.

- [35] B. Kieback, A. Neubrand, and H. Riedel, "Processing techniques for functionally graded materials," *Mater. Sci. Eng. A*, vol. 362, no. 1–2, pp. 81–106, 2003.
- [36] G. Håkansson, L. Hultman, J. E. Sundgren, J. E. Greene, and W. D. Münz, "Microstructures of TiN films grown by various physical vapour deposition techniques," *Surf. Coatings Technol.*, vol. 48, no. 1, pp. 51–67, 1991.
- [37] Z. J. Zhou, S. X. Song, J. Du, Z. H. Zhong, and C. C. Ge, "Performance of W/Cu FGM based plasma facing components under high heat load test," *J. Nucl. Mater.*, vol. 363–365, no. 1–3, pp. 1309–1314, 2007.
- [38] V. Kumar and B. Kandasubramanian, "Processing and design methodologies for advanced and novel thermal barrier coatings for engineering applications," *Particuology*, vol. 27, pp. 1–28, 2016.
- [39] N. Oxman, "Variable property rapid prototyping: Inspired by nature, where form is characterized by heterogeneous compositions, the paper presents a novel approach to layered manufacturing entitled variable property rapid prototyping," *Virtual Phys. Prototyp.*, vol. 6, no. 1, pp. 3–31, 2011.
- [40] G. H. Loh, E. Pei, D. Harrison, and M. D. Monzón, "An overview of functionally graded additive manufacturing," *Addit. Manuf.*, vol. 23, pp. 34–44, 2018.
- [41] A. O. Aremu *et al.*, "A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing," *Addit. Manuf.*, vol. 13, pp. 1–13, 2017.
- [42] R. B. Wicker and E. W. MacDonald, "Multi-material, multi-technology stereolithography: This feature article covers a decade of research into tackling one of the major challenges of the stereolithography technique, which is including multiple materials in one construct," *Virtual Phys. Prototyp.*, vol. 7, no. 3, pp. 181–194, 2012.
- [43] S. Tamas-Williams and I. Todd, "Design for additive manufacturing with site-specific properties in metals and alloys," *Scr. Mater.*, vol. 135, pp. 105–110, 2017.
- [44] V. Birman and L. W. Byrd, "Modeling and analysis of functionally graded materials and structures," *Appl. Mech. Rev.*, vol. 60, no. 1–6, pp. 195–216, 2007.
- [45] P. M. and P. M. J.Y. Hascoet, "Manufacturing of complex parts with continuous Functionally Graded Material (FGM)," p. 234, 2011.
- [46] M. Vaezi, S. Chianrabutra, B. Mellor, and S. Yang, "Multiple material additive manufacturing - Part 1: A review: This review paper covers a decade of research on multiple material additive manufacturing technologies which can produce complex geometry parts with different materials," *Virtual Phys. Prototyp.*, vol. 8, no. 1, pp. 19–50, 2013.
- [47] P. Taylor, T. Huang, M. S. Mason, G. E. Hilmas, and M. C. Leu, "Virtual and Physical Prototyping Freeze-form extrusion fabrication of ceramic parts Freeze-form extrusion fabrication of ceramic parts," no. December 2014, pp. 37–41.
- [48] H. Chung and S. Das, "Functionally graded Nylon-11/silica nanocomposites produced by selective laser sintering," *Mater. Sci. Eng. A*, vol. 487, no. 1–2, pp. 251–257, 2008.
- [49] B. E. Carroll *et al.*, "Functionally graded material of 304L stainless steel and inconel 625 fabricated by directed energy deposition: Characterization and thermodynamic modeling," *Acta Mater.*, vol. 108, pp. 46–54, 2016.
- [50] H. Yang *et al.*, "Performance evaluation of ProJet multi-material jetting 3D printer," *Virtual*

- Phys. Prototyp.*, vol. 12, no. 1, pp. 95–103, 2017.
- [51] D. V. Kaweesa, D. R. Spillane, and N. A. Meisel, “Investigating the Impact of Functionally Graded Materials on Fatigue Life of Material Jetted Specimens,” pp. 578–592, 2017.
- [52] R. Udriou and I. C. Braga, “Polyjet technology applications for rapid tooling,” *MATEC Web Conf.*, vol. 112, pp. 1–6, 2017.
- [53] Y. L. Yap, C. Wang, S. L. Sing, V. Dikshit, W. Y. Yeong, and J. Wei, “Material jetting additive manufacturing: An experimental study using designed metrological benchmarks,” *Precis. Eng.*, vol. 50, pp. 275–285, 2017.
- [54] S. Lim, R. A. Buswell, T. T. Le, S. A. Austin, A. G. F. Gibb, and T. Thorpe, “Developments in construction-scale additive manufacturing processes,” *Autom. Constr.*, vol. 21, no. 1, pp. 262–268, 2012.
- [55] W. Han, M. A. Jafari, S. C. Danforth, and A. Safari, “Tool path-based deposition planning in fused deposition processes,” *J. Manuf. Sci. Eng. Trans. ASME*, vol. 124, no. 2, pp. 462–472, 2002.
- [56] A. L. Marsan, S. Allen, P. Kulkarni, and D. Dutta, “An Integrated Software System for Process Planning for Layered Manufacturing,” *Solid Free. Fabr. Symp.*, pp. 661–668, 1997.
- [57] R. van Weeren *et al.*, “Quality of Parts Processed by Fused Deposition,” *Solid Free. Fabr.*, pp. 314–321, 1995.
- [58] B. N. Turner, R. Strong, and S. A. Gold, “A review of melt extrusion additive manufacturing processes: I. Process design and modeling,” *Rapid Prototyp. J.*, vol. 20, no. 3, pp. 192–204, 2014.
- [59] L. G. Blok, M. L. Longana, H. Yu, and B. K. S. Woods, “An investigation into 3D printing of fibre reinforced thermoplastic composites,” *Addit. Manuf.*, vol. 22, no. April, pp. 176–186, 2018.
- [60] T. A. Osswald, J. Puentes, and J. Kattinger, “Fused filament fabrication melting model,” *Addit. Manuf.*, vol. 22, no. April, pp. 51–59, 2018.
- [61] J. T. Belter and A. M. Dollar, “Strengthening of 3D printed fused deposition manufactured parts using the fill compositing technique,” *PLoS One*, vol. 10, no. 4, pp. 1–19, 2015.
- [62] All3DP, “3D Slicer Settings for Beginners - 8 Things You Need to Know.” [Online]. Available: <https://all3dp.com/3d-slicer-settings-beginners-8-things-need-know/>.
- [63] F. Schurig and B. S. C. Science, “Slicing Algorithms for 3D-Printing,” no. Grimm 2004, pp. 1–13, 2015.
- [64] Jing Hu, “Study On STL-Based Slicing Process For 3D Printing,” *Solid Free. Fabr.*, pp. 885–895, 2017.
- [65] J. M. Hodgson, Gary, Alessandro Ranellucci, “Flow Math,” *Slic3r Manual*. [Online]. Available: <https://manual.slic3r.org/advanced/flow-math>.
- [66] C. Cookbook, “CNCCookbook’s G-Code Tutorial.” [Online]. Available: <https://www.cnccookbook.com/cnc-g-code-arc-circle-g02-g03/>.

8 ΥΠΟΜΝΗΜΑ

ΠΑΡΑΚΑΤΩ ΠΑΡΑΤΙΘΟΝΤΑΙ ΟΙ ΚΩΔΙΚΕΣ ΡΥΘΜΟΝ ΓΙΑ ΚΑΘΕ ΚΟΜΜΑΤΙ ΤΗΣ ΠΕΙΡΑΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΑΝΑΛΥΤΙΚΟΤΕΡΑ.

- Case_Study_1.py

```
1. import copy
2. import math
3.
4.
5. class Point:
6.     n = 0
7.     x = 0
8.     y = 0
9.
10.    def to_string(self):
11.        return "[" + "X" + str("{:.3f}".format(self.x)) + "," + "Y" + str("{:.3f}".format(self
    .y)) + "]"
12.
13.    def distance(self, p):
14.        return math.sqrt((p.x - self.x) ** 2 + (p.y - self.y) ** 2)
15.
16.
17. class Line:
18.     # Flag to decide if line is supposed to be drawn
19.     isle = False
20.     dist = 0
21.     lf = 0
22.
23.     # Constructor. Used by python to pass initial arguments
24.     def __init__(self, a, b):
25.         self.a = Point()
26.         self.b = Point()
27.         self.a.n = a
28.         self.b.n = b
29.
30.     def to_string(self):
31.         return "[S" + str(self.a.n) + "," + "S" + str(self.b.n) + "]"
32.
33.     def line_distance(self):
34.         self.dist = self.a.distance(self.b)
35.         return self.dist
36.
37.
38. # Class that symbolizes the whole grid.
39. class Grid:
40.     Le = []
41.     Lt = []
42.     lines = []
43.     points = []
44.     active_point = Point()
45.     F = 1200
46.     nozzle_diameter = 0.200
47.     Wp = nozzle_diameter
48.     Hp = 0.200
49.     Af = 2.74899
50.     Ap = 0
51.     lp = 0
52.     E = 0
53.
54.     # Append Active point in points array.
55.     def save_active_point(self):
56.         self.active_point.n = len(self.points) + 1
```

```

57.         self.points.append(copy.deepcopy(self.active_point))
58.
59.     # Initialize the active point and save it to points
60.     def initialize(self, starting_x, starting_y):
61.         self.active_point.x = starting_x
62.         self.active_point.y = starting_y
63.         self.save_active_point()
64.
65.     # Active point movement
66.     def up(self, distance):
67.         self.active_point.y = self.active_point.y + distance
68.         self.save_active_point()
69.
70.     def down(self, distance):
71.         self.active_point.y = self.active_point.y - distance
72.         self.save_active_point()
73.
74.     def right(self, distance):
75.         self.active_point.x = self.active_point.x + distance
76.         self.save_active_point()
77.
78.     def left(self, distance):
79.         self.active_point.x = self.active_point.x - distance
80.         self.save_active_point()
81.
82.     def up_left(self, distance):
83.         self.active_point.x = self.active_point.x - distance
84.         self.active_point.y = self.active_point.y + distance
85.         self.save_active_point()
86.
87.     # Task 1
88.     def calculate_lines(self):
89.
90.         for i in range(1, 5):
91.             l = Line(i, i + 1)
92.             l.isLe = True
93.             self.lines.append(l)
94.
95.         for i in range(5, 61):
96.             l = Line(i, i + 1)
97.             if i % 2 == 0:
98.                 l.isLe = True
99.             else:
100.                l.isLe = False
101.                self.lines.append(l)
102.
103.         for l in self.lines:
104.             if l.isLe:
105.                 self.Le.append(l)
106.
107.             else:
108.                 self.Lt.append(l)
109.
110.     def print_lines(self):
111.         j = 1
112.         for l in self.Le:
113.             print("Le" + str(j) + l.to_string())
114.             j = j + 1
115.         print
116.         j = 1
117.         for l in self.Lt:
118.             print("Lt" + str(j) + l.to_string())
119.             j = j + 1
120.         print
121.
122.     def save_lines_to_file(self):

```

```

123.     h = len(self.lines)
124.     i = 1
125.     f = open('cs1_line_extrusion_transition_list.txt', 'w')
126.     j = 1
127.     f.write("Le = { ")
128.     for l in self.Le:
129.         f.write(l.to_string())
130.         if i < h:
131.             f.write(", ")
132.             i = i + 1
133.         j = j + 1
134.     f.write("}")
135.
136.     j = 1
137.     f.write("\n" + "Lt = {")
138.     for l in self.Lt:
139.         f.write(l.to_string())
140.         if i < h:
141.             f.write(", ")
142.             i = i + 1
143.         j = j + 1
144.     f.write("}")
145.
146.     # Task 2
147.
148.     def calculate_points(self, X_dist, Y_dist, starting_x, starting_y):
149.         gradeLength = X_dist / float(3)
150.         first_gradeLength = gradeLength / float(8)
151.         second_gradeLength = gradeLength / float(4)
152.         third_gradeLength = gradeLength / float(2)
153.
154.         self.initialize(starting_x, starting_y)
155.
156.         # Perimeter
157.         self.left(X_dist)
158.         self.down(Y_dist)
159.         self.right(X_dist)
160.         self.up(Y_dist)
161.
162.         for i in range(1, 5):
163.             self.left(first_gradeLength)
164.             self.down(Y_dist)
165.             self.left(first_gradeLength)
166.             self.up(Y_dist)
167.
168.         for i in range(1, 3):
169.             self.left(second_gradeLength)
170.             self.down(Y_dist)
171.             self.left(second_gradeLength)
172.             self.up(Y_dist)
173.
174.         self.left(third_gradeLength)
175.         self.down(Y_dist)
176.
177.         self.up_left(third_gradeLength)
178.
179.         self.right(X_dist)
180.         self.up(third_gradeLength)
181.         self.left(X_dist)
182.         self.up(third_gradeLength)
183.         self.right(X_dist)
184.         self.up(second_gradeLength)
185.         self.left(2 * gradeLength)
186.
187.         for i in range(1, 2):
188.             self.down(third_gradeLength)

```

```

189.         self.right(Y_dist)
190.         self.down(third_gradeLength)
191.         self.left(Y_dist)
192.
193.     self.down(third_gradeLength)
194.     self.right(Y_dist)
195.     self.down(first_gradeLength)
196.     self.left(gradeLength)
197.     self.up(second_gradeLength)
198.     self.right(gradeLength)
199.
200.     for i in range(1, 4):
201.         self.up(second_gradeLength)
202.         self.left(gradeLength)
203.         self.up(second_gradeLength)
204.         self.right(gradeLength)
205.
206.     def print_points(self):
207.         for p in self.points:
208.             print("P" + str(p.n) + p.to_string())
209.
210.     def save_points_to_file(self):
211.         h = len(self.points)
212.         i = 1
213.         f = open('cs1_point_XY_list.txt', 'w')
214.         f.write("point_XY_list = { ")
215.         for p in self.points:
216.             f.write(p.to_string())
217.             if i < h:
218.                 f.write(", ")
219.                 i = i + 1
220.         f.write("}")
221.         f.close()
222.
223.     def pass_points_to_lines(self):
224.         for line in self.lines:
225.             line.a.x = self.points[line.a.n - 1].x
226.             line.a.y = self.points[line.a.n - 1].y
227.             line.b.x = self.points[line.b.n - 1].x
228.             line.b.y = self.points[line.b.n - 1].y
229.
230.     def print_line_distances(self):
231.         i = 1
232.         for line in self.Le:
233.             print("[Le" + str(i) + ", " + str("{:.3f}".format(line.line_distance())) + "]")
234.             i += 1
235.
236.     def save_line_distances_to_file(self):
237.         f = open('cs1_path_length_list.txt', 'w')
238.         f.write('path_length_list = {')
239.         i = 1
240.         h = len(self.Le)
241.         for line in self.Le:
242.             f.write("[Le" + str(i) + ", " + str("{:.3f}".format(line.line_distance())) + "]")
243.
244.             if i < h:
245.                 f.write(",")
246.                 i += 1
247.         f.write("}")
248.         f.close()
249.
250.     def calculate_extrusion(self):
251.         self.Ap = (self.Wp * self.Hp) - ((1 - (math.pi / 4)) * (self.Hp ** 2))
252.         for l in self.Le:
253.             self.lp = l.line_distance()
254.             l.lf = (self.Ap * self.lp) / self.Af

```



```

254.         self.E = l.lf + self.E
255.         l.E = self.E
256.         self.E = 0
257.         for l in self.lines:
258.             if l.isLe:
259.                 self.lp = l.line_distance()
260.                 l.lf = (self.Ap * self.lp) / self.Af
261.                 self.E = l.lf + self.E
262.                 l.E = self.E
263.
264.     def save_gcode_extrusion_list_to_file(self):
265.         f = open('cs1_extrusion_parameter_list.gcode', 'w')
266.         i = 1
267.         h = len(self.Le)
268.         for p in self.Le:
269.             f.write("G1 X" + str("{:.3f}".format(p.a.x)) + " Y" + str("{:.3f}".format(p.a.y))
+ " E" + str(
270.                 "{:.3f}".format(p.E) + " F" + str(self.F))
271.             if i < h:
272.                 f.write("\n")
273.                 i += 1
274.         f.close()
275.
276.     # Task 6
277.     def save_gcode_transition_list(self):
278.         f = open('cs1_transition_parameter_list.gcode', 'w')
279.         i = 1
280.         h = len(self.Lt)
281.         for p in self.Lt:
282.             f.write(
283.                 "G X" + str("{:.3f}".format(p.a.x)) + " Y" + str("{:.3f}".format(p.a.y)) + " E
0" + " F" + str(self.F))
284.             if i < h:
285.                 f.write("\n")
286.                 i += 1
287.         f.close()
288.
289.     def save_gcode_to_file(self):
290.         f = open('case_study_1.gcode', 'w')
291.         l = self.lines[0]
292.         f.write("G0 X" + str("{:.3f}".format(l.a.x)) + " Y" + str("{:.3f}".format(l.a.y)) + "
E" + str(
293.             "{:.3f}".format(l.E) + " F" + str(self.F) + "\n")
294.         for l in self.lines:
295.             if l.isLe:
296.                 f.write("G1 X" + str("{:.3f}".format(l.b.x)) + " Y" + str("{:.3f}".format(l.b.
y)) + " E" + str(
297.                     "{:.3f}".format(l.E) + " F" + str(self.F) + "\n")
298.             else:
299.                 f.write("G0 X" + str("{:.3f}".format(l.b.x)) + " Y" + str("{:.3f}".format(l.b.
y)) + " E0" + " F" + str(
300.                     self.F) + "\n")
301.         f.close()
302.
303.
304. # Code
305.grid = Grid()
306.
307. # Task 1
308.grid.calculate_lines()
309.grid.save_lines_to_file()
310.
311. # Task 2
312.grid.calculate_points(60, 40, 170, 160)
313.grid.save_points_to_file()
314.

```

```

315.# Task 3
316.grid.pass_points_to_lines()
317.grid.save_line_distances_to_file()
318.
319.# Task 5
320.grid.calculate_extrusion()
321.grid.save_gcode_extrusion_list_to_file()
322.
323.# Task 6
324.grid.save_gcode_transition_list()
325.grid.save_gcode_to_file()

```

- Case_Study_2.py

```

1. import copy
2. import math
3.
4.
5. class Point:
6.     n = 0
7.     x = 0
8.     y = 0
9.
10.    def to_string(self):
11.        return "[" + "X" + str("{:.3f}".format(self.x)) + "," + "Y" + str("{:.3f}".format(self
.y)) + "]"
12.
13.    def distance(self, p):
14.        return math.sqrt((p.x - self.x) ** 2 + (p.y - self.y) ** 2)
15.
16.
17. class Line:
18.     # Flag to decide if line is supposed to be drawn
19.     isle = False
20.     dist = 0
21.     lf = 0
22.
23.     # Constructor. Used by python to pass initial arguments
24.     def __init__(self, a, b):
25.         self.a = Point()
26.         self.b = Point()
27.         self.a.n = a
28.         self.b.n = b
29.
30.     def to_string(self):
31.         return "[S" + str(self.a.n) + ", " + "S" + str(self.b.n) + "]"
32.
33.     def line_distance(self):
34.         self.dist = self.a.distance(self.b)
35.         return self.dist
36.
37.
38. # Class that symbolizes the whole grid.
39. class Grid:
40.     Le = []
41.     Lt = []
42.     lines = []
43.     points = []
44.     active_point = Point()
45.     F = 1200
46.     nozzle_diameter = 0.200
47.     Wp = nozzle_diameter
48.     Hp = 0.200
49.     Af = 2.74899
50.     Ap = 0

```

```

51.     lp = 0
52.     E = 0
53.     Z1 = 1.000
54.     Z2 = 1.100
55.     F2 = 900
56.     F3 = 1800
57.     F4 = 1000
58.
59.     # Append Active point in points array.
60.     def save_active_point(self):
61.         self.active_point.n = len(self.points) + 1
62.         self.points.append(copy.deepcopy(self.active_point))
63.
64.     # Initialize the active point and save it to points
65.     def initialize(self, starting_x, starting_y):
66.         self.active_point.x = starting_x
67.         self.active_point.y = starting_y
68.         self.save_active_point()
69.
70.     # Active point movement
71.     def up(self, distance):
72.         self.active_point.y = self.active_point.y + distance
73.         self.save_active_point()
74.
75.     def down(self, distance):
76.         self.active_point.y = self.active_point.y - distance
77.         self.save_active_point()
78.
79.     def right(self, distance):
80.         self.active_point.x = self.active_point.x + distance
81.         self.save_active_point()
82.
83.     def left(self, distance):
84.         self.active_point.x = self.active_point.x - distance
85.         self.save_active_point()
86.
87.     def up_left(self, distance):
88.         self.active_point.x = self.active_point.x - distance
89.         self.active_point.y = self.active_point.y + distance
90.         self.save_active_point()
91.
92.     # Task 1
93.     def calculate_lines(self):
94.
95.         for i in range(1, 5):
96.             l = Line(i, i + 1)
97.             l.isLe = True
98.             self.lines.append(l)
99.
100.        for i in range(5, 301):
101.            l = Line(i, i + 1)
102.            if i % 2 == 0:
103.                l.isLe = True
104.            else:
105.                l.isLe = False
106.            self.lines.append(l)
107.
108.        for l in self.lines:
109.            if l.isLe:
110.                self.Le.append(l)
111.
112.            else:
113.                self.Lt.append(l)
114.
115.    def print_lines(self):
116.        j = 1

```

```

117.     for l in self.Le:
118.         print("Le" + str(j) + l.to_string())
119.         j = j + 1
120.     print
121.     j = 1
122.     for l in self.Lt:
123.         print("Lt" + str(j) + l.to_string())
124.         j = j + 1
125.     print
126.
127.     def save_lines_to_file(self):
128.         h = len(self.lines)
129.         i = 1
130.         f = open('cs2_line_extrusion_transition_list.txt', 'w')
131.         j = 1
132.         f.write("Le = { ")
133.         for l in self.Le:
134.             f.write(l.to_string())
135.             if i < h:
136.                 f.write(", ")
137.                 i = i + 1
138.             j = j + 1
139.         f.write("}")
140.
141.         j = 1
142.         f.write("\n" + "Lt = {")
143.         for l in self.Lt:
144.             f.write(l.to_string())
145.             if i < h:
146.                 f.write(", ")
147.                 i = i + 1
148.             j = j + 1
149.         f.write("}")
150.
151.         # Task 2
152.
153.     def calculate_points(self, X_dist, Y_dist, starting_x, starting_y):
154.         gradeLength = X_dist / float(3)
155.         first_gradeLength = gradeLength / float(8)
156.         second_gradeLength = gradeLength / float(4)
157.         third_gradeLength = gradeLength / float(2)
158.
159.         self.initialize(starting_x, starting_y)
160.
161.         # Perimeter
162.         self.left(X_dist)
163.         self.down(Y_dist)
164.         self.right(X_dist)
165.         self.up(Y_dist)
166.
167.         for i in range(1, 21):
168.             self.left(first_gradeLength)
169.             self.down(Y_dist)
170.             self.left(first_gradeLength)
171.             self.up(Y_dist)
172.
173.         for i in range(1, 11):
174.             self.left(second_gradeLength)
175.             self.down(Y_dist)
176.             self.left(second_gradeLength)
177.             self.up(Y_dist)
178.
179.         for i in range(1, 5):
180.             self.left(third_gradeLength)
181.             self.down(Y_dist)
182.             self.left(third_gradeLength)

```

```

183.         self.up(Y_dist)
184.
185.         self.left(third_gradeLength)
186.         self.down(Y_dist)
187.         self.up_left(third_gradeLength)
188.
189.         for i in range(1, 10):
190.             self.right(X_dist)
191.             self.up(third_gradeLength)
192.             self.left(X_dist)
193.             self.up(third_gradeLength)
194.
195.         self.right(X_dist)
196.         self.up(second_gradeLength)
197.         self.left(2 * gradeLength)
198.
199.         for i in range(1, 9):
200.             self.down(third_gradeLength)
201.             self.right(Y_dist)
202.             self.down(third_gradeLength)
203.             self.left(Y_dist)
204.
205.         self.down(third_gradeLength)
206.         self.right(Y_dist)
207.         self.down(third_gradeLength)
208.         self.left(Y_dist)
209.         self.down(third_gradeLength)
210.         self.right(Y_dist)
211.         self.down(first_gradeLength)
212.         self.left(gradeLength)
213.         self.up(second_gradeLength)
214.         self.right(gradeLength)
215.
216.         for i in range(1, 20):
217.             self.up(second_gradeLength)
218.             self.left(gradeLength)
219.             self.up(second_gradeLength)
220.             self.right(gradeLength)
221.
222.     def print_points(self):
223.         for p in self.points:
224.             print("P" + str(p.n) + p.to_string())
225.
226.     def save_points_to_file(self):
227.         h = len(self.points)
228.         i = 1
229.         f = open('cs2_point_XY_list.txt', 'w')
230.         f.write("point_XY_list = { ")
231.         for p in self.points:
232.             f.write(p.to_string())
233.             if i < h:
234.                 f.write(", ")
235.                 i = i + 1
236.         f.write(" }")
237.         f.close()
238.
239.     def pass_points_to_lines(self):
240.         for line in self.lines:
241.             line.a.x = self.points[line.a.n - 1].x
242.             line.a.y = self.points[line.a.n - 1].y
243.             line.b.x = self.points[line.b.n - 1].x
244.             line.b.y = self.points[line.b.n - 1].y
245.
246.     def print_line_distances(self):
247.         i = 1
248.         for line in self.Le:

```

```

249.         print("[Le" + str(i) + ", " + str("{:.3f}".format(line.line_distance())) + "]")
250.         i += 1
251.
252.     def save_line_distances_to_file(self):
253.         f = open('cs2_path_length_list.txt', 'w')
254.         f.write('path_length_list = {')
255.         i = 1
256.         h = len(self.Le)
257.         for line in self.Le:
258.             f.write("[Le" + str(i) + ", " + str("{:.3f}".format(line.line_distance())) + "]")
259.
260.             if i < h:
261.                 f.write(",")
262.                 i += 1
263.         f.write("}")
264.         f.close()
265.
266.     def calculate_extrusion(self):
267.         self.Ap = (self.Wp * self.Hp) - ((1 - (math.pi / 4)) * (self.Hp ** 2))
268.         for l in self.Le:
269.             self.lp = l.line_distance()
270.             l.lf = (self.Ap * self.lp) / self.Af
271.             self.E = l.lf + self.E
272.             l.E = self.E
273.         self.E = 0
274.         for l in self.lines:
275.             if l.isLe:
276.                 self.lp = l.line_distance()
277.                 l.lf = (self.Ap * self.lp) / self.Af
278.                 self.E = l.lf + self.E
279.                 l.E = self.E
280.
281.     def save_gcode_extrusion_list_to_file(self):
282.         f = open('cs2_extrusion_parameter_list.gcode', 'w')
283.         i = 1
284.         h = len(self.Le)
285.         for p in self.Le:
286.             f.write("G1 X" + str("{:.5f}".format(p.a.x)) + " Y" + str("{:.5f}".format(p.a.y))
287.             + " E" + str(
288.                 "{:.5f}".format(p.E) + " F" + str(self.F))
289.             if i < h:
290.                 f.write("\n")
291.                 i += 1
292.         f.close()
293.
294.     # Task 6
295.     def save_gcode_transition_list(self):
296.         f = open('cs2_transition_parameter_list.gcode', 'w')
297.         i = 1
298.         h = len(self.Lt)
299.         for p in self.Lt:
300.             f.write(
301.                 "G X" + str("{:.5f}".format(p.a.x)) + " Y" + str("{:.5f}".format(p.a.y)) + " E
302.                 0" + " F" + str(self.F))
303.             if i < h:
304.                 f.write("\n")
305.                 i += 1
306.         f.close()
307.
308.     def save_gcode_to_file(self):
309.         f = open('case_study_2.gcode', 'w')
310.         l = self.lines[0]
311.         f.write("G1 X" + str("{:.3f}".format(l.a.x)) + " Y" + str("{:.3f}".format(l.a.y)) + "
312.         F" + str(self.F3) + "\n")
313.         f.write("G1 Z" + str("{:.3f}".format(self.Z1)) + " F" + str(self.F2) + "\n")
314.         f.write("G1 E0.0000 F" + str(self.F4) + "\n")

```

```

311.         f.write("G92 E0" + "\n")
312.         for l in self.lines:
313.             if l.isLe:
314.                 f.write("G1 Z" + str("{:.3f}".format(self.Z1) + " F" + str(self.F2) + "\n"))
315.                 f.write("G1 X" + str("{:.3f}".format(l.b.x)) + " Y" + str("{:.3f}".format(l.b.
y)) + " E" + str(
316.                     "{:.4f}".format(l.E) + " F" + str(self.F) + "\n")
317.             else:
318.                 f.write("G1 Z" + str("{:.3f}".format(self.Z2)) + " F" + str(self.F2) + "\n")
319.                 f.write("G1 X" + str("{:.3f}".format(l.b.x)) + " Y" + str("{:.3f}".format(l.b.
y)) + " F" + str(
320.                     self.F) + "\n")
321.         f.close()
322.
323.
324. # Code
325. grid = Grid()
326.
327. # Task 1
328. grid.calculate_lines()
329. grid.save_lines_to_file()
330.
331. # Task 2
332. grid.calculate_points(60, 40, 170, 160)
333. grid.save_points_to_file()
334.
335. # Task 3
336. grid.pass_points_to_lines()
337. grid.save_line_distances_to_file()
338.
339. # Task 5
340. grid.calculate_extrusion()
341. grid.save_gcode_extrusion_list_to_file()
342.
343. # Task 6
344. grid.save_gcode_transition_list()
345. grid.save_gcode_to_file()

```

- Case_Study_3.py

```

1. import math
2.
3.
4. class Point:
5.     x = 0
6.     y = 0
7.
8.     def __init__(self, x, y):
9.         self.x = x
10.        self.y = y
11.
12.        def to_string(self):
13.            return "[" + "X" + str("{:.3f}".format(self.x)) + "," + "Y" + str("{:.3f}".format(self
.y)) + "]"
14.
15.        def distance(self, p):
16.            return math.sqrt((p.x - self.x) ** 2 + (p.y - self.y) ** 2)
17.
18.
19. class Circle:
20.     # Constructor. Used to pass initial arguments
21.     def __init__(self, r, center_X, center_Y, F, E):
22.         self.r = r
23.         self.center_X = center_X
24.         self.center_Y = center_Y

```

```

25.         self.F = F
26.         self.E = E
27.         self.perimeter = 2 * math.pi * r
28.         self.points = []
29.
30.         # G Code command to draw circle. Always uses the initial center of the first circle.
31.         def gcode_draw_circle(self):
32.             return "G1 X" + str("{:.3f}".format(self.center_X)) + " Y" + str(
33.                 "{:.3f}".format(self.center_Y + self.r)) + " F" + str(self.F) + "\n" \
34.                 "G2 X" + str(
35.                     "{:.3f}".format(self.center_X)) + " Y" + str(
36.                         "{:.3f}".format(self.center_Y + self.r)) + " I0.000" + " J" + str(
37.                             "{:.3f}".format(-self.r)) + " E" + str("{:.3f}".format(self.E)) + "\n"
38.
39.         def calculate_circle_points(self, n):
40.             for i in range(0, n):
41.                 angle = i * (360 / float(n))
42.                 self.points.append(Point(self.center_X + self.r * math.cos(math.radians(angle) + (
43. math.pi / 2)),
44.                                         self.center_Y + self.r * math.sin(math.radians(angle) + (
45. math.pi / 2))))
46.
47.         def print_points(self):
48.             print("Points of circle with R = " + str(self.r))
49.             i = 1
50.             for p in self.points:
51.                 print(str(i) + " X" + str("{:.3f}".format(p.x)) + " Y" + str("{:.3f}".format(p.y))
52. )
53.                 i += 1
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.

```



```

86.     def calculate_circle_extrusion(self, r):
87.         self.Ap = (self.Wp * self.Hp) - ((1 - (math.pi / 4)) * (self.Hp ** 2))
88.         self.lp = self.get_perimeter(r)
89.         lf = (self.Ap * self.lp) / self.Af
90.         self.E = lf + self.E
91.         return self.E
92.
93.     def calculate_extrusion(self, distance):
94.         self.Ap = (self.Wp * self.Hp) - ((1 - (math.pi / 4)) * (self.Hp ** 2))
95.         self.lp = distance
96.         lf = (self.Ap * self.lp) / self.Af
97.         self.E = lf + self.E
98.         return self.E
99.
100.    def calculate_all_points(self):
101.        # First grade of circles with 12 points
102.        for i in range(0, 6):
103.            self.circles[i].calculate_circle_points(12)
104.
105.        # Second grade of circles with 24 points
106.        for i in range(6, 12):
107.            self.circles[i].calculate_circle_points(24)
108.
109.        # Third grade of circles with 48 points
110.        for i in range(12, 18):
111.            self.circles[i].calculate_circle_points(48)
112.
113.        # Fourth grade of circles with 96 points
114.        for i in range(18, 27):
115.            self.circles[i].calculate_circle_points(96)
116.
117.    def save_Gcode(self):
118.        f = open('case_study_3.gcode', 'w')
119.        current_R = self.initial_R
120.        # Move to the center
121.        f.write(self.gcode_move_no_write(self.center_X, self.center_Y))
122.        # Draw first set of circles
123.        self.circles.append(
124.            Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_extrusion(current_R)))
125.        f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
126.
127.        # First grade with every set of circles having a distance of 0.40
128.        grade = 0.40
129.        for c in range(2):
130.            current_R += grade
131.            self.circles.append(
132.                Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_extrusion(current_R)))
133.            f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
134.
135.        # Second grade with every set of circles having a distance of 0.80
136.        grade = 0.60
137.        for c in range(3):
138.            current_R += grade
139.            self.circles.append(
140.                Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_extrusion(current_R)))
141.            f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
142.
143.        # Third grade with every set of circles having a distance of 1.00
144.        grade = 0.80
145.        for c in range(3):
146.            current_R += grade
147.            self.circles.append(

```

```

148.         Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_
    extrusion(current_R))
149.         f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
150.
151.         # Fourth grade with every set of circles having a distance of 1.00
152.         grade = 1.00
153.         for c in range(3):
154.             current_R += grade
155.             self.circles.append(
156.                 Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_
    extrusion(current_R))
157.                 f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
158.
159.         # Fifth grade with every set of circles having a distance of 1.20
160.         grade = 1.20
161.         for c in range(4):
162.             current_R += grade
163.             self.circles.append(
164.                 Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_
    extrusion(current_R))
165.                 f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
166.
167.         # Sixth grade with every set of circles having a distance of 1.40
168.         grade = 1.40
169.         for c in range(5):
170.             current_R += grade
171.             self.circles.append(
172.                 Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_
    extrusion(current_R))
173.                 f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
174.
175.         # Seventh grade with every set of circles having a distance of 1.60
176.         grade = 1.60
177.         for c in range(3):
178.             current_R += grade
179.             self.circles.append(
180.                 Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_
    extrusion(current_R))
181.                 f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
182.
183.         # Eighth grade with every set of circles having a distance of 1.80
184.         grade = 1.80
185.         for c in range(3):
186.             current_R += grade
187.             self.circles.append(
188.                 Circle(current_R, self.center_X, self.center_Y, self.F, self.calculate_circle_
    extrusion(current_R))
189.                 f.write(self.circles[len(self.circles) - 1].gcode_draw_circle())
190.
191.         self.calculate_all_points()
192.
193.         # Lines from 54th to 1st circle
194.         inner_circle = self.circles[0]
195.         outer_circle = self.circles[26]
196.         step = 8
197.         j = step
198.         for i in range(0, 10, 2):
199.             f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))
200.             f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points
    [i + 1].y))
201.             f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))
202.             f.write(self.gcode_move_no_write(outer_circle.points[j + step].x, outer_circle.poi
    nts[j + step].y))
203.             j += step * 2

```

```

204.
205.     i = 10
206.     f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))
207.     f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points[i +
1].y))
208.     f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))
209.
210.     # Lines from 54th to 12th circle
211.     f.write(self.gcode_move_no_write(outer_circle.points[0].x, outer_circle.points[0].y))

212.     inner_circle = self.circles[6]
213.     outer_circle = self.circles[26]
214.     step = 4
215.     j = step
216.     for i in range(0, 22, 2):
217.         f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))
218.         f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points
[i + 1].y))
219.         f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))
220.         f.write(self.gcode_move_no_write(outer_circle.points[j + step].x, outer_circle.poi
nts[j + step].y))
221.         j += step * 2
222.
223.     i = 22
224.     f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))
225.     f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points[i +
1].y))
226.     f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))
227.
228.     # Lines from 54th to 24th circle
229.     f.write(self.gcode_move_no_write(outer_circle.points[0].x, outer_circle.points[0].y))

230.     inner_circle = self.circles[12]
231.     outer_circle = self.circles[26]
232.     step = 2
233.     j = step
234.     for i in range(0, 46, 2):
235.         f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))
236.         f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points
[i + 1].y))
237.         f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))
238.         f.write(self.gcode_move_no_write(outer_circle.points[j + step].x, outer_circle.poi
nts[j + step].y))
239.         j += step * 2
240.
241.     i = 46
242.     f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))
243.     f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points[i +
1].y))
244.     f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))
245.
246.     # Lines from 54th to 36th circle
247.     f.write(self.gcode_move_no_write(outer_circle.points[0].x, outer_circle.points[0].y))

248.     inner_circle = self.circles[18]
249.     outer_circle = self.circles[26]
250.     step = 1
251.     j = step
252.     for i in range(0, 94, 2):
253.         f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))

```

```

254.         f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points
[i + 1].y))
255.         f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))

256.         f.write(self.gcode_move_no_write(outer_circle.points[j + step].x, outer_circle.poi
nts[j + step].y))
257.         j += step * 2
258.
259.         i = 94
260.         f.write(self.gcode_move_write(inner_circle.points[i].x, inner_circle.points[i].y))
261.         f.write(self.gcode_move_no_write(inner_circle.points[i + 1].x, inner_circle.points[i +
1].y))
262.         f.write(self.gcode_move_write(outer_circle.points[j].x, outer_circle.points[j].y))
263.
264.
265.circles = ConcentricCircles(1, 95, 95, 1200)
266.circles.save_Gcode()

```

- Arc_Linearization.py

```

1. import math
2. import re
3.
4.
5. class Point:
6.     x = 0
7.     y = 0
8.
9.     def __init__(self, x, y):
10.         self.x = x
11.         self.y = y
12.
13.     def to_string(self):
14.         return "[" + "X" + str("{:.3f}".format(self.x)) + "," + "Y" + str("{:.3f}".format(self.y
)) + "]"
15.
16.     def distance(self, p):
17.         return math.sqrt((p.x - self.x) ** 2 + (p.y - self.y) ** 2)
18.
19.
20. class Extrusion:
21.     Wp = 0.200
22.     Hp = 0.200
23.     Af = 2.74899
24.     Ap = 0
25.     lp = 0.05
26.     E = 0
27.     F = 1200
28.     newLines = ''
29.     FEED = 'rF1200'
30.     G0 = 'G0'
31.     distance = 0.05
32.     multi = 20
33.
34.     def getPointFromLine(self, line):
35.         m = re.search('[X]\d*\.\d*', line)
36.         x = float(m.group(0)[1:])
37.         m = re.search('[Y]\d*\.\d*', line)
38.         y = float(m.group(0)[1:])
39.         return Point(x, y)
40.
41.     # Append Extrusion parameter in every line
42.     def AddExtrusion(self):
43.         with open('Arc_Linearization.txt', 'r') as f:
44.             lines = [line.rstrip() for line in f]

```

```

45.         for line in lines:
46.             if self.FEED not in line and self.G0 not in line:
47.                 self.newLines = self.newLines + line + ' ' + ' E' + str(
48.                     "{:.4f}".format(self.calculateArcExtrusion(self.distance))) + '\n'
49.             else:
50.                 self.newLines = self.newLines + line + ' F' + str(self.F) + '\n'
51.         f.close()
52.
53.         self.E = 0
54.         p0 = Point(95.0, 126.0)
55.         with open('Only_Lines.txt', 'r') as f:
56.             lines = [line.rstrip() for line in f]
57.             for line in lines:
58.                 if 'G00' not in line:
59.                     p1 = self.getPointFromLine(line)
60.                     self.newLines = self.newLines + line + ' ' + self.FEED + ' E' + str(
61.                         "{:.4f}".format(self.calculateExtrusion(p0.distance(p1)))) + ' F' + str(
self.F) + '\n'
62.                 else:
63.                     p0 = self.getPointFromLine(line)
64.                     self.newLines = self.newLines + line + ' F' + str(self.F) + '\n'
65.         f.close()
66.
67.         print self.newLines
68.         with open('Arc_Linearization_final.gcode', 'w') as f:
69.             f.write(self.newLines)
70.             f.close()
71.
72.         # Calculate the amount of material that is extruded
73.         def calculateArcExtrusion(self, distance):
74.             self.Ap = (self.Wp * self.Hp) - ((1 - (math.pi / 4)) * (self.Hp ** 2))
75.             self.lp = distance
76.             lf = (self.Ap * self.lp) / self.Af
77.             self.E = (lf*self.multi) + self.E
78.             return self.E
79.
80.         def calculateExtrusion(self, distance):
81.             self.Ap = (self.Wp * self.Hp) - ((1 - (math.pi / 4)) * (self.Hp ** 2))
82.             self.lp = distance
83.             lf = (self.Ap * self.lp) / self.Af
84.             self.E = lf + self.E
85.             return self.E
86.
87.
88. e = Extrusion()
89. e.AddExtrusion()

```