ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΤΩΝ ΚΑΤΕΡΓΑΣΙΩΝ

Αυτόματη Εκπόνηση Φασεολογίων Τόρνευσης με Σχεδιασμό Βάσει Μορφολογικών Χαρακτηριστικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΠΑΡΜΠΟΥΝΗΣ ΕΛΕΥΘΕΡΙΟΣ

Επιβλέπων : Γ.-Χ. Βοσνιάκος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2018

Περίληψη

Τα συστήματα CAD/CAM λόγω της τεχνολογικής τους ανάπτυξης έχουν μπει όλο και περισσότερο στην καθημερινότητα των μηχανουργικών κατεργασιών. Αφενός, τα συστήματα CAM αποτελούν βασικό στοιχείο για την ανάπτυξη και την εξέλιξη ενός μηχανουργείου καθώς συμβάλουν στην ταχύτερη και ορθότερη κατεργασία των μηχανουργικών αντικειμένων. Αφετέρου, τα συστήματα CAD αποτελούν βασικό εργαλείο για τον μηχανολόγο μηχανικό και τον σχεδιαστή μηχανολογικών προϊόντων, καθώς αποτελούν την γλώσσα με την οποία επικοινωνεί ο μηχανολόγος με τον μηχανουργό που θα του κατασκευάσει τα εξαρτήματα.

Με το πέρασμα των χρόνων η ανάγκη για αλληλεπίδραση αυτών των δύο συστημάτων ολοένα και αυξάνεται διότι η επικοινωνία μεταξύ αυτών αποτελεί καίριο παράγοντα για την αποδοτικότητα της παραγωγής ενός προϊόντος. Η τελευταία ξεκινάει από το σχεδιασμό και καταλήγει στην κατασκευή του προϊόντος. Σκοπός της παρούσας διπλωματικής εργασίας λοιπόν είναι η αυτοματοποίηση αυτής της αλληλεπίδρασης με σκοπό την ελαχιστοποίηση του χρόνου που απαιτείται για την μετάβαση ενός αντικειμένου από το CAD σύστημα στο CAM σύστημα.

Αρχικά, στα κεφάλαια 1 και 2, γίνεται μια εισαγωγή στο αντικείμενο των συστημάτων CAD/CAM και αναλύονται περισσότερο οι λειτουργίες τους και περιγράφονται έννοιες όπως είναι τα μορφολογικά χαρακτηριστικά, τα φασεολόγια και οι κατεργασίες.

Στη συνέχεια, στο Κεφάλαιο 3, περιγράφεται συνοπτικά ο αλγόριθμος που έχει δημιουργηθεί για τον σκοπό της διπλωματικής εργασίας και έπειτα αναλύεται η λειτουργία του βήμα προς βήμα. Ο αναγνώστης σε αυτό το μέρος της εργασίας λαμβάνει μια πρώτη ιδέα σχετικά με τον τρόπο δράσης που μπορεί να υιοθετήσει ένας μηχανικός εφαρμόζοντας τις προτάσεις της διπλωματικής εργασίας.

Ακολουθεί το Κεφάλαιο 4 στο οποίο υπάρχει μία αναλυτική περιγραφή της εκτέλεσης του αλγορίθμου η οποία έγινε σε πραγματικό περιβάλλον εργασίας. Σε αυτό το κεφάλαιο ο αναγνώστης κατανοεί πλήρως το πώς λειτουργεί και το πώς μπορεί να εφαρμοστεί ο αλγόριθμος, καθώς επίσης βλέπει και το τελικό αποτέλεσμα.

Τέλος, στο Κεφάλαιο 5, παρουσιάζονται κάποιες προτάσεις βελτίωσης του αλγορίθμου που θα τον κάνει σε πρώτο στάδιο πιο παραγωγικό και σε δεύτερο στάδιο θα διευρύνει το πεδίο εφαρμογής του.

Λέξεις Κλειδιά: Συστήματα CAD/CAM, Μορφολογικά Χαρακτηριστικά, Φασεολόγιο, Μηχανουργείο, Σχεδιασμός Αντικειμένου

Abstract

CAD / CAM systems due to their technological development have become more and more the routine of machining processes. On the one hand, CAM systems are a key element for the development and evolution of a machinery workshop as they contribute to faster and more accurate machining of the machining parts. On the other hand, CAD systems are a basic tool for the mechanical engineer and designer of mechanical products as they are the language with which the mechanical engineer communicates with the engineer who will manufacture the products.

Over the years, the need for interaction between these two systems has been increasing and increasing because communication between these systems is a key factor in the efficiency of production of a mechanical product, which starts from design and ends up in the construction. Therefore, the purpose of this diploma thesis is to automate this interaction in order to minize the time that is required in order to move an object from the CAD system to the CAM system.

Initially, in chapters 1 and 2 there is an introduction to CAD/CAM systems' object and also an analysis of their functions. Moreover, there are descripted concepts such as morphological features, phases and machining processes.

Chapter 3 then summarizes the algorithm created for the purpose of this diploma thesis and then analyzes the step-by-step operation. The reader in this part of the thesis takes a first idea of how a mechanical engineer can perform by applying the thesis proposals.

Next is Chapter 4, in which there is a detailed description of the implementation of the algorithm that was done in a real work environment. In this chapter, the reader fully understands how the algorithm works and how it can be applied in real situations, and also sees the final result.

Finally, in Chapter 5, there are some suggestions for improving the algorithm that, in first stage, will make it more productive, and at a second stage they will broaden its scope.

Keywords: CAD/CAM Systems, Morphological Features, Phases, Machinery Workshop, Product Design

Πίνακας Περιεχομένων

Περίληψη _____ *i*

Abstract _____ ii

Πίνακας Περιεχομένων _____ iii

Κεφάλαιο 1: Εισαγωγή _____ *1*

1.1Αντικείμενο _____ 1

1.2 Ιστορικό και Κίνητρο _____ 1

1.3 Δομή Εργασίας _____ 3

Κεφάλαιο 2: Θεωρητικό Υπόβαθρο _____ *5*

2.1 Ανάπτυξη της Τεχνολογίας CAD _____ 5

2.2 Ανάπτυξη της Τεχνολογίας CAM _____ 7

2.3 Σχεδιασμός Βάσει Μορφολογικών Χαρακτηριστικών (Feature-Based Modeling) _____ 9

Κεφάλαιο 3: Προγραμματιστικό Μέρος _____ *13*

3.1 Γενική Περιγραφή του Αλγορίθμου _____ 13

3.2 Ανάλυση Αλγορίθμου _____ 15

3.3 Αλγόριθμος _____ 26

Κεφάλαιο 4: Εφαρμογή Αλγορίθμου _____ *27*

4.1 Σχεδιαστικός Προσδιορισμός Τεμαχίου _____ 27

4.2 Αποτέλεσμα Εφαρμογής Αλγορίθμου _____ 39

Κεφάλαιο 5: Συμπεράσματα και Προτάσεις _____ *45*

5.1 Προτεινόμενη Χρήση Αλγορίθμου _____ 45

5.2 Προτάσεις Βελτίωσης _____ 47

Κεφάλαιο 6: Βιβλιογραφία _____ *50*

Παράρτημα _____ *54*

Κεφάλαιο 1: Εισαγωγή

1.1 Αντικείμενο

Σκοπός της παρούσας εργασίας αποτελεί η ανάπτυξη ενός αλγορίθμου ο οποίος θα εφαρμόζεται στο σχεδιαστικό λογισμικό του χρήστη, ο οποίος χρήστης θα έχει σχεδιάσει ένα αντικείμενο προς κατασκευή με την διαδικασία της τόρνευσης. Ο αλγόριθμος αυτός θα παράγει τον G κώδικα για το συγκεκριμένο αντικείμενο. Ο G κώδικας είναι η γλώσσα η οποία «κατανοεί» μία εργαλειομηχανή CNC και ουσιαστικά αποτελείται από εντολές ελέγχου της μηχανής, όπως η αλλαγή ενός εργαλείο ή η εκκίνηση της ατράκτου, και από εντολές μετακίνησης του εργαλείου υπο ορισμένες συνθήκες.

Ο αλγόριθμος σχεδιάστηκε έτσι ώστε να προτρέπει τον σχεδιαστή να λειτουργεί με έναν συγκεκριμένο τρόπο, ένας τρόπος ο οποίος αντιπροσωπεύει σε μεγάλο βαθμό τον τρόπο λειτουργίας μιας εργαλειομηχανής CNC.

1.2 Ιστορικό και Κίνητρο

Οι υπολογιστές έχουν μεγάλη συμμετοχή στην ανάπτυξη των προϊόντων καθ' όλο τον κύκλο ζωής του προϊόντος, από τον σχεδιασμό του, την ανάλυσή του μέχρι και την κατασκευή του. Από τις αρχές της δεκαετίας του 1950, όπου και ξεκίνησε η ιδέα του Computer Aided Design (CAD), τα συστήματα αυτά έχουν επεκταθεί σε κάθε πτυχή του σχεδιασμού και της ανάπτυξης ενός προϊόντος. Τα συστήματα CAD ξεκίνησαν από ένα απλό εργαλείο σχεδιασμού δύο διαστάσεων (2D Drafting) και έχουν εξελιχθεί σε πολύ ισχυρά εργαλεία που βασίζονται σε τεχνολογία μοντελισμού στερεού. Περίπου την ίδια εποχή, εμπνεύστηκε η δημιουργία των Computer Aided Manufacturing (CAM) συστημάτων από τις εργαλειομηχανές αριθμητικού ελέγχου (Numeric Control machines, NC). Η ανάπτυξη των CAM συστημάτων έγινε ξεχωριστά από τα CAD συστήματα καθώς επίσης και από διαφορετικές ομάδες ανθρώπων. Η ανάπτυξη της τεχνολογίας CAD και CAM έχει αυξήσει δραματικά την αποδοτικότητα σε κάθε επιμέρους τομέα (Besant, 1986). Ωστόσο, η ανεξάρτητη ανάπτυξη αυτών των δύο έχει κρατήσει πίσω την συνολική ανάπτυξη της αποδοτικότητας από τον σχεδιασμό στην παραγωγή. Η επικοινωνία μεταξύ των συστημάτων CAD και CAM αποτελεί ένα εμπόδιο για περαιτέρω ανάπτυξη της αποδοτικότητας της παραγωγής.

Η πρώτη προσπάθεια διάσπασης αυτής της απομόνωσης των δύο συστημάτων ήταν η επαναχρησιμοποίηση του μοντέλου του προϊόντος που παράχθηκε στο σύστημα CAD στα συστήματα CAM. Δημιουργήθηκε η ικανότητα των συστημάτων CAM να χειρίζονται απ' ευθείας τα μοντέλα των συστημάτων CAD, είτε αυτά ήταν σε μοντέλα σύρματος (wireframe models) είτε μοντέλα στερεού (Gibbs, 1998). Όμως, το μοντέλο του CAD που δινόταν σαν είσοδος παρουσίαζε μόνο το

προϊόν όπως πρέπει να είναι στην τελική του μορφή. Οι παραδοσιακές μηχανές CNC που χρησιμοποιούνταν έπαιρναν σαν είσοδο NC προγράμματα, δηλαδή G και M εντολές, για να κατευθυνθούν εκεί που έπρεπε. Οι προγραμματιστές NC έπρεπε να γνωρίζουν τα ενδιάμεσα στάδια της μορφής του προϊόντος για να μπορέσουν να δημιουργήσουν κώδικες μηχανής για κάθε βήμα κατεργασίας. Αυτά τα ενδιάμεσα στάδια της μορφής του προϊόντος κατά την διάρκεια της κατεργασίας του δεν είναι διαθέσιμα στο μοντέλο του CAD συστήματος και πρέπει να ληφθούν είτε από την δημιουργία νέων μοντέλων είτε από την επεξεργασία του τελικού μοντέλου. Απαιτείται μεγάλη παρεμβολή από τον χρήστη και πολύ καλή γνώση των μηχανών CNC και των διεργασιών τους για να μπορέσει κανείς να δημιουργήσει κώδικες μηχανής.

Οι τεχνολογίες CAD/CAM έχουν συνεχίσει να εξελίσσονται. Η τωρινή τάση είναι τα συστήματα που βασίζονται στα features (feature-based systems). Τα features παίζουν καθοριστικό ρόλο στην ενσωμάτωση των CAD/CAM συστημάτων. Η αυτόματη αναγνώριση των features έχει επιτευχθεί σε έναν πολύ ικανοποιητικό βαθμό και έχει εφαρμοστεί στα συστήματα CAPP (Computer Aided Process Planning). Τα συστήματα CAPP που είναι βασισμένα στα features ερμηνεύουν το μοντέλο του προϊόντος σε αντικείμενα κατεργασίας και χρησιμοποιούν αυτά τα αντικείμενα για να παράγουν κατασκευαστικές οδηγίες με σκοπό την παραγωγή του προϊόντος. Με την αυτόματη αναγνώριση των features, τα συστήματα CAPP μπορούν να αναγνωρίσουν τα αντικείμενα κατεργασίας απ' ευθείας από το μοντέλο στερεού που έχει παραχθεί από ένα σύστημα CAD και να δημιουργήσουν σχέδια επεξεργασίας για το στερεό μοντέλο. Ένα προϊόν πρέπει να κατασκευαστεί για να έρθει στην ζωή. Ωστόσο, τα σχέδια κατεργασίας που παράγονται από ένα τέτοιο σύστημα δεν μπορούν να χρησιμοποιηθούν σαν σημείο εκκίνησης από τα συστήματα CAM για να δημιουργήσουν εκείνα με τη σειρά τους τροχιές εργαλείων και NC προγράμματα. Χρειάζεται πάλι ο χρήστης να δημιουργήσει χειροκίνητα τα βήματα κατεργασίας και να ορίσει την γεωμετρία της κατεργασίας για το κάθε βήμα. Χρειάζεται περεταίρω αναβάθμιση για την ύπαρξη μίας αδιάλειπτης CAD/CAM ενσωμάτωσης.

Στο πανεπιστήμιο του Κάνσας, στο τμήμα των Μηχανολόγων Μηχανικών, έχει γίνει έρευνα και έχει εφαρμοστεί αυτόματη αναγνώριση των features για τα στερεά μοντέλα για την δημιουργία σχεδίων κατεργασίας τα οποία βασίζονται στα στερεά μοντέλα και στις τεχνικές πληροφορίες των εργαλείων των μηχανών. Η έρευνα αυτή έχει γίνει με σκοπό την σύνδεση των συστημάτων CAPP και των CAM συστημάτων μέσω των αντικειμένων κατεργασίας (machining features), έτσι ώστε να μειωθεί η διεπαφή της επικοινωνίας του χρήστη και να αυτοματοποιηθεί η διαδικασία προετοιμασίας των τροχιών των εργαλείων (Brooks, 1997). Το συστήματα αναμένεται να είναι εύκολο στην επέκτασή του και στην ακριβή μεταφορά των δεδομένων.

1.3 Δομή Εργασίας

Η παρούσα εργασία αποτελείται από τέσσερα βασικά μέρη:

Το πρώτο μέρος αποτελείται από τα δύο πρώτα κεφάλαια. Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή σχετικά με το αντικείμενο της εργασίας, τι πραγματεύεται δηλαδή η εργασία και σε ποιους τομείς της μηχανολογίας και του αυτοματισμού εμπλέκεται. Επίσης, σε αυτό το κεφάλαιο γίνεται και μία αναφορά σε ιστορικά γεγονότα και σε στοιχεία που αποτελούν κίνητρο για αυτή την εργασία. Στο δεύτερο κεφάλαιο αναλύεται το θεωρητικό υπόβαθρο που αποτελεί την βάση πάνω στην οποία αναπτύχθηκε το σκεπτικό και η υλοποίηση του αλγορίθμου.

Το δεύτερο μέρος είναι ουσιαστικά το τρίτο κεφάλαιο στο οποίο παρατίθεται λεπτομερώς ο κώδικας που έχει αναπτυχθεί. Αρχικά γίνεται μια περιγραφή για τις ενέργειες που εκτελεί ο κώδικας καθώς επίσης και τον τρόπο λειτουργίας που θα πρέπει να ακολουθήσει ο χρήστης που καλείται να τον χρησιμοποιήσει. Έπειτα, αναλύεται εκτενώς ο κώδικας και οι εντολές που χρησιμοποιούνται και στο τέλος του κεφαλαίου καταγράφεται ο κώδικας αυτούσιος.

Το τρίτο μέρος της εργασίας είναι η παρουσίαση μιας εφαρμογής του αλγορίθμου που έγινε στο κεφάλαιο τέσσερα. Σκοπός αυτής της παρουσίασης είναι η πλήρης εφαρμογή της διαδικασίας που ακολουθείται για την κατασκευή ενός εξαρτήματος τόρνευσης, η οποία ξεκινάει από τον σχεδιασμό του αντικειμένου, συνεχίζεται στην εφαρμογή του αλγορίθμου και ολοκληρώνεται με την κατεργασία του αντικειμένου στην εργαλειομηχανή.

Τέταρτο και τελευταίο μέρος της εργασίας είναι το κεφάλαιο πέντε στο οποίο προτείνονται διάφοροι τρόποι εφαρμογής του αλγορίθμου καθώς επίσης και ορισμένες περιπτώσεις στις οποίες θα μπορούσε ο αλγόριθμος να εφαρμοστεί. Τέλος, αναφέρονται προτάσεις βελτίωσης του αλγορίθμου.

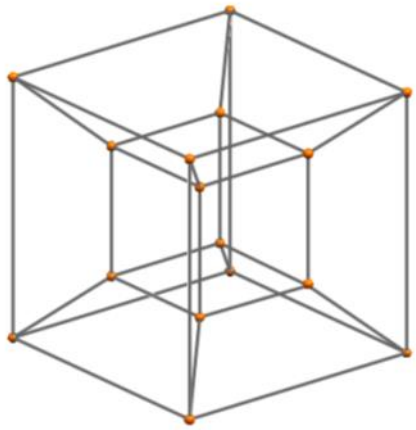
Κεφάλαιο 2: Θεωρητικό Υπόβαθρο

2.1 Ανάπτυξη της Τεχνολογίας CAD

Ο όρος «CAD σύστημα» χρησιμοποιείται για την περιγραφή κάθε λογισμικού συστήματος το οποίο είναι ικανό να ορίζει εξαρτήματα με γεωμετρία. Χρησιμοποιείται ο υπολογιστής για να βοηθήσει στην διαδικασία του σχεδιασμού και της ανάπτυξης προϊόντων (Besant, 1986). Τα συστήματα CAD προέρχονται από τα πρόωρα γραφικά υπολογιστικά συστήματα, και εξελίχτηκαν με την ανάπτυξη των διαδραστικών γραφικών των υπολογιστών και της τεχνολογίας γεωμετρικής μοντελοποίησης. Οι υπολογιστές βέβαια έχουν από πολύ καιρό που χρησιμοποιούνται για τους υπολογισμούς των μηχανικών σε δουλειές μεγάλων κλιμάκων πριν από την εμφάνιση των CAD συστημάτων. Σημείο καμπής για τα CAD συστήματα αποτελεί η ανάπτυξη του συστήματος Sketchpad στο M.I.T. το 1963 από τον Dr. Ivan Sutherland (Besant, 1986). Το Sketchpad θεωρείται ο πρόγονος των μοντέρνων CAD συστημάτων καθώς επίσης θεωρείται και μία σημαντική ανακάλυψη για τα γραφικά των υπολογιστών σε γενικό επίπεδο. Το Sketchpad ήταν το πρώτο σύστημα το οποίο έδινε στο χρήστη την δυνατότητα να αλληλεπιδρά γραφικά με τον υπολογιστή ζωγραφίζοντας πάνω σε μία CRT οθόνη με ένα στυλό φωτός. Παρουσίασε ένα πρωτότυπο ενός γραφικού περιβάλλοντος χρήστη, το οποίο αποτελεί ένα απαραίτητο χαρακτηριστικό των μοντέρνων CAD συστημάτων.

Η τεχνολογία των CAD συστημάτων έχει σημειώσει ραγδαία εξέλιξη από την στιγμή που δημιουργήθηκαν. Στις αρχές της δεκαετίας του 1970 τα συστήματα CAD ήταν πιο πολύ λογισμικά ιχνογράφησης τα οποία χρησιμοποιούνταν για την δημιουργία σχεδίων δύο διαστάσεων (2D drawings) τα οποία έμοιαζαν πολύ στα χειροποίητα σχέδια. Ο όρος CAD τότε χρησιμοποιούνταν πιο πολύ σαν Computer Aided Drafting, καθώς ο όρος Drafting χρησιμοποιείται για τα σχέδια δύο διαστάσεων. Η γεωμετρία που μπορούσε να χρησιμοποιήσει τότε ο χρήστης ήταν περιορισμένη σε πολύ απλή γεωμετρία, όπως γραμμές, κυκλικά τόξα και ελλειπτικά τόξα. Η ανάπτυξη στον προγραμματισμό και στην τεχνολογία των εξαρτημάτων των υπολογιστών, και κυρίως στον μοντελισμό στερεών στη δεκαετία του 1970, είχαν ως αποτέλεσμα οι CAD εφαρμογές να γίνουν πιο εύστροφες στις σχεδιαστικές λειτουργίες τους. Καθώς η τεχνολογία του γεωμετρικού μοντελισμού αναπτύχθηκε από τον απλό σχεδιασμό δύο διαστάσεων, στον τρισδιάστατο σχεδιασμό σύρματος (3D Wire-frame), στις τρισδιάστατες επιφάνειες και τώρα στον τρισδιάστατο μοντελισμό στερεού, έτσι αναπτύχθηκαν και τα συστήματα CAD. Η ραγδαία ανάπτυξη των συστημάτων CAD από τον απλό σχεδιασμό δύο διαστάσεων στον πολύπλοκο μοντελισμό τρισδιάστατων στερεών έχει ως αποτέλεσμα αυτά τα συστήματα να χρησιμοποιούνται ευρέως σήμερα σε όλες τις δραστηριότητες της μηχανικής, από τον σχεδιασμό της σύλληψης μιας ιδέας και στην λεπτομερή μηχανική μελέτη, καθώς επίσης και στην δυναμική ανάλυση των εξαρτημάτων και στον σχεδιασμό των συγκροτημάτων.

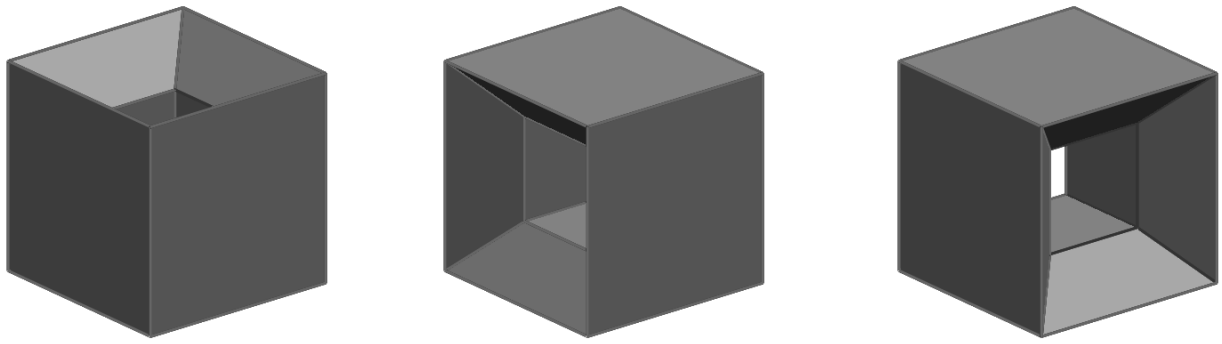
Η άφιξη του μοντελισμού τρισδιάστατων στερεών σήμανε και την αρχή μιας νέας εποχής το 1970 στα συστήματα CAD. Ο μοντελισμός στερεών δημιουργεί σαφή και ολοκληρωμένο προσδιορισμό της γεωμετρικής αναπαράστασης των αντικειμένων σε αντίθεση με την αναπαράσταση των μοντέλων σύρματος (Wire-frame models). Το μοντέλα σύρματος είναι ασαφή με την έννοια ότι για ένα και μοναδικό τέτοιο μοντέλο μπορεί να είναι δυνατές διάφορες ερμηνείες. Ένα πολύ καλό παράδειγμα για το συγκεκριμένο θέμα αποτελεί η «Εικόνα 2.1» (Hoffmann, 1989). Στο μοντέλο του παραδείγματος υπάρχει μια τετραγωνική οπή στο κέντρο του εξαρτήματος αλλά δεν μπορούμε με



Εικόνα 2.1 "Παράδειγμα Wire-frame Model"

σαφήνεια να προσδιορίσουμε τον προσανατολισμό αυτής της οπής με το μοντέλο σύρματος. Υπάρχουν τρεις διαφορετικές περιπτώσεις για τον προσανατολισμό της οπής, όπως φαίνεται και στην «Εικόνα2.2».

Επίσης, οπτικά είναι αδύνατο να προσδιορίσεις ογκομετρικές πληροφορίες του μοντέλου από τα μοντέλα σύρματος ή τα μοντέλα επιφανειών, σε αντίθεση με τα στερεά μοντέλα όπου αυτό δεν



Εικόνα 2.2 "Παράδειγμα Απεικόνισης Στερεού Μοντέλου"

αποτελεί πρόβλημα. Τα στερεά μοντέλα έχουν σαφής ερμηνείες και περιέχουν ολοκληρωμένες πληροφορίες, ως εκ τούτου χρησιμοποιούνται για την δημιουργία μηχανολογικών σχεδίων, καθώς

επίσης σε αυτά τα μοντέλα μπορούν να γίνουν μηχανολογικές αναλύσεις. Υπάρχουν αρκετές διαφορετικές προσεγγίσεις για τον μοντελισμό στερεού. Οι δύο πιο κοινές από αυτές είναι η κατασκευαστική στερεάς γεωμετρίας (constructive solid geometry «CSG») και η αναπαράσταση οριοθέτησης (boundary representation «B-rep»).

Ο μοντελισμός στερεού έχει γίνει ένα ώριμο και διάσημο εργαλείο στον σχεδιασμό μηχανολογικών εξαρτημάτων καθώς επίσης και σε άλλους τομείς. Η ανάπτυξη των πυρήνων του στερεού μοντελισμού οριοθέτησης (B-rep), όπως είναι το Parasolid και το ACIS, στα τέλη της δεκαετίας του 1980 έχει παίξει έναν σημαντικό ρόλο στην ανάπτυξη των μοντέρνων συστημάτων CAD. Στις μέρες μας, τα συστήματα CAD δεν περιορίζονται μόνο στον σχεδιασμό και την αναπαράσταση, αλλά επιχειρούν να εισχωρήσουν και σε πιο «νοητικές» περιοχές της πραγματογνωμοσύνης των σχεδιαστών, όπως είναι οι μηχανολογικές αναλύσεις και οι προσομοιώσεις συγκροτημάτων.

2.2 Ανάπτυξη της Τεχνολογίας CAM

Η τεχνολογία CAM πυροδοτήθηκε από την εφεύρεση των εργαλειομηχανών NC. Οι εργαλειομηχανές NC αναπτύχθηκαν με σκοπό να κατασκευάζουν περίπλοκα σχήματα με ακρίβεια και με επαναλαμβανόμενο τρόπο. Οι μηχανές αυτές λοιπόν κατευθύνονται από προγράμματα, κομμάτια κώδικα, τα οποία ακολουθούν το βιομηχανικό πρότυπο δεδομένων RS274D, διεθνώς γνωστό και ως ISO 6983. Το πρότυπο αυτό ορίζει κάποιους M και G κώδικες οι οποίοι προσδιορίζουν μια σειρά κινήσεων του κοπτικού εργαλείου καθώς επίσης την φορά περιστροφής, την ταχύτητα κοπής και διάφορες βοηθητικές λειτουργίες της μηχανής όπως είναι η ροή του ψυκτικού υγρού κοπής (Lee, 1999). Τα προγράμματα NC είναι μεγάλα σε πλήθος χαρακτήρων και πρέπει να προσδιορίζουν κάθε μεμονωμένη κίνηση του κοπτικού εργαλείου. Επίσης, αυτά τα προγράμματα είναι δύσκολο να δημιουργηθούν και να επεξεργαστούν χειροκίνητα. Στην «Εικόνα 2.3» δίνεται ένα παράδειγμα ενός προγράμματος NC για την δημιουργία μιας οπής. Απλά προγράμματα NC, όπως οι διεργασίες από

```
N0010 G40 G17 G90 G70
N0020 G91 G28 Z0.0
N0030 T01 M06
N0040 G00 G90 X4.2445 Y-9.8098 S500 M03
N0050 G43 Z5.9207 H00 M08
N0060 G81 Z5.7644 R5.9207 F10.
N0070 G80
N0080 M09
N0090 G00 Z8.
N0100 X3.6 Y-9.2
N0110 M02
```

Εικόνα 2.3 "Παράδειγμα G Κώδικα"

σημείο σε σημείο, μπορούν να δημιουργηθούν χειροκίνητα, συνήθως με την βοήθεια ενός υπολογιστή πράξεων. Για πιο περίπλοκα προγράμματα, ωστόσο, η χειροκίνητη δημιουργία προγραμμάτων NC

από μηχανολογικά σχέδια είναι μια πολύ χρονοβόρα διαδικασία και το ανθρώπινο λάθος μπορεί να γίνει πολύ εύκολα. Σε αυτό το σημείο έρχονται τα προγράμματα CAM τα οποία αναπτύχθηκαν για να χρησιμοποιούνται οι υπολογιστές με σκοπό να προετοιμάζουν και να παράγουν προγράμματα για τις εργαλειομηχανές NC.

Η πρώτη δημιουργία ενός CAM συστήματος προέκυψε όταν στην δεκαετία του 1950 αναπτύχθηκε στο Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT) το εργαλείο Automatically Programmed Tool (APT) το οποίο χρησιμοποιήθηκε για να βοηθήσει στον έλεγχο των μηχανών NC. Το APT είναι μια διεθνής γλώσσα προγραμματισμού για τις μηχανές NC και έχει χρησιμοποιηθεί ευρέως στον τομέα της βιομηχανίας (Tien-Chien Chang, 1998). Παρέχει έναν κατάλληλο τρόπο προσδιορισμού των γεωμετρικών στοιχείων και παραγωγής σημείων κοπής για προγράμματα NC μέσω ηλεκτρονικών υπολογιστών. Αρχικά, η γλώσσα APT μπορούσε να χειριστεί μόνο σχετικά εύκολες γεωμετρίες, όπως είναι σημεία, γραμμές, κύκλοι, επίπεδα, τετραγωνικές επιφάνειες κ.α. Στην συνέχεια, έγινε εφικτή και η αντιμετώπιση πιο περίπλοκων γεωμετριών. Με την βοήθεια της APT, τα πρώτα συστήματα CAM μπορούσαν να δημιουργήσουν ένα σχέδιο για το κομμάτι προς επεξεργασία και να μετατρέψουν το σχέδιο σε πρόγραμμα NC έτσι ώστε οι εργαλειομηχανές NC στην συνέχεια να μπορέσουν να κατασκευάσουν το κομμάτι από τα κομμάτια του κώδικα που παράχθηκε. Η γλώσσα APT αναπτύχθηκε πριν την δημιουργία των γραφικών διεπαφών του χρήστη, οπότε βασίζεται σε κείμενο για τον προσδιορισμό της γεωμετρίας και των τροχιών του κοπτικού εργαλείου τα οποία χρειάζονται για την επεξεργασία ενός κομματιού.

Παρόλο που η APT έχει αρκετά πλεονεκτήματα σε σχέση με την χειροκίνητη προσέγγιση για την δημιουργία NC προγραμμάτων, η χρήση της περιλαμβάνει τον προσδιορισμό εντολών για ολοκληρωμένες γεωμετρίες και θέσεις εργαλείου, γεγονός το οποίο θέτει ένα μεγάλο ποσοστό για πιθανά λάθη κατά την όλη διαδικασία. Για την αντιμετώπιση αυτού του προβλήματος, την δεκαετία του 1980, δημιουργήθηκαν και έγιναν γνωστά τα συστήματα CAM που βασίζονται στα γραφικά. Αυτά τα συστήματα μπορούν να περιγράψουν την γεωμετρία σε μορφή σημείων, γραμμών, τόξων και ούτω καθεξής, σε αντίθεση με την μετάφραση κειμένου προσανατολισμένης σημειογραφίας που γινόταν πριν (Tien-Chien Chang, 1998). Έτσι, ο χρήστης μπορεί πιο γρήγορα να ορίσει την γεωμετρία καθώς επίσης μπορεί να χρησιμοποιήσει αναλυτικές γραφικές απεικονίσεις για να ορίσει, να επαληθεύσει και να επεξεργαστεί με γρήγορο τρόπο τις πραγματικές κινήσεις του κοπτικού εργαλείου. Οι γραφικές απεικονίσεις δίνουν επίσης την δυνατότητα στο σύστημα να απεικονίσουν στην οθόνη το ακριβές αποτέλεσμα της πορείας του κοπτικού εργαλείου, δίνοντας την ευκαιρία στον χρήστη να αξιολογήσει τις εκάστοτε κατεργασίες και να αποφύγει τυχόν προβλήματα σε δουλειές οι οποίες γίνονται για πρώτη φορά και είναι στο στάδιο πρωτοτυποποίησης.

Παρά το γεγονός ότι τα συστήματα CAM έχουν ξεκινήσει ανεξάρτητα από τα συστήματα CAD, επηρεάζονται πάρα πολύ στην μετέπειτα ανάπτυξή τους από τα συστήματα CAD. Όταν πρωτοεμφανίστηκαν τα συστήματα CAM που είναι βασισμένα στα γραφικά είχαν σχεδιαστεί να

επαναχρησιμοποιούν τα μοντέλα της γεωμετρίας του κομματιού που είχε παραχθεί από ένα σύστημα CAD. Τα γεωμετρικά μοντέλα που χρησιμοποιούνταν για την παραγωγή τροχιών των εργαλείων κοπής, και κατ επέκταση προγραμμάτων NC, αναπτύχθηκαν από δισδιάστατα σχέδια, και τρισδιάστατα μοντέλα σύρματος, και μοντέλα επιφανειών σε στερεά μοντέλα, παράλληλα με την ανάπτυξη των CAD συστημάτων. Η ουσία σε όλο αυτό είναι ότι εάν γίνει προσπάθεια παραγωγής τροχιάς εργαλείου κοπής από ένα τρισδιάστατο μοντέλο σύρματος δεν υπάρχει τρόπος να ελεγχθεί η ύπαρξη τυχόν παρεμβολών μεταξύ του εργαλείου κοπής, του κομματιού και των εξαρτημάτων στήριξης. Όταν όμως χρησιμοποιείται ένα στερεό μοντέλο, μπορεί να αναπαραχθεί ολόκληρο το περιβάλλον εργασίας, καθώς επίσης το αρχικό κομμάτι προς επεξεργασία, ο όγκος του τελικού του κομματιού, τα εργαλεία και τα εξαρτήματα συγκράτησης, με αποτέλεσμα την δημιουργία μιας τροχιάς εργαλείου η οποία δεν θα έχει καμία παρεμβολή με όλα τα προηγούμενα. Επίσης, υπάρχει η δυνατότητα ενσωμάτωσης κανόνων γεωμετρικών συλλογισμών και γνώσης μεταλλοτεχνίας για την επιλογή των ταχυτήτων κοπής, των εργαλείων και της σειράς των κατεργασιών.

Με την συνεχή ανάγκη για βελτίωση στην ευκολία χρήσης αυτών των συστημάτων και την αύξηση της παραγωγικότητάς τους, ενσωματώνονται ολοένα και καλύτεροι αυτοματισμοί σε όλους τους τομείς των συστημάτων CAM, από την διεπαφή του χρήστη μέχρι και τους Post Processors. Η μέθοδος με την οποία είναι φτιαγμένο ένα CAM σύστημα να παράγει NC Code, του επιτρέπει να δημιουργεί πιο σταθερά προγράμματα και σε πολύ μικρό χρόνο. Με τις τελευταίες εξελίξεις στον χώρο έχει δημιουργηθεί η πρόκληση ανάπτυξης ενός CNC ανοιχτής αρχιτεκτονικής. Ο χειρισμός ανοιχτής αρχιτεκτονικής σημαίνει μια κοινή αρχιτεκτονική των μερών των συστημάτων και των διεπαφών, με αποτέλεσμα την αβίαστη πρόσβαση σε όλα τα δεδομένα μέσα στον χειρισμό, ένα φιλικό περιβάλλον διεπαφής για τον χρήστη, και μια βελτιωμένη επικοινωνία μεταξύ της εργαλειομηχανής.

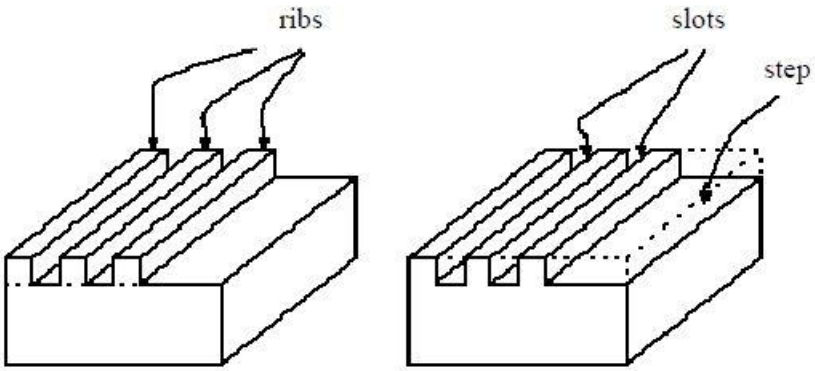
2.3 Σχεδιασμός Βάσει Μορφολογικών Χαρακτηριστικών (Feature-Based Modeling)

Είναι ευρέως αποδεκτό ότι η τεχνολογία των μορφολογικών χαρακτηριστικών παίζει έναν πολύ σημαντικό ρόλο στην επίτευξη της σύνδεσης των CAD/CAM συστημάτων. Οι μεθοδολογίες που βασίζονται στα μορφολογικά χαρακτηριστικά χωρίζονται σε δύο ομάδες, στην μέθοδο αναγνώρισης των μορφολογικών χαρακτηριστικών και στον σχεδιασμό με βάση τα μορφολογικά χαρακτηριστικά. Η μέθοδος της αναγνώρισης των μορφολογικών χαρακτηριστικών αναλύει την τοπολογία και την γεωμετρία ενός κομματιού και τα ταυτίζει με τα πιο κατάλληλα χαρακτηριστικά από μία ομάδα βασικών και προκαθορισμένων μορφολογικών χαρακτηριστικών. Με αυτό τον τρόπο, ένα μοντέλο το οποίο αποτελείται από χαμηλού επιπέδου στοιχεία (π.χ. ευθείες, σημεία, επιφάνειες) μετατρέπεται σε ένα μοντέλο από υψηλού επιπέδου στοιχεία (π.χ. οπές, ποκέτες, αύλακες) Έχουν δημοσιευτεί πολλές

έρευνες στον τομέα της αναγνώρισης μορφολογικών χαρακτηριστικών και έχουν υιοθετηθεί πολλές μέθοδοι. Το κύριο πλεονέκτημα όλων αυτών των μεθόδων είναι ότι δεν βάζουν κανέναν περιορισμό στους σχεδιαστές και ότι είναι ανεξάρτητες από τα συστήματα CAD και τους τύπους των αρχείων που αυτά παράγουν.

Ένα μορφολογικό χαρακτηριστικό είναι ένα σύνολο τοπολογικών και γεωμετρικών στοιχείων τα οποία έχουν ενωθεί μεταξύ τους με σκοπό να δίνουν χρήσιμες πληροφορίες στους μηχανικούς σχετικά με το προϊόν. Οι χρήστες μεταξύ τους έχουν διαφορετική άποψη σχετικά με το ποιες πληροφορίες είναι σημαντικές για ένα δοσμένο σχήμα βάση κάποιων παραγόντων, όπως είναι ο σχεδιασμός, η ανάλυση, η συνδεσμολογία, και η κατασκευή (Shah, 1995).

Ένα μοντέλο ενός προϊόντος μπορεί να δημιουργηθεί χρησιμοποιώντας ένα σετ από μορφολογικά χαρακτηριστικά, η μέθοδος γνωστή και ως feature-based design. Ένα μορφολογικό χαρακτηριστικό είναι ένα σχήμα το οποίο έχει σημασία για τον σχεδιαστή μηχανικό (Salomons, 1995). Ο σχεδιασμός με τα ήδη υπάρχοντα μορφολογικά χαρακτηριστικά μειώνει σημαντικά τον όγκο της δουλειάς του σχεδιαστή μηχανικού. Είναι πολύ δύσκολο να παρέχεται ένα σετ χαρακτηριστικών για κάθε πιθανή



Εικόνα 2.4 "Παράδειγμα Μορφολογικών Χαρακτηριστικών"

περίπτωση και θα είχε και σαν αποτέλεσμα την ύπαρξη μίας τεράστιας και δυσκίνητης βιβλιοθήκης. Ένα πλεονέκτημα όμως του σχεδιασμού βάση μορφολογικών χαρακτηριστικών είναι ότι τα ήδη υπάρχοντα χαρακτηριστικά μπορούν να επεξεργαστούν και να επαναχρησιμοποιηθούν με ευκολία. Ο σχεδιαστής μπορεί πολύ απλά να προσθέσει, να διαγράψει, ή να επεξεργαστεί τα χαρακτηριστικά ενός μοντέλου και να δημιουργήσει ένα νέο μοντέλο ουσιαστικά. Η τροποποίηση των χαρακτηριστικών είναι πιο ομαλή διαδικασία συγκριτικά με την άμεση αλλαγή της υποκείμενης γεωμετρίας (Chen, 1995).

Από την πλευρά του σχεδιασμού, τα μορφολογικά χαρακτηριστικά μπορεί να είναι είτε προσθήκη στερεού είτε αφαίρεση στερεού σε ένα στερεό μοντέλο. Τα σχεδιαστικά χαρακτηριστικά δεν μπορούν να χρησιμοποιηθούν πάντα απ’ ευθείας στη δημιουργία των φασεολογίων και στην κατασκευή των

προϊόντων. Η «Εικόνα 2.4» παρουσιάζει ένα συχνό παράδειγμα σχετικά με την διαφορά ανάμεσα σε σχεδιαστικά χαρακτηριστικά και σε χαρακτηριστικά μηχανικών κατεργασιών (machining features). Το προϊόν μπορεί να σχεδιαστεί προσθέτοντας νεύρα (ribs) στην ορθογωνική βάση του προϊόντος. Ωστόσο, τα χαρακτηριστικά μηχανικών κατεργασιών για το συγκεκριμένο προϊόν είναι τα λεγόμενα slots, καθώς επίσης και ένα step που ουσιαστικά αντιπροσωπεύει το υλικό που πρέπει να αφαιρεθεί για να δημιουργηθεί το προϊόν.

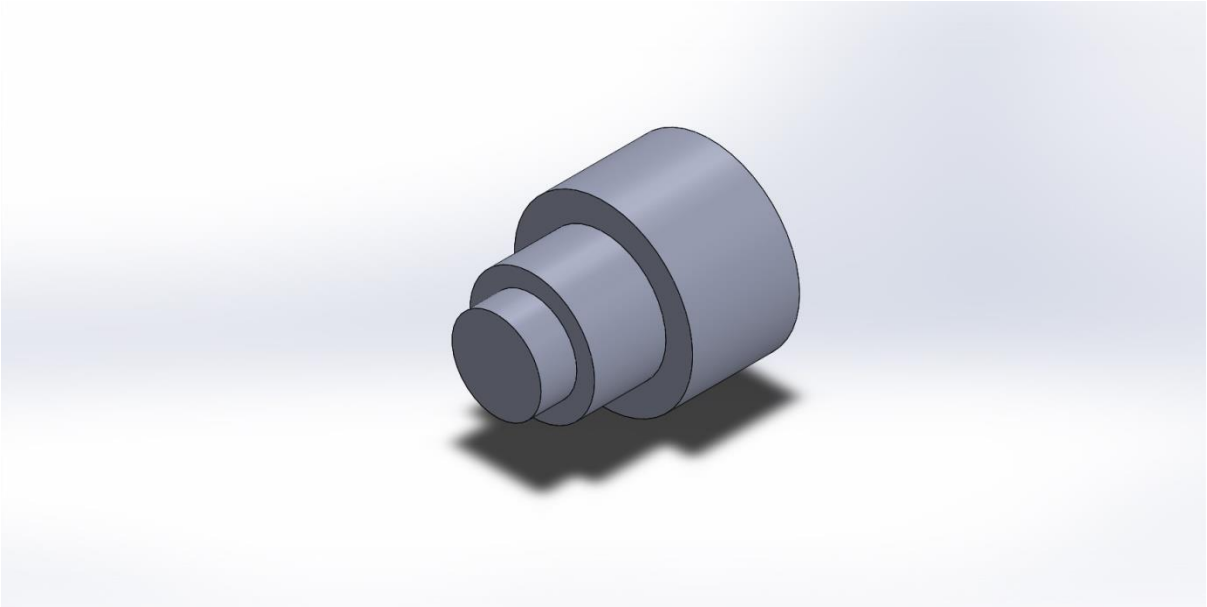
Ένας τρόπος αντιμετώπισης των διαφορετικών όψεων σχετικά με τα σχεδιαστικά χαρακτηριστικά και τα χαρακτηριστικά μηχανικών κατεργασιών είναι η χρησιμοποίηση των μηχανικών κατεργασιών και στον σχεδιαστικό και στον κατασκευαστικό τομέα.

Ο σχεδιασμός βάση μορφολογικών χαρακτηριστικών χτίζει ένα μοντέλο χρησιμοποιώντας προκαθορισμένα μορφολογικά χαρακτηριστικά τα οποία υπάρχουν σε μία βιβλιοθήκη. Η γεωμετρία αυτών των χαρακτηριστικών είναι καθορισμένη αλλά οι διαστάσεις τους έχουν οριστεί σαν μεταβλητές, με σκοπό να πάρουν τιμή από τον σχεδιαστή όταν το κάθε μορφολογικό χαρακτηριστικό κληθεί να χρησιμοποιηθεί στη διαδικασία του μοντελισμού. Δεδομένου ότι τα μορφολογικά χαρακτηριστικά συνδέονται στενά με την εκάστοτε εφαρμογή σχεδίασης, όλη η βασική γεωμετρία και οι τεχνικές πληροφορίες μπορούν να συμπεριληφθούν στο μοντέλο. Υπάρχουν δύο βασικές μεθοδολογίες σχεδιασμού βάση μορφολογικών χαρακτηριστικών, «καταστροφή» με χρήση χαρακτηριστικών μηχανικών κατεργασιών και «σύνθεση» με χρήση σχεδιαστικών χαρακτηριστικών. Στην διαδικασία της πρώτης μεθόδου, ο χρήστης ξεκινάει με τον σχεδιασμό ενός αρχικού μοντέλου το οποίο αντιπροσωπεύει το υλικό που είναι προς επεξεργασία. Το σχεδιαστικό μοντέλο αναπτύσσεται αφαιρώντας μορφολογικά χαρακτηριστικά από το αρχικό μοντέλο, τα οποία αντιστοιχούν στο υλικό που πρέπει να αφαιρεθεί από το αρχικό κομμάτι με μηχανικές κατεργασίες. Στην δεύτερη μέθοδο, ένα σχεδιαστικό μοντέλο μπορεί να αναπτυχθεί είτε αφαιρώντας μορφολογικά χαρακτηριστικά, όπως και στην πρώτη μέθοδο, είτε ακόμη και προσθέτοντας μορφολογικά χαρακτηριστικά.

Κεφάλαιο 3: Προγραμματιστικό Μέρος

3.1 Γενική Περιγραφή του Αλγορίθμου

Ένα κομμάτι τόρνευσης αποτελείται ουσιαστικά από δύο προφίλ, ένα εξωτερικό προφίλ και ένα εσωτερικό. Η διαδικασία κατασκευής ενός κομματιού τόρνευσης ξεκινάει συνήθως από την εξωτερική κατεργασία, όπου ο τεχνίτης ξεκινάει την αφαίρεση υλικού με σκοπό αρχικά να φτάσει στην αμέσως μικρότερη διάμετρο από την αρχική που υπάρχει και σε ένα συγκεκριμένο μήκος. Έπειτα, προχωράει στην αφαίρεση υλικού προς την επόμενη μικρότερη διάμετρο και σε ένα μικρότερο μήκος από το προηγούμενο, και αυτό συνεχίζεται μέχρι να φτάσει στην μικρότερη διάμετρο που υπάρχει στο μηχανολογικό σχέδιο. Αντίστοιχα, στην εσωτερική κατεργασία ο τεχνίτης ξεκινάει την αφαίρεση υλικού από την μικρότερη διάμετρο που υπάρχει και για ένα συγκεκριμένο



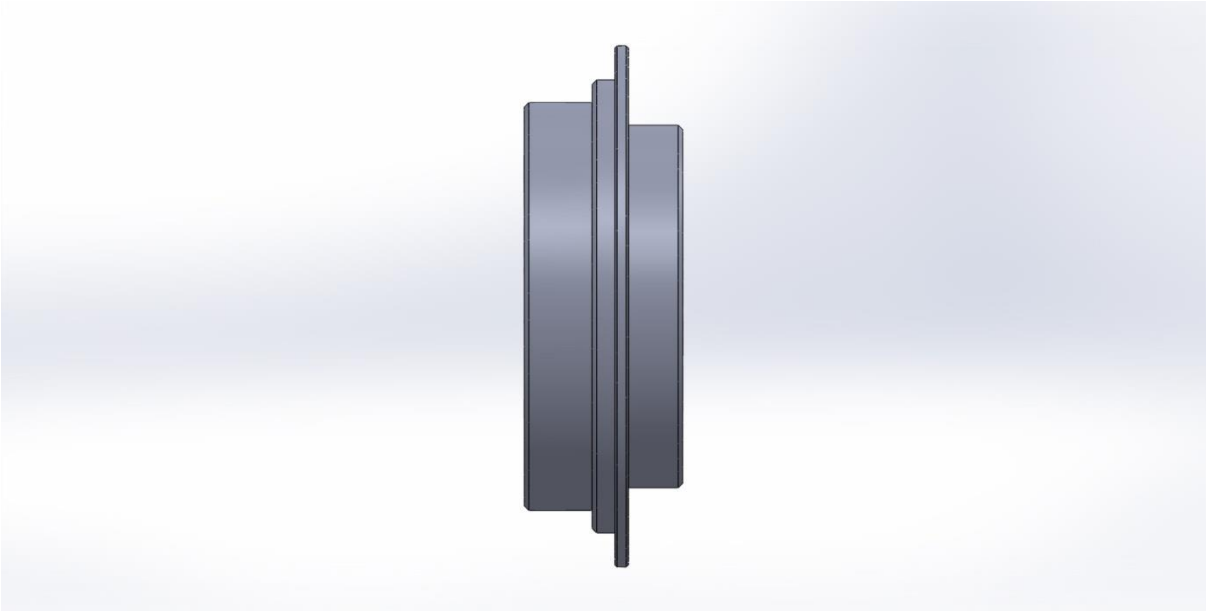
Εικόνα 3.1 "Αντικείμενο Τόρνευσης Μίας Φάσης"

μήκος. Η διαδικασία δηλαδή είναι ακριβώς αντίστροφη.

Μία εργαλειομηχανή CNC ακολουθεί αυτήν ακριβώς την διαδικασία γνωρίζοντας τις συντεταγμένες του κάθε προφίλ του αντικειμένου. Ο προγραμματιστής της μηχανής θα πρέπει να πληκτρολογήσει ακριβώς με συντεταγμένες του προφίλ της κάθε κατεργασίας καθώς επίσης και τον τρόπο με τον οποίο θα κινηθεί το εργαλείο από το ένα σημείο προς το επόμενο διότι το προφίλ αποτελείται και από καμπύλα τμήματα εκτός από ευθεία τμήματα.

Για να μπορέσει ο αλγόριθμος να εξάγει έναν σωστό G κώδικα θα πρέπει και ο σχεδιαστής του αντικειμένου στο σχεδιαστικό πρόγραμμα να ακολουθήσει την φιλοσοφία της κατεργασίας τόρνευσης στον σχεδιασμό του. Θα πρέπει ουσιαστικά να σχεδιάσει με ένα boss feature το αρχικό κομμάτι που θα παραλάβει η εργαλειομηχανή προς κατεργασία και στην συνέχεια να το διαμορφώσει με features αφαίρεσης υλικού (π.χ. Cut-Extrude), όπως ακριβώς θα έκανε και η εργαλειομηχανή. Εφόσον σχεδιαστούν τα δύο βασικά προφίλ που χρειάζονται για να προσδιοριστεί το αντικείμενο ο σχεδιαστής στη συνέχεια θα μπορούσε να επεξεργαστεί τις ακμές της κάθε αλλαγής διαμέτρου του κομματιού και να προσθέσει ένα Fillet Feature ή ένα Chamfer Feature όπου αυτό είναι απαραίτητο.

Μόλις ο σχεδιασμός του αντικειμένου έχει ολοκληρωθεί, ο σχεδιαστής είναι σε θέση να εκτελέσει τον αλγόριθμο. Σε αυτό το σημείο ο σχεδιαστής πρέπει να γνωρίζει ότι τα κομμάτια τόρνευσης συνήθως έχουν δύο φάσεις κατασκευής, ανάλογα από τα προφίλ από τα οποία αποτελούνται. Για παράδειγμα, στην ***Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.*** βλέπουμε ένα ντικείμενο το οποίο μπορεί να δημιουργηθεί με μία φάση κατεργασίας. Μπορεί ουσιαστικά να γίνει η



Εικόνα 3.2 "Αντικείμενο Τόρνευσης Δύο Φάσεων"

συγκράτησή του από την μεγαλύτερη διάμετρο που υπάρχει και η εργαλειομηχανή να επεξεργαστεί και να διαμορφώσει τις δύο μικρότερες διαμέτρους.

Στην *Εικόνα 3.2* φαίνεται ένα αντικείμενο το οποίο χρειάζεται δύο φάσεις κατεργασίας. Αυτό συμβαίνει διότι εκτός από τις δύο διαμέτρους που υπάρχουν στα αριστερά του αντικειμένου όπως το βλέπουμε υπάρχει και μία δεξιά της μεγαλύτερης διαμέτρου. Οπότε, για να κατασκευαστεί το συγκεκριμένο αντικείμενο θα πρέπει να κατεργαστούν πρώτα οι δύο διάμετροι μαζί, στην συνέχεια να γίνει η συγκράτηση του κομματιού σε μία από τις δύο διαμέτρους που δημιουργήθηκαν με σκοπό

να δημιουργηθεί και η τελευταία που έμεινε. Θα μπορούσε να γίνει και το αντίθετο, να δημιουργηθεί πρώτα αυτή που είναι μόνη της και στην συνέχεια οι άλλες δύο, η επιλογή είναι του κατασκευαστή ο οποίος επιλέγει ποιο είναι κατά τη γνώμη του το καλύτερο σενάριο.

Οπότε, σε αυτό το σημείο ο σχεδιαστής πριν εκτελέσει τον αλγόριθμο θα πρέπει να έχει επιλέξει από το Feature Tree τα Features που θέλει να κατεργαστούν στην πρώτη φάση κατεργασίας. Στην επιλογή του θα πρέπει να συμπεριλάβει, εκτός από τα Extrude-Cuts που αντιπροσωπεύουν τις διαμέτρους κατεργασίας, και τα Fillet Features ή Chamfer Features που υπάρχουν στις αντίστοιχες διαμέτρους. Εδώ πρέπει να σημειωθεί ότι ο σχεδιαστής πρέπει να προσέξει στην δημιουργία αυτών των Features καθώς δεν πρέπει να υπάρχουν π.χ. σε ένα Fillet Feature ακμές που υπάρχουν σε διαφορετικές φάσεις κατεργασίας. Πρέπει δηλαδή το κάθε Fillet ή Chamfer να έχει εφαρμοστεί σε ακμές που βρίσκονται στην ίδια φάση κατεργασίας. Επίσης, στην επιλογή των Features ο σχεδιαστής θα πρέπει να διαλέξει και το αρχικό Boss-Extrude ούτως ώστε ο αλγόριθμος να μπορέσει να διαβάσει και να αντιληφτεί τις διαστάσεις του αρχικού κομματιού με σκοπό να ξεκινήσει από το σωστό σημείο την κάθε κατεργασία.

Συνοψίζοντας λοιπόν του κανόνες σχεδίασης έχουμε τα παρακάτω:

- Δημιουργία ενός Boss-Extrude που θα αντιπροσωπεύει το αρχικό Stock του αντικειμένου
- Διαμόρφωση του κάθε προφίλ κατεργασίας με χρήση εντολών Cut-Extrude
- Κάθε εντολή Cut-Extrude θα πρέπει να έχει αρχική επιφάνεια εφαρμογής την επιφάνεια του προσώπου του αντικειμένου, και το μήκος του Cut-Extrude ουσιαστικά να δίνεται από το σημείο Z0
- Το κάθε Chamfer Feature ή Fillet Feature θα πρέπει να περιέχει ακμές που βρίσκονται στην ίδια φάση κατεργασίας

Όταν ο σχεδιαστής λοιπόν επιλέξει τα Features που θέλει και εκτελέσει τον αλγόριθμο θα δημιουργηθούν δύο αρχεία *.txt* στην θέση «C:», το ένα θα ονομάζεται *«G-Code Phase 1»* και θα περιλαμβάνει τον G κώδικα για την πρώτη φάση κατεργασίας, και το δεύτερο θα ονομάζεται *«G-Code Phase 2»* και θα περιλαμβάνει τον G κώδικα για την δεύτερη φάση κατεργασίας. Στο αρχείο της δεύτερης φάσης κατεργασίας θα περιλαμβάνονται τα Features που έχουν μείνει εκτός επιλογής πριν την εκτέλεση του αλγορίθμου. Σε περίπτωση που το κομμάτι προς κατεργασία δεν χρειάζεται δεύτερη φάση, οπότε ο σχεδιαστής ουσιαστικά έχει επιλέξει όλα τα Features από το Feature Tree, το αρχείο της δεύτερης φάσης κατεργασίας θα είναι κενό.

3.2 Ανάλυση Αλγορίθμου

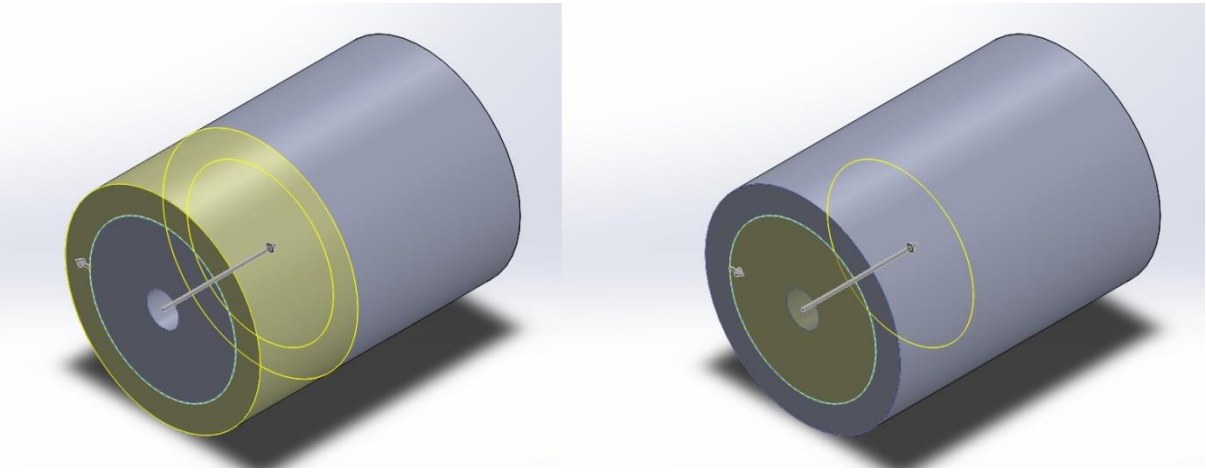
Στην παρούσα παράγραφο περιγράφεται αναλυτικά το πώς λειτουργεί ο αλγόριθμος. Πιο συγκεκριμένα, ο αλγόριθμος αποτελείται από έξι τομείς οι οποίοι είναι οι ακόλουθοι:

1. Καθορισμός του μεγέθους των πινάκων
2. Διαχωρισμός των Features
3. Ταξινόμηση των πινάκων βάση των διαμέτρων
4. Δημιουργία τοπολογικών πινάκων
5. Δημιουργία αρχείου "1st Phase G Code"
6. Δεύτερη Φάση

3.2.1 Καθορισμός του Μεγέθους των Πινάκων

Το πρώτο βήμα του αλγορίθμου είναι η σάρωση όλων των επιλεγμένων στοιχείων με σκοπό την αναγνώρισή τους και την μετέπειτα ταξινόμησή τους. Για την αναγνώριση των στοιχείων έχουν δημιουργηθεί τέσσερις πίνακες, ο "varExtFeatID" ο οποίος αντιστοιχεί σε Features τόρνευσης εξωτερικής κατεργασίας, ο "varIntFeatID" ο οποίος αντιστοιχεί σε Features τόρνευσης εσωτερικής κατεργασίας, ο "varFilletID" ο οποίος αντιστοιχεί σε Features όπου σε μια ακμή ή περισσότερες του αντικειμένου έχει εφαρμοστεί ένα Fillet και ο "varChamferID" ο οποίος αντιστοιχεί σε Features όπου σε μια ακμή ή περισσότερες του αντικειμένου έχει εφαρμοστεί ένα Chamfer. Έτσι λοιπόν υπάρχει μία επαναληπτική διαδικασία η οποία λειτουργεί ως εξής.

Για το πρώτο επιλεγμένο στοιχείο ελέγχεται αρχικά εάν ο τύπος του στοιχείου αυτού είναι Feature. Εάν δεν είναι τότε εμφανίζεται στον χρήστη ένα μήνυμα το οποίο γράφει πως τα επιλεγμένα στοιχεία πρέπει να είναι μόνο Features και όχι κάτι άλλο. Σε αυτήν την περίπτωση ο αλγόριθμος σταματάει την λειτουργία του και ο χρήστης πρέπει να επιλέξει ξανά τα Features, αυτή την φορά με τον σωστό τρόπο, και να τρέξει τον αλγόριθμο από την αρχή. Με αυτόν τον τρόπο εξασφαλίζεται πως ο χρήστης δεν θα επιλέξει κάποιο άλλο στοιχείο από το Feature Tree το οποίο θα δημιουργήσει πρόβλημα στον αλγόριθμο. Τέτοια στοιχεία μπορεί να είναι το Origin, κάποιο Plane ή ακόμη και το υλικό του Part.



Εικόνα 3.3 "Παράδειγμα Επιλογής Flip Side to Cut"

Μόλις λοιπόν το επιλεγμένο στοιχείο περάσει αυτόν τον έλεγχο ακολουθεί ο επόμενος ο οποίος είναι ο προσδιορισμός του είδους του Feature, εάν αυτό δηλαδή είναι ένα Cut-Extrude ή ένα Fillet ή ένα Chamfer.

Στο πρώτο σενάριο, αυτό του Cut-Extrude, ακολουθεί ένας ακόμη έλεγχος. Προσδιορίζεται η φορά κοπής του εκάστοτε Cut-Extrude με σκοπό την αντιστοίχιση του Feature σε εξωτερική κατεργασία ή εσωτερική, όπως φαίνεται και από την εικόνα . Οπότε, στην περίπτωση που ο προσανατολισμός του Cut-Extrude αντιστοιχεί σε εξωτερική κατεργασία ο μετρητής των στοιχείων του πίνακα "varExtFeatID" αυξάνεται κατά ένα και στην αντίθετη περίπτωση ο μετρητής των στοιχείων του πίνακα "varIntFeatID" αυξάνεται κατά ένα.

Εάν το επιλεγμένο Feature δεν είναι κάποιο Cut-Extrude τότε ελέγχεται εάν είναι Fillet. Σε περίπτωση που ο έλεγχος είναι ορθός, τότε ο μετρητής των στοιχείων του πίνακα "varFilletID" αυξάνεται κατά ένα, και σε διαφορετική περίπτωση ο μετρητής των στοιχείων του πίνακα "varChamferID" αυξάνεται κατά ένα. Μόλις ολοκληρωθούν οι παραπάνω έλεγχοι η επαναληπτική διαδικασία περνάει στο επόμενο επιλεγμένο στοιχείο και εκτελεί ξανά τους ίδιους ελέγχους μέχρι να ελεγχτούν όλα τα επιλεγμένα στοιχεία.

Μόλις τελειώσει αυτή η διαδικασία έπεται η αρχικοποίηση των τεσσάρων αυτών πινάκων καθώς τώρα είναι γνωστό το μέγεθος του καθενός. Επίσης, σε αυτό το σημείο, αρχικοποιείται και ένας ακόμη πίνακας, ο "varSelectedFeatID", του οποίου το μέγεθος είναι το άθροισμα όλως των επιλεγμένων Features. Αυτός ο πίνακας θα χρησιμοποιηθεί στην συνέχεια του αλγορίθμου ως βάση για την συνέχεια της δεύτερης φάσης κατεργασίας.

3.2.2 Διαχωρισμός των Features

Εφόσον λοιπόν έχει προσδιοριστεί το μέγεθος του κάθε πίνακα και ξέρουμε πλέον το πλήθος του κάθε είδους Feature που έχουμε επόμενο βήμα είναι η συμπλήρωση του κάθε πίνακα με τα ID του εκάστοτε Feature.

Η διαδικασία αυτή έχει την ίδια βάση λειτουργίας με τον έλεγχο που γινόταν στην *παράγραφο 3.2.1*. Ξεκινάει ο αλγόριθμος από το πρώτο επιλεγμένο στοιχείο. Η πρώτη ενέργεια που γίνεται είναι να καταχωρηθεί το ID του Feature στον πίνακα "varSelectedFeatID".

Στην συνέχεια ξεκινάνε οι ίδιοι έλεγχοι που υπάρχουν και προηγουμένως. Εάν το επιλεγμένο Feature είναι Cut-Extrude και ο προσανατολισμός του αντιστοιχεί σε εξωτερική κατεργασία, τότε το ID του καταχωρείται στην πρώτη θέση του πίνακα "varExtFeatID" και ενεργοποιείται για τον συγκεκριμένο πίνακα η επόμενη κενή θέση. Εάν το Cut-Extrude αντιστοιχεί σε εσωτερική κατεργασία, τότε το ID του καταχωρείται στην πρώτη θέση του πίνακα "varIntFeatID" και ενεργοποιείται για τον συγκεκριμένο πίνακα η επόμενη κενή θέση. Έπειτα ελέγχεται εάν το Feature είναι Fillet και το ID του

καταχωρείται στον αντίστοιχο πίνακα καθώς επίσης το ίδιο συμβαίνει και σε περίπτωση που το Feature είναι Chamfer.

Η ενεργοποίηση της επόμενης θέσης για τον κάθε πίνακα γίνεται με την χρήση ενός δείκτη θέσης για τον κάθε πίνακα οποίος ξεκινάει με την τιμή μηδέν και κάθε φορά που καταχωρείται κάποιο στοιχείο στον εκάστοτε πίνακα ο δείκτης αυτός αυξάνεται κατά ένα. Έτσι, την επόμενη φορά που θα χρειαστεί να γίνει κάποια καταχώριση στον κάθε πίνακα εξασφαλίζεται ότι θα μπει στην επόμενη θέση και όχι πάνω στην ίδια.

Σε αυτό το σημείο του αλγορίθμου υπάρχει και ένας επιπλέον έλεγχος στην επαναληπτική διαδικασία ο οποίος είχε παραληφθεί στην αρχική επαναληπτική διαδικασία. Ο χρήστης στην επιλογή των Features που θα κάνει για την πρώτη φάση κατεργασίας θα πρέπει να έχει επιλέξει και το Boss-Extrude το οποίο αντιστοιχεί στο αρχικό Stock του αντικειμένου. Μέχρι στιγμής αυτό το Feature δεν έχει αναφερθεί κάπου στον αλγόριθμο παρόλο το γεγονός πως έχει περάσει από την αρχική επαναληπτική διαδικασία. Εδώ λοιπόν, ελέγχεται αυτό το Feature και καθορίζει δύο πράγματα. Αρχικά, καταχωρείται σε μία μεταβλητή το ολικό μήκος του Stock. Στη συνέχεια, προσδιορίζεται εάν το Stock είναι συμπαγές υλικό ή διάτρητο. Σε περίπτωση που είναι συμπαγές, καταχωρείται σε μια άλλη μεταβλητή η διάμετρός του, και στην περίπτωση που είναι διάτρητο καταχωρείται σε μία μεταβλητή η εξωτερική του διάμετρος και σε μια άλλη η εσωτερική του διάμετρος. Έτσι λοιπόν, έχει προσδιοριστεί πλήρως το αρχικό Stock του αντικειμένου και κατ επέκταση ο αλγόριθμος ξέρει από πού θα ξεκινήσει τον κάθε κύκλο κατεργασίας, είτε σε εξωτερική είτε σε εσωτερική κατεργασία εάν αυτή υπάρχει.

3.2.3 Ταξινόμηση των Πινάκων Βάσει των Διαμέτρων

Επόμενο βήμα στον αλγόριθμο είναι η ταξινόμηση των πινάκων βάση των διαμέτρων των Features. Σκοπός αυτού του μέρους του αλγορίθμου είναι να μπουν τα Features σε μία σωστή σειρά για να δημιουργηθεί στην πορεία το προφίλ των συντεταγμένων που χρειάζεται να περιγράψει ο G κώδικας. Πρέπει ουσιαστικά τα Cut-Extrude της εξωτερικής κατεργασίας να μπουν σε μία αύξουσα σειρά με βάση την τελική τους διάμετρο, και τα Cut-Extrude της εσωτερικής κατεργασίας να μπουν σε μία φθίνουσα σειρά με βάση την τελική τους διάμετρο. Επομένως, αυτό το μέρος του αλγορίθμου εφαρμόζεται μόνο στους δύο πρώτους πίνακες που έχουν δημιουργηθεί και όχι στους “varFilletID” και “varChamferID” καθώς οι διάμετροι αυτών των Features δεν έχει νόημα να ταξινομηθούν και επίσης στο κάθε Feature μπορεί να υπάρχουν παραπάνω από μία διάμετρος.

Έτσι λοιπόν, υπάρχει μία επαναληπτική διαδικασία στην οποία ξεκινάει και ελέγχεται το κάθε στοιχείο του πίνακα “varExtFeatID”. Από το κάθε ID ο αλγόριθμος μπορεί να εξάγει πληροφορίες για το εκάστοτε Feature, στην συγκεκριμένη περίπτωση εξάγει την διάμετρο του Cut-Extrude και μπορεί με αυτό τον τρόπο να ταξινομήσει τον πίνακα. Αντίστοιχα, υπάρχει μια επαναληπτική διαδικασία για την ταξινόμηση του πίνακα “varIntFeatID”.

3.2.4 Δημιουργία Τοπολογικών Πινάκων

Όπως έχει προαναφερθεί, για την δημιουργία του προφίλ της κάθε κατεργασίας πρέπει να υπάρχουν οι συντεταγμένες του κάθε σημείου, και απαραίτητα με την σωστή σειρά ανάλογα με το εάν είναι εξωτερική ή εσωτερική κατεργασία. Για τον σωστό προσδιορισμό αυτών των συντεταγμένων δημιουργήθηκαν δύο τοπολογικοί πίνακες, ο "varExtTop" που αντιστοιχεί στις συντεταγμένες των σημείων της εξωτερικής κατεργασίας και ο "varIntTop" που αντιστοιχεί στις συντεταγμένες των σημείων της εσωτερικής κατεργασίας.

Κάθε Cut-Extrude από τα οποία αποτελούνται τα προφίλ κατεργασίας έχει μία συγκεκριμένη διάμετρο και ένα συγκεκριμένο μήκος, το οποίο για τις κατεργασίες τόρνου προσδιορίζεται πάντα από το πρόσωπο του αντικειμένου που θεωρείται το σημείο 0 κατά τον Z άξονα. Αυτές οι δύο διαστάσεις είναι οι βασικές που χρειάζονται για να προσδιοριστεί ένα τέτοιο στοιχείο. Επίσης, υπάρχουν πολλές περιπτώσεις όπου κάποιο Feature στην αρχική του ακμή ή στην τελική του έχει υποστεί κάποια επεξεργασία, όπως για παράδειγμα ένα Fillet ή ένα Chamfer. Σε αυτές τις περιπτώσεις, για τον σωστό προσδιορισμό των συντεταγμένων του προφίλ θα πρέπει αυτά τα στοιχεία να είναι γνωστά. Έτσι λοιπόν ο κάθε τοπολογικός πίνακας αποτελείται τόσες γραμμές όσες είναι και τα Cut-Extrude που αποτελούν το κάθε προφίλ, και από οχτώ στήλες οι οποίες είναι οι εξής:

1. Διάσταση Διαμέτρου
2. Μήκος Στοιχείου
3. Ακτίνα Chamfer στην πρώτη ακμή
4. Μήκος Chamfer στην πρώτη ακμή
5. Ακτίνα Fillet στην πρώτη ακμή
6. Ακτίνα Chamfer στην τελική ακμή
7. Μήκος Chamfer στην τελική ακμή
8. Ακτίνα Fillet στην τελική ακμή

Σύμφωνα λοιπόν με αυτά έπεται η διαδικασία συμπλήρωσης των στοιχείων των δύο αυτών πινάκων η οποία ξεκινάει όπως πάντα με την αρχικοποίησή τους, όπου το κάθε στοιχείο του πίνακα παίρνει την τιμή 0. Στην συνέχεια ξεκινάει η συμπλήρωση των στοιχείων για τον "varExtTop" με μία επαναληπτική διαδικασία η οποία ξεκινάει να ελέγχει με την σειρά τα στοιχεία του πίνακα "varExtFeatID".

Ξεκινώντας από το πρώτο στοιχείο αυτού του πίνακα ο αλγόριθμος καταχωρεί στις δύο πρώτες στήλες της πρώτης γραμμής του πίνακα "varExtTop" τα στοιχεία σχετικά με την διάμετρο του Feature και με το μήκος του. Έπειτα πρέπει να διαπιστωθεί εάν στις ακμές αυτού του Feature υπάρχει κάποιο Chamfer ή κάποιο Fillet. Για να γίνει αυτό, ο αλγόριθμος γνωρίζοντας την διάμετρο του Cut-Extrude στο οποίο βρίσκεται ελέγχει τις διαμέτρους στις οποίες έχουν εφαρμοστεί τα Chamfers σε πρώτη φάση και τα Fillets σε δεύτερη φάση. Μόλις βρεθούν δύο ίδιοι διάμετροι ο αλγόριθμος ξέρει

ότι το συγκεκριμένο Cut-Extrude έχει υποστεί ένα Chamfer ή ένα Fillet, ανάλογα με το τι βρήκε. Οπότε, αυτό που χρειάζεται ακόμη για να προσδιοριστεί πλήρως η γραμμή του "varExtTop" είναι σε ποια ακμή από τις δύο του Cut-Extrude έχει εφαρμοστεί η επεξεργασία. Αυτό γίνεται συγκρίνοντας το σημείο του κέντρου της ακμής με τις συντεταγμένες του Cut-Extrude. Πιο αναλυτικά, όταν αναφερόμαστε στον όρο «ακμή» ουσιαστικά αναφερόμαστε σε έναν κύκλο, διότι στο προφίλ ενός κομματιού τόρνευσης κάθε οριζόντια γραμμή αντιστοιχεί σε έναν κύκλο. Έτσι λοιπόν, για να εφαρμοστεί ένα Chamfer ή ένα Fillet ο χρήστης καλείται να επιλέξει έναν κύκλο στον οποίο θα γίνει η συγκεκριμένη κατεργασία. Οπότε, για να βρει ο αλγόριθμος σε ποια από τις δύο ακμές του Cut-Extrude έχει εφαρμοστεί η επεξεργασία αρκεί να συγκρίνει το σημείο Z του κέντρου του κύκλου στο οποίο εφαρμόστηκε η επεξεργασία με τις συντεταγμένες του Cut-Extrude.

Για παράδειγμα, στην πρώτη διάμετρο της εξωτερικής κατεργασίας, εάν υπάρχει ένα Fillet στην πρώτη ακμή του Cut-Extrude το σημείο Z του Fillet θα είναι το 0, και αντίστοιχα εάν υπάρχει ένα Chamfer στην τελική ακμή του Cut-Extrude το σημείο Z του Chamfer θα ίσο με το μήκος του Cut-Extrude. Στον έλεγχο των επόμενων Cut-Extrude πέραν του πρώτου, εάν έχει διαπιστωθεί ότι υπάρχει ένα Fillet για παράδειγμα, για να βρεθεί σε ποια ακμή έχει εφαρμοστεί αρκεί να γίνει σύγκριση του σημείου Z του Fillet με το μήκος του Cut-Extrude. Σε περίπτωση που αυτοί οι δύο αριθμοί δεν είναι ίδιοι συνεπάγεται ότι το Fillet έχει εφαρμοστεί στην αρχική ακμή του Cut-Extrude της οποίας το σημείο Z είναι ίδιο με το μήκος του προηγούμενο Cut-Extrude.

Η ίδια διαδικασία ακολουθείται για τον πίνακα "varIntTop". Με αυτόν τον τρόπο λοιπόν για κάθε Cut-Extrude γίνεται γνωστή η διάμετρός του, το μήκος του, εάν υπάρχει κάποιο Chamfer στην πρώτη ακμή του και ποιες είναι οι διαστάσεις του, εάν υπάρχει κάποιο Fillet στην πρώτη ακμή του και ποιο είναι το Radius του, εάν υπάρχει κάποιο Chamfer στην τελική ακμή του και ποιες είναι οι διαστάσεις του και εάν υπάρχει κάποιο Fillet στην τελική ακμή του και ποιο είναι το Radius του.

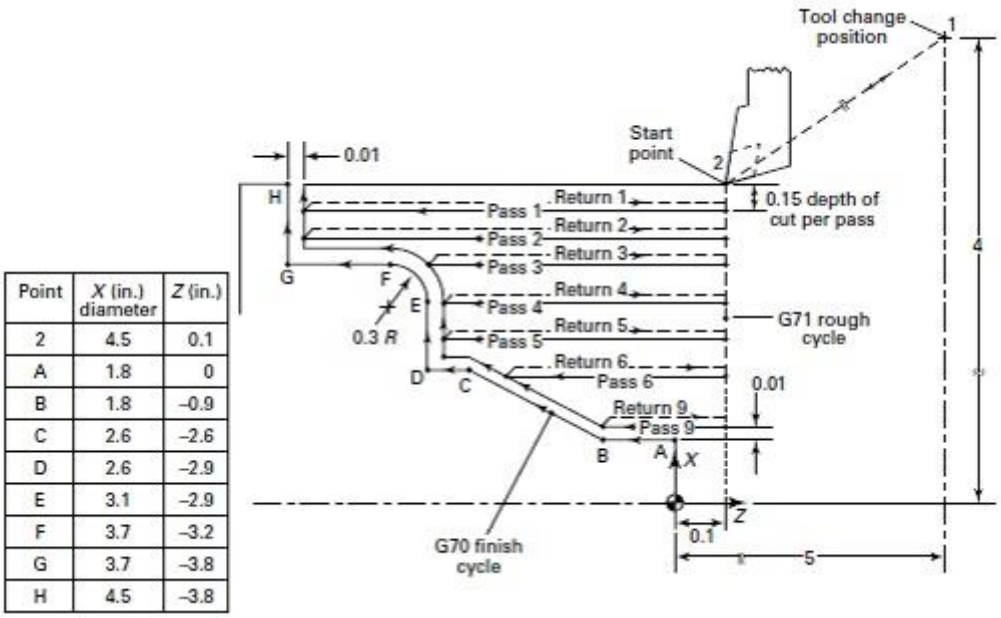
3.2.5 Δημιουργία Αρχείου "1st Phase G Code"

Αφού λοιπόν έχει ολοκληρωθεί η συμπλήρωση των δύο τοπολογικών πινάκων ο αλγόριθμος έχει στην μνήμη του όλες τις συντεταγμένες που χρειάζεται για να δημιουργήσει τον G κώδικα. Πρώτο βήμα σε αυτή την φάση του αλγορίθμου, ξεκινώντας από την εξωτερική κατεργασία του αντικειμένου, είναι η δημιουργία του αρχείου "G-Code Phase 1.txt". Αφού δημιουργηθεί το αρχείο ο αλγόριθμος μπορεί να γράψει μέσα στο αυτό το αρχείο. Τα πρώτα πράγματα που πρέπει να τυπωθούν είναι οι αρχικές διαδικασίες που λαμβάνουν χώρα κάθε φορά που ξεκινάει να εκτελείται ένας G κώδικας σε μία εργαλειομηχανή. Αυτές οι διαδικασίες είναι η μετακίνηση της μηχανής σε ένα σημείο μακριά από το κομμάτι προς κατεργασία, η ενεργοποίηση του εργαλείου με το οποίο η μηχανή θα ξεκινήσει την κατεργασία, η εκκίνηση της ατράκτου σύμφωνα με μία ταχύτητα κοπής ένα με σταθερές στροφές, το άνοιγμα των ψυκτικών υγρών του εργαλείου εάν αυτό κρίνεται απαραίτητο και η μετακίνηση του εργαλείου στο αρχικό σημείο του κύκλου κατεργασίας. Οι τιμές για όλα τα

παραπάνω καθορίζονται από τον χρήστη μέσω ενός παραθύρου εισαγωγής δεδομένων το οποίο εμφανίζεται στην οθόνη όταν ο αλγόριθμος ξεκινήσει την εκτέλεσή του.

Στην παρούσα εργασία έχει ληφθεί υπόψη η λειτουργία ενός συγκεκριμένου μηχανήματος CNC, γεγονός που σημαίνει ότι ο αλγόριθμος έχει προγραμματιστεί να δημιουργεί έναν G κώδικα που ακολουθεί τα πρότυπα του Controller του συγκεκριμένου μηχανήματος. Αυτό συνεπάγεται την ύπαρξη κάποιων εντολών και ενός τρόπου λειτουργίας που σε πιο πρόσφατους Controllers δεν χρειάζονται.

Αφού λοιπόν έχουν τυπωθεί οι αρχικές διαδικασίες που χρειάζονται και το εργαλείο έχει μετακινηθεί στο σημείο εκκίνησης της κατεργασίας σειρά έχει η τύπωση του κύκλου κατεργασίας για το εξωτερικό προφίλ. Για την εκτέλεση αυτού του είδους κατεργασιών υπάρχει δύο G κώδικες, ο G71 και ο G70. Ο G71 είναι ο κύκλος ξεχονδρίσματος, ο οποίος ξεκινάει από ένα σημείο κοντά στο αρχικό Stock, για παράδειγμα 1mm πάνω από την διάμετρο του Stock και 1mm έξω από το πρόσωπό του. Χρειάζεται να γνωρίζει τις συντεταγμένες του προφίλ που πρέπει να δημιουργηθεί με την σωστή σειρά καθώς και το είδος της μετακίνησης που θα ακολουθήσει από το ένα σημείο προς το άλλο. Επίσης, θα πρέπει να έχει σαν είσοδο το βήμα με το οποίο θα κατεβαίνει ακτινικά μέχρι να συναντήσει τις επιθυμητές διαμέτρους καθώς επίσης την ταχύτητα με την οποία θα τρέχει το εργαλείο καθώς κατεργάζεται το υλικό. Τέλος, θα πρέπει να προσδιοριστεί και το ποσό του υλικού που θα αφήσει ο κύκλος σε διάμετρο και σε πρόσωπα για το τελικό φινίρισμα. Ο G70 είναι ο κώδικας του φινιρίσματος και ακολουθεί τον G71, δηλαδή ότι προφίλ έχει δηλωθεί στον G71 αυτό θα ακολουθήσει και ο G70, με ένα πάσο στην όλη κατεργασία για φινίρισμα του εξαρτήματος.



Point	X (in.) diameter	Z (in.)
2	4.5	0.1
A	1.8	0
B	1.8	–0.9
C	2.6	–2.6
D	2.6	–2.9
E	3.1	–2.9
F	3.7	–3.2
G	3.7	–3.8
H	4.5	–3.8

Εικόνα 3.4 "Περιγραφή Κύκλου G71 και G70"

Στον συγκεκριμένο Controller για να λειτουργήσουν αυτοί οι δύο κύκλοι θα πρέπει να είναι γνωστός ο αριθμός των γραμμών στις οποίες περιγράφονται οι κινήσεις που θα κάνει το εργαλείο για να δημιουργήσει το προφίλ του αντικειμένου. Για τον λόγο αυτό, πριν ξεκινήσει η διαδικασία τύπωσης του κύκλου στο αρχείο .txt, έχει δημιουργηθεί ένας πίνακας “CycleT” του οποίου το μέγεθος είναι άγνωστο. Οπότε, επειδή δεν είναι γνωστό από πριν το πλήθος των γραμμών των εντολών για το προφίλ, γράφονται οι γραμμές μία προς μία σε αυτόν τον πίνακα, κάθε φορά που καταχωρείται μία γραμμή το μέγεθος του πίνακα αυξάνεται κατά ένα, και μόλις ολοκληρωθεί αυτή η διαδικασία και προσδιοριστεί το μέγεθος του πίνακα μπορεί να προχωρήσει η τύπωση του κύκλου.

Η διαδικασία συμπλήρωσης του πίνακα “CycleT” χωρίζεται σε τρία στάδια, πρώτο στάδιο είναι η αρχή του κύκλου, δεύτερο στάδιο είναι η συνέχεια του κύκλου και τρίτο στάδιο είναι το τελείωμα του κύκλου.

Στο πρώτο στάδιο πρέπει να προσδιοριστεί εάν το προφίλ της εξωτερικής κατεργασίας ξεκινάει με Chamfer, ή με Fillet ή με τίποτα από τα δύο. Οπότε, υπάρχει ο πρώτος έλεγχος στον οποίο ελέγχεται το “varExtTop(0,2)”, δηλαδή το τρίτο νούμερο της πρώτης γραμμής του πίνακα που αντιστοιχεί στην ακτίνα του Chamfer. Εάν αυτό το νούμερο είναι διάφορο του μηδενός τότε σημαίνει πως αυτή η ακμή έχει Chamfer οπότε ο κώδικας ξεκινάει τις κινήσεις του εργαλείου με ανάλογο τρόπο. Εάν αυτός ο αριθμός είναι ίσος με το μηδέν ο κώδικας προχωράει στον έλεγχο του στοιχείου “varExtTop(0,4), στο πέμπτο στοιχείο της πρώτης γραμμής που αντιστοιχεί στην ακτίνα του Fillet της πρώτης ακμής. Εάν αυτός ο αριθμός είναι διάφορος του μηδενός τότε ο κώδικας ξεκινάει τις κινήσεις του εργαλείου με ανάλογο τρόπο. Εάν και το αυτός ο αριθμός είναι ίσος με το μηδέν τότε σημαίνει ότι στην πρώτη

ακμή δεν έχουμε κάποια επεξεργασία οπότε το εργαλείο πηγαίνει κατ' ευθείαν στην τελική διάμετρο του Cut-Extrude.

Επίσης, στο πρώτο στάδιο λαμβάνεται υπόψη και το τελείωμα του πρώτου Cut-Extrude, όπου γίνονται αντίστοιχοι έλεγχοι στα στοιχεία "varExtTop(0,5)" και "varExtTop(0,7)". Οπότε, με την ολοκλήρωση του πρώτου σταδίου της συμπλήρωσης του πίνακα "CycleT" το εργαλείο έχει κατεργαστεί σωστά το πρώτο Cut-Extrude της εξωτερικής κατεργασίας και είναι έτοιμο να μετακινηθεί στο επόμενο, όπου ξεκινάει το δεύτερο στάδιο.

Στο δεύτερο στάδιο της συμπλήρωσης του πίνακα "CycleT", για κάθε στοιχείο του πίνακα "varExtTop" γίνονται οι αντίστοιχοι έλεγχοι στην αρχική τους ακμή και στην τελική τους. Υπάρχει δηλαδή μία επαναληπτική διαδικασία η οποία ελέγχει τα στοιχεία "varExtTop(i,2)", "varExtTop(i,4)", "varExtTop(i,5)" και "varExtTop(i,7)", όπως ακριβώς και στο πρώτο στάδιο, για κάθε i που ανήκει στον πίνακα "varExtTop". Σύμφωνα λοιπόν με αυτά ο αλγόριθμος δημιουργεί τις κατάλληλες εντολές μετακίνησης, "G1" "G2" "G3", και έτσι δημιουργείται το προφίλ της εξωτερικής κατεργασίας μέχρι και το τελευταίο Cut-Extrude που υπάρχει. Σε αυτό το σημείο, το εργαλείο έχει έρθει σε μία συντεταγμένη η οποία είναι ή η διάμετρος και το μήκος του τελευταίου Cut-Extrude ή οι συντεταγμένες της επεξεργασίας που έχει υποστεί η τελευταία ακμή του τελευταίου Cut-Extrude. Μένει λοιπόν να μετακινηθεί το εργαλείο σε μία διάμετρο μεγαλύτερης αυτής του αρχικού Stock για να βγει από το κομμάτι που κατεργάζεται και να ολοκληρωθεί η διαδικασία.

Στην παρούσα φάση είναι που εκτελείται το τρίτο στάδιο συμπλήρωσης του πίνακα "CycleT". Σε αυτό το στάδιο ελέγχεται εάν έχει υποστεί κάποιο Chamfer ή κάποιο Fillet η ακμή που υπάρχει στο μήκος του τελευταίου Cut-Extrude και σε διάμετρο ίση με αυτή του Stock. Έτσι, ο αλγόριθμος εξασφαλίζει ότι θα κατεργαστεί και η τελευταία ακμή του εξαρτήματος και δεν θα βγει απλά το εργαλείο εκτός Stock αφήνοντας κάποιο κομμάτι του εξαρτήματος ακατέργαστο.

Με την ολοκλήρωση του τρίτου σταδίου έχει ολοκληρωθεί και η συμπλήρωση του πίνακα "CycleT" οπότε τώρα ο αλγόριθμος γνωρίζει ακριβώς ποιες και πόσες είναι οι εντολές που χρειάζονται για να προσδιοριστεί με G κώδικα το προφίλ της εξωτερικής κατεργασίας. Ακολουθεί λοιπόν η τύπωση του κύκλου G71, ο οποίος θα έχει την ακόλουθη μορφή

G71 P1 Q_ U_ W_ D_ F_

N1 ….

N2 …

.

.

.N_ ….

Όπου P και Q αντιστοιχούν στον αριθμό της πρώτης και της τελευταίας εντολής που βρίσκονται μέσα στον κύκλο και ξεκινάνε με το γράμμα N. U είναι το υλικό που θα αφήσει ο κύκλος για φινίρισμα ακτινικά, W είναι το υλικό που θα αφήσει ο κύκλος για φινίρισμα στα πρόσωπα, D είναι η ακτίνα του υλικού που θα αφαιρεί κάθε φορά ο κύκλος για να φτάσει στις επιθυμητές διαμέτρους και F είναι το πόσο γρήγορα θα τρέχει το εργαλείο στον άξονα Z κάθε φορά που κατεργάζεται υλικό, και αντιστοιχεί σε χιλιοστά ανά περιστροφή.

Μόλις τυπωθεί στο αρχείο ο κύκλος G71 σειρά έχει το φινίρισμα του εξαρτήματος με τον κύκλο G70. Σε αυτό το σημείο, ελέγχεται εάν ο χρήστης έχει επιλέξει στο παράθυρο εισαγωγής δεδομένων τη χρήση διαφορετικού εργαλείου για το φινίρισμα. Σε περίπτωση που έχει επιλέξει διαφορετικό εργαλείο ο αλγόριθμος εκτελεί μία σειρά εντολών για την αλλαγή του εργαλείου, επιλέγει το σωστό εργαλείο με βάση τα δεδομένα που έχει δώσει ο χρήστης και επαναφέρει το εργαλείο στην αρχική θέση του κύκλου. Τότε, μπορεί να εκτελεστεί η εντολή G70 που είναι της μορφής

G70 P1 Q_

όπου P και Q αντιστοιχούν στους αριθμούς των εντολών που βρίσκονται μέσα στον κύκλο G71.

Σε περίπτωση που ο χρήστης έχει επιλέξει το φινίρισμα να γίνει με το εργαλείο που έγινε το ξεχόνδρισμα τότε ο κύκλος G70 εκτελείται άμεσα χωρίς κάποια μετακίνηση του εργαλείου, καθώς ο κύκλος G71 στο τέλος του φέρνει το εργαλείο στην αρχική θέση από την οποία ξεκίνησε και είναι η ίδια θέση από την οποία πρέπει να ξεκινήσει και ο G70. Στο τέλος αυτού του κύκλου η δουλειά του εκάστοτε εργαλείου έχει τελειώσει, οπότε μεταφέρεται στο σημείο αλλαγής εργαλείου.

Μόλις ολοκληρωθεί και ο κύκλος φινιρίσματος έχει ολοκληρωθεί και η εξωτερική κατεργασία της πρώτης φάσης του αντικειμένου. Ο αλγόριθμος σε αυτό το σημείο είναι έτοιμος να προχωρήσει στην εσωτερική κατεργασία αυτής της φάσης. Εφ' όσον η μηχανή βρίσκεται στο σημείο αλλαγής εργαλείου από την προηγούμενη εντολή, η πρώτη εντολή της εσωτερικής κατεργασίας είναι να πάρει το αντίστοιχο εργαλείο ξεχονδρίσματος χωρίς να γίνει πρώτα κάποια μετακίνηση. Μόλις το σωστό εργαλείο ενεργοποιηθεί, μεταφέρεται στο σημείο εκκίνησης του κύκλου και από εδώ και στο εξής εκτελείται η ίδια διαδικασία με αυτήν της εξωτερικής κατεργασίας. Ο πίνακας "CycleT" ξαναγράφεται από την αρχή και μόλις συμπληρωθεί με τις καινούριες κινήσεις που χρειάζονται να εκτελεστούν για την εσωτερική κατεργασία τυπώνεται ο νέος κύκλος G71. Η διαφορά σε αυτήν την περίπτωση είναι ότι η μεταβλητή U αυτού του κύκλου θα πρέπει να καταχωρηθεί με αρνητικό πρόσημο αυτή τη φορά καθώς δηλώνει τιμή ακτίνας, και στην εσωτερική κατεργασία το ξεχόνδρισμα θέλουμε να μας αφήσει μικρότερη διάμετρο από την τελική για να έρθει το εργαλείο του φινιρίσματος να κόψει και να φινίρει.

Και στην εσωτερική κατεργασία ο χρήστης έχει την επιλογή να δώσει εντολή για διαφορετικό εργαλείο στο φινίρισμα, όπως και στην εξωτερική κατεργασία. Οπότε και εδώ, μόλις τυπωθεί ο κύκλος G71 ελέγχεται εάν έχει επιλεγεί διαφορετικό εργαλείο ή όχι και ο αλγόριθμος εκτελεί τις

αντίστοιχες ενέργειες. Με το τελείωμα και του κύκλου G70 το εργαλείο μετακινείται στην θέση αλλαγής εργαλείου και ο G κώδικας σε αυτό το σημείο έχει τελειώσει, οπότε εκτελείται η εντολή "M30".

Εδώ τελειώνει και η πρώτη φάση κατεργασίας του εξαρτήματος και ο αλγόριθμος είναι έτοιμος να προχωρήσει στην 2η φάση.

3.2.6 Δεύτερη Φάση

Στην δεύτερη φάση κατεργασίας του αντικειμένου οι ενέργειες που πρέπει να γίνουν ουσιαστικά για τον προσδιορισμό των Features, τον διαχωρισμό τους, την ταξινόμησή τους και τον προσδιορισμό των δύο προφίλ κατεργασίας είναι οι ίδιες. Η ουσία σε αυτό το κομμάτι του αλγορίθμου είναι ο τρόπος με τον οποίο θα βρεθούν ποια Features δεν έχουν επιλεγεί, ποια Features δηλαδή πρέπει να ληφθούν υπόψη για την δεύτερη φάση.

Για να γίνει λοιπόν αυτό ο αλγόριθμος ξεκινάει και ελέγχει ένα ένα όλα τα στοιχεία του Feature Tree συγκρίνοντας το ID του στοιχείου που βρίσκει με όλα τα ID από τον πίνακα "varSelectedFeatID", ο οποίος είχε δημιουργηθεί στην αρχή της πρώτης φάσης κατεργασίας και περιέχει όλα τα ID από τα Features που είχαν επιλεγεί. Εφ' όσον ολοκληρωθεί η σύγκριση του ID του ενός στοιχείου με όλα τα ID του "varSelectedFeatID" θα υπάρχουν δύο πιθανά σενάρια, ή το ID που ελέγχεται θα έχει βρεθεί μέσα στον πίνακα "varSelectedFeatID" ή δεν θα έχει βρεθεί. Εάν έχει βρεθεί, σημαίνει ότι το στοιχείο το οποίο ελέγχει εκείνη την στιγμή είναι ένα από τα επιλεγμένα Features, συνεπώς δεν πρέπει να ληφθεί υπόψη για την δεύτερη φάση κατεργασίας και μπορεί να προχωρήσει στον έλεγχο του επόμενου στοιχείου. Εάν δεν έχει βρεθεί σημαίνει ότι το συγκεκριμένο στοιχείο υπάρχει πιθανότητα να είναι κάποιο Feature που προορίζεται για την δεύτερη φάση κατεργασίας.

Επόμενος έλεγχος που πρέπει να γίνει για το συγκεκριμένο στοιχείο που έχει βρεθεί είναι για το εάν αυτό είναι κάποιο Feature από αυτά που μας ενδιαφέρουν για την κατεργασία, για το εάν δηλαδή είναι ένα Cut-Extrude, ένα Chamfer ή ένα Fillet και δεν είναι κάποιο άλλο στοιχείο, όπως για παράδειγμα ένα Plane. Οπότε, κάθε φορά που επαληθεύεται ότι το συγκεκριμένο στοιχείο είναι ένα από τα Features που πρέπει να ληφθούν υπόψη για την δεύτερη φάση κατεργασίας, το ID τους καταχωρείται στον πίνακα "var2ndPhaseID". Μόλις ολοκληρωθεί ο έλεγχος όλων των στοιχείων του Feature Tree, ο πίνακας "var2ndPhaseID" θα περιέχει όλα τα ID από τα Features που πρέπει να ληφθούν υπόψη για την δεύτερη φάση κατεργασίας και ο αλγόριθμος θα είναι σε θέση να εκτελέσει από την αρχή τα βήματα από *3.2.1* έως και *3.2.5*.

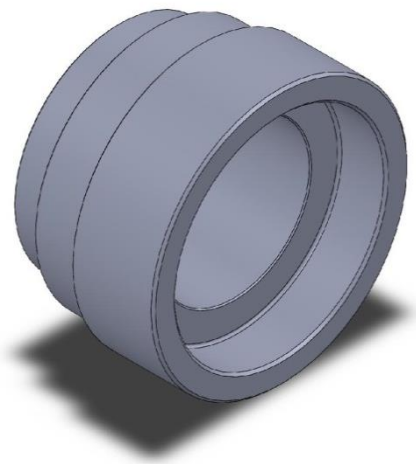
3.3 Αλγόριθμος

Ο κώδικας που έχει δημιουργηθεί παρατίθεται σο Παράρτημα. Σημειώνεται ότι όσες γραμμές ξεκινούν με πρώτο χαρακτήρα «'» είναι σχόλια και δεν εκτελούνται.

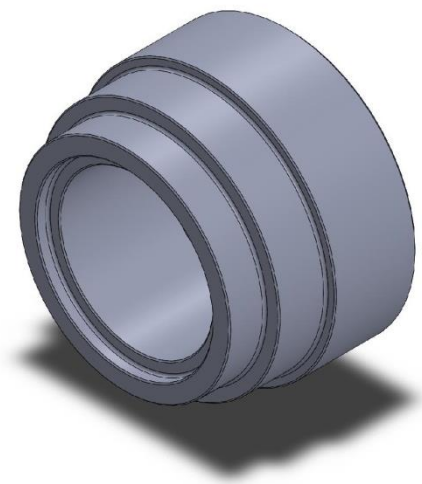
Κεφάλαιο 4: Εφαρμογή Αλγορίθμου

4.1 Σχεδιαστικός Προσδιορισμός Τεμαχίου

Στο παρόν κεφάλαιο της διπλωματικής εργασίας παρουσιάζεται αναλυτικά ο τρόπος που μπορεί να σχεδιαστεί ένα συγκεκριμένο αντικείμενο σύμφωνα με τις οδηγίες που έχουν προσδιοριστεί στο προηγούμενο κεφάλαιο καθώς επίσης παρουσιάζεται και η εκτέλεση του αλγορίθμου με σκοπό την δημιουργία του G κώδικα για το συγκεκριμένο αντικείμενο. Στην συνέχεια, υπάρχει φωτογραφικό υλικό από την διαδικασία κατεργασίας του αντικειμένου σύμφωνα με τα βήματα που πρέπει να ακολουθηθούν καθώς και με την εκτέλεση του G κώδικα που παράχθηκε από τον αλγόριθμο της διπλωματικής εργασίας. Το αντικείμενο το οποίο είναι προς κατεργασία είναι αυτό της «Εικόνας 4.1» και «Εικόνας 4.2».

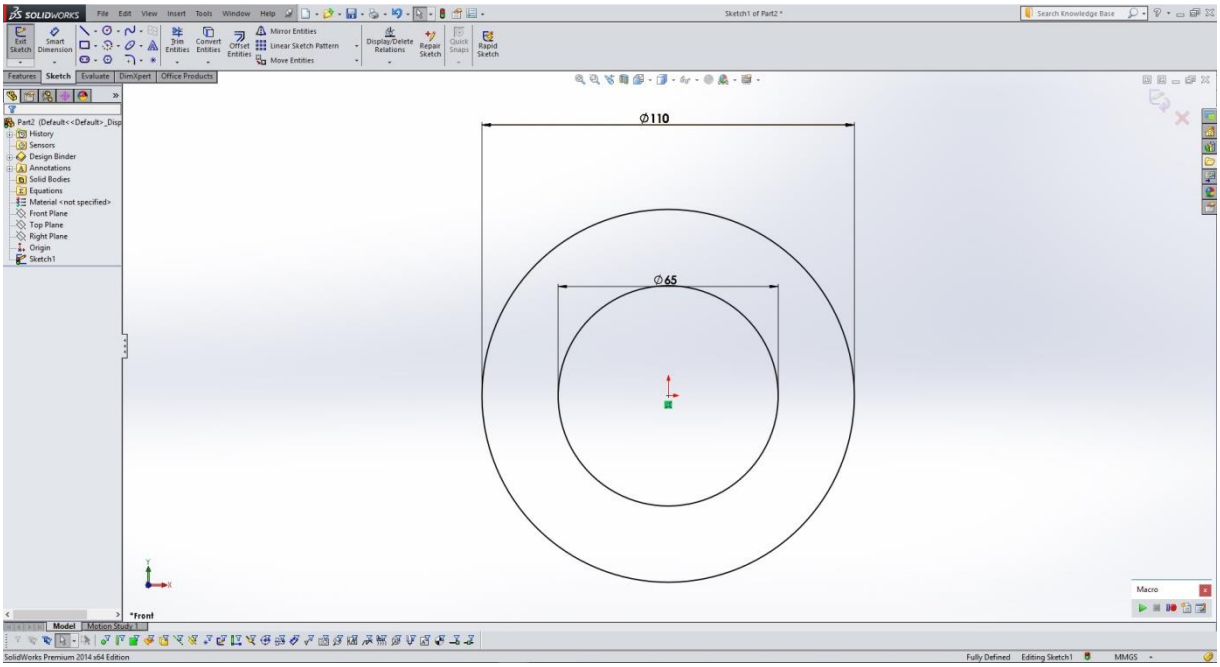


Εικόνα 4.1 " Όψη 1 Αντικειμένου Προς Κατεργασία"



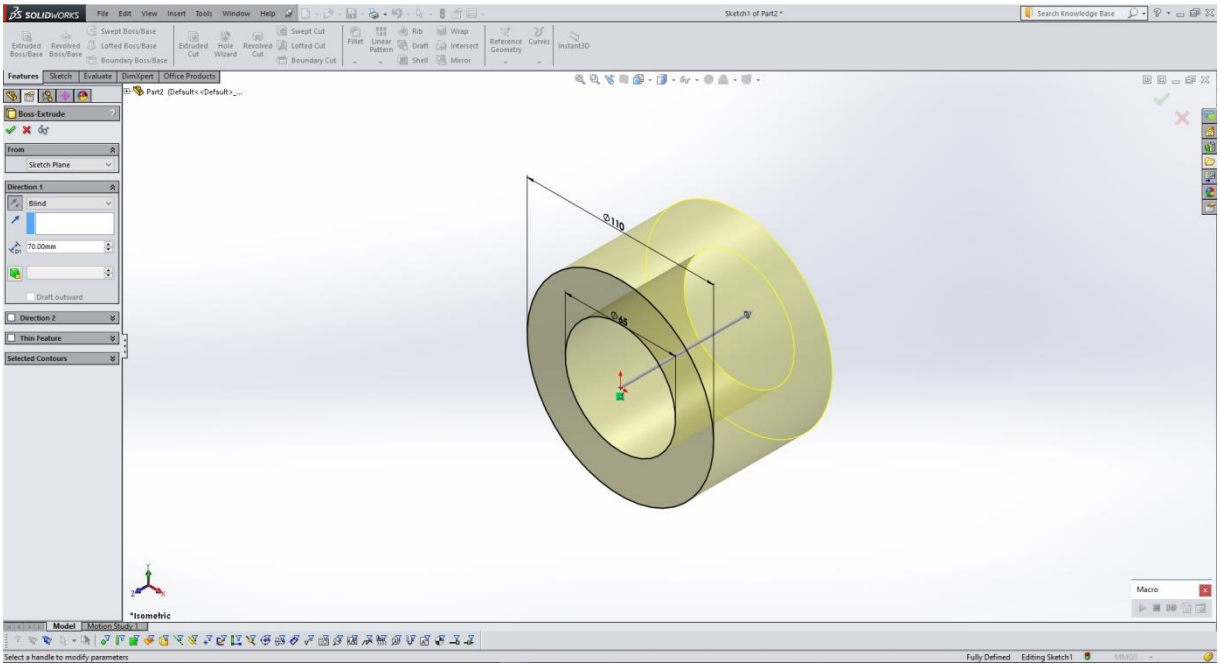
Εικόνα 4.2 " Όψη 2 Αντικειμένου Προς Κατεργασία"

Πρώτο βήμα για τον σχεδιασμό του αντικειμένου είναι ο σχεδιασμός του αρχικού Stock το οποίο προορίζεται για κατεργασία στην CNC εργαλειομηχανή. Αυτό γίνεται ξεκινώντας με την δημιουργία ενός Sketch στο οποίο σχεδιάζονται οι δύο κύκλοι που αποτελούν τις διαστάσεις του Stock, όπως φαίνεται στην «Εικόνα 4.3».



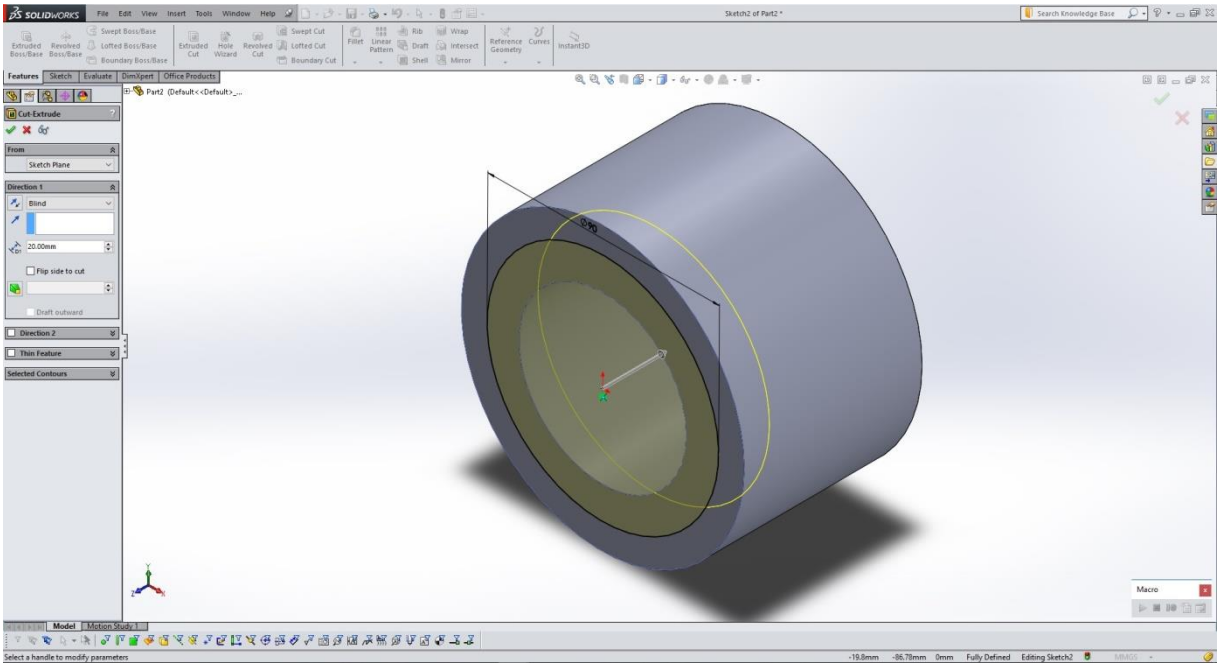
Εικόνα 4.3 "Sketch του Boss-Extrude Feature"

Ακολουθεί η εκτέλεση της εντολής Boss-Extrude στην οποία προσδιορίζεται το μήκος του αρχικού Stock που έχουμε, όπως φαίνεται στην «Εικόνα 4.4». Με την ολοκλήρωση αυτού του βήματος έχει ολοκληρωθεί και ο πλήρης προσδιορισμός του αρχικού Stock.



Εικόνα 4.4 "Εντολή Boss-Extrude"

Σε αυτό το σημείο ο σχεδιαστής είναι σε θέση να ξεκινήσει να διαμορφώνει το εξάρτημα με την χρήση εντολών Cut-Extrude. Ξεκινώντας λοιπόν με την πρώτη φάση του αντικειμένου έπεται ο προσδιορισμός της εσωτερικής κατεργασίας που υπάρχει, καθώς σε αυτή την φάση δεν υπάρχει κάποιο Cut-Extrude για διαμόρφωση εξωτερικής διαμέτρου. Οπότε έχουμη την εκτέλεση της εντολής Cut-Extrude που φαίνεται στην «Εικόνα 4.5».

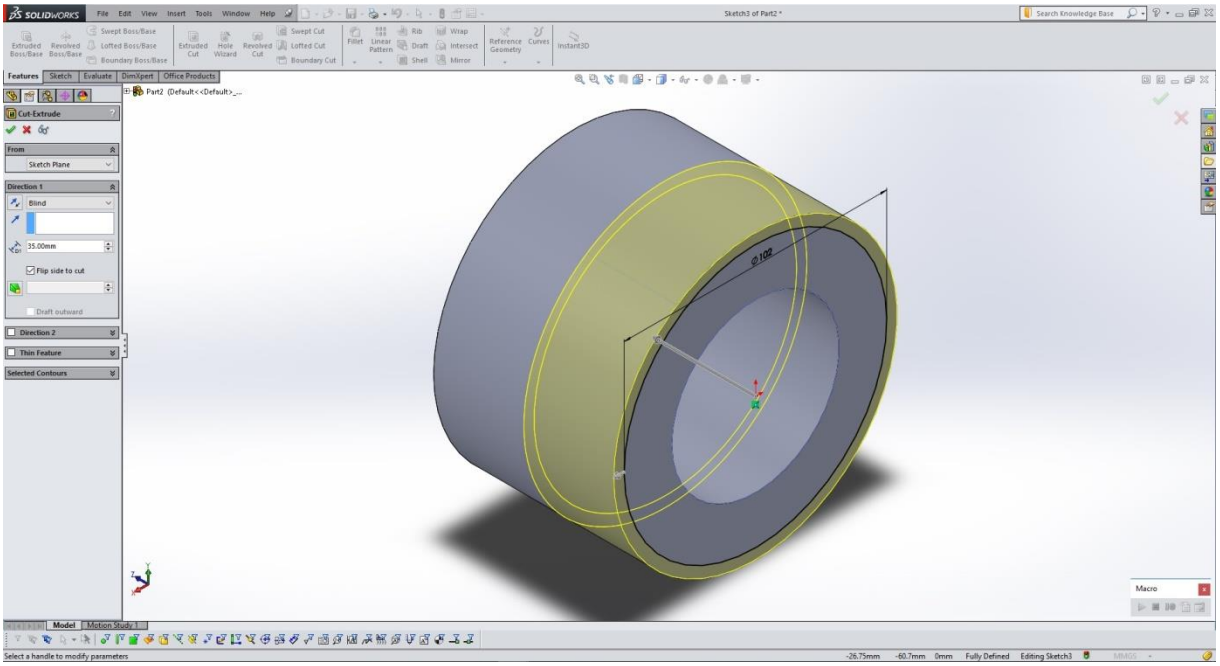


Εικόνα 4.5 "Εντολή Cut-Extrude Εσωτερικής Κατεργασίας"

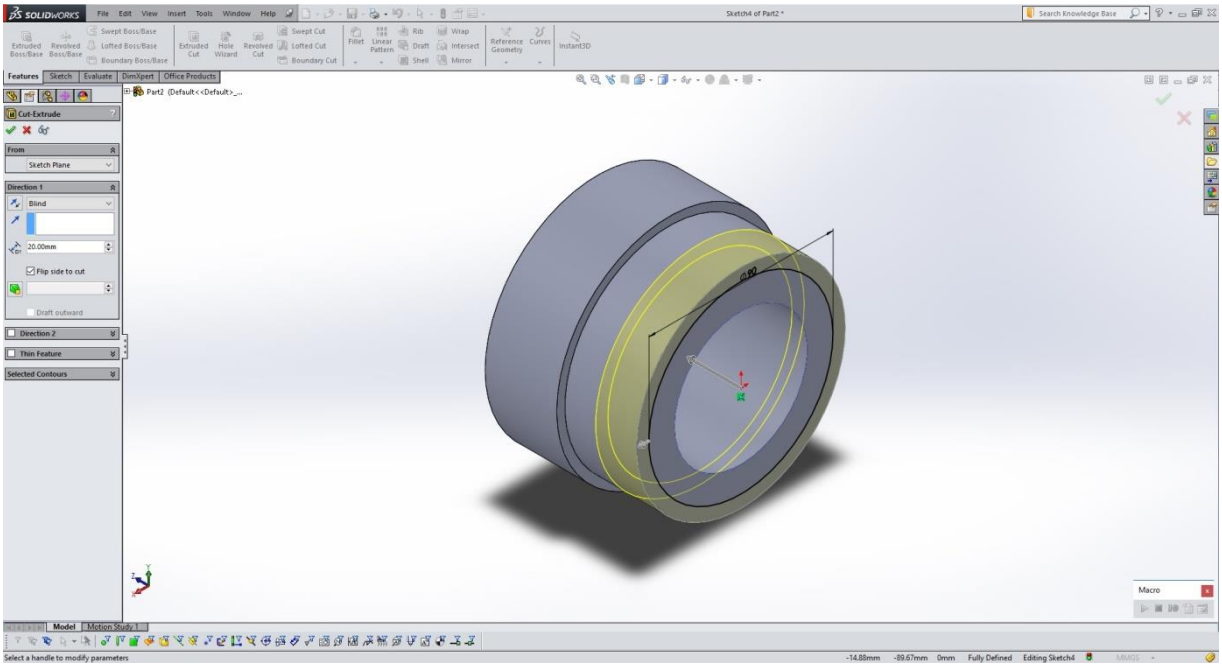
Στη συνέχεια, ο σχεδιαστής μπορεί να προχωρήσει στον προσδιορισμό της δεύτερης φάσης κατεργασίας ξεκινώντας με τον σχεδιασμό των εξωτερικών διαμέτρων. Ακολουθεί λοιπόν ο σχεδιασμός της αμέσως μικρότερης διαμέτρου από αυτής του Stock. Έτσι, έχουμε την εκτέλεση της εντολής όπως φαίνεται στην «Εικόνα 4.6»

Επόμενο βήμα είναι ο σχεδιασμός της επόμενης μικρότερης διαμέτρου, όπως φαίνεται στην «Εικόνα 4.7». Σε αυτό το σημείο πρέπει να παρατηρηθεί ότι όλα τα Sketch που δημιουργούνται για το εκάστοτε Feature βρίσκονται το «πρόσωπο» του αντικειμένου, δηλαδή το κέντρο του κύκλου βρίσκετα πάντα στο σημείο Z0 και από εκεί προσδριορίζεται το μήκος του Feature. Το στοιχείο αυτό είναι βασικό κομμάτι στον τρόπο σχεδιασμού του αντικειμένου καθώς πάνω σε αυτό το στοιχείο βασίζεται και ο αλγόριθμος για να εξάγει σωστά τα δεδομένα που πρέπει.

Επίσης, πρέπει να τονιστεί η χρήση του “Flip Side to Cut”, ενός στοιχείου της εντολής Cut Extrude, το οποίο είναι ανενεργό για διαμέτρους εσωτερικής κατεργασίας και ενεργό για διαμέτρους εξωτερικής κατεργασίας. Βάση αυτού του στοιχείου ο αλγόριθμος κατανοεί και ξεχωρίζει σωστά τις αντίστοιχες κατεργασίες.

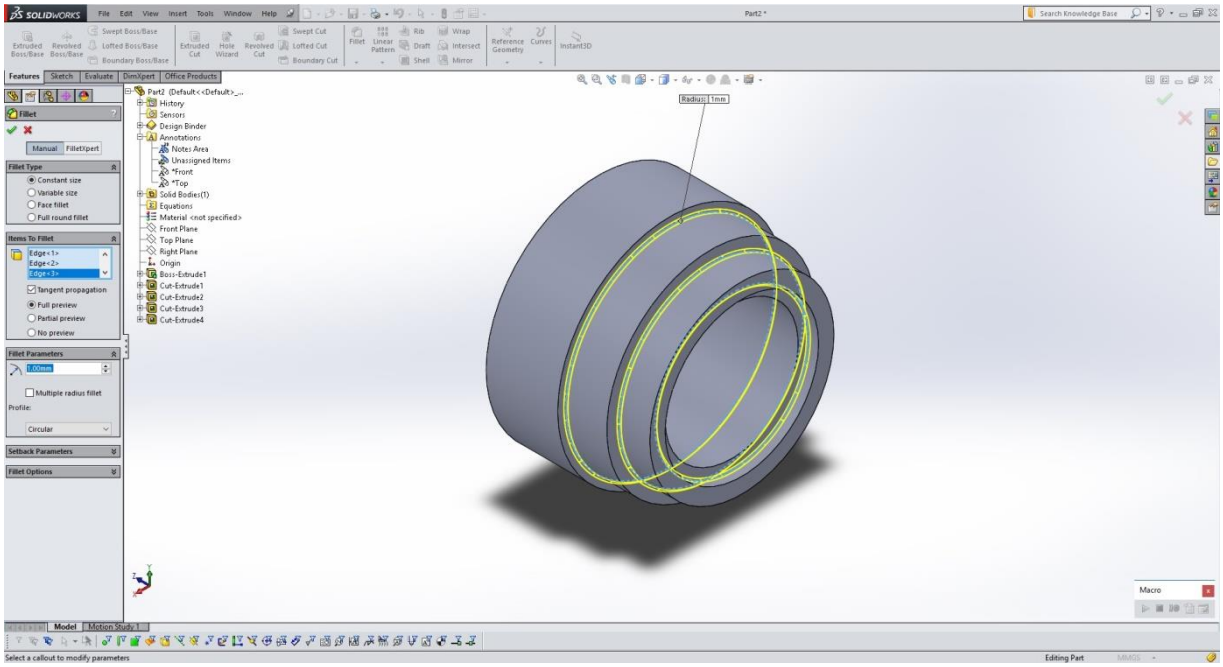


Εικόνα 4.6 "Εντολή Cut-Extrude Εξωτερικής Κατεργασίας"



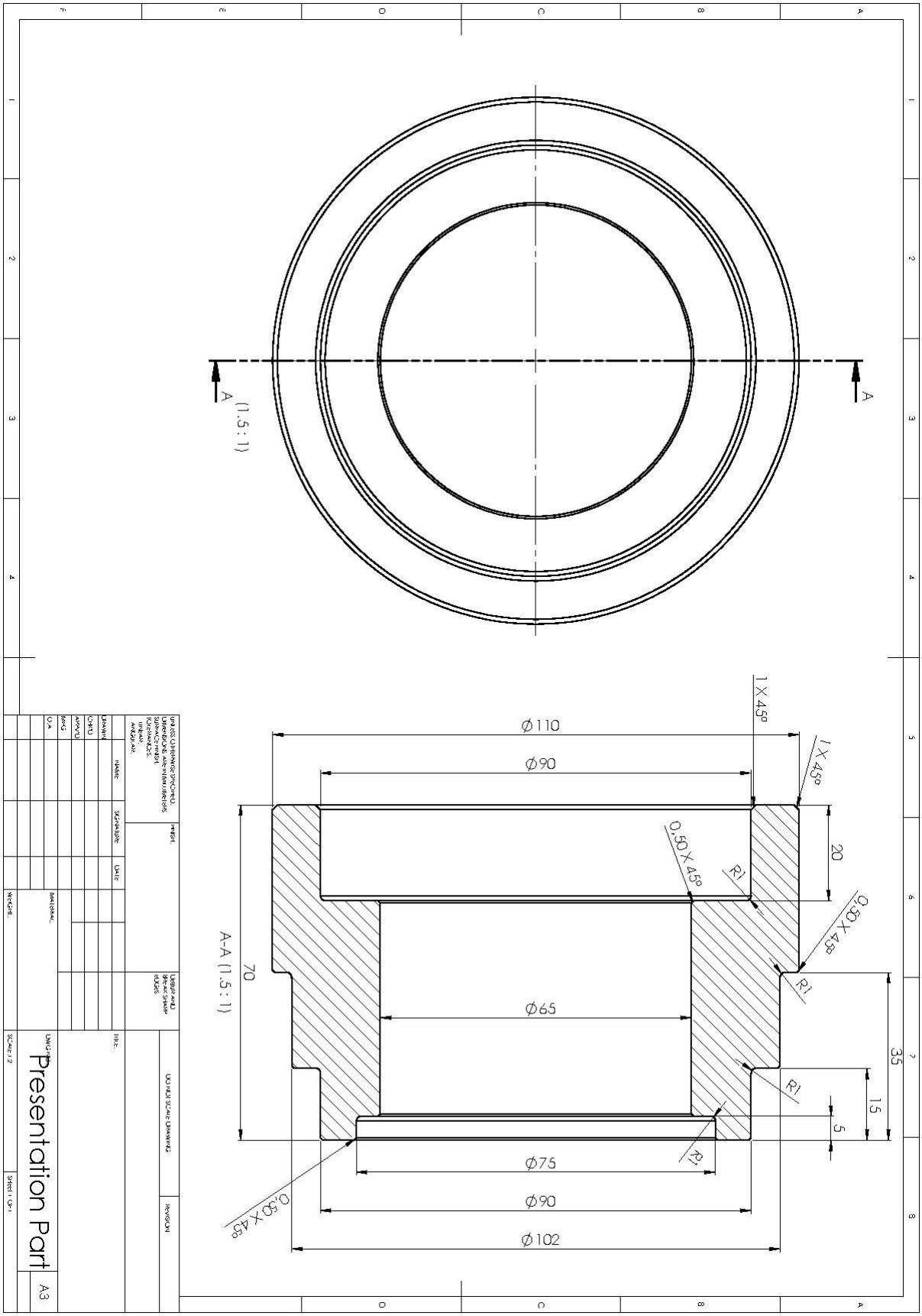
Εικόνα 4.7 "2η Εντολή Cut-Extrude Εξωτερικής Κατεργασίας"

Με αυτή λοιπόν την διαδικασία σχεδιάζεται και το υπόλοιπο αντικείμενο. Μόλις ο σχεδιαστής ολοκληρώσει τον σχεδιασμό των διαμέτρων του αντικειμένου μπορεί να προχωρήσει στην επεξεργασία των ακμών του με χρήση των εντολών Chamfer και Fillet. Σε αυτό το σημείο πρέπει να ο σχεδιαστής να προσέξει οι ακμές που θα επιλέξει για το κάθε Feature που δημιουργεί να ανήκουν στην ίδια φάση κατεργασίας, όπως ακριβώς φαίνεται στην «Εικόνα 4.8».



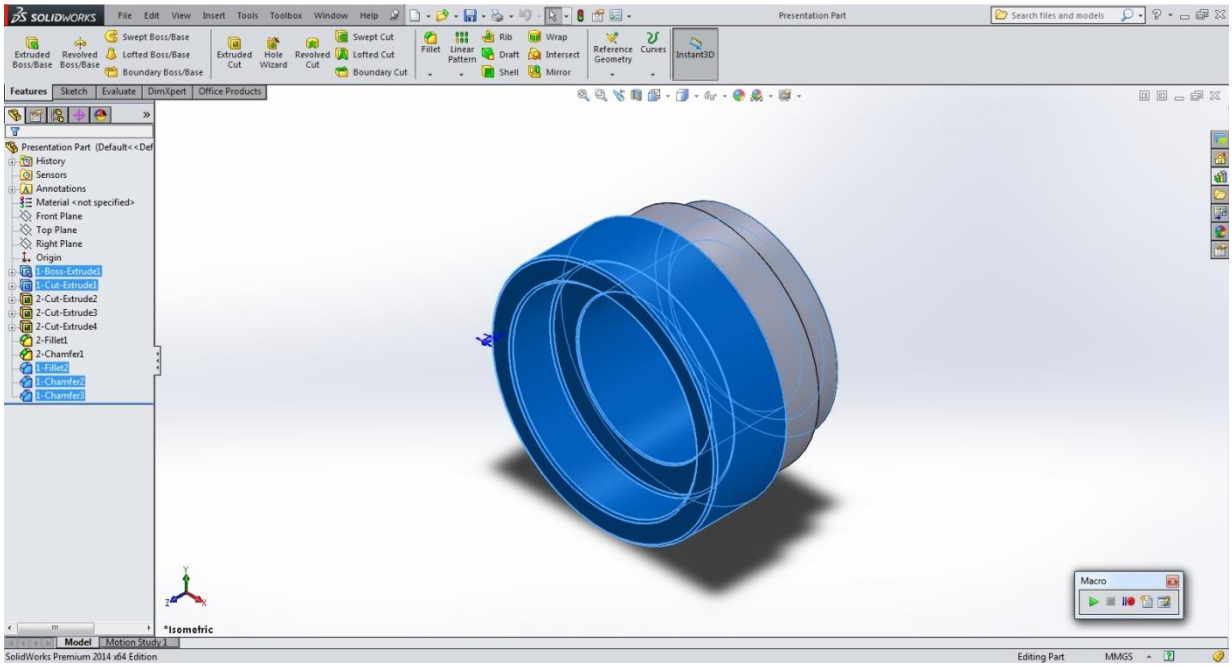
Εικόνα 4.8 "Εντολή Fillet"

Με την χρήση λοιπόν κι αυτών των εντολών ο σχεδιασμός του αντικειμένου έχει ολοκληρωθεί πλήρως και το αντικείμενο προς κατεργασία που προκύπτει είναι αυτό της «Εικόνας 4.9»



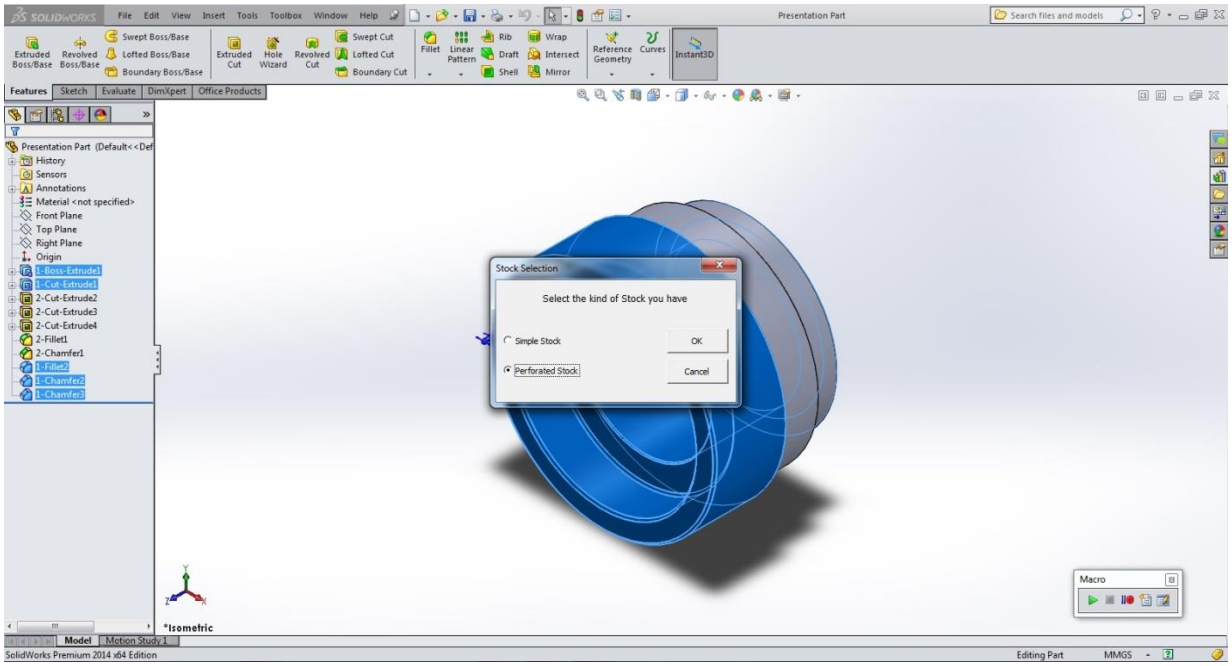
Εικόνα 4.9 "Μηχανολογικό Σχέδιο Αντικειμένου"

Μόλις λοιπόν ο σχεδιαστής ολοκληρώσει τον σχεδιασμό του αντικειμένου είναι σε θέση να εκτελέσει τον αλγόριθμο. Όπως έχει αναφερθεί και σε προηγούμενη παράγραφο της εργασίας, πρώτο βήμα για την εκτέλεση του αλγορίθμου είναι η επιλογή των Features από το Feature Tree τα οποία πρέπει να επεξεργαστούν στην πρώτη φάση κατεργασίας. Για την διευκόλυνσή του, ο σχεδιαστής θα μπορούσε να μετανομάσει τα Features προσθέτοντας μπροστά από το όνομα του κάθε Feature τον αριθμό «1» ή «2» ο οποίος αντιστοιχεί στην φάση κατεργασίας στην οποία βρίσκεται το κάθε Feature, με σκοπό την ευκολότερη επιλογή τους στην συνέχεια όταν θα χρειαστεί να εκτελέσει στον αλγόριθμο. Το πρώτο βήμα λοιπόν φαίνεται στην «Εικόνα 4.10».



Εικόνα 4.10 "Επιλογή Features 1ης Φάσης"

Καθώς είναι επιλεγμένα τα Features της πρώτης φάσης κατεργασίας, ο χρήστης μπορεί να εκτελέσει τον αλγόριθμο. Από το παράθυρο «Macro» που φαίνεται κάτω δεξιά στην «Εικόνα 4.10» πατώντας το κουμπί «Play» ο χρήστης μπορεί να επιλέξει την μακροεντολή που θέλει, στην συγκεκριμένη περίπτωση η μακροεντολή, που ουσιαστικά αποτελεί τον αλγόριθμο που έχει δημιουργηθεί, έχει το όνομα «G-CODE.swp». Ξεκινώντας λοιπόν η μακροεντολή εμφανίζεται το πρώτο από τα δύο συνολικά παράθυρα που εμφανίζει η μακροεντολή για την αλληλεπίδρασή της με τον χρήστη. Σε αυτό το πρώτο παράθυρο, το οποίο φαίνεται στην «Εικόνα 4.11», ο χρήστης καλείται αν επιλέξει τον τύπο του αρχικού Stock, εάν αυτό δηλαδή είναι διάτρητο ή όχι. Για το αντικείμενο της παρουσίασης η σωστή επιλογή είναι το «Perforated Stock», καθώς το αρχικό Stock αποτελείται από μία οπή διαμέτρου 65mm, οπότε είναι διάτρητο.



Εικόνα 4.11 "Παράθυρο Προσδιορισμού Τύπου Stock"

Μόλις λοιπόν ο χρήστης επιλέξει το «Perforated Stock» και πατήσει «OK», στη συνέχεια εμφανίζεται το δεύτερο παράθυρο, το οποίο παρατείθεται στην «Εικόνα 4.12». Σε αυτό το παράθυρο ο χρήστης καλείται να επιλέξει τις συνθήκες κοπής του ξεχονδρίσματος και του φινιρίσματος του αντικειμένου καθώς επίσης και το εάν θα χρησιμοποιηθούν διαφορετικά εργαλεία για ξεχόνδρισμα και διαφορετικά για φινίρισμα. Όσο αναφορά τις συνθήκες κοπής, ο χρήστης έχει τις ακόλουθες επιλογές:

Για το ξεχόνδρισμα

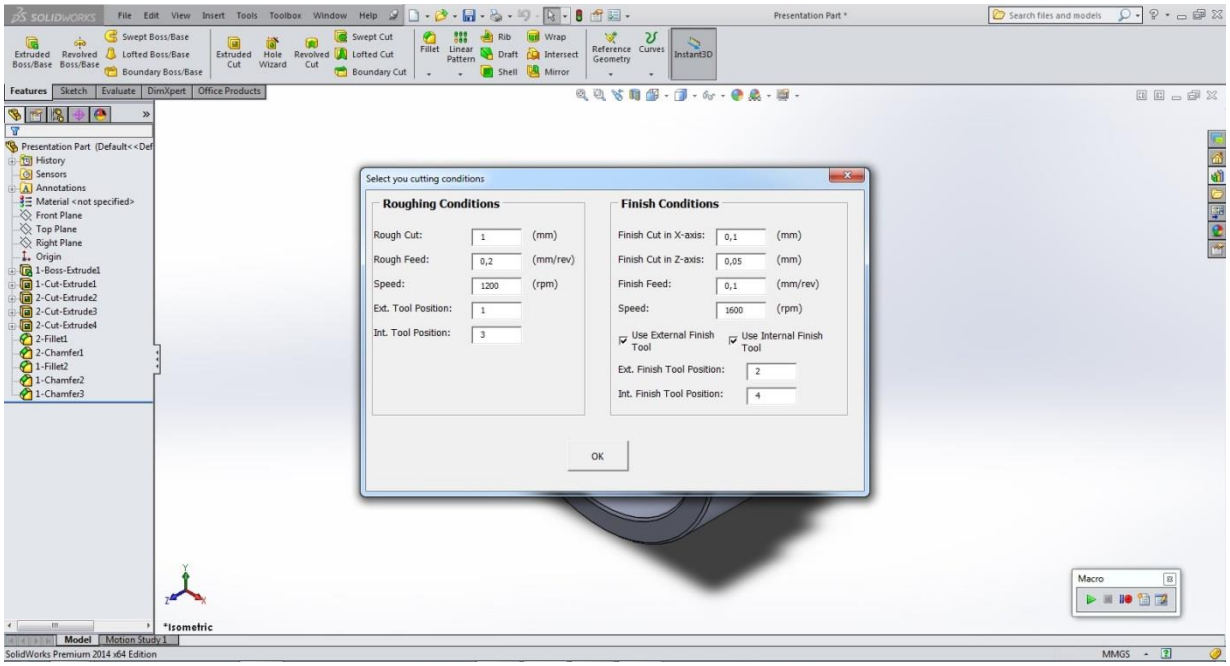
- “Rough Cut”: Το μέγεθος της κάθε κοπής που θα κάνει το εργαλείο μέχρι να φτάσει στην επιθυμητή διάμετρο. Η τιμή αυτής της επιλογής αντιστοιχεί σε ακτίνα και όχι σε διάμετρο, δηλαδή εάν ο χρήστης σε αυτό το πεδίο συμπληρώσει την τιμή 0,5 σημαίνει ότι το εργαλείο εάν για παράδειγμα ξεκινήσει τον κύκλο του από μία διάμετρο 60mm για να φτάσει σε μία διάμετρο 50mm η πρώτη κοπή που θα γίνει θα είναι σε διάμετρο 59mm, και όχι σε 59,5mm. Η επόμενη σε διάμετρο 58mm και ούτω καθεξής μέχρι να φτάσει στην διάμετρο των 50mm
- “Rough Feed”: Αντιπροσωπεύει το μήκος κατά του άξονα Z το οποίο θα μετακινείται το εργαλείο σε κάθε περιστροφή της ατράκτου όταν θα βρίσκεται σε κίνηση κοπής
- “Speed”: Αντιστοιχεί στις στροφές της ατράκτου ανά λεπτό
- “Ext. Tool Position”: Αντιστοιχεί σε ποια θέση εργαλείου της μηχανής βρίσκεται το εργαλείο ξεχονδρίσματος εξωτερικής κατεργασίας (π.χ. στην θέση 1)
- “Int. Tool Position”: Αντιστοιχεί σε ποια θέση εργαλείου της μηχανής βρίσκεται το εργαλείο ξεχονδρίσματος εσωτερικής κατεργασίας

Για το φινίρισμα

- “Finish Cut in X-axis”: Αντιστοιχεί στην ακτίνα στην οποία θα σταματήσει την κατεργασία το εργαλείο του ξεχονδρίσματος για να έρθει να κατεργαστεί το εργαλείο του φινιρίσματος. Για παράδειγμα, εάν η τιμή αυτού του πεδίου είναι «0,2» και έχουμε μια τελική διάμετρο στα 50mm, τότε το εργαλείο του ξεχονδρίσματος θα σταματήσει την κατεργασία του στην διάμετρο των 50,4mm και από εκεί θα συνεχίσει την κατεργασία το εργαλείο του φινιρίσματος.
- “Finish Cut in Z-axis”: Αντίστοιχα με την προηγούμενη επιλογή αλλά αυτή απευθύνεται στον άξονα Z
- “Finish Feed”: Το μήκος το οποίο θα μετακινείται το εργαλείο φινιρίσματος ανά περιστροφή της ατράκτου καθώς βρίσκεται σε κίνηση κοπής. Αυτή η τιμή κατά κανόνα πρέπει να είναι μικρότερη της αντίστοιχης του ξεχονδρίσματος με σκοπό το εργαλείο να κινείται με μικρότερη ταχύτητα καθώς κατεργάζεται και κατ επέκταση να δημιουργεί πιο γυαλιστερή επιφάνεια
- “Speed”: Όπως και στο ξεχόνδρισμα έτσι κι εδώ αντιστοιχεί στις στροφές της ατράκτου ανά λεπτό
- “Ext. Finish Tool Position”: Αντιστοιχεί σε ποια θέση εργαλείου της μηχανής βρίσκεται το εργαλείο φινιρίσματος εξωτερικής κατεργασίας
- “Int. Finish Tool Position”: Αντιστοιχεί σε ποια θέση εργαλείου της μηχανής βρίσκεται το εργαλείο φινιρίσματος εσωτερικής κατεργασίας

Για να λειτουργήσουν οι δύο τελευταίες επιλογές θα πρέπει να έχουν τσεκαριστεί τα πεδία “Use External Finish Tool” και “Use Internal Finish Tool” αντίστοιχα. Εάν δεν έχουν επιλεγεί τα αντίστοιχα αυτά τα πεδία, ότι τιμή υπάρχει στην αντίστοιχη επιλογή αγνοείται από τον αλγόριθμο και το φινίρισμα γίνεται με το ίδιο εργαλείο που έγινε και το ξεχόνδρισμα.

Μόλις λοιπόν ο χρήστης συμπληρώσει τα απαραίτητα πεδία στο δεύτερο παράθυρο που εμφανίζεται, είναι σε θέση να πατήσει το κουμπί «OK» και να συνεχίσει η εκτέλεση του αλγορίθμου. Έπειτα, ο αλγόριθμος επεξεργάζεται τις εισόδους που έχει σύμφωνα με τις διαδικασίες που έχουν περιγραφεί στο προηγούμενο κεφάλαιο και εξάγει τα δύο «.txt» αρχεία τα οποία περιέχουν τον G κώδικα για το αντικείμενο της παρουσίασης. Το περιεχόμενο των δύο αυτών αρχείων παρουσιάζεται παρακάτω.



Εικόνα 4.12 "Παράθυρο Προσδιορισμού Συνθηκών Κοπής"

Περιεχόμενο Αρχείου «G-Code Phase 1»:

```
O0900
G90
G53 G0 X0 Z0
T0303
G41
S1200 M3
M8
G0 X65 Z1
G71 P10 Q16 U-0,1 W0,05 D1 F0,2
N10 G0 X92
N11 G1 Z0 F0,1
N12 X90 Z-1
N13 G1 Z-19 F0,1
N14 G3 X88 Z-20 R1 F0,1
N15 G1 X66 F0,1
N16 X65 Z-20,5
G0 X65 Z1
M9
G53 G0 X0 Z0
G40
T0300
T0404
G41
S1600 M3
```

```
M8
G0 X65 Z1
G70 P10 Q16
M9
G53 G0 X0 Z0
T0400
G40
M30
```

Περιεχόμενο Αρχείου «G-Code Phase 1»:

```
O0901
G90
G53 G0 X0 Z0
T0101
G42
S1200 M3
M8
G0 X110 Z1
G71 P10 Q20 U0,1 W0,05 D1 F0,2
N10 G0 X89
N11 G1 Z0 F0,1
N12 X90 Z-0,5
N13 G1 Z-14 F0,1
N14 G2 X92 Z-15 R1 F0,1
N15 G1 X101 F0,1
N16 X102 Z-15,5
N17 G1 Z-34 F0,1
N18 G2 X104 Z-35 R1 F0,1
N19 G1 X109 F0,1
N20 X110 Z-35,5
G0 X110 Z1
M9
G53 G0 X0 Z0
G40
T0100
T0202
G42
S1600 M3
M8
G0 X110 Z1
G70 P10 Q20
M9
G53 G0 X0 Z0
T0200
G40
```

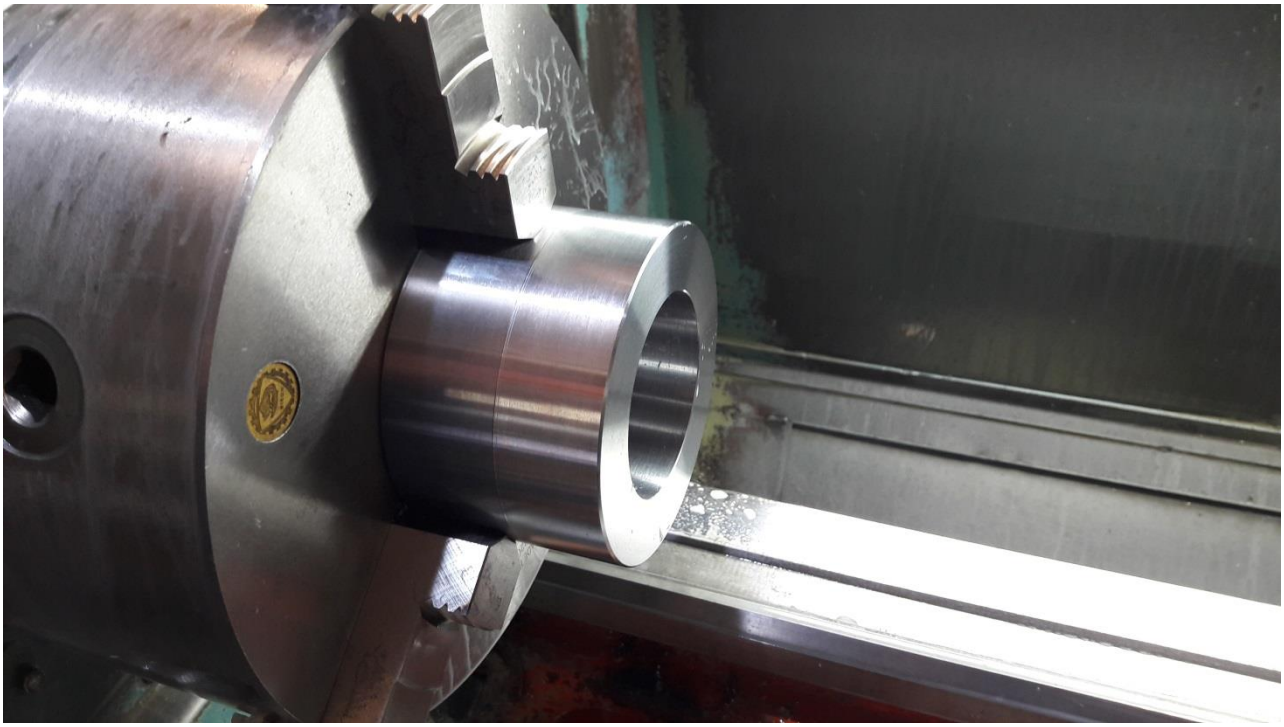
```
T0303
G41
S1200 M3
M8
G0 X65 Z1
G71 P10 Q16 U-0,1 W0,05 D1 F0,2
N10 G0 X76
N11 G1 Z0 F0,1
N12 X75 Z-0,5
N13 G1 Z-4 F0,1
N14 G3 X73 Z-5 R1 F0,1
N15 G1 X66 F0,1
N16 X65 Z-5,5
G0 X65 Z1
M9
G53 G0 X0 Z0
G40
T0300
T0404
G41
S1600 M3
M8
G0 X65 Z1
G70 P10 Q16
M9
G53 G0 X0 Z0
T0400
G40
M30
```

4.2 Αποτέλεσμα Εφαρμογής Αλγορίθμου

Αφού εκτέλεστηκαν όλα τα βήματα που παρουσιάστηκαν στην Παράγραφο 4.1 και δημιουργήθηκαν οι δύο G κώδικες, επόμενο βήμα ήταν η κατεργασία του αντικειμένου. Έτσι λοιπόν, ο τεχνίτης ετοίμασε το αντικείμενο σύμφωνα με τις διαστάσεις που πρέπει να έχει το αρχικό Stock όπως αυτό έχει σχεδιαστεί. Έπειτα, τοποθετήθικαν τα κατάλληλα εργαλεία στην εργαλειομηχανή και στις σωστές θέσεις, σύμφωνα με τις οδηγίες του σχεδιαστή. Τέλος, τα δύο αρχεία με τον G κώδικα που δημιουργήθηκαν από τον αλγόριθμο της παρούσας διπλωματικής εργασίας, μεταφέρθηκαν μέσω καλωδίου RS232 στην μνήμη της εργαλειομηχανής.

Μετά την ολοκλήρωση των παραπάνω ενεργειών ο χειριστής της εργαλειομηχανής είναι έτοιμος να κατεργαστεί το αντικείμενο. Ακολουθεί φωτογραφικό υλικό με τις φάσεις κατεργασίας του αντικειμένου πάνω στην εργαλειομηχανή καθώς επίσης και με το τελικό αποτέλεσμα.

Εικόνα 4.13 "Φωτογραφία Εξαρτήματος στο Μηχάνημα Πριν την Κατεργασία"

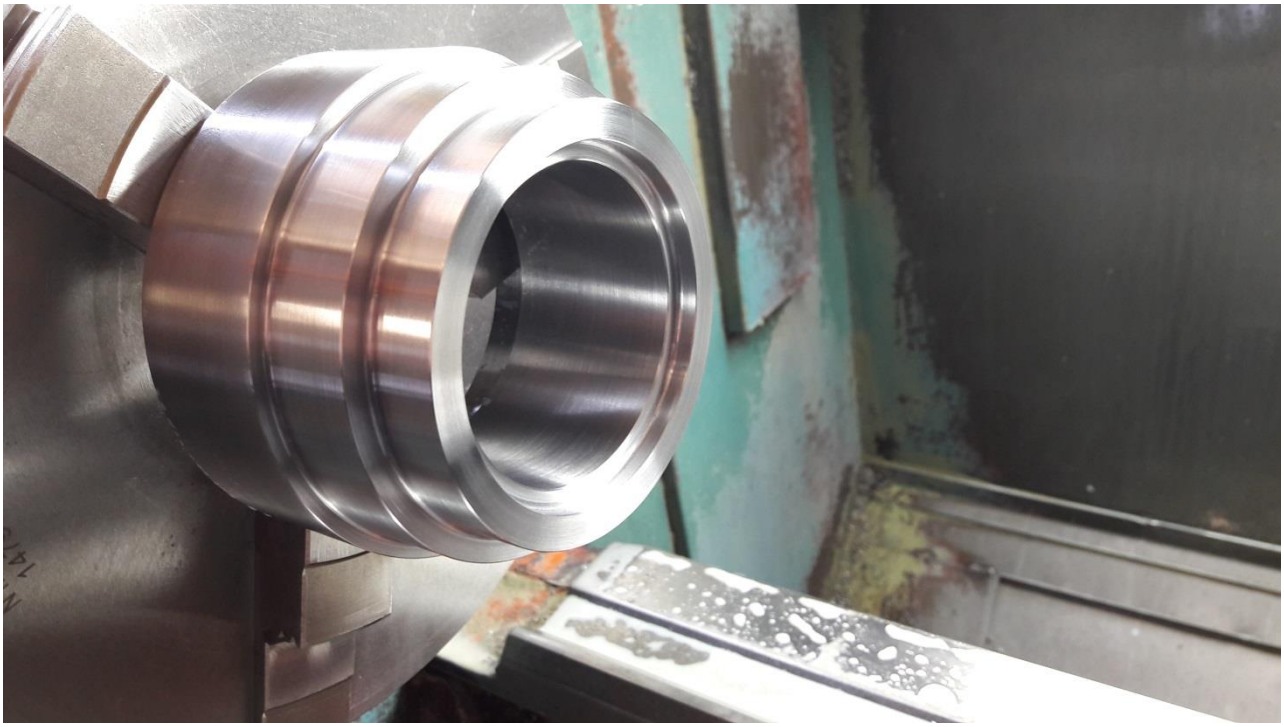


Εικόνα 4.15 "Φωτογραφία Εξαρτήματος στο Μηχάνημα Πριν την Κατεργασία"

Εικόνα 4.14 "Φωτογραφία Εξαρτήματος Μετά την 1η Φάση Κατεργασίας"

Εικόνα 4.16 "Φωτογραφία Εξαρτήματος Μετά την 1η Φάση Κατεργασίας"

Εικόνα 4.17 "Φωτογραφία Εξαρτήματος Μετά την 2η Φάση Κατεργασίας"

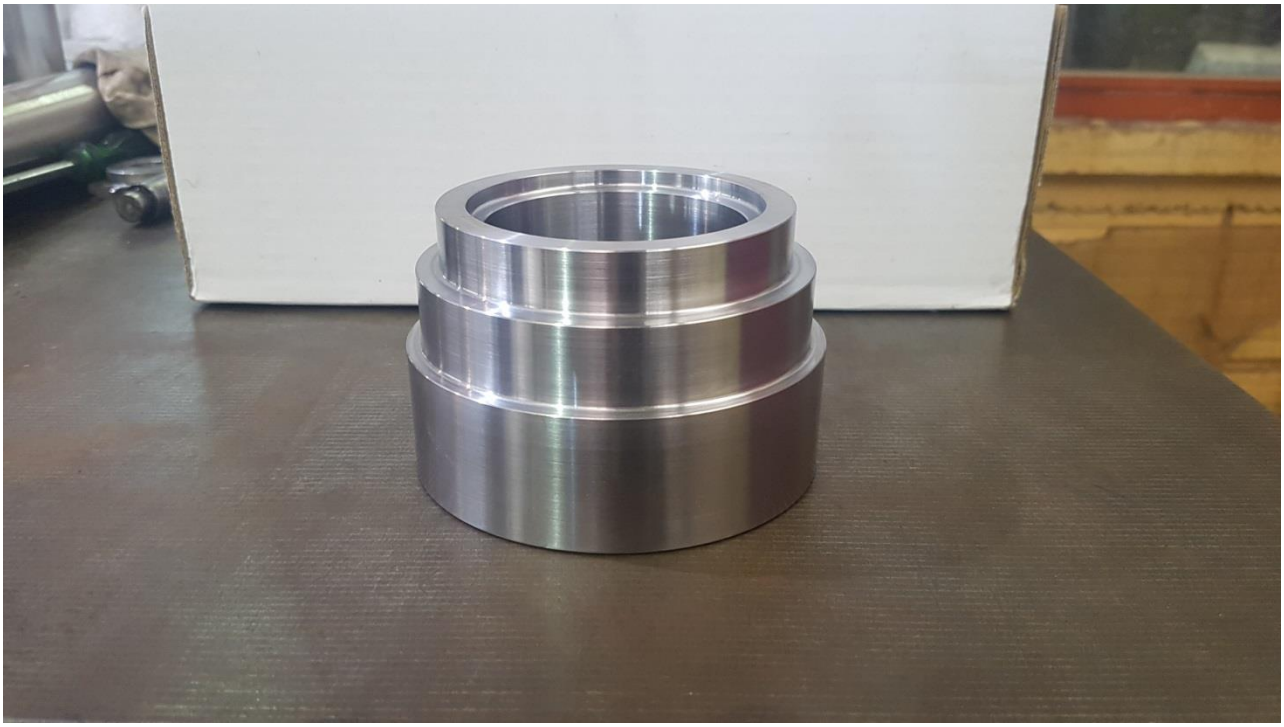


Εικόνα 4.18 "Φωτογραφία Εξαρτήματος Μετά την 2η Φάση Κατεργασίας"

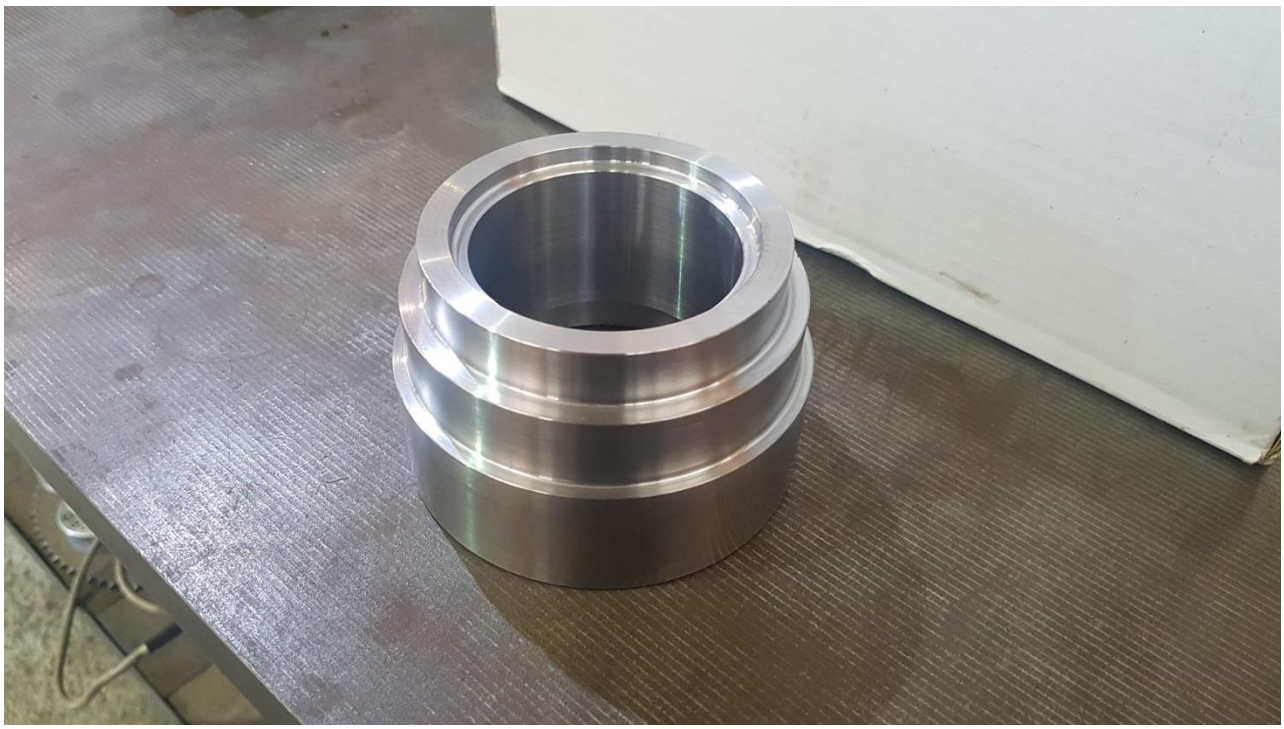
Εικόνα 4.19 "Φωτογραφία Εξαρτήματος Μετά την 2η Φάση Κατεργασίας"

Εικόνα 4.20 "Φωτογραφία Ολοκληρωμένου Εξαρτήματος"

Εικόνα 4.21 "Φωτογραφία Ολοκληρωμένου Εξαρτήματος"



Εικόνα 4.22 "Φωτογραφία Ολοκληρωμένου Εξαρτήματος"



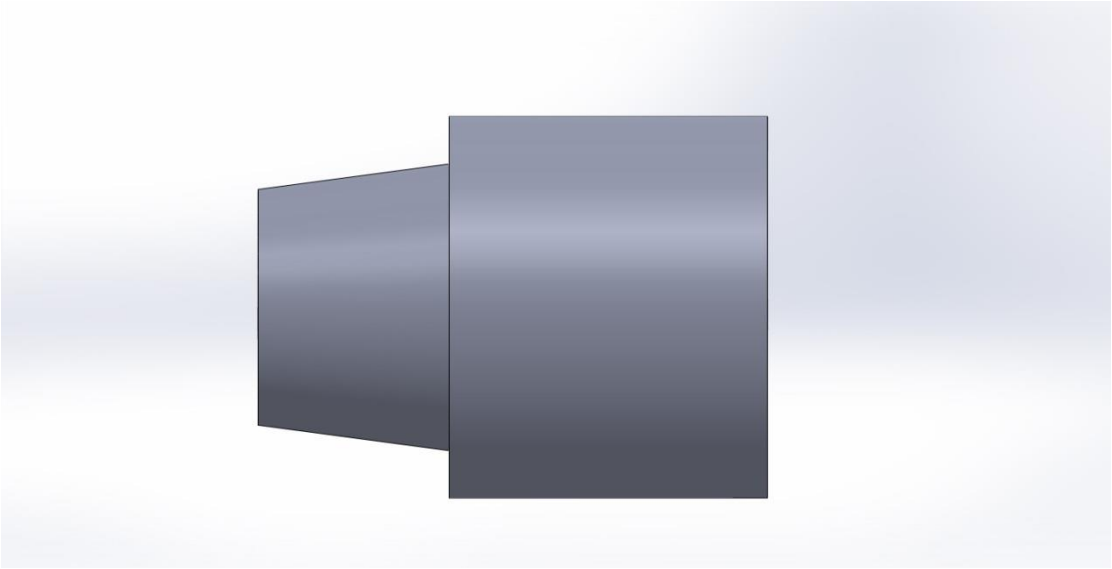
Εικόνα 4.23 "Φωτογραφία Ολοκληρωμένου Εξαρτήματος"

Κεφάλαιο 5: Συμπεράσματα και Προτάσεις

5.1 Προτεινόμενη Χρήση Αλγορίθμου

Ένα προϊόν του μηχανουργείου για να φτάσει στην τελική του μορφή πρέπει να ξεκινήσει από τον σχεδιασμό του, στη συνέχεια εφόσον κατασκευαστεί σε μία εργαλειομηχανή CNC θα πρέπει να γίνει ο προγραμματισμός της μηχανής σε G κώδικα, και τέλος να φτάσουμε στην κατεργασία του αντικειμένου. Όταν πρόκειται για ένα εξάρτημα που χρειάζεται κατεργασία φρέζας τα πράγματα είναι λίγο πιο πολύπλοκα σε σχέση με τον τόρνο. Παραταύτα, ο προγραμματισμός της εργαλειομηχανής για την κατασκευή ενός εξαρτήματος τόρνευσης παραμένει αρκετά χρονοβόρος.

Με τον αλγόριθμο της παρούσας εργασίας, ο χρόνος που απαιτείται για την κατασκευή ενός κομματιού τόρνευσης μειώνεται κατά ένα μεγάλο ποσοστό καθώς μηδενίζεται ο χρόνος που απαιτείται για τον προγραμματισμό της εργαλειομηχανής. Από την στιγμή που ο σχεδιαστής εξοικειωθεί με τον τρόπο σχεδίασης που προτείνεται για την χρήση του αλγορίθμου θα έχει εξοικονομήσει χρόνο και χρήματα για την επιχείρησή του.

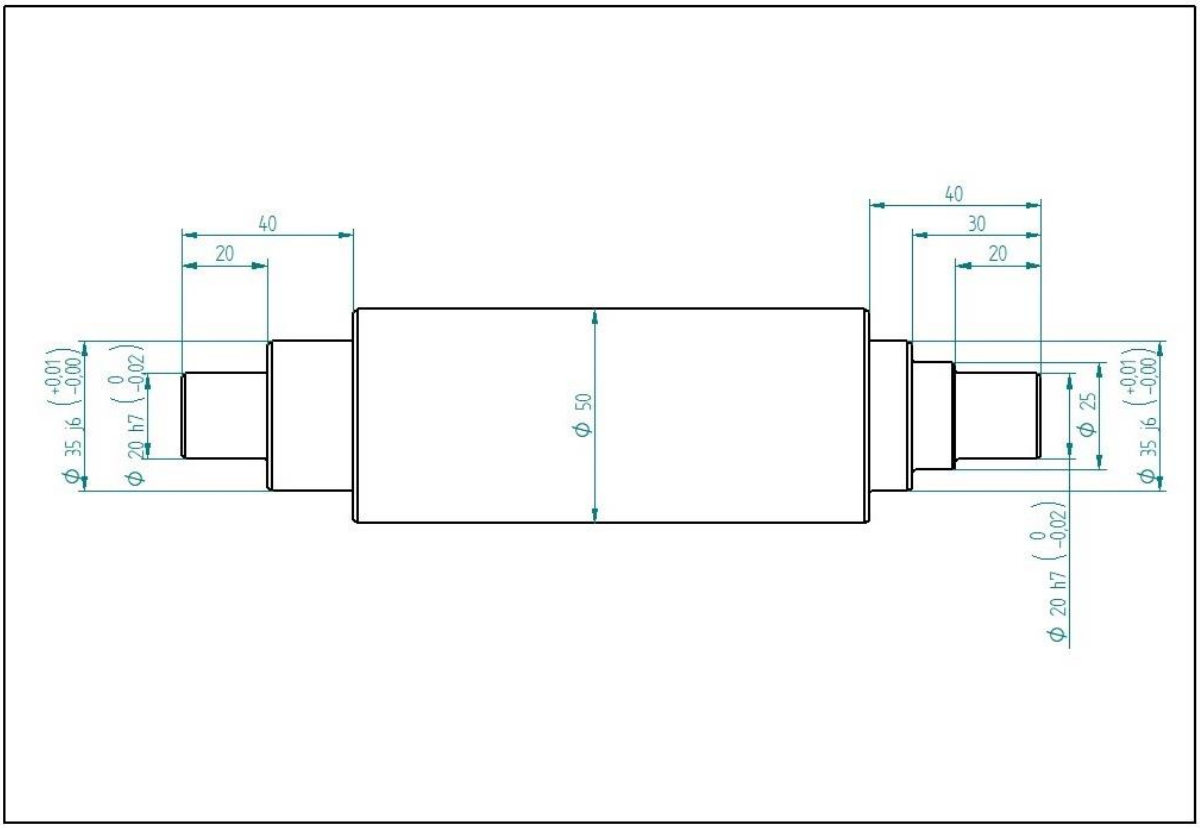


Εικόνα 5.1 "Παράδειγμα Αντικειμένου με Κωνική Επιφάνεια"

Στην μορφή που βρίσκεται ο αλγόριθμος, η χρήση για την οποία προτείνεται είναι για την κατασκευή απλών εξαρτημάτων τόρνευσης. Εξαρτήματα δηλαδή που αποτελούνται από απλές διαμέτρους, σπασίματα γωνιών (chamfers) και ράδια στις ακμές του (Fillets). Με αυτά τα μορφολογικά χαρακτηριστικά καλύπτεται μία μεγάλη γκάμα εξαρτημάτων τόρνευσης καθώς όταν μιλάμε για σπασίματα γωνιών, επειδή ο αλγόριθμος έχει την δυνατότητα να αναγνωρίζει και τις δύο διαστάσεις του σπασίματος, δεν περιορίζεται σε αυτά με 45 μοίρες κλίση. Οπότε μπορούμε να πούμε ότι ο

αλγόριθμος έχει τη δυνατότητα να εξυπηρετήσει και εξαρτήματα τα οποία περιέχουν μεγάλες κωνικές επιφάνειες, αρκεί βέβαια ο σχεδιαστής να σχεδιάσει σωστά το αντικείμενο, δηλαδή η κωνική επιφάνεια να γίνει με ένα Chamfer Feature και όχι με ένα Cut Extrude.

Μία ακόμη εφαρμογή του αλγορίθμου θα ήταν και σε εξαρτήματα που έχουν απαιτήσεις μικρών ανοχών σε συγκεκριμένες επιφάνειες. Για παράδειγμα, όταν ο μηχανολόγος σχεδιάζει έναν άξονα ο οποίος θα μεταφέρει κίνηση, υπάρχουν ανοχές στις διαμέτρους στις οποίες αφορούν τα ρουλεμάν που θα χρησιμοποιηθούν στον άξονα καθώς επίσης και στη διάμετρο που θα χρησιμοποιηθεί για την σύνδεση του άξονα με τον κινητήρα από τον οποίο θα πάρει κίνηση και στη διάμετρο που θα χρησιμοποιηθεί για να δώσει την κίνηση. Όλες αυτές οι ανοχές είναι σε εκατοστά του χιλιοστού, οπότε αυτός ο άξονας απαιτεί ειδική προσέγγιση από τον κατασκευαστή..



Εικόνα 5.2 "Μηχανολογικό Σχέδιο με Ανοχές"

Η διαδικασία κατασκευής ενός τέτοιου άξονα είναι ότι κατασκευάζεται όλος άξονας στις τελικές του διαστάσεις ενώ οι διάμετροι που έχουν κάποια ανοχή δεν φτάνουν στην τελική τους διάσταση, αλλά μένουν σε μεγαλύτερη διάμετρο ή σε μικρότερη ανάλογα με τον εάν η ανοχή αυτή είναι σε εξωτερική ή σε εσωτερική κατεργασία. Για παράδειγμα, εάν ο κατασκευαστής πρέπει να ταιριάξει μία διάμετρο στα 20mm, θα την φέρει στα 20.2mm και στη συνέχεια θα προχωρήσει στην τελική διάμετρο που

απαιτεί ακρίβεια εκατοστού του χιλιοστού. Αυτό συμβαίνει διότι όταν η εργαλειομηχανή προγραμματίζεται να μετακινηθεί σε μία συγκεκριμένη συντεταγμένη, δεν θα μεταφερθεί με ακρίβεια εκατοστού του χιλιοστού σε αυτή τη θέση, μπορεί για παράδειγμα να μετακινηθεί στα 20.03mm. Οπότε σε περίπτωση που η ανοχή για την συγκεκριμένη διάσταση είναι στα 20mm (-0.01 έως -0.03) θα πρέπει να γίνει νέα κατεργασία. Επειδή λοιπόν ο κατασκευαστής δεν μπορεί να γνωρίζει από πριν σε ποια διάσταση ακριβώς θα μετακινηθεί η εργαλειομηχανή ακολουθεί την διαδικασία που περιγράψαμε, έρχεται σε μια διαφορετική διάμετρο και στην συνέχεια έρχεται στην τελική του εξαρτήματος.

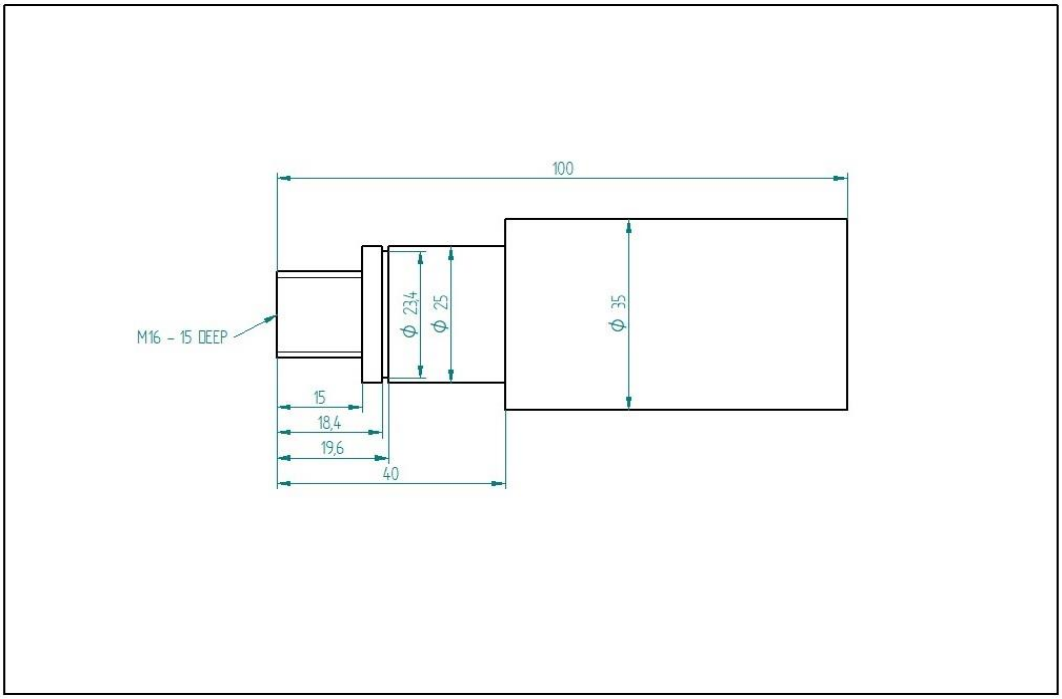
«Εικόνα άξονα με ανοχές»

Σε αυτή λοιπόν την περίπτωση ο αλγόριθμος θα μπορούσε να εξοικονομήσει χρόνο. Από την στιγμή λοιπόν που ο χρήστης έχει τελειώσει με τον σχεδιασμό του άξονα, θα μπορούσε να κάνει ένα αντίγραφο του σχεδίου στο οποίο με μερικές ενέργειες ακόμη θα μπορούσε να αλλάξει τις διαμέτρους στις οποίες χρειάζεται ανοχή. Έτσι, ο προγραμματιστής της μηχανής θα είχε να προγραμματίσει μόνο τις κινήσεις για το ταίριασμα των διαμέτρων με τις ανοχές, καθώς όλο το υπόλοιπο πρόγραμμα του άξονα θα έβγαινε έτοιμο από τον αλγόριθμο.

Οπότε, μπορούμε ουσιαστικά να μιλήσουμε για ένα εργαλείο το οποίο μπορεί να εξοικονομήσει αρκετό χρόνο στην καθημερινή λειτουργία ενός μηχανουργείου καθώς επίσης να μειώσει και την πιθανότητα λάθους στον προγραμματισμό της εργαλειομηχανής, φαινόμενο το οποίο είναι αρκετά συχνό όταν ο προγραμματισμός γίνεται από τον χρήστη.

5.2 Προτάσεις Βελτίωσης

Όπως αναφέρθηκε προηγουμένως, ο αλγόριθμος στην παρούσα του μορφή είναι ιδανικός για απλά εξαρτήματα τόρνευσης καθώς και για εξαρτήματα με ακρίβειες τα οποία όμως αποτελούνται από απλές γεωμετρικές μορφές. Τα εξαρτήματα που δεν μπορούν να καλυφθούν από τον αλγόριθμο είναι αυτά που περιέχουν πιο πολύπλοκη γεωμετρία, όπως για παράδειγμα ένα εξάρτημα που περιέχει ένα αυλάκι για ασφάλεια, ή ένα μεγάλο ‘ξεμπουκάρισμα’ ανάμεσα σε δύο διαμέτρους εσωτερικής κατεργασίας ή ακόμη και ένα σπείρωμα.

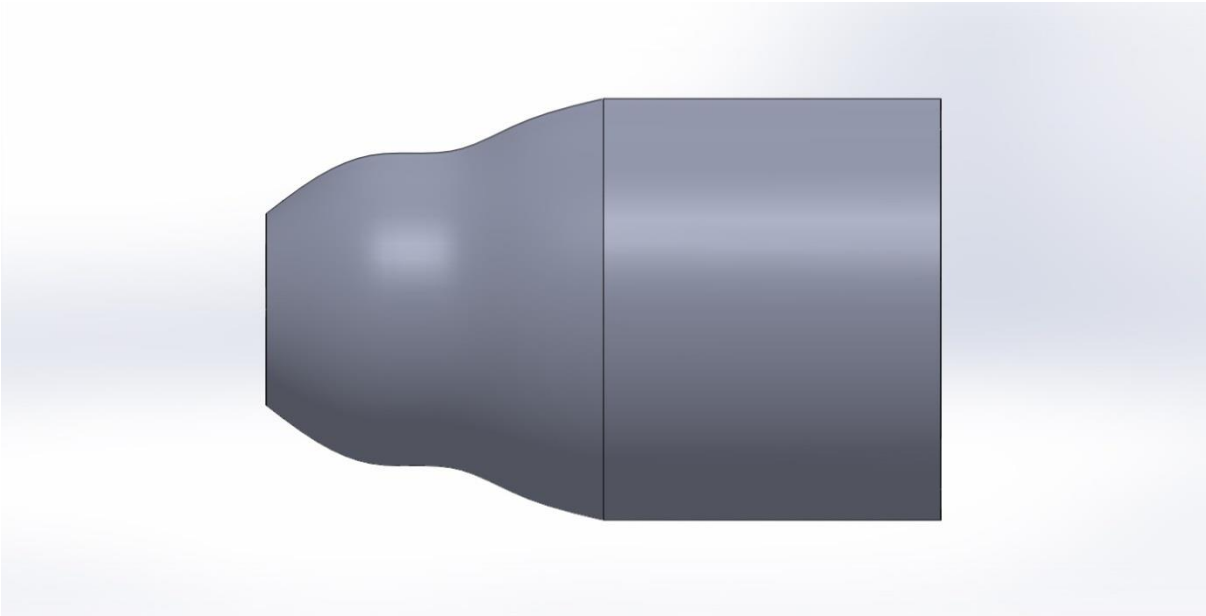


Εικόνα 5.3 "Παράδειγμα Αντικειμένου με Κάθετη Εγκοπή"

Οπότε, σαν πρώτο στάδιο βελτίωσης του αλγορίθμου θα μπορούσε να ήταν η προσθήκη αυτών των μορφολογικών χαρακτηριστικών στις δυνατότητές του. Έτσι ο αλγόριθμος θα ήταν σε θέση να εκτελέσει και άλλης φύσης κατεργασίες σε σύγκριση με τον κύκλο G71 που χρησιμοποιήθηκε, ο οποίος χρησιμοποιεί κινήσεις κατά τον άξονα Z για να φτάσει στην τελική διαμόρφωση που απαιτείται. Για παράδειγμα, όταν πρέπει να διαμορφωθεί ένα αυλάκι για ασφάλεια η πιο κατάλληλη κατεργασία για αυτού του είδους το μορφολογικό χαρακτηριστικό είναι κινήσεις κατά τον X άξονα, που συνεπάγεται την χρήση διαφορετικού κύκλου καθώς επίσης θα χρειάζεται να προσδιοριστεί και διαφορετικά η γεωμετρία του χαρακτηριστικού αυτού. Διαφορετικός κύκλος επίσης θα μπορούσε να χρησιμοποιηθεί και στην περίπτωση του σπειρώματος, όπου εκεί θα χρειαστούν να προσδιοριστούν κι άλλες μεταβλητές, όπως για παράδειγμα το βήμα του σπειρώματος, η διάμετρος του πυρήνα και άλλα.

Μία ακόμη βελτίωση που θα μπορούσε να γίνει στον αλγόριθμο θα ήταν και η προσθήκη στα μορφολογικά χαρακτηριστικά ενός προφίλ το οποίο να αποτελείται καμπύλες, όπως B-Splines. Σε αυτή την περίπτωση θα ήταν πολύ ενδιαφέρον ο τρόπος προσέγγισης που θα πρέπει να αναπτυχθεί στον αλγόριθμο για τον προσδιορισμό των συντεταγμένων του προφίλ καθώς επίσης και για το εάν θα μπορεί να χρησιμοποιηθεί κάποιος κύκλος ή εάν θα πρέπει να γίνει κατεύθυνση του εργαλείου σημείο προς σημείο για όλο το φάσμα της κατεργασίας του ξεχονδρίσματος.

Ακόμη, μία περαιτέρω βελτίωση που θα μπορούσε να εφαρμοστεί στον συγκεκριμένο αλγόριθμο θα μπορούσε να ήταν η προσθήκη περισσοτέρων επιλογών σχετικά με τον τρόπο κατεργασίας του αντικειμένου. Τέτοιες επιλογές για παράδειγμα θα μπορούσε να ήταν η προσθήκη επιλογής ή όχι της



Εικόνα 5.4 "Παράδειγμα Αντικειμένου με Προφίλ από Καμπύλες"

ταχύτητας κοπής (Constant Surface Speed), ένα είδος κατεργασίας το οποίο χρησιμοποιείται κατά κόρον στην τόρνευση. Μία άλλη προσθήκη που θα μπορούσε να βελτιώσει τον αλγόριθμο θα ήταν η επιλογή της μορφής του εργαλείου που καλείται να εκτελέσει την κατεργασία, εάν δηλαδή το κοπτικό εργαλείο είναι τριγωνικής μορφής ή ρόμβος 80 μοιρών ή 55 μοιρών ή ακόμη και κύκλος. Σε πολλές περιπτώσεις, ειδικά με την χρήση του κύκλου G71, η επιλογή αυτή δεν φέρει αλλαγές στον τρόπο κίνησης του εργαλείου που είναι και το ζητούμενο από τον αλγόριθμο. Έχει όμως μεγάλη διαφορά στην φύση της κατεργασίας και στο τελικό αποτέλεσμα του εξαρτήματος και θα ήταν θετικό στοιχείο η ύπαρξη αυτής της επιλογής από τον χρήστη καθώς θα έχει τη δυνατότητα να δώσει συγκεκριμένες οδηγίες για το ποιο εργαλείο θα πρέπει να χρησιμοποιηθεί από τον κατασκευαστή για τις εκάστοτε κατεργασίες.

Τέλος, καταλήγουμε στο συμπέρασμα ότι ο αλγόριθμος που παρουσιάστηκε σε αυτήν την εργασία είναι ένα πολύ χρήσιμο εργαλείο που θα μπορούσε να εφαρμοστεί στην καθημερινότητα ενός μηχανουργείου, καθώς εξοικονομεί αρκετό χρόνο στην παραγωγή ενός εξαρτήματος τόρνευσης και επίσης ελαχιστοποιεί τις πιθανότητες λάθους στην λειτουργία των εργαλειομηχανών. Παρατηρείται επίσης ότι υπάρχει αρκετό περιθώριο βελτίωσης στο εύρος των αναγκών που μπορεί να καλύψει ο συγκεκριμένος αλγόριθμος καθώς επίσης και στην ελευθερία της παραμετροποίησης των κατεργασιών που παρέχει στον χρήστη. Όλες οι παραπάνω βελτιώσεις αποτελούν κινήσεις που σε περίπτωση εφαρμογής του στο μέλλον θα μπορούμε να μιλάμε για έναν εντελώς καινούριο και επαναστατικό τρόπο λειτουργίας του μηχανουργείου και της επικοινωνίας γενικότερα μεταξύ του μηχανολόγου και του κατασκευαστή.

Κεφάλαιο 6: Βιβλιογραφία

[1] Anderson, D.C. and Chang, T.C., *"Automated process planning using object-oriented feature-based design",* Advanced Geometric Modelling for Engineering Applications, pp 247-260, 1990

[2] Alberto J. Alvares, Joao Carlos Espindola Ferreira, *"A system for the design and manufacture of feature-based parts through the Internet",* International Journal of Advanced Manufacturing Technology, Vol.35, pp 646-664, 2008

[3] Alting L, Zhang H, *"Computer aided process planning: The state-of-the-art survey"*, International Journal of Production Research, pp 553–85, 1989

[4] Bronsvoort, W. F., Noort, A., van den Berg, J. and Hoek, G. F. M., *"Product development with multipleview feature modelling"*, CD-ROM Proceedings of the IFIP Conference on Feature Modelling and Advanced Design-For-The-Life-Cycle Systems, pp 12-14, 2001

[5] Chen, X. and Hoffmann, C. M., *"On editability of feature-based design. Computer-Aided Design",* pp 905-914, 1995

[6] Christoph M. Hoffmann, *"Geometric and Solid Modeling"*, Morgan Kaufmann Publishers, San Francisco, CA, 1989

[7] Chwastyk, P & Kolosowski M, *"Integration CAD/CAPP/CAM systems in design process of innovative products"*, Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, Volume 23, pp 397-400, 2012

[8] Chwastyk P, Kolosowski M, *"Choice of Variant of Manufacturing Process with Estimation of Cost Production",* Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Editor B. Katalinic, pp 511-512, 2011

[9] Cicirello V, Regli W C, *"An approach to feature based comparison of solid models of machined parts",* Journal of Artificial Intelligence for engineering design analysis and manufacturing, Vol. 16, pp 385-399, 2002

[10] Colin B. Besant, *"Computer-Aided Design and Manufacture"*, 3rd ed. Ellis Horwood Ltd, 1986

[11] Febransyah, A., *"A feature-based approach to automating high-level process planning"*, PhD thesis, North Carolina State University, 2001

[12] Feng SC and Song EY, *"A manufacturing process information model for design and process planning integration",* Journal of Manufacturing Systems, pp 1-15, 2003

[13] H. J. de St. Germain, S. R. Stark, W. B. Thompson, and T. C. Henderson, *"Constraint optimization and feature-based model construction for reverse engineering"*, Proceedings of the ARPA Image Understanding Workshop, 1996

[14] Han, J. H., Pratt, M. and Regli, W. C, *"Manufacturing feature recognition from solid models: a status report"*, IEEE Transactions on Robotics and Automation, pp 782-796, 2000

[15] Huifen, W., Youliang, Z., Jian, C., Lee, S. F. and Kwong, W. C., *"Feature-based collaborative design",* Journal of Materials Processing Technology, pp 613-618, 2003

[16] Ismail N, Abu Bakar N, Juri AH, *"Recognition of cylindrical based features using edge boundary technique for integrated manufacturing",* Robotics and Computer Integrated Manufacturing Vol. 20, pp 417-422, 2004

[17] Ismail N, Tan C F, Wong S V, Osman M R & Sulaiman S, *"Proc Student Conference an Research and Development",* pp 90-93, 2002

[18] J. Britanik and M. Marefat, *"Hierarchical plan merging with applications to process planning"*, International Joint Conference on Artificial Intelligence, 1995

[19] Jung Hyun Han, Inho Han and Juneho Yi, *"Manufacturing feature recognition toward integration with process planning"*, IEEE Transactions on Systems, Man and Cybernetics – part B: Cybernetics, pp 373-380, 2001

[20] K. Lee, *"Principles of CAD/CAM/CAE systems"*, Addison-Wesley, 1999

[21] Karnik M V, Gupta S K, Magrab EB, *"Geometric Algorithms for containment analysis of rotational parts",* Computer aided design, Vol. 37, pp 213-230, 2005

[22] Kim YS, Wang E., *"Recognition of machining features for cast then machined parts",* Computer-Aided Design, pp 71–87, 2002

[23] KwangHoon Lee, Chris A. McMahon and Kwan H. Lee, *"Design of a Feature-based Multi-viewpoint Design Automation System"*, International Journal of CAD/CAM Vol. 3, No. 2, pp. 67-75, 2003

[24] Li B, Liu J., *"Detail feature recognition and decomposition in solid model",* Computer-Aided Design, pp 405–414, 2002

[25] Madsen D A, Madsen P A, *"Engineering Drawing & Design",* Clifton Park, 2012

[26] Miao H, Sridharan N, Shah JJ, *"CAD–CAM integration using machining features",* International Journal of Computer Integrated Manufacturing, pp 296-318, 2002

[27] Omid F V & Mahmoud H, *"Int J Comput Integr Manuf",* pp 1095-1117, 2010

[28] Otto. W. Salomons, *"Computer Support in the Design of Mechanical Products: Constraint specification and satisfaction in feature based design for manufacturing" Ph.D. thesis,* University of Twente, Netherlands, 1995

[29] Park SC., *"Knowledge capturing methodology in process planning",* Computer-Aided Design, pp 1109–1117, 2003

[30] Pratt MJ, Anderson BD, Ranger T., *"Towards the standardized exchange of parameterized feature-based CAD models",* Computer-Aided Design, pp 1251–1265, 2005

[31] Pratt, M. J. and Anderson, B. D., *"A shape modelling applications programming interface for the STEP standard,"* Computer-Aided Design, pp 531-543, 2001

[32] S. K. Gupta and D. S. Nau, *"A systematic approach for analyzing the manufacturability of machined parts"*, International Journal of Computer Aided Design, pp 323-342, 1995

[33] S. L. Brooks, K. E. Hummel, and M. L. Wolf, *"XCUT: a rule-based expert system for the automated process planning of machined parts"*, Proceedings of ASME Winter Annual Meeting, pp 181-194, 1997

[34] S. M. Amaitik and S. E. Kilic, *"STEP-based feature modeller for computer-aided process planning"*, International Journal of Production Research, 2005

[35] S. Sivakumar & V. Dhanalakshmi, *"A feature-based system for CAD/CAM integration through STEP ile for cylindrical parts",* Indian Journal of Engineering & Material Sciences Vol 20, pp 21-26, 2013

[36] Shah, J. J. and Mantyla, M, *"Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications",* John Wiley and Sons, 1995

[37] Shah, J.J. and Rogers, M.T., *"Expert form feature modelling shel"l*, Comput.-Aided Des, pp515-524, 1988

[38] Shakeri M., *"Implementation of an automated operation planning and optimum operation sequencing and tool selection algorithms",* Computers in Industry, pp 223–236, 2004

[39] Somashekar Subrahmanyam and Michael Wozny, *"An overview of automatic feature recognition techniques for computer-aided process planning"*, Computers in Industry, pp 1-21, 1995

[40] Stefano PD, Bianconi F, Angelo LD, *"An approach for feature semantics recognition in geometric models",* Computer-Aided Design, pp 993–1009, 2004

[41] Sunil V B & Pande S S, *"Comp Aid Des",* pp 502-517, 2008

[42] Sunil V B, Agarwal R and Pande S S, *"Comput Ind",* pp 686-701, 2010

[43] Vamsi Krishna P, Shankar N.V.S. & Surendra Babu B, *"Feature based modeling and automated process plan generation for turning components",* Advances in Production Engineering & Management 6, pp 153-162, 2011

[44] Tien-Chien Chang, Richard A. Wysk and Hus-Pin Wang, *"Computer-Aided Manufacturing"*, 2nd ed. Prentice Hall, 1998

[45] W D Li, S K Ong and A Y C Nee, *"Recognizing manufacturing features from a designby-feature model",* Computer aided design Vol. 36, pp 849-868, 2002

[46] Wang F, Mills JJ, Devarajan V, *"A conceptual approach managing design resource",* Computer in Industry, pp 169–183, 2002

[47] Willian F. Gibbs, *"CAD/CAM integration – what's the right choice for you?"*, Modern Machine Shop, 1998

[48] Wong, T.N. and Wong, K.W., *"A feature-based design system for computer-aided process planning"*, J. Mater. Process. Tech, pp 2839-2862, 1995

[49] Xionghui Zhou, Yanjie Qiu, Guangru Hua, Huifeng Wang, Xueyu Ruan, *"A feasible approach to the integration of CAD and CAPP",* Computer-Aided Design 39, pp 324-338, 2007

[50] Yan, Yamazaki and Liu, *"Recognition of machining features and feature topologies from NC programs",* Computer aided design Vol. 32, pp 605-616, 2000

[51] Zhou Z D, Xie S Q & Wang W Z, *"Int J Comput Integr Manuf",* pp 43-61, 2008

Παράρτημα

```
Dim swApp As SldWorks.SldWorks
Dim swModel As SldWorks.ModelDoc2
Dim swModelEx As SldWorks.ModelDocExtension
Dim swPart As SldWorks.PartDoc
Dim swSelMgr As SldWorks.SelectionMgr
Dim swFeat As SldWorks.Feature
Dim swSubFeat As SldWorks.Feature
Dim swExtFeat As SldWorks.ExtrudeFeatureData2
Dim swDim As SldWorks.Dimension
Dim swChamFeat As SldWorks.ChamferFeatureData2
Dim swFillet As SldWorks.SimpleFilletFeatureData2
Dim swEdge As SldWorks.Edge
Dim swCurve As SldWorks.Curve
Dim fso As New FileSystemObject
Dim ts As TextStream

Dim intExtC As Integer
Dim intIntC As Integer
Dim intFilletC As Integer
Dim intChamferC As Integer
Dim LineC As Integer

Dim varSelectedFeatID As Variant
Dim varExtFeatID As Variant
Dim varIntFeatID As Variant
Dim varFilletID As Variant
Dim varChamferID As Variant
Dim varExtTop As Variant
Dim varIntTop As Variant
Dim vEdges As Variant
Dim varStockDFeat(0, 4) As Variant
Dim varStockSDFeat(0, 4) As Variant

Dim i As Integer
Dim u As Integer
Dim w As Integer
Dim j As Integer
Dim k As Integer
Dim l As Integer
Dim m As Integer

Dim StockD As Double
Dim StockSD As Double
Dim StockDepth As Double

Public StockType As String
Public dblRoughFeed As Double
Public dblFinishFeed As Double
Public dblRSpeed As Double
Public dblFSpeed As Double
Public dblFinishU As Double
Public dblFinishW As Double
Public dblRoughCut As Double
Public intExRTPos As Integer
Public intExFTPos As Integer
```

```
Public intInRTPos As Integer
Public intInFTPos As Integer

Public blnThreadCheck As Boolean
Public strThreadFeat As String
Public intThreadTool As Integer
Public intThreadCutSpeed As Integer
Public dblThreadX As Double
Public dblThreadZ As Double
Public dblThreadK As Double
Public dblThreadD As Double
Public dblThreadF As Double
Public intThreadA As Integer
Dim varThreadID As Variant
Dim intThreadIDCheck As Integer

Dim errorcode As Long
Dim intA As Integer
Dim intB As Integer
Dim temp As Variant
Dim edgenum As Integer
Dim bret As Boolean
Dim ChamfD As Double
Dim CenterZ As Double
Dim FilletR As Double
Dim FilletD As Double
Dim strPath As String
Dim ZPos As Double
Dim CycleT(99) As String
```

```
Sub main()

    Set swApp = Application.SldWorks
    Set swModel = swApp.ActiveDoc
    Set swSelMgr = swModel.SelectionManager
    Set swModelEx = swModel.Extension

    'Counters  Initilazation-------------------------------------------------------
------------
    '-------------------------------------------------------------------------------
------------
    intExtC = 0
    intIntC = 0
    intFilletC = 0
    intChamferC = 0
    '-------------------------------------------------------------------------------
------------

    'Matrix Size Determination-----------------------------------------------------
------------
    '-------------------------------------------------------------------------------
------------

    For i = 1 To swSelMgr.GetSelectedObjectCount2(-1)
        If swSelMgr.GetSelectedObjectType3(i, -1) = 22 Then
            Set swFeat = swSelMgr.GetSelectedObject6(i, -1)
            If swFeat.GetTypeName = "Cut" Then
                Set swExtFeat = swFeat.GetDefinition
                If swExtFeat.FlipSideToCut Then
                    intExtC = intExtC + 1
                Else
                    intIntC = intIntC + 1
                End If
            ElseIf swFeat.GetTypeName = "Fillet" Then
                intFilletC = intFilletC + 1
            ElseIf swFeat.GetTypeName = "Chamfer" Then
                intChamferC = intChamferC + 1
            End If
        Else
            MsgBox "Please select only Features"
            MsgBox "The Application will close and you have to restart it"
            End
        End If
    Next i

    If intExtC <> 0 Then
        ReDim varExtFeatID(intExtC - 1) As Variant
    End If
    If intIntC <> 0 Then
        ReDim varIntFeatID(intIntC - 1) As Variant
    End If
    If intFilletC <> 0 Then
        ReDim varFilletID(intFilletC - 1) As Variant
    End If
    If intChamferC <> 0 Then
        ReDim varChamferID(intChamferC - 1) As Variant
    End If
    ReDim varSelectedFeatID(intExtC + intIntC + intFilletC + intChamferC + 1) As
Variant
```

```
    '-------------------------------------------------------------------------------
----

    'Feature  Seperation-------------------------------------------------------------
----
    '-------------------------------------------------------------------------------
----
    j = 0           'position counter for external feature ID
    k = 0           'position counter for internal feature ID
    l = 0           'position counter fot fillet feature ID
    m = 0           'position counter for chamfer feature ID

    For i = 0 To 4
        varStockDFeat(0, i) = 0
        varStockSDFeat(0, i) = 0
    Next i

    frmStockSelection.Show

    For i = 1 To swSelMgr.GetSelectedObjectCount2(-1)
        Set swFeat = swSelMgr.GetSelectedObject6(i, -1)
        varSelectedFeatID(i - 1) = swModelEx.GetPersistReference3(swFeat)
        If swFeat.GetTypeName = "Cut" Then
            Set swExtFeat = swFeat.GetDefinition
            If swExtFeat.FlipSideToCut Then
                varExtFeatID(j) = swModelEx.GetPersistReference3(swFeat)

                j = j + 1
            Else
                varIntFeatID(k) = swModelEx.GetPersistReference3(swFeat)

                k = k + 1
            End If
        ElseIf swFeat.GetTypeName = "Fillet" Then
            varFilletID(l) = swModelEx.GetPersistReference3(swFeat)

            l = l + 1
        ElseIf swFeat.GetTypeName = "Chamfer" Then
            varChamferID(m) = swModelEx.GetPersistReference3(swFeat)

            m = m + 1
        ElseIf swFeat.GetTypeName = "Extrusion" Then
            Set swExtFeat = swFeat.GetDefinition
            StockDepth = swExtFeat.GetDepth(True) * 1000
            If StockType = "Stock" Then
                Set swSubFeat = swFeat.GetFirstSubFeature
                Set swDim = swSubFeat.Parameter("D1")
                varStockDFeat(0, 0) = swDim.Value
            ElseIf StockType = "Perforated Stock" Then
                Set swSubFeat = swFeat.GetFirstSubFeature
                Set swDim = swSubFeat.Parameter("D1")
                varStockDFeat(0, 0) = swDim.Value
                Set swDim = swSubFeat.Parameter("D2")
                varStockSDFeat(0, 0) = swDim.Value
            End If
        End If
    Next i
    '-------------------------------------------------------------------------------
------------
```

```
    'Sorting Arrays by Diameter-------------------------------------------------------
------------
    '-----------------------------------------------------------------------------
------------

    'External Feature Sort
    For i = 0 To j - 2
        For u = i + 1 To j - 1
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varExtFeatID(i),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intA = swDim.Value

            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varExtFeatID(u),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intB = swDim.Value

            If intA > intB Then
                temp = varExtFeatID(u)
                varExtFeatID(u) = varExtFeatID(i)
                varExtFeatID(i) = temp
            End If
        Next u
    Next i

    'Internal Feature Sort
    For i = 0 To k - 2
        For u = i + 1 To k - 1
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varIntFeatID(i),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intA = swDim.Value

            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varIntFeatID(u),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intB = swDim.Value

            If intA < intB Then
                temp = varIntFeatID(u)
                varIntFeatID(u) = varIntFeatID(i)
                varIntFeatID(i) = temp
            End If
        Next u
    Next i
    '-----------------------------------------------------------------------------
--------------------

    'Create topology arrays------------------------------------------------------
--------------------
    '-----------------------------------------------------------------------------
--------------------
```

```
    'Arrays Initianalization
    If j <> 0 Then
        ReDim varExtTop(UBound(varExtFeatID), 8) As Variant
        For i = 0 To UBound(varExtTop, 1)
            For u = 0 To UBound(varExtTop, 2)
                varExtTop(i, u) = 0
            Next u
        Next i
    End If

    If k <> 0 Then
        ReDim varIntTop(UBound(varIntFeatID), 8) As Variant
        For i = 0 To UBound(varIntTop, 1)
            For u = 0 To UBound(varIntTop, 2)
                varIntTop(i, u) = 0
            Next u
        Next i
    End If




    'External Feature Topology
    If j <> 0 Then
        For i = 0 To UBound(varExtFeatID)
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varExtFeatID(i),
errorcode)
            Set swExtFeat = swFeat.GetDefinition
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")

            varExtTop(i, 0) = swDim.Value
            varExtTop(i, 1) = swExtFeat.GetDepth(True) * 1000

            If m <> 0 Then
                For u = 0 To UBound(varChamferID)
                    Set                          swFeat                          =
swModelEx.GetObjectByPersistReference3(varChamferID(u), errorcode)
                    Set swChamFeat = swFeat.GetDefinition
                    bret = swChamFeat.AccessSelections(swModel, Nothing)
                    edgenum = swChamFeat.GetEdgeCount
                    vEdges = swChamFeat.Edges

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
                        ChamfD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If ChamfD = Round(varExtTop(i, 0), 3) Then
                            If CenterZ = -varExtTop(i, 1) Then
                                varExtTop(i,                  5)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                                varExtTop(i,                  6)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                            Else
                                varExtTop(i,                  2)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
```

```
                                varExtTop(i,                    3)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                            End If
                        ElseIf ChamfD = Round(varStockDFeat(0, 0), 3) Then
                            varStockDFeat(0,                    2)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                            varStockDFeat(0,                    3)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                        End If

                    Next w
                swChamFeat.ReleaseSelectionAccess
                Next u
            End If

            If l <> 0 Then
                For u = 0 To UBound(varFilletID)
                    Set                             swFeat                          =
swModelEx.GetObjectByPersistReference3(varFilletID(u), errorcode)
                    Set swFillet = swFeat.GetDefinition
                    bret = swFillet.AccessSelections(swModel, Nothing)
                    vEdges = swFillet.Edges
                    FilletR = swFillet.GetRadius(vEdges(0))

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
                        FilletD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If FilletD = Round(varExtTop(i, 0), 3) Then
                            If CenterZ = -Round(varExtTop(i, 1), 3) Then
                                varExtTop(i, 7) = FilletR * 1000
                            Else
                                varExtTop(i, 4) = FilletR * 1000
                            End If
                        ElseIf FilletD = Round(varStockDFeat(0, 0), 3) Then
                            varStockDFeat(0, 4) = FilletR * 1000
                        End If
                    Next w
                    swFillet.ReleaseSelectionAccess
                Next u
            End If

        Next i
    End If


    'Internal Feature Topology
    If k <> 0 Then
        For i = 0 To UBound(varIntFeatID)
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varIntFeatID(i),
errorcode)
            Set swExtFeat = swFeat.GetDefinition
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")

            varIntTop(i, 0) = swDim.Value
            varIntTop(i, 1) = swExtFeat.GetDepth(True) * 1000
```

```
            If m <> 0 Then
                For u = 0 To UBound(varChamferID)
                    Set                          swFeat                          =
swModelEx.GetObjectByPersistReference3(varChamferID(u), errorcode)
                    Set swChamFeat = swFeat.GetDefinition
                    bret = swChamFeat.AccessSelections(swModel, Nothing)
                    edgenum = swChamFeat.GetEdgeCount
                    vEdges = swChamFeat.Edges

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
                        ChamfD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If ChamfD = Round(varIntTop(i, 0), 3) Then
                            If CenterZ = -varIntTop(i, 1) Then
                                varIntTop(i,                  5)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                                varIntTop(i,                  6)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                            Else
                                varIntTop(i,                  2)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                                varIntTop(i,                  3)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                            End If
                        ElseIf ChamfD = Round(varStockSDFeat(0, 0), 3) Then
                            varStockSDFeat(0,                  2)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                            varStockSDFeat(0,                  3)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                        End If
                    Next w
                swChamFeat.ReleaseSelectionAccess
                Next u
            End If

            If l <> 0 Then
                For u = 0 To UBound(varFilletID)
                    Set                          swFeat                          =
swModelEx.GetObjectByPersistReference3(varFilletID(u), errorcode)
                    Set swFillet = swFeat.GetDefinition
                    bret = swFillet.AccessSelections(swModel, Nothing)
                    vEdges = swFillet.Edges
                    FilletR = swFillet.GetRadius(vEdges(0))

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
                        FilletD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If FilletD = Round(varIntTop(i, 0), 3) Then
                            If CenterZ = -varIntTop(i, 1) Then
                                varIntTop(i, 7) = FilletR * 1000
                            Else
                                varIntTop(i, 4) = FilletR * 1000
```

```
                            End If
                        ElseIf FilletD = Round(varStockSDFeat(0, 0), 3) Then
                            varStockSDFeat(0, 4) = FilletR * 1000
                        End If
                    Next w
                    swFillet.ReleaseSelectionAccess
                Next u
            End If

        Next i
    End If


    'Create 1st Phase G Code-------------------------------------------------------
-----------------------------
    '-----------------------------------------------------------------------------
-----------------------------

    frmCutConditions.Show

    'Set the file path for the txt file
    strPath = "C:\G-Code Phase 1.txt"

    If j <> 0 Or k <> 0 Then
        'Create the new txt file
        Set ts = fso.CreateTextFile(strPath, True)

        'Write in File
        ts.WriteLine "O0900"
        ts.WriteLine "G90"
        ts.WriteLine "G53 G0 X0 Z0"

        'External Process
        If j <> 0 Then

            ts.WriteLine "T0" & intExRTPos & "0" & intExRTPos
            ts.WriteLine "G42"
            ts.WriteLine "S" & dblRSpeed & " M3"
            ts.WriteLine "M8"
            ts.WriteLine "G0 X" & varStockDFeat(0, 0) & " Z1"

            LineC = 10
            ZPos = 0

            'Start of Cycle
            If varExtTop(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varExtTop(0, 0) - 2 *
varExtTop(0, 3)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) & " Z-" &
varExtTop(0, 2)
                LineC = LineC + 1
                If varExtTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
```

```
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) + 2 *
varExtTop(0, 5) & " Z-" & varExtTop(0, 1)
                    LineC = LineC + 1
                ElseIf varExtTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(0, 0) +
2 * varExtTop(0, 7) & " Z-" & varExtTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            ElseIf varExtTop(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varExtTop(0, 0) - 2 *
varExtTop(0, 4)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G3 X" & varExtTop(0, 0) & " Z-
" & varExtTop(0, 4) & " R" & varExtTop(0, 4)
                LineC = LineC + 1
                If varExtTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) + 2 *
varExtTop(0, 5) & " Z-" & varExtTop(0, 1)
                    LineC = LineC + 1
                ElseIf varExtTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(0, 0) +
2 * varExtTop(0, 7) & " Z-" & varExtTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            Else
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varExtTop(0, 0)
                LineC = LineC + 1
                If varExtTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) + 2 *
varExtTop(0, 5) & " Z-" & varExtTop(0, 1)
                    LineC = LineC + 1
                ElseIf varExtTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(0, 0) +
2 * varExtTop(0, 7) & " Z-" & varExtTop(0, 1) & " F" & dblFinishFeed
```

```
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            End If

            'Cycle Continue
            For i = 1 To UBound(varExtTop, 1)
                If varExtTop(i, 2) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varExtTop(i, 0) -
2 * varExtTop(i, 2) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) & "
Z-" & varExtTop(i, 3) + varExtTop(i - 1, 1)
                    LineC = LineC + 1
                    If varExtTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) +
2 * varExtTop(i, 5) & " Z-" & varExtTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(i,
0) + 2 * varExtTop(i, 7) & " Z-" & varExtTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                ElseIf varExtTop(i, 4) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varExtTop(i, 0) -
2 * varExtTop(i, 4) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varExtTop(i, 0) &
" Z-" & varExtTop(i, 4) + varExtTop(i - 1, 1) & " R" & varExtTop(i, 4) & " F" &
dblFinishFeed
                    LineC = LineC + 1
                    If varExtTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) +
2 * varExtTop(i, 5) & " Z-" & varExtTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(i,
0) + 2 * varExtTop(i, 7) & " Z-" & varExtTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
```

```
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varExtTop(i, 0) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                    If varExtTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) +
2 * varExtTop(i, 5) & " Z-" & varExtTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(i,
0) + 2 * varExtTop(i, 7) & " Z-" & varExtTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                End If
            Next i

            'End of Cycle
            If varStockDFeat(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockDFeat(0, 0) -
2 * varStockDFeat(0, 2) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varStockDFeat(0, 0) & "
Z-" & varStockDFeat(0, 3) + varExtTop(UBound(varExtTop, 1), 1)
                CycleT(LineC - 9) = "G0 X" & varStockDFeat(0, 0) & " Z1"
            ElseIf varStockDFeat(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockDFeat(0, 0) -
2 * varStockDFeat(0, 4) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G3 X" & varStockDFeat(0, 0) &
" Z-" & varStockDFeat(0, 4) + varExtTop(UBound(varExtTop, 1), 1) & " R" &
varStockDFeat(0, 4) & " F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockDFeat(0, 0) & " Z1"
            Else
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockDFeat(0, 0) &
" F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockDFeat(0, 0) & " Z1"
            End If

            'Write Cycle in Text File
            ts.WriteLine "G71 P10 Q" & LineC & " U" & dblFinishU & " W" &
dblFinishW & " D" & dblRoughCut & " F" & dblRoughFeed

            For i = 0 To LineC - 9
                ts.WriteLine CycleT(i)
            Next i
```

```
            'Finish Cycle
            If intExFTPos <> 0 Then
                ts.WriteLine "M9"
                ts.WriteLine "G53 G0 X0 Z0"
                ts.WriteLine "G40"
                ts.WriteLine "T0" & intExRTPos & "00"
                ts.WriteLine "T0" & intExFTPos & "0" & intExFTPos
                ts.WriteLine "G42"
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "M8"
                ts.WriteLine "G0 X" & varStockDFeat(0, 0) & " Z1"
                ts.WriteLine "G70 P10 Q" & LineC
            Else
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "G70 P10 Q" & LineC
            End If

            'End of Tool 1
            ts.WriteLine "M9"
            ts.WriteLine "G53 G0 X0 Z0"

            If intExFTPos <> 0 Then
                ts.WriteLine "T0" & intExFTPos & "00"
            Else
                ts.WriteLine "T0" & intExRTPos & "00"
            End If
            ts.WriteLine "G40"
        End If

        'Internal Process
        If k <> 0 Then

            ts.WriteLine "T0" & intInRTPos & "0" & intInRTPos
            ts.WriteLine "G41"
            If varStockSDFeat(0, 0) = 0 Then
                varStockSDFeat(0, 0) = varIntTop(k - 1, 0)
            End If

            ts.WriteLine "S" & dblRSpeed & " M3"
            ts.WriteLine "M8"
            ts.WriteLine "G0 X" & varStockSDFeat(0, 0) & " Z1"

            LineC = 10
            ZPos = 0

            'Start of Cycle
            If varIntTop(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varIntTop(0, 0) + 2 *
varIntTop(0, 3)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) & " Z-" &
varIntTop(0, 2)
                LineC = LineC + 1
                If varIntTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 6) & " F" & dblFinishFeed
```

```
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) - 2 *
varIntTop(0, 5) & " Z-" & varIntTop(0, 1)
                    LineC = LineC + 1
                ElseIf varIntTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(0, 0) -
2 * varIntTop(0, 7) & " Z-" & varIntTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            ElseIf varIntTop(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varIntTop(0, 0) + 2 *
varIntTop(0, 4)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G2 X" & varIntTop(0, 0) & " Z-
" & varIntTop(0, 4) & " R" & varIntTop(0, 4) & " F" & dblFinishFeed
                LineC = LineC + 1
                If varIntTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) - 2 *
varIntTop(0, 5) & " Z-" & varIntTop(0, 1)
                    LineC = LineC + 1
                ElseIf varIntTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(0, 0) -
2 * varIntTop(0, 7) & " Z-" & varIntTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            Else
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varIntTop(0, 0)
                LineC = LineC + 1
                If varIntTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) - 2 *
varIntTop(0, 5) & " Z-" & varIntTop(0, 1)
                    LineC = LineC + 1
                ElseIf varIntTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
```

```
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(0, 0) -
2 * varIntTop(0, 7) & " Z-" & varIntTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            End If

            'Cycle Continue
            For i = 1 To UBound(varIntTop, 1)
                If varIntTop(i, 2) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varIntTop(i, 0) +
2 * varIntTop(i, 2) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) & "
Z-" & varIntTop(i, 3) + varIntTop(i - 1, 1)
                    LineC = LineC + 1
                    If varIntTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) -
2 * varIntTop(i, 5) & " Z-" & varIntTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(i,
0) - 2 * varIntTop(i, 7) & " Z-" & varIntTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                ElseIf varIntTop(i, 4) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varIntTop(i, 0) +
2 * varIntTop(i, 4) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varIntTop(i, 0) &
" Z-" & varIntTop(i, 4) + varIntTop(i - 1, 1) & " R" & varIntTop(i, 4) & " F" &
dblFinishFeed
                    LineC = LineC + 1
                    If varIntTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) -
2 * varIntTop(i, 5) & " Z-" & varIntTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(i,
0) - 2 * varIntTop(i, 7) & " Z-" & varIntTop(i, 1) & " F" & dblFinishFeed
```

```
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varIntTop(i, 0) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                    If varIntTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) -
2 * varIntTop(i, 5) & " Z-" & varIntTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(i,
0) - 2 * varIntTop(i, 7) & " Z-" & varIntTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                End If
            Next i

            'End of Cycle

            If varStockSDFeat(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockSDFeat(0, 0) +
2 * varStockSDFeat(0, 2) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varStockSDFeat(0, 0) & "
Z-" & varStockSDFeat(0, 3) + varIntTop(UBound(varIntTop, 1), 1)
                CycleT(LineC - 9) = "G0 X" & varStockSDFeat(0, 0) & " Z1"
            ElseIf varStockSDFeat(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockSDFeat(0, 0) +
2 * varStockSDFeat(0, 4) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G2 X" & varStockSDFeat(0, 0) &
" Z-" & varStockSDFeat(0, 4) + varIntTop(UBound(varIntTop, 1), 1) & " R" &
varStockSDFeat(0, 4) & " F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockSDFeat(0, 0) & " Z1"
            Else
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockSDFeat(0, 0) &
" F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockSDFeat(0, 0) & " Z1"
            End If


            'Write Cycle in Text File
            ts.WriteLine "G71 P10 Q" & LineC & " U-" & dblFinishU & " W" &
dblFinishW & " D" & dblRoughCut & " F" & dblRoughFeed
```

```
            For i = 0 To LineC - 9
                ts.WriteLine CycleT(i)
            Next i

            'Finish Cycle
            If intInFTPos <> 0 Then
                ts.WriteLine "M9"
                ts.WriteLine "G53 G0 X0 Z0"
                ts.WriteLine "G40"
                ts.WriteLine "T0" & intInRTPos & "00"
                ts.WriteLine "T0" & intInFTPos & "0" & intInFTPos
                ts.WriteLine "G41"
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "M8"
                ts.WriteLine "G0 X" & varStockSDFeat(0, 0) & " Z1"
                ts.WriteLine "G70 P10 Q" & LineC
            Else
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "G70 P10 Q" & LineC
            End If

            'End of Tool 2 and of Phase 1
            ts.WriteLine "M9"
            ts.WriteLine "G53 G0 X0 Z0"
            If intInFTPos <> 0 Then
                ts.WriteLine "T0" & intInFTPos & "00"
            Else
                ts.WriteLine "T0" & intInRTPos & "00"
            End If
            ts.WriteLine "G40"

        End If


        ts.WriteLine "M30"
        ts.Close
    End If

    '2nd  Phase-----------------------------------------------------------------------
---------------------
    '-------------------------------------------------------------------------------
---------------------
    Dim var2ndPhaseID(99) As Variant
    Dim found As Boolean
    Dim a As Integer
    Dim strPossibleID As String
    Dim IDcheck As Integer

    Set swFeat = swModel.FirstFeature
    a = 0
    found = False

    'Counters  Initilazation-----------------------------------------------------------
------------
    '-------------------------------------------------------------------------------
------------
    intExtC = 0
    intIntC = 0
```

```
    intFilletC = 0
    intChamferC = 0

    For i = 1 To 4
        varStockDFeat(0, i) = 0
        varStockSDFeat(0, i) = 0
    Next i

    '-------------------------------------------------------------------------------
------------


    While Not swFeat Is Nothing
        strPossibleID = swModelEx.GetPersistReference3(swFeat)
        For i = 0 To UBound(varSelectedFeatID)
            IDcheck                                                              =
swModelEx.IsSamePersistentID(swModelEx.GetPersistReference3(swFeat),
varSelectedFeatID(i))
            If IDcheck = 1 Then
                found = True
            End If
        Next i

        If found = False Then
            If swFeat.GetTypeName = "Cut" Then
                Set swExtFeat = swFeat.GetDefinition
                If swExtFeat.FlipSideToCut Then
                    intExtC = intExtC + 1
                Else
                    intIntC = intIntC + 1
                End If
                var2ndPhaseID(a) = swModelEx.GetPersistReference3(swFeat)
                a = a + 1
            ElseIf swFeat.GetTypeName = "Fillet" Then
                intFilletC = intFilletC + 1
                var2ndPhaseID(a) = swModelEx.GetPersistReference3(swFeat)
                a = a + 1
            ElseIf swFeat.GetTypeName = "Chamfer" Then
                intChamferC = intChamferC + 1
                var2ndPhaseID(a) = swModelEx.GetPersistReference3(swFeat)
                a = a + 1
            End If

        Else: found = False
        End If

        Set swFeat = swFeat.GetNextFeature
    Wend

    If intExtC <> 0 Then
        ReDim varExtFeatID(intExtC - 1) As Variant
    End If
    If intIntC <> 0 Then
        ReDim varIntFeatID(intIntC - 1) As Variant
    End If
    If intFilletC <> 0 Then
        ReDim varFilletID(intFilletC - 1) As Variant
    End If
    If intChamferC <> 0 Then
```

```
        ReDim varChamferID(intChamferC - 1) As Variant
    End If



    'Feature  Seperation-------------------------------------------------------------
----
    '---------------------------------------------------------------------------
----
    j = 0           'position counter for external feature ID
    k = 0           'position counter for internal feature ID
    l = 0           'position counter fot fillet feature ID
    m = 0           'position counter for chamfer feature ID

    For i = 0 To a - 1
        Set   swFeat   =   swModelEx.GetObjectByPersistReference3(var2ndPhaseID(i),
errorcode)
        If swFeat.GetTypeName = "Cut" Then
            Set swExtFeat = swFeat.GetDefinition
            If swExtFeat.FlipSideToCut Then
                varExtFeatID(j) = swModelEx.GetPersistReference3(swFeat)
                j = j + 1
            Else
                varIntFeatID(k) = swModelEx.GetPersistReference3(swFeat)
                k = k + 1
            End If
        ElseIf swFeat.GetTypeName = "Fillet" Then
            varFilletID(l) = swModelEx.GetPersistReference3(swFeat)
            l = l + 1
        ElseIf swFeat.GetTypeName = "Chamfer" Then
            varChamferID(m) = swModelEx.GetPersistReference3(swFeat)
            m = m + 1
        End If
    Next i

    'Sorting Arrays by Diameter------------------------------------------------------
------------
    '---------------------------------------------------------------------------
------------

    'External Feature Sort
    For i = 0 To j - 2
        For u = i + 1 To j - 1
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varExtFeatID(i),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intA = swDim.Value

            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varExtFeatID(u),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intB = swDim.Value

            If intA > intB Then
                temp = varExtFeatID(u)
                varExtFeatID(u) = varExtFeatID(i)
                varExtFeatID(i) = temp
```

```
            End If
        Next u
    Next i

    'Internal Feature Sort
    For i = 0 To k - 2
        For u = i + 1 To k - 1
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varIntFeatID(i),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intA = swDim.Value

            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varIntFeatID(u),
errorcode)
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")
            intB = swDim.Value

            If intA < intB Then
                temp = varIntFeatID(u)
                varIntFeatID(u) = varIntFeatID(i)
                varIntFeatID(i) = temp
            End If
        Next u
    Next i
    '-------------------------------------------------------------------------------
--------------------


    'Create topology arrays-------------------------------------------------------
--------------------
    '-------------------------------------------------------------------------------
--------------------

    'Arrays Initianalization
    If j <> 0 Then
        ReDim varExtTop(UBound(varExtFeatID), 8) As Variant
        For i = 0 To UBound(varExtTop, 1)
            For u = 0 To UBound(varExtTop, 2)
                varExtTop(i, u) = 0
            Next u
        Next i
    End If

    If k <> 0 Then
        ReDim varIntTop(UBound(varIntFeatID), 8) As Variant
        For i = 0 To UBound(varIntTop, 1)
            For u = 0 To UBound(varIntTop, 2)
                varIntTop(i, u) = 0
            Next u
        Next i
    End If


    'External Feature Topology
    If j <> 0 Then
        For i = 0 To UBound(varExtFeatID)
```

```
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varExtFeatID(i),
errorcode)
            Set swExtFeat = swFeat.GetDefinition
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")

            varExtTop(i, 0) = swDim.Value
            varExtTop(i, 1) = swExtFeat.GetDepth(True) * 1000

            If m <> 0 Then
                For u = 0 To UBound(varChamferID)
                    Set                          swFeat                          =
swModelEx.GetObjectByPersistReference3(varChamferID(u), errorcode)
                    Set swChamFeat = swFeat.GetDefinition
                    bret = swChamFeat.AccessSelections(swModel, Nothing)
                    edgenum = swChamFeat.GetEdgeCount
                    vEdges = swChamFeat.Edges

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
                        ChamfD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If ChamfD = Round(varExtTop(i, 0), 3) Then
                            If CenterZ = -(StockDepth - varExtTop(i, 1)) Then
                                varExtTop(i,                  5)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                                varExtTop(i,                  6)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                            Else
                                varExtTop(i,                  2)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                                varExtTop(i,                  3)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                            End If
                        ElseIf ChamfD = Round(varStockDFeat(0, 0), 3) Then
                            varStockDFeat(0,                  2)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                            varStockDFeat(0,                  3)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                        End If
                    Next w
                swChamFeat.ReleaseSelectionAccess
                Next u
            End If

            If l <> 0 Then
                For u = 0 To UBound(varFilletID)
                    Set                          swFeat                          =
swModelEx.GetObjectByPersistReference3(varFilletID(u), errorcode)
                    Set swFillet = swFeat.GetDefinition
                    bret = swFillet.AccessSelections(swModel, Nothing)
                    vEdges = swFillet.Edges
                    FilletR = swFillet.GetRadius(vEdges(0))

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
```

```
                        FilletD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If FilletD = Round(varExtTop(i, 0), 3) Then
                            If CenterZ = -(StockDepth - varExtTop(i, 1)) Then
                                varExtTop(i, 7) = FilletR * 1000
                            Else
                                varExtTop(i, 4) = FilletR * 1000
                            End If
                        ElseIf FilletD = Round(varStockDFeat(0, 0), 3) Then
                            varStockDFeat(0, 4) = FilletR * 1000
                        End If
                    Next w
                    swFillet.ReleaseSelectionAccess
                Next u
            End If

        Next i
    End If

    'Internal Feature Topology
    If k <> 0 Then
        For i = 0 To UBound(varIntFeatID)
            Set  swFeat  =  swModelEx.GetObjectByPersistReference3(varIntFeatID(i),
errorcode)
            Set swExtFeat = swFeat.GetDefinition
            Set swSubFeat = swFeat.GetFirstSubFeature
            Set swDim = swSubFeat.Parameter("D1")

            varIntTop(i, 0) = swDim.Value
            varIntTop(i, 1) = swExtFeat.GetDepth(True) * 1000

            If m <> 0 Then
                For u = 0 To UBound(varChamferID)
                    Set                          swFeat                          =
swModelEx.GetObjectByPersistReference3(varChamferID(u), errorcode)
                    Set swChamFeat = swFeat.GetDefinition
                    bret = swChamFeat.AccessSelections(swModel, Nothing)
                    edgenum = swChamFeat.GetEdgeCount
                    vEdges = swChamFeat.Edges

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
                        ChamfD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If ChamfD = Round(varIntTop(i, 0), 3) Then
                            If CenterZ = -(StockDepth - varIntTop(i, 1)) Then
                                varIntTop(i,                  5)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                                varIntTop(i,                  6)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                            Else
                                varIntTop(i,                  2)                  =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                                varIntTop(i,                  3)                  =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                            End If
```

```
                        ElseIf ChamfD = Round(varStockSDFeat(0, 0), 3) Then
                            varStockSDFeat(0,                 2)                 =
swChamFeat.GetEdgeChamferDistance(1) * 1000
                            varStockSDFeat(0,                 3)                 =
swChamFeat.GetEdgeChamferDistance(0) * 1000
                        End If
                    Next w
                swChamFeat.ReleaseSelectionAccess
                Next u
            End If

            If l <> 0 Then
                For u = 0 To UBound(varFilletID)
                    Set                          swFeat                          =
swModelEx.GetObjectByPersistReference3(varFilletID(u), errorcode)
                    Set swFillet = swFeat.GetDefinition
                    bret = swFillet.AccessSelections(swModel, Nothing)
                    vEdges = swFillet.Edges
                    FilletR = swFillet.GetRadius(vEdges(0))

                    For w = 0 To UBound(vEdges)
                        Set swEdge = vEdges(w)
                        Set swCurve = swEdge.GetCurve
                        FilletD = Round(swCurve.CircleParams(6) * 2000, 3)
                        CenterZ = Round(swCurve.CircleParams(2) * 1000, 3)

                        If FilletD = Round(varIntTop(i, 0), 3) Then
                            If CenterZ = -(StockDepth - varIntTop(i, 1)) Then
                                varIntTop(i, 7) = FilletR * 1000
                            Else
                                varIntTop(i, 4) = FilletR * 1000
                            End If
                        ElseIf FilletD = Round(varStockSDFeat(0, 0), 3) Then
                            varStockSDFeat(0, 4) = FilletR * 1000
                        End If
                    Next w
                    swFillet.ReleaseSelectionAccess
                Next u
            End If

        Next i
    End If



    'Create 2st Phase G Code-------------------------------------------------------
-----------------------------
    '-----------------------------------------------------------------------------
-----------------------------

    'Set the file path for the txt file
    strPath = "C:\G-Code Phase 2.txt"

    If j <> 0 Or k <> 0 Then
        'Create the new txt file
        Set ts = fso.CreateTextFile(strPath, True)

        'Write in File
        ts.WriteLine "O0901"
```

```
        ts.WriteLine "G90"
        ts.WriteLine "G53 G0 X0 Z0"

        'External Process
        If j <> 0 Then

            ts.WriteLine "T0" & intExRTPos & "0" & intExRTPos
            ts.WriteLine "G42"
            ts.WriteLine "S" & dblRSpeed & " M3"
            ts.WriteLine "M8"
            ts.WriteLine "G0 X" & varStockDFeat(0, 0) & " Z1"

            LineC = 10
            ZPos = 0

            'Start of Cycle
            If varExtTop(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varExtTop(0, 0) - 2 *
varExtTop(0, 3)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) & " Z-" &
varExtTop(0, 2)
                LineC = LineC + 1
                If varExtTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) + 2 *
varExtTop(0, 5) & " Z-" & varExtTop(0, 1)
                    LineC = LineC + 1
                ElseIf varExtTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(0, 0) +
2 * varExtTop(0, 7) & " Z-" & varExtTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            ElseIf varExtTop(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varExtTop(0, 0) - 2 *
varExtTop(0, 4)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G3 X" & varExtTop(0, 0) & " Z-
" & varExtTop(0, 4) & " R" & varExtTop(0, 4) & " F" & dblFinishFeed
                LineC = LineC + 1
                If varExtTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) + 2 *
varExtTop(0, 5) & " Z-" & varExtTop(0, 1)
```

```
                    LineC = LineC + 1
                ElseIf varExtTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(0, 0) +
2 * varExtTop(0, 7) & " Z-" & varExtTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            Else
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varExtTop(0, 0)
                LineC = LineC + 1
                If varExtTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(0, 0) + 2 *
varExtTop(0, 5) & " Z-" & varExtTop(0, 1)
                    LineC = LineC + 1
                ElseIf varExtTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) -
varExtTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(0, 0) +
2 * varExtTop(0, 7) & " Z-" & varExtTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            End If

            'Cycle Continue
            For i = 1 To UBound(varExtTop, 1)
                If varExtTop(i, 2) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varExtTop(i, 0) -
2 * varExtTop(i, 2) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) & "
Z-" & varExtTop(i, 3) + varExtTop(i - 1, 1)
                    LineC = LineC + 1
                    If varExtTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) +
2 * varExtTop(i, 5) & " Z-" & varExtTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(i,
0) + 2 * varExtTop(i, 7) & " Z-" & varExtTop(i, 1) & " F" & dblFinishFeed
```

```
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                ElseIf varExtTop(i, 4) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varExtTop(i, 0) -
2 * varExtTop(i, 4) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varExtTop(i, 0) &
" Z-" & varExtTop(i, 4) + varExtTop(i - 1, 1) & " R" & varExtTop(i, 4) & " F" &
dblFinishFeed
                    LineC = LineC + 1
                    If varExtTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) +
2 * varExtTop(i, 5) & " Z-" & varExtTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(i,
0) + 2 * varExtTop(i, 7) & " Z-" & varExtTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varExtTop(i, 0) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                    If varExtTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varExtTop(i, 0) +
2 * varExtTop(i, 5) & " Z-" & varExtTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) - varExtTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G2 X" & varExtTop(i,
0) + 2 * varExtTop(i, 7) & " Z-" & varExtTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varExtTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                End If
            Next i
```

```
            'End of Cycle

            If varStockDFeat(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockDFeat(0, 0) -
2 * varStockDFeat(0, 2) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varStockDFeat(0, 0) & "
Z-" & varStockDFeat(0, 3) + varExtTop(UBound(varExtTop, 1), 1)
                CycleT(LineC - 9) = "G0 X" & varStockDFeat(0, 0) & " Z1"
            ElseIf varStockDFeat(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockDFeat(0, 0) -
2 * varStockDFeat(0, 4) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G3 X" & varStockDFeat(0, 0) &
" Z-" & varStockDFeat(0, 4) + varExtTop(UBound(varExtTop, 1), 1) & " R" &
varStockDFeat(0, 4) & " F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockDFeat(0, 0) & " Z1"
            Else
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockDFeat(0, 0) &
" F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockDFeat(0, 0) & " Z1"
            End If


            'Write Cycle in Text File
            ts.WriteLine "G71 P10 Q" & LineC & " U" & dblFinishU & " W" &
dblFinishW & " D" & dblRoughCut & " F" & dblRoughFeed

            For i = 0 To LineC - 9
                ts.WriteLine CycleT(i)
            Next i

            'Finish Cycle
            If intExFTPos <> 0 Then
                ts.WriteLine "M9"
                ts.WriteLine "G53 G0 X0 Z0"
                ts.WriteLine "G40"
                ts.WriteLine "T0" & intExRTPos & "00"
                ts.WriteLine "T0" & intExFTPos & "0" & intExFTPos
                ts.WriteLine "G42"
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "M8"
                ts.WriteLine "G0 X" & varStockDFeat(0, 0) & " Z1"
                ts.WriteLine "G70 P10 Q" & LineC
            Else
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "G70 P10 Q" & LineC
            End If

            'End of Tool 1
            ts.WriteLine "M9"
            ts.WriteLine "G53 G0 X0 Z0"
            If intExFTPos <> 0 Then
                ts.WriteLine "T0" & intExFTPos & "00"
            Else
                ts.WriteLine "T0" & intExRTPos & "00"
            End If
            ts.WriteLine "G40"
        End If
```

```
        'Internal Process
        If k <> 0 Then

            ts.WriteLine "T0" & intInRTPos & "0" & intInRTPos
            ts.WriteLine "G41"
            If varStockSDFeat(0, 0) = 0 Then
                varStockSDFeat(0, 0) = varIntTop(k - 1, 0)
            End If

            ts.WriteLine "S" & dblRSpeed & " M3"
            ts.WriteLine "M8"
            ts.WriteLine "G0 X" & varStockSDFeat(0, 0) & " Z1"

            LineC = 10
            ZPos = 0

            'Start of Cycle
            If varIntTop(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varIntTop(0, 0) + 2 *
varIntTop(0, 3)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) & " Z-" &
varIntTop(0, 2)
                LineC = LineC + 1
                If varIntTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) - 2 *
varIntTop(0, 5) & " Z-" & varIntTop(0, 1)
                    LineC = LineC + 1
                ElseIf varIntTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(0, 0) -
2 * varIntTop(0, 7) & " Z-" & varIntTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            ElseIf varIntTop(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varIntTop(0, 0) + 2 *
varIntTop(0, 4)
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G1 Z0 F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G2 X" & varIntTop(0, 0) & " Z-
" & varIntTop(0, 4) & " R" & varIntTop(0, 4) & " F" & dblFinishFeed
                LineC = LineC + 1
                If varIntTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
```

```
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) - 2 *
varIntTop(0, 5) & " Z-" & varIntTop(0, 1)
                    LineC = LineC + 1
                ElseIf varIntTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(0, 0) -
2 * varIntTop(0, 7) & " Z-" & varIntTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            Else
                CycleT(LineC - 10) = "N" & LineC & " G0 X" & varIntTop(0, 0)
                LineC = LineC + 1
                If varIntTop(0, 5) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 6) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(0, 0) - 2 *
varIntTop(0, 5) & " Z-" & varIntTop(0, 1)
                    LineC = LineC + 1
                ElseIf varIntTop(0, 7) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) -
varIntTop(0, 7) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(0, 0) -
2 * varIntTop(0, 7) & " Z-" & varIntTop(0, 1) & " F" & dblFinishFeed
                    LineC = LineC + 1
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(0, 1) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                End If
            End If

            'Cycle Continue
            For i = 1 To UBound(varIntTop, 1)
                If varIntTop(i, 2) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varIntTop(i, 0) +
2 * varIntTop(i, 2) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) & "
Z-" & varIntTop(i, 3) + varIntTop(i - 1, 1)
                    LineC = LineC + 1
                    If varIntTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) -
2 * varIntTop(i, 5) & " Z-" & varIntTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
```

```
                        CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(i,
0) - 2 * varIntTop(i, 7) & " Z-" & varIntTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                ElseIf varIntTop(i, 4) <> 0 Then
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varIntTop(i, 0) +
2 * varIntTop(i, 4) & " F" & dblFinishFeed
                    LineC = LineC + 1
                    CycleT(LineC - 10) = "N" & LineC & " G2 X" & varIntTop(i, 0) &
" Z-" & varIntTop(i, 4) + varIntTop(i - 1, 1) & " R" & varIntTop(i, 4) & " F" &
dblFinishFeed
                    LineC = LineC + 1
                    If varIntTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) -
2 * varIntTop(i, 5) & " Z-" & varIntTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(i,
0) - 2 * varIntTop(i, 7) & " Z-" & varIntTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                Else
                    CycleT(LineC - 10) = "N" & LineC & " G1 X" & varIntTop(i, 0) &
" F" & dblFinishFeed
                    LineC = LineC + 1
                    If varIntTop(i, 5) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 6) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " X" & varIntTop(i, 0) -
2 * varIntTop(i, 5) & " Z-" & varIntTop(i, 1)
                        LineC = LineC + 1
                    ElseIf varExtTop(i, 7) <> 0 Then
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) - varIntTop(i, 7) & " F" & dblFinishFeed
                        LineC = LineC + 1
                        CycleT(LineC - 10) = "N" & LineC & " G3 X" & varIntTop(i,
0) - 2 * varIntTop(i, 7) & " Z-" & varIntTop(i, 1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    Else
                        CycleT(LineC - 10) = "N" & LineC & " G1 Z-" & varIntTop(i,
1) & " F" & dblFinishFeed
                        LineC = LineC + 1
                    End If
                End If
```

```
            Next i

            'End of Cycle

            If varStockSDFeat(0, 2) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockSDFeat(0, 0) +
2 * varStockSDFeat(0, 2) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " X" & varStockSDFeat(0, 0) & "
Z-" & varStockSDFeat(0, 3) + varIntTop(UBound(varIntTop, 1), 1)
                CycleT(LineC - 9) = "G0 X" & varStockSDFeat(0, 0) & " Z1"
            ElseIf varStockSDFeat(0, 4) <> 0 Then
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockSDFeat(0, 0) +
2 * varStockSDFeat(0, 4) & " F" & dblFinishFeed
                LineC = LineC + 1
                CycleT(LineC - 10) = "N" & LineC & " G2 X" & varStockSDFeat(0, 0) &
" Z-" & varStockSDFeat(0, 4) + varIntTop(UBound(varIntTop, 1), 1) & " R" &
varStockSDFeat(0, 4) & " F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockSDFeat(0, 0) & " Z1"
            Else
                CycleT(LineC - 10) = "N" & LineC & " G1 X" & varStockSDFeat(0, 0) &
" F" & dblFinishFeed
                CycleT(LineC - 9) = "G0 X" & varStockSDFeat(0, 0) & " Z1"
            End If


            'Write Cycle in Text File
            ts.WriteLine "G71 P10 Q" & LineC & " U-" & dblFinishU & " W" &
dblFinishW & " D" & dblRoughCut & " F" & dblRoughFeed

            For i = 0 To LineC - 9
                ts.WriteLine CycleT(i)
            Next i

            'Finish Cycle
            If intInFTPos <> 0 Then
                ts.WriteLine "M9"
                ts.WriteLine "G53 G0 X0 Z0"
                ts.WriteLine "G40"
                ts.WriteLine "T0" & intInRTPos & "00"
                ts.WriteLine "T0" & intInFTPos & "0" & intInFTPos
                ts.WriteLine "G41"
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "M8"
                ts.WriteLine "G0 X" & varStockSDFeat(0, 0) & " Z1"
                ts.WriteLine "G70 P10 Q" & LineC
            Else
                ts.WriteLine "S" & dblFSpeed & " M3"
                ts.WriteLine "G70 P10 Q" & LineC
            End If

            'End of Tool 2 and of Phase 2
            ts.WriteLine "M9"
            ts.WriteLine "G53 G0 X0 Z0"
            If intInFTPos <> 0 Then
                ts.WriteLine "T0" & intInFTPos & "00"
            Else
                ts.WriteLine "T0" & intInRTPos & "00"
            End If
```

```
            ts.WriteLine "G40"

        End If


        ts.WriteLine "M30"
        ts.Close
    End If

End Sub
```