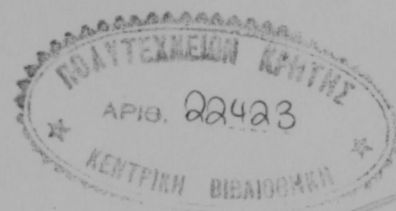


62-
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ



Διπλωματική Εργασία

ΣΤΕΡΕΑ ΑΠΕΙΚΟΝΙΣΗ
Εφαρμογή στο AutoCAD



ΣΠΥΡΟΣ ΕΛΕΥΣΙΝΙΩΤΗΣ

Χανιά Δεκέμβρης 1990

62

ΣΤΕΡΕΑ ΑΠΕΙΚΟΝΙΣΗ
Εφαρμογή στο AutoCAD

Π Ρ Ο Λ Ο Γ Ο Σ

Η εργασία αυτή ξεκίνησε από μια ιδέα του κ. Η. Στράγκα. Αντικείμενό της είναι η δημιουργία λογισμικού το οποίο δίνει την δυνατότητα στο σχεδιαστικό πακέτο AutoCAD να κάνει σχέσεις Bool (ένωση, τομή, αφαίρεση) μεταξύ παραλληλεπιπέδων και κυλίνδρων.

Το πρώτο κεφάλαιο της εργασίας αυτής αποτελεί μια εισαγωγή που προσδιορίζει το τι είναι σχεδιασμός και παραγωγή με τη βοήθεια υπολογιστών.

Στο δεύτερο κεφάλαιο γίνεται μια αναλυτική παρουσίαση των αλγόριθμων δημιουργίας στερεών σωμάτων και των μεθόδων που λειτουργούν σήμερα τα συστήματα στερεάς απεικόνισης.

Και τέλος στο τρίτο κεφάλαιο γίνεται η ανάλυση και παρουσίαση των συγκεκριμένων δυνατοτήτων του προγράμματος Bool που δίνει την δυνατότητα στο AutoCAD να κάνει τις πράξεις ένωση, τομή και αφαίρεση μεταξύ παραλληλεπιπέδων και κυλίνδρων. Και η τοποθέτηση της εντολής αυτής στα μενού του AutoCAD.

Πριν ξεκινήσω την εργασία θα ήθελα να ευχαριστήσω:

Τον Αναπληρωτή Καθηγητή του Πολυτεχνείου Κρήτης Ηλία Στράγκα τόσο για τις υποδείξεις του, όσο και για την επίβλεψη της εργασίας.

Τον Επίκουρο Καθηγητή του Πολυτεχνείου Κρήτης Νίκο Μαρμαρά και τον Αναπληρωτή Καθηγητή Γιώργο Σταυρακάκη για την συμμετοχή τους στην επιτροπή αξιολόγησης αυτής της εργασίας.

Και τέλος τον Τεχνικό Διευθυντή του Κέντρου Καινοτομιών Αθήνας (ΚΕΚΑ) Α.Ε. Νίκο Μπιλάλη για τη βοήθειά του.

Σπύρος Ελευσινιώτης

Πειραιάς 5/12/1990

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΑΡΑΓΩΓΗ

ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ

1.1 Γενικά	2
1.2 Ο ρόλος του ηλεκτρονικού υπολογιστή	3
1.2.1 Συστήματα με γραφικά και διάλογο	4
1.2.2 Συστήματα χωρίς γραφικά και χωρίς διάλογο	4
1.3 Η διαδικασία σχεδιασμού και παραγωγής	5
1.3.1 Σχεδιασμός	5
1.3.2 Παραγωγή	7
1.4 Ο ρόλος του ανθρώπινου παράγοντα	9

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

ΣΤΕΡΕΑ ΑΠΕΙΚΟΝΙΣΗ

2.1 Μέθοδοι δημιουργίας στερεών	12
2.1.1 Συρματοειδή μοντέλα	12
2.1.2 Μοντέλα με βάση γράφους	15

2.1.3 Μοντέλα Bool	17
2.1.4 Σάρωση	21
2.1.5 Αποσύνθεση	23
2.1.6 Γενικευμένα μοντέλα	26
2.2 Μέθοδοι στερεάς απεικόνισης	28
2.2.1 Συνδετική στερεά απεικόνιση	29
2.2.2 Οριακή αναπαράσταση	30
2.2.3 Υποδιαίρεση σε στοιχεία	31
2.3 Σύγκριση των μεθόδων συνδετικής στερεάς απεικόνισης και οριακής αναπαράστασης	32

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

ΣΧΕΣΕΙΣ BOOL ΣΤΟ AutoCAD

3.1 Γενικά	36
3.2 Το AutoCAD	37
3.3 Η AutoLISP	38
3.4 Η βάση δεδομένων του AutoCAD	43
3.4.1 Η βάση δεδομένων για έναν κύλινδρο	44
3.4.2 Η βάση δεδομένων για ένα παραλληλεπίπεδο	46
3.5 Η εύρεση τομών στην εντολή Bool	49
3.6 Βασική δομή της λειτουργίας της εντολής Bool	52
3.7 Ένωση	56
3.7.1 Ένωση δύο παραλληλεπιπέδων	56
3.7.2 Ένωση παραλληλεπιπέδου κυλίνδρου	58

3.7.3 Ένωση δύο κυλίνδρων	61
3.8 Τομή	63
3.8.1 Τομή δύο παραλληλεπιπέδων	63
3.8.2 Τομή παραλληλεπιπέδου κυλίνδρου	65
3.8.3 Τομή δύο κυλίνδρων	68
3.9 Αφαίρεση	70
3.9.1 Αφαίρεση δύο παραλληλεπιπέδων	70
3.9.2 Αφαίρεση παραλληλεπιπέδου κυλίνδρου	72
3.9.3 Αφαίρεση δύο κυλίνδρων	76
3.10 Η εντολή Bool στα μενού του AutoCAD	78
3.11 Αξιοποίηση της εντολής Bool	80

ΒΙΒΛΙΟΓΡΑΦΙΑ	82
------------------------	----

ΠΑΡΑΡΤΗΜΑ Α

Λίστα του προγράμματος BOOL.LSP	85
---	----

ΠΑΡΑΡΤΗΜΑ Β

Κωδικοί των ενοτήτων POLYLINE και VERTEX	101
--	-----

ΠΑΡΑΡΤΗΜΑ Γ

Εντολές στο αρχείο ACAD.MNU (μενού του AutoCAD)	103
---	-----

ΠΑΡΑΡΤΗΜΑ Δ

Ελληνο-αγγλική μετάφραση όρων	105
---	-----

ΠΑΡΑΡΤΗΜΑ Ε

Κατατεθέντα σήματα (trademarks)	107
---	-----

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΑΡΑΓΩΓΗ

ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

1. ΓΕΝΙΚΑ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΑΡΑΓΩΓΗ

ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ

1.1 ΓΕΝΙΚΑ

Η πρώτη εργαλειομηχανή αριθμητικού ελέγχου (Numerical Control , NC) αναπτύχθηκε στο πανεπιστήμιο MIT των ΗΠΑ το 1955.

Λίγα χρόνια αργότερα, συγκεκριμένα το 1963, αναπτύχθηκε στο ίδιο πανεπιστήμιο το πρώτο σύστημα σχεδιασμού με τη βοήθεια υπολογιστή.

Η σύνδεση του συστήματος σχεδιασμού ενός κομματιού με το σύστημα παραγωγής του, το γνωστό από την αγγλική ορολογία σαν CAD-CAM (Computer Aided Design - Computer Aided Manufacturing), άρχισε αρκετά να επιτευχθεί και έγινε για πρώτη φορά μόλις στις αρχές του 1980.

Σήμερα ο σχεδιασμός με υπολογιστές είναι ένα από τα ευρύτερα χρησιμοποιούμενα πεδία εφαρμογών και είναι στενά συνδεδεμένος με την παραγωγή.

Στις πρώτες φάσεις ανάπτυξης αυτών των εφαρμογών βοήθησαν πολύ τα συστήματα γραφικών στους υπολογιστές και τα γεωμετρικά μοντέλλα.

1.2 Ο ΡΟΛΟΣ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ

Ο Η/Υ χρησιμοποιήθηκε πρώτα στην απλή σχεδίαση ενός αντικειμένου. Τα πρώτα συστήματα CAD ήταν δύο διαστάσεων και χρησιμοποιήθηκαν μόνο για την παροχή των κατασκευαστικών σχεδίων των αντικειμένων που ήταν για παραγωγή, δηλαδή είχαμε απλά σχεδίαση με τον υπολογιστή αντί για σχεδίαση με το χέρι.

Αργότερα αναπτύχθηκαν διάφορα ειδικά πακέττα για την εκτέλεση της ανάλυσης της απόδοσης του αντικειμένου, όπως για παράδειγμα δομική ανάλυση, ανάλυση χώρων, στατική και δυναμική αντοχή, θερμική ανάλυση, κινηματική ανάλυση κλπ. Άλλα πακέττα εκτελούν την προσομείωση της λειτουργίας του προϊόντος, για παράδειγμα την κίνηση ρομποτικού βραχίονα κλπ.

Ας δούμε τώρα πώς μπορούν να χαρακτηριστούν τα σημερινά συστήματα CAD-CAM ανάλογα με το πώς χρησιμοποιούνται οι υπολογιστές. Τα δύο άκρα είναι:

- Συστήματα με γραφικά και διάλογο.
- Συστήματα χωρίς γραφικά και χωρίς διάλογο.

1.2.1 ΣΥΣΤΗΜΑΤΑ ΜΕ ΓΡΑΦΙΚΑ ΚΑΙ ΔΙΑΛΟΓΟ

Είναι τα πιο πολύ χρησιμοποιούμενα συστήματα, όπου τα τερματικά του υπολογιστή (οθόνη, πληκτρολόγιο, ποντίκι, ψηφιακή πλακέτα, φωτεινή πένα κλπ.) χρησιμοποιούνται για να δώσουμε τα δεδομένα στον Η/Υ. Λέγοντας διάλογο εννοούμε πως όταν καθόμαστε στο τερματικό, ο υπολογιστής μεταφράζει και εκτελεί τις εντολές τόσο γρήγορα ώστε να μην διακόπτεται ο ειρμός των σκέψεών μας, αλλά αντίθετα η δημιουργικότητά μας βοηθάται από τις απαντήσεις του υπολογιστή.

Συστήματα με γραφικά και διάλογο είναι τα πιο τυπικά από τα σημερινά συστήματα CAD-CAM. Όμως τους λείπουν μερικές πολύ χρήσιμες δυνατότητες πολύπλοκης επεξεργασίας, όπως λεπτομερειακά προγράμματα πεπερασμένων στοιχείων (finite elements).

1.2.2 ΣΥΣΤΗΜΑΤΑ ΧΩΡΙΣ ΓΡΑΦΙΚΑ ΚΑΙ ΧΩΡΙΣ ΔΙΑΛΟΓΟ

Στα συστήματα αυτά δίνουμε τα δεδομένα από ένα αρχείο δεδομένων το οποίο έχουμε δημιουργήσει από πριν. Για παράδειγμα, εάν θέλουμε να φτιάξουμε μία πόρτα, δίνουμε στο αρχείο όλες τις πληροφορίες που χρειάζονται: τις διαστάσεις, τη θέση και το μέγεθος της κλειδαριάς, το μέγεθος των ανοιγμάτων, το υλικό κατασκευής κλπ.

Τα συστήματα αυτά λέγονται συστήματα επεξεργασίας δέσμης (batch processing) και συνήθως απαιτούν κάποιο χρόνο για να δουλέψουν. Αυτό που αρχεί δεν είναι ο υπολογιστής, αλλά διάφορες

άλλες καθυστερήσεις μέχρι να έρθει η δική μας σειρά να πάρουμε τα αποτελέσματα του προγράμματός μας. Αν και συχνά πολλοί τα θεωρούν αντιπαραγωγικά και ξεπερασμένα, τα συστήματα αυτά είναι τα μόνα που επιτρέπουν την χρήση και κατ' επέκταση την αξιοποίηση των δυνατοτήτων των υπερυπολογιστών (supercomputers).

Σήμερα το ιδανικότερο σύστημα CAD-CAM, όπως καταλαβαίνουμε, θα αποτελείται από έναν συνδυασμό των δύο αυτών συστημάτων που αναπτύχθηκαν παραπάνω.

1.3 Η ΔΙΑΔΙΚΑΣΙΑ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

Η διαδικασία σχεδιασμού και παραγωγής περιλαμβάνει όλα εκείνα τα στάδια ξεκινώντας από τις αρχικές προδιαγραφές ενός προϊόντος (ανάλυση αναγκών, συλλογή πληροφοριών κλπ) και φτάνοντας μέχρι και το στάδιο της πώλησής του.

Εδώ θα γίνει μια αναφορά μόνο στο τι εννοούμε συγκεκριμένα λέγοντας "σχεδιασμός" και τι λέγοντας "παραγωγή".

1.3.1 ΣΧΕΔΙΑΣΜΟΣ

Με τον όρο "σχεδιασμός" μπορούμε να εννοήσουμε μια διαδικασία επιλογής, περιγραφής και δημιουργίας μιας φόρμας (μορφή) που εκπληρώνει μια λειτουργία. Ακόμα μπορούμε να εννοήσουμε και τη διαδικασία της δημιουργίας ενός έργου για τον

αισθητικό του αντί για τον πρακτικό του σκοπό.

Εμείς με τον όρο "σχεδιασμός" θα εννοούμε την τεχνική σχεδιομελέτη, η οποία μάλιστα είναι μια επαναληπτική διαδικασία. Μπορούμε να την ορίσουμε σαν ένα σύστημα επεξεργασίας πληροφοριών, γραφικό και αναλυτικό, με σκοπό την περιγραφή μιας σύνθετης μορφής. Οι αρχικές πληροφορίες μπορεί να είναι η δήλωση μιας ανάγκης ή ο ορισμός ενός προβλήματος, καθώς και ο τρόπος με τον οποίο ικανοποιείται η ανάγκη αυτή.

Συνήθως η σχεδιομελέτη έχει στιχμές μεγάλης δημιουργικότητας που ακολουθούνται από μακρές διαδικασίες επιλογής, περιγραφής, μετατροπής και ανάλυσης διάφορων μορφών, μέχρι να φτάσουμε σε μια τελική μορφή η οποία να ανταποκρίνεται στις λειτουργικές ανάγκες.

Εκτός βέβαια από αυτά, η διαδικασία της σχεδιομελέτης είναι και ένας αγώνας με δύο αντιπάλους, τον χρόνο και το κόστος. Τους κανόνες του παιχνιδιού τους καθορίζουν βασικά η αγορά, οι παραγωγικές μας δυνατότητες και ο πελάτης.

Όπως αναφέρθηκε πιο πριν, η σχεδιομελέτη είναι μια επαναληπτική διαδικασία, την οποία τελειώνουμε όταν το κόστος μιας ακόμα επανάληψης (δηλαδή κόστος ανθρώπων, μηχανών και υλικών) ξεπερνάει το αναμενόμενο κέρδος από την βελτίωση του προϊόντος.

1.3.2 ΠΑΡΑΓΩΓΗ

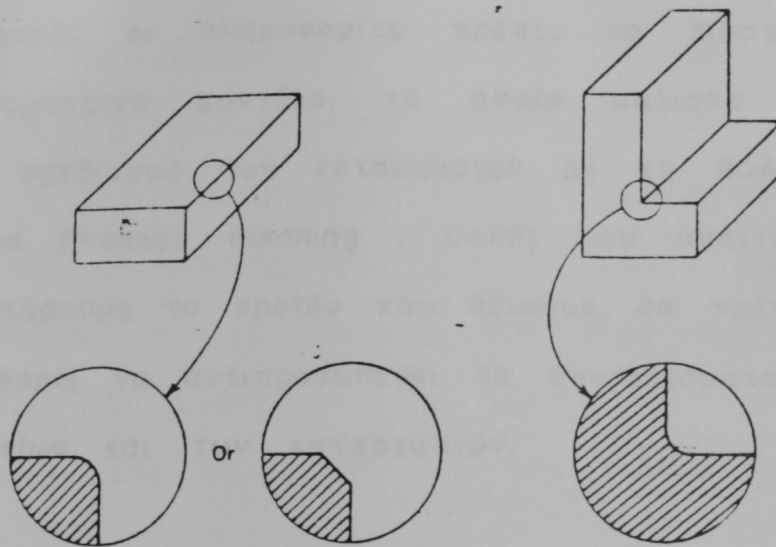
Η διαδικασία της παραγωγής έρχεται να επιβάλει τους δικούς της περιορισμούς. Για παράδειγμα δεν μπορούμε να κατασκευάσουμε ένα αντικείμενο και να είναι ακριβώς στις διαστάσεις που το θέλουμε. Ένα σύνολο περιορισμών από τα μηχανήματα, τις διαδικασίες κατεργασίας, τις ιδιότητες του υλικού και τον ανθρώπινο παράγοντα, συνομωτούν στο να μειώσουν την ακρίβεια της κατασκευής ενός κομματιού.

Παίρνοντας υπόψη όλους αυτούς τους περιορισμούς, ο σχεδιομελετητής πρέπει να αποφασίσει πόσο πολύπλοκο θα είναι το μοντέλο και το σύστημα που θα χρησιμοποιήσει. Βέβαια όσες περισσότερες παραμέτρους λαμβάνει υπόψη του το σύστημα και το μοντέλο, δηλαδή όσο πιο λεπτομερές είναι, τόσο και η επεξεργασία γίνεται πιο αρχή και το κόστος μεγαλύτερο.

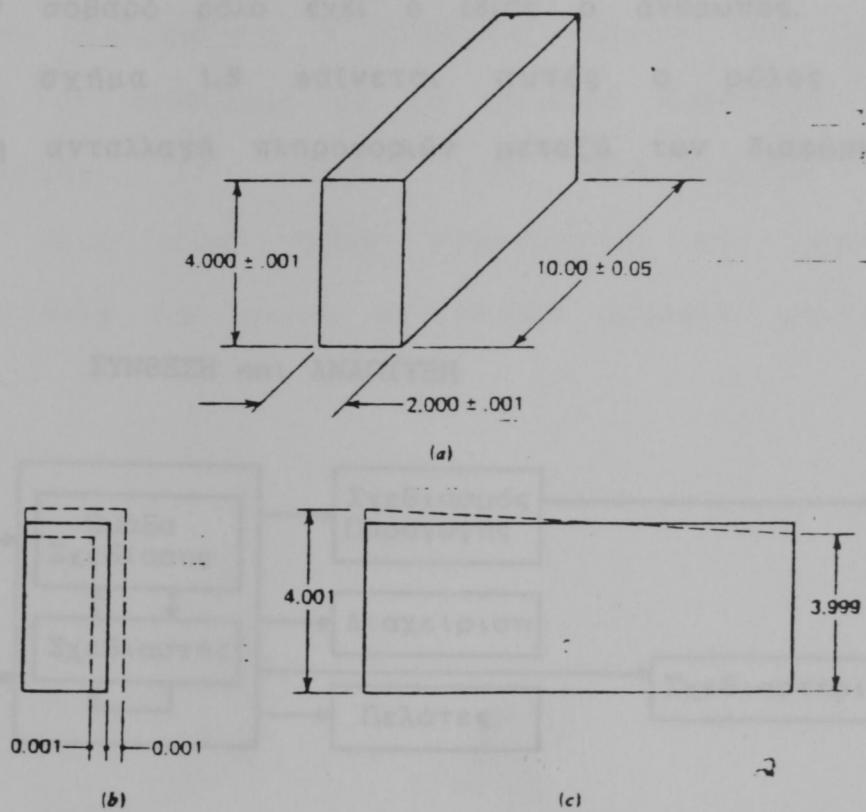
Πρέπει να πούμε ότι μικρές μετατροπές στο μοντέλο, όπως φιλέτο (fillet) ή απότμηση (chamfer) (βλέπε σχήμα 1.1) , μπορούν πολύ εύκολα να γίνουν από το ίδιο το πρόγραμμα και συνεπώς σπάνια ο σχεδιομελετητής ασχολείται με αυτές.

Για να πάρουμε υπόψη όμως τις δυσκολίες κατασκευής ορίζουμε σε κάθε διάσταση μια ανοχή, δηλαδή μια περιοχή αποδεκτών τιμών (βλέπε σχήμα 1.2). Όσο πιο ακριβείς θέλουμε να είναι οι διαστάσεις ενός κομματιού (δηλαδή όσο μικρότερες ανοχές αποδεχόμαστε), τόσο πιο ακριβό γίνεται το κομμάτι.

Σχήμα 1.2 Ορισμός ανοχών.



Σχήμα 1.1 Μικρές μετατροπές στο μοντέλο.



Σχήμα 1.2 Ορισμός ανοχών.

Όλες αυτές οι πληροφορίες πρέπει να εισαχθούν κατάλληλα μέσα στο γεωμετρικό μοντέλο, το οποίο μάλιστα αποκτά μεγάλη σημασία στον σχεδιασμό των καταργασιών με τη βοήθεια υπολογιστή (Computer Aided Process Planning , CAPP) που απαιτεί ένα κομμάτι. Έτσι για να πάρουμε το προϊόν που θέλουμε, θα πρέπει ένα σύστημα CAD-CAM να μπορεί να αντιπροσωπεύει τα αποτελέσματα της επέμβασης και των εργαλείων και των καταργασιών.

1.4 Ο ΡΟΛΟΣ ΤΟΥ ΑΝΘΡΩΠΙΝΟΥ ΠΑΡΑΓΟΝΤΑ

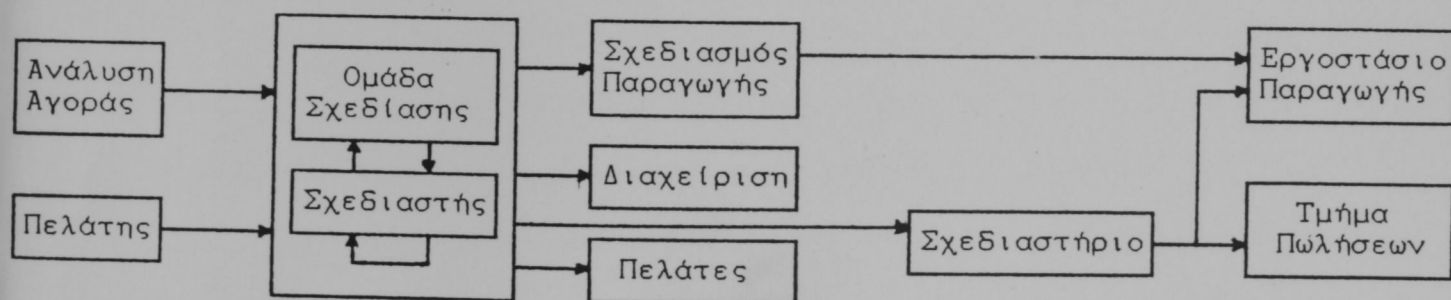
Στην όλη διαδικασία σχεδιασμού και παραγωγής με τη βοήθεια υπολογιστών σοβαρό ρόλο έχει ο ίδιος ο άνθρωπος.

Στο σχήμα 1.3 φαίνεται αυτός ο ρόλος καθώς και η απαιτούμενη ανταλλαγή πληροφοριών μεταξύ των διαφόρων ομάδων.

ΠΡΟΔΙΑΓΡΑΦΕΣ

ΣΥΝΘΕΣΗ και ΑΝΑΠΤΥΞΗ

ΠΑΡΑΓΩΓΗ



Σχήμα 1.3 Ο ρόλος του ανθρώπινου παράγοντα στην διαδικασία CAD-CAM.

Η ανταλλαγή πληροφοριών γίνεται συνήθως με τη μορφή εγγράφων, σχεδίων ή φυσικών μοντέλων.

Σήμερα, η δυσκολότερη φάση αυτής της διαδικασίας βρίσκεται στην δημιουργία αλγορίθμων, για την ανάπτυξη και σύνθεση των εναλλακτικών λύσεων - μια περιοχή που λίγο έχει αναπτυχθεί - επειδή απαιτείται η "γνώση" και η "φαντασία" του σχεδιομελετητή για την παραγωγή, την αξιολόγηση και την σύνθεση των εναλλακτικών λύσεων.

Η περιοχή του CAD-CAM εξελίσσεται και αλλάζει διεθνώς με ταχύτατους ρυθμούς. Η αλματώδης ανάπτυξη των δυνατοτήτων των ηλεκτρονικών υπολογιστών, η τεράστια μείωση του κόστους και του όγκου τους, σε συνδιασμό με τη ραγδαία ανάπτυξη του λογισμικού, έχουν αλλάξει τη μορφή των συστημάτων CAD-CAM. Η αξιοποίησή τους είναι πλέον απρόσκοπτη, όχι μόνο από μεγάλες βιομηχανίες αλλά και μικρά μελετητικά γραφεία και μηχανουργεία.

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

2.1 ΜΕΘΟΔΟΙ ΑΝΑΠΟΤΥΓΜΑΤΟΣ ΣΤΕΡΕΩΝ

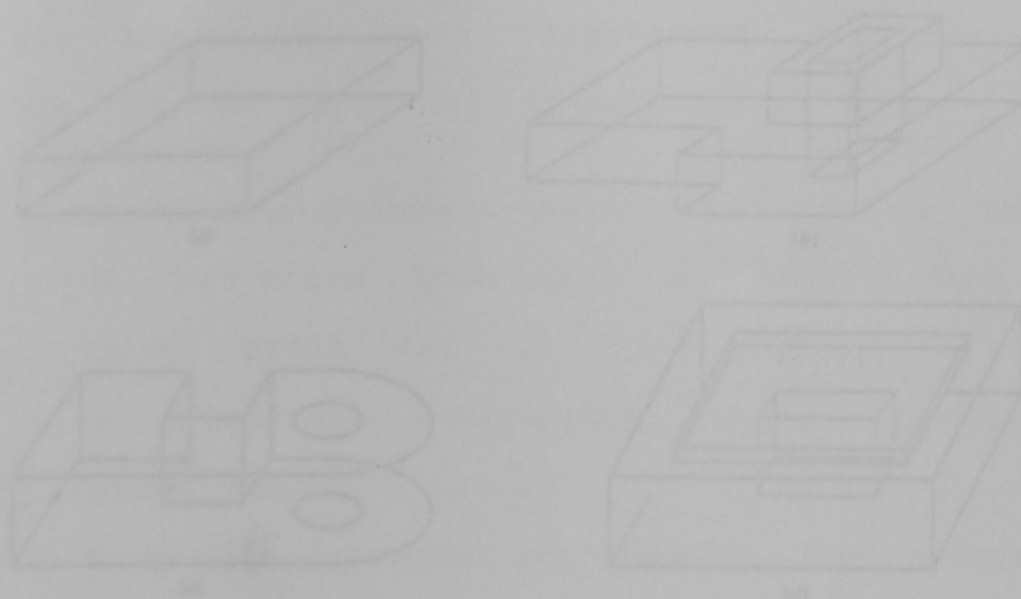
Στα συνέχεια ακολουθούν οι μεθόδους για την παραγωγή στερεών με τη βοήθεια της φωτογραφίας.

ΣΤΕΡΕΑ ΑΠΕΙΚΟΝΙΣΗ

2.1.1 ΣΥΝΘΕΣΗ ΣΤΕΡΕΩΝ

Τα πρώτα συστήματα CAD που αναπτύχθηκαν για την παραγωγή στερεών με τη βοήθεια της φωτογραφίας, ήταν βασισμένα στην αρχή της οπτικής παραμόρφωσης.

Το σύστημα αυτό, που αναπτύχθηκε από τον J. J. Burges, βασίζεται στην αρχή της οπτικής παραμόρφωσης. Η μέθοδος αυτή, που αναπτύχθηκε από τον J. J. Burges, βασίζεται στην αρχή της οπτικής παραμόρφωσης.



Σχήμα 2.1. Καλά ορισμένα συρματοειδή μοντέλα.

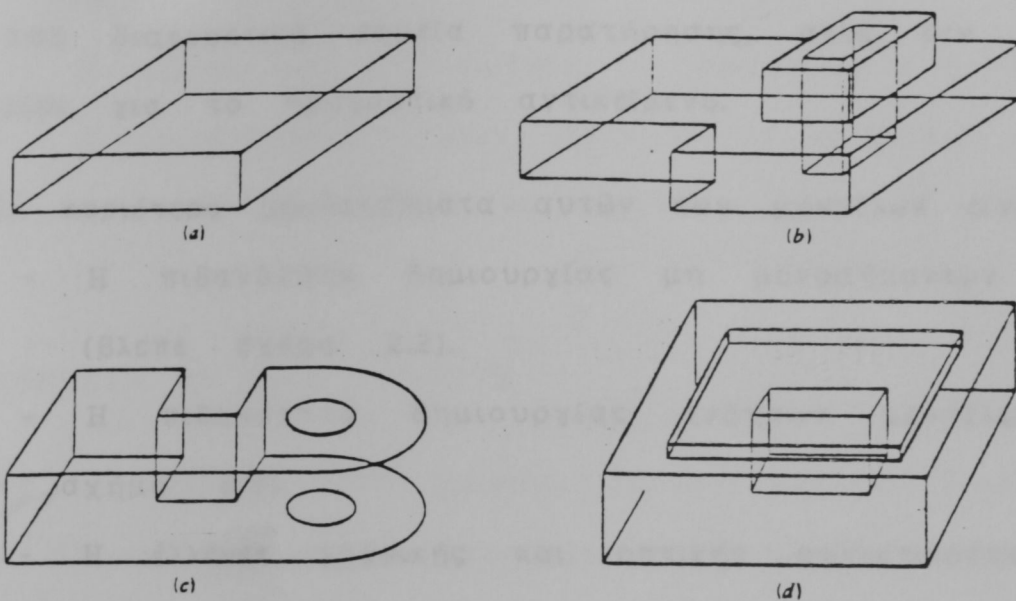
2.1 ΜΕΘΟΔΟΙ ΔΗΜΙΟΥΡΓΙΑΣ ΣΤΕΡΕΩΝ

Στη συνέχεια περιγράφονται οι βασικές μέθοδοι που χρησιμοποιούνται σήμερα για την κατασκευή στερεών.

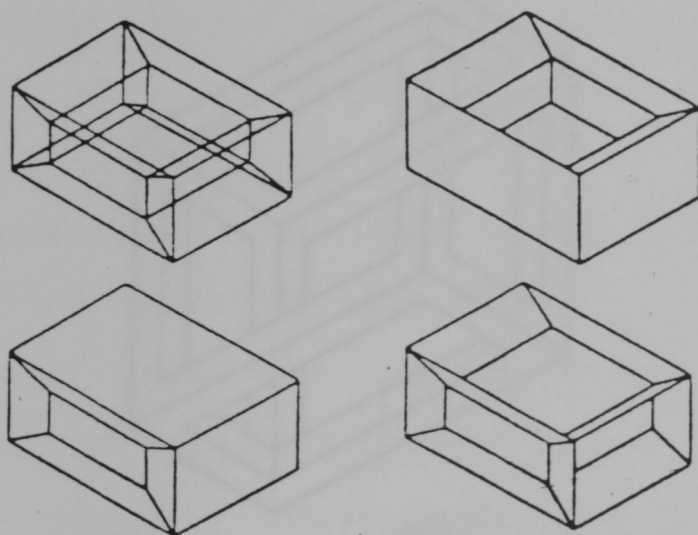
2.1.1 ΣΥΡΜΑΤΟΕΙΔΗ ΜΟΝΤΕΛΑ

Τα πρώτα συστήματα CAD που αναπτύχθηκαν ήταν συρματοειδή και διδιάστατα. Σήμερα βέβαια είναι τρισδιάστατα και αποτελούν την απλούστερη μέθοδο απεικόνισης ενός αντικειμένου.

Το συρματοειδές μοντέλο αποτελείται από έναν πεπερασμένο αριθμό κορυφών και από τις ακμές που συνδέουν ορισμένα ζεύγη κορυφών (βλέπε σχήμα 2.1). Η σχεδίαση αυτών των κορυφών και των ακμών μας δίνει μια αναπαράσταση του αντικειμένου.



Σχήμα 2.1 Καλά ορισμένα συρματοειδή μοντέλα.

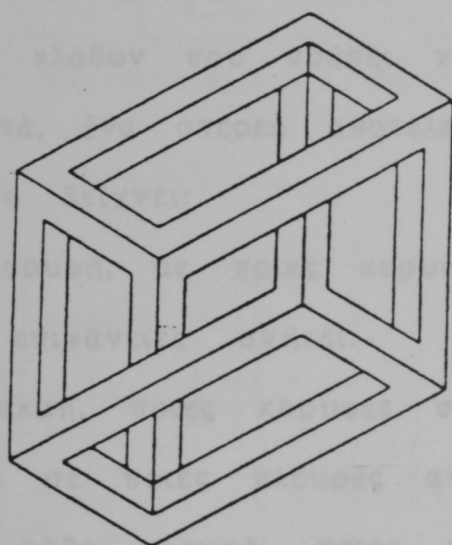


Σχήμα 2.2 Μη μονοσήμαντο συρματοειδές μοντέλο.

Το μοντέλο αυτό μπορεί να χρησιμοποιηθεί για την παραγωγή όψεων από διαφορετικά σημεία παρατήρησης, αλλά δεν μας δίνει πληροφορίες για το πραγματικό αντικείμενο.

Τα κυριότερα μειονεκτήματα αυτών των μοντέλων είναι:

- Η πιθανότητα δημιουργίας μη μονοσήμαντων μοντέλων (βλεπε σχήμα 2.2).
- Η πιθανότητα δημιουργίας ανόητων μοντέλων (βλέπε σχήμα 2.3).
- Η έλλειψη γραφικής και οπτικής συνεκτικότητας, όπως στην παράσταση της σιλουέτας ενός αντικειμένου.
- Ο υπερβολικός αριθμός λεπτομερειών.



Σχήμα 2.3 Ανόητο συρματοειδές μοντέλο.

Όμως τα συρματοειδή μοντέλα είναι αναντικατάστατα σε χρήσεις όπως:

- Η μελέτη του αντικειμένου με πεπερασμένα στοιχεία.
- Η προετοιμασία εντολών για CAM με αριθμητικό έλεγχο.
- Η σχεδιομελέτη τυπωμένων κυκλωμάτων.

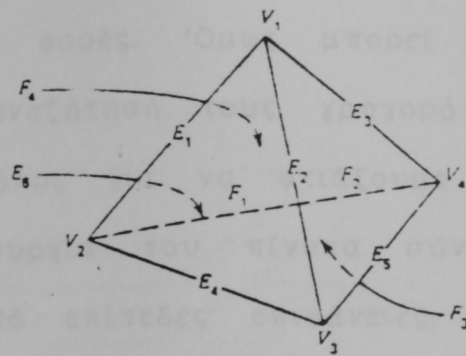
2.1.2 ΜΟΝΤΕΛΑ ΜΕ ΒΑΣΗ ΓΡΑΦΟΥΣ

Στα μοντέλα με βάση γραφούς έχουμε όλα τα δεδομένα που συνδέουν πλευρές, ακμές και κορυφές με έμφαση στην τοπολογική δομή.

Γράφος είναι ένα σύνολο κόμβων. Οι κόμβοι συνδέονται με τους κλάδους. Ο αριθμός των κλάδων που περνάνε από έναν κόμβο ορίζει το βαθμό. Ένας δρόμος από έναν κόμβο σε έναν άλλο αποτελεί μια ακολουθία κλάδων που πρέπει να διασχυστεί.

Στα μοντέλα αυτά, ένα στερεό αποτελείται από έναν κατάλογο (βλέπε σχήμα 2.4) που δείχνει:

- Για κάθε κορυφή, με ποιες κορυφές συνδέεται, σε ποιες ακμές και επιφάνειες ανήκει.
- Για κάθε ακμή, ποιες κορυφές συνδέει, με ποιες ακμές συνορεύει και σε ποιες πλευρές ανήκει.
- Τέλος για κάθε πλευρά, ποιες κορυφές και ακμές της ανήκουν καθώς και με ποιες πλευρές συνορεύει.



Vertices	Edges	Faces
V_1	E_1	F_1
V_2, V_3, V_4 E_1, E_2, E_3 F_1, F_2, F_4	V_1, V_2 E_2, E_3, E_4, E_6 F_1, F_4	V_1, V_2, V_3 E_1, E_4, E_2 F_2, F_3, F_4
V_2	E_2	F_2
V_1, V_3, V_4 E_1, E_4, E_6 F_1, F_3, F_4	V_1, V_3 E_1, E_3, E_4, E_5 F_1, F_2	V_1, V_3, V_4 E_2, E_5, E_3 F_1, F_3, F_4
V_3	E_3	F_3
V_1, V_2, V_4 E_2, \dots	V_1, V_4 E_1, E_2, \dots	V_4, V_3, V_2 E_4, \dots

Σχήμα 2.4 Μοντέλο με βάση γράφο.

	Vertex							
Vertex	1	2	3	4	5	6	7	8
1	0	1	0	1	1	0	0	0
2	1	0	1	0	0	1	0	0
3	0	1	0	1	0	0	1	0
4	1	0	1	0	0	0	0	1
5	1	0	0	0	0	1	0	1
6	0	1	0	0	1	0	1	0
7	0	0	1	0	0	1	0	1
8	0	0	0	1	1	0	1	0

1 = Connectivity
0 = No connectivity

Σχήμα 2.5 Πίνακας σύνδεσης ενός κύβου.

Προφανώς κάποια δεδομένα δεν χρειάζονται επειδή δηλώνονται περισσότερες από μία φορές. Όμως μπορεί να μας είναι χρήσιμα γιατί καθιστούν την αναζήτησή τους γρηγορότερη.

Ένας άλλος τρόπος για να φτιάξουμε ένα μοντέλο με βάση χάρφο, είναι η δημιουργία του πίνακα σύνδεσης. Αν μάλιστα το στερεό αποτελείται από επίπεδες επιφάνειες, τότε είναι δυνατό να γίνει μεγάλη οικονομία στην αποθήκευση στοιχείων. Ο πίνακας σύνδεσης είναι ένας διδιάστατος πίνακας όπως στο παράδειγμα του σχήματος 2.5.

2.1.3 MONTELA BOOL

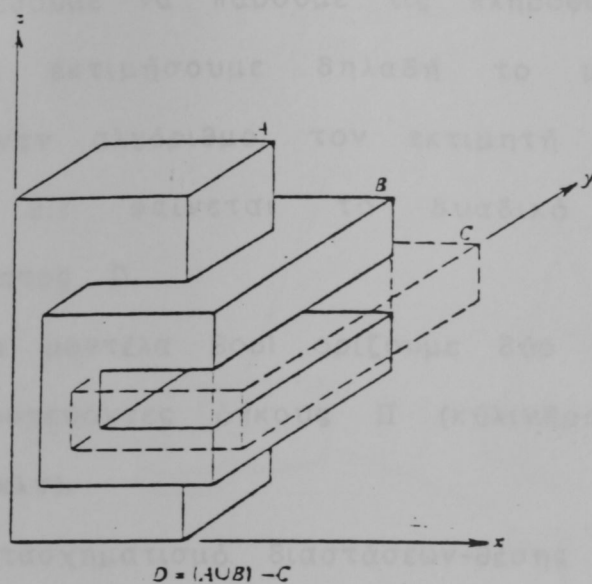
Οι γνωστές πράξεις από τη θεωρία συνόλων - ένωση, τομή, αφαίρεση - ορίζονται και σαν σχέσεις μεταξύ σωμάτων, δίνοντάς μας τα μοντέλα Bool.

Το βασικότερο χαρακτηριστικό ενός τέτοιου μοντέλου βρίσκεται στο ότι είναι διαδικαστικό.

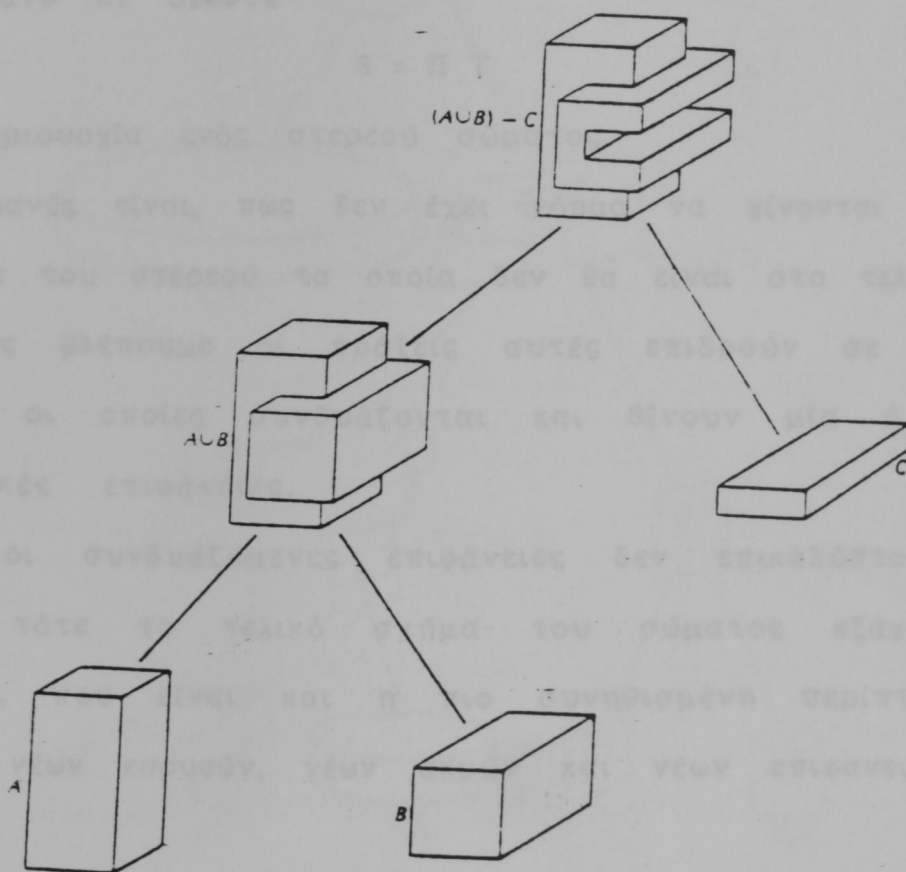
Ας δούμε ένα απλό παράδειγμα (βλέπε σχήμα 2.6). Θεωρούμε ότι γνωρίζουμε τις διαστάσεις των σωμάτων A, B, C και έχουμε ίσως έναν κατάλογο των κορυφών τους. Το μοντέλο Bool είναι

$$D = (A \cup B) - C$$

Βέβαια το μοντέλο αυτό δεν μας δίνει πληροφορίες για το αποτέλεσμα των πράξεων. Ποιες είναι δηλαδή οι επιφάνειες, οι ακμές και οι κορυφές που δημιουργούν το σώμα D.



Σχήμα 2.6 Απλό διαδικαστικό μοντέλο.



Σχήμα 2.7 Δυαδικό δέντρο.

Για να μπορέσουμε να πάρουμε τις πληροφορίες που μας είναι αναγκαίες, για να εκτιμήσουμε δηλαδή το μοντέλο, πρέπει να χρησιμοποιήσουμε έναν αλγόριθμο, τον εκτιμητή ορίων.

Στο σχήμα 2.7 φαίνεται το δυαδικό δέντρο για την δημιουργία του σώματος D.

Γενικά για τα μοντέλα Bool ορίζουμε δύο πράγματα:

- Τους πρωτεύοντες όγκους Π (κύλινδρος, παραλληλεπίπεδο, σφαίρα κλπ).
- Έναν μετασχηματισμό διαστάσεων-θέσης T .

Στο σχήμα 2.8 φαίνονται οι διαδοχικές πράξεις Bool που γίνονται πάνω σε όγκους

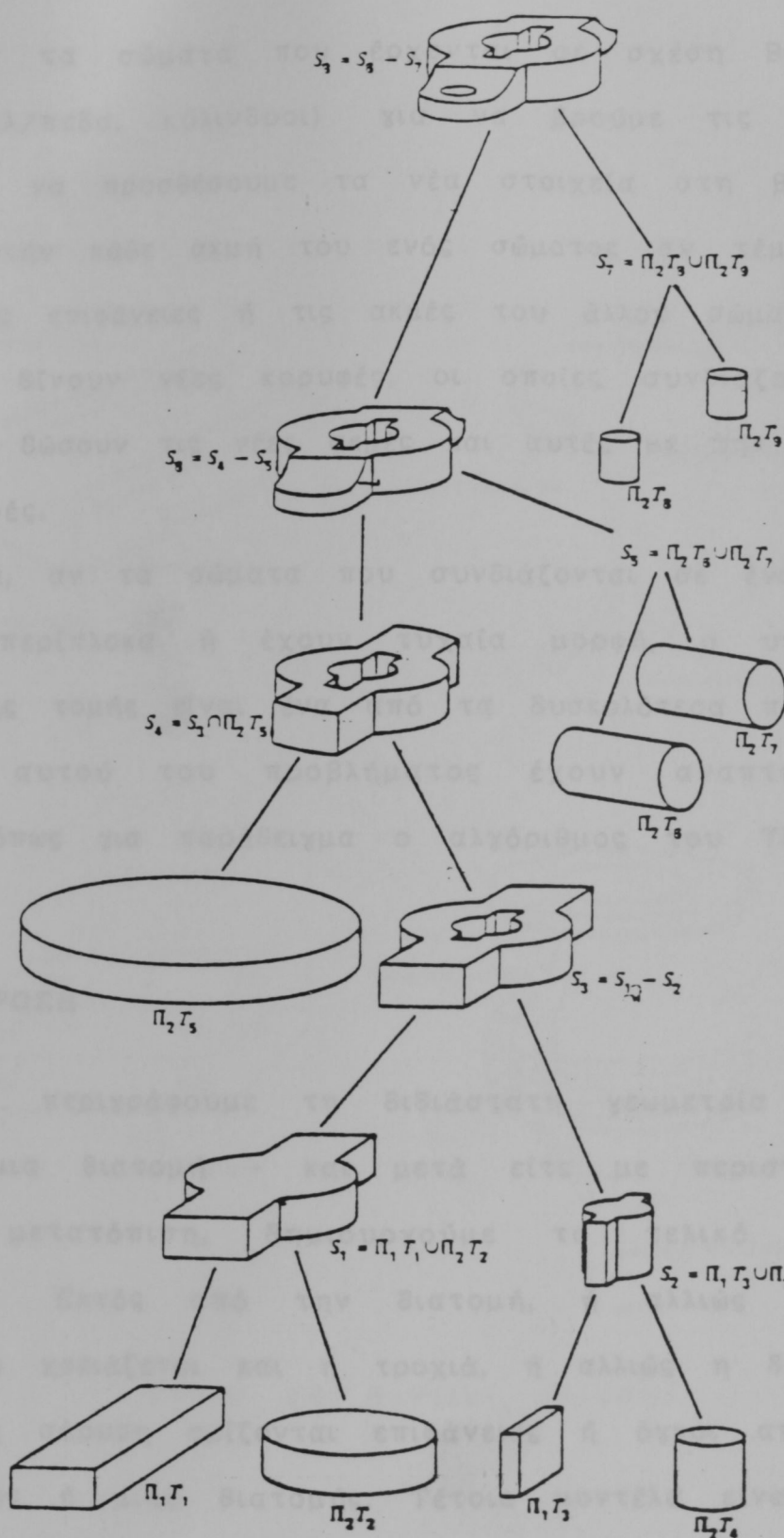
$$S = \Pi T$$

για τη δημιουργία ενός στερεού σώματος.

Προφανές είναι, πως δεν έχει νόημα να γίνονται πράξεις Bool σε τμήματα του στερεού τα οποία δεν θα είναι στο τελικό στερεό.

Όπως βλέπουμε οι πράξεις αυτές επιδρούν σε δύο οριακές επιφάνειες, οι οποίες συνδυάζονται και δίνουν μία ή περισσότερες νέες οριακές επιφάνειες.

Αν οι συνδυαζόμενες επιφάνειες δεν επικαλύπτονται ή δεν τέμνονται τότε το τελικό σχήμα του σώματος εξάγεται εύκολα. Διαφορετικά, που είναι και η πιο συνηθισμένη περίπτωση, έχουμε δημιουργία νέων κορυφών, νέων ακμών και νέων επιφανειών.



Σχήμα 2.8 Μοντέλο Bool κατασκευής στερεού σώματος.

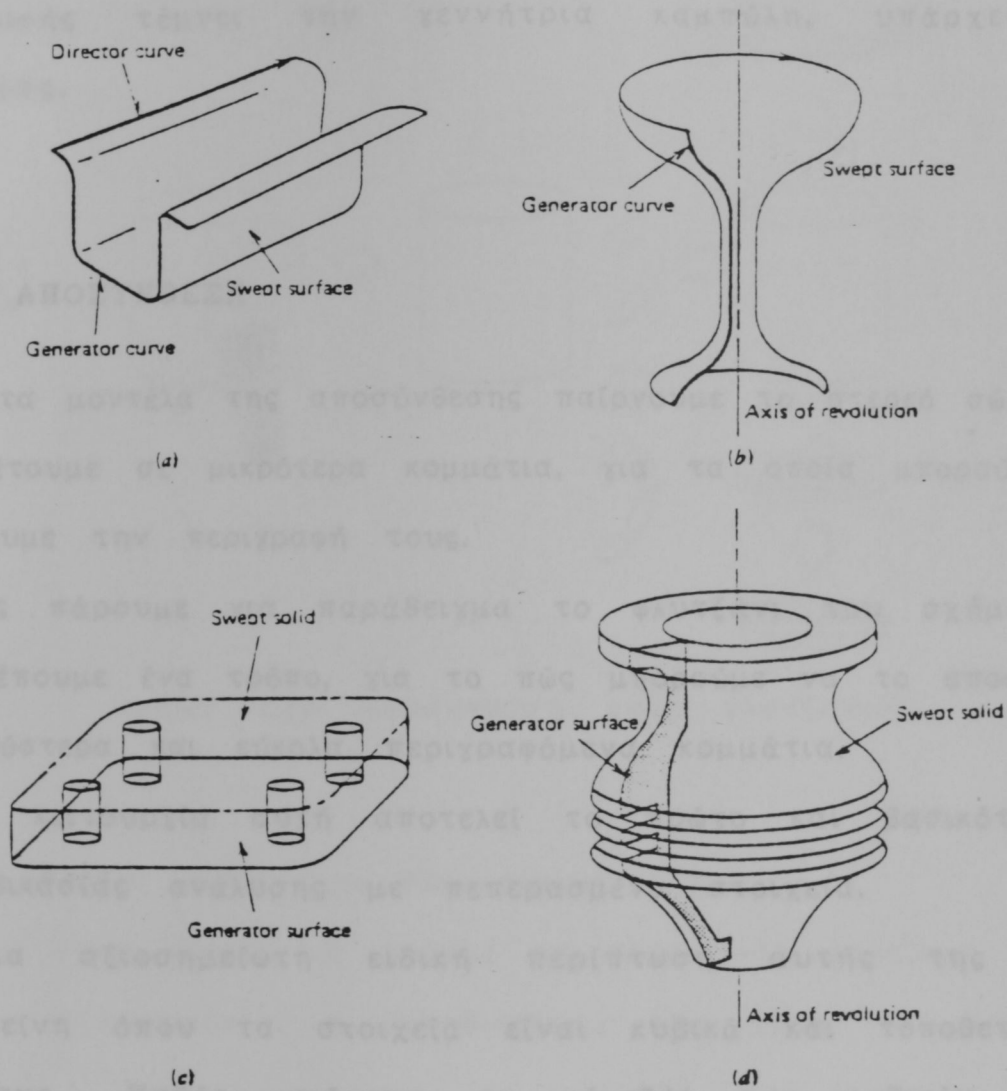
Όταν τα σώματα που έρχονται σε σχέση Bool είναι απλά (π.χ. παραλ/πεδα, κύλινδροι) για να βρούμε τις επικαλυπτόμενες πλευρές και να προσθέσουμε τα νέα στοιχεία στη βάση δεδομένων, εξετάζουμε την κάθε ακμή του ενός σώματος αν τέμνεται, με κάθε μια από τις επιφάνειες ή τις ακμές του άλλου σώματος. Τα σημεία τομής μας δίνουν νέες κορυφές, οι οποίες συνδιάζονται κατάλληλα για να μας δώσουν τις νέες ακμές και αυτές με την σειρά τους τις νέες πλευρές.

Γενικά, αν τα σώματα που συνδιάζονται σε ένα μοντέλο Bool είναι πιο περίπλοκα ή έχουν τυχαία μορφή, ο υπολογισμός των καμπυλών της τομής είναι ένα από τα δυσκολότερα προβλήματα. Για την λύση αυτού του προβλήματος έχουν αναπτυχθεί διάφοροι αλγόριθμοι, όπως για παράδειγμα ο αλγόριθμος του Timmer.

2.1.4 ΣΑΡΩΣΗ

Πρώτα περιγράφουμε τη διδιάστατη γεωμετρία του τεμάχιου - δηλαδή μια διατομή - και μετά είτε με περιστροφή είτε με παράλληλη μετατόπιση, δημιουργούμε το τελικό σώμα (βλέπε σχήμα 2.9). Εκτός από την διατομή, ή αλλιώς την γεννήτρια καμπύλη, μας χρειάζεται και η τροχιά, ή αλλιώς η διεύθυνση.

Με τη σάρωση ορίζονται επιφάνειες ή όγκοι από την κίνηση μιας καμπύλης ή μιας διατομής. Τέτοια μοντέλα είναι χρήσιμα για την παρουσίαση σωμάτων που έχουν είτε σταθερή διατομή, είτε κάποιον άξονα συμμετρίας, καθώς και για την εύρεση παρεμβολών.



Σχήμα 2.9 Μοντέλα σάρωσης.

Η σάρωση είναι μια σύνθετη λειτουργία που συνίσταται από τις απλούστερες λειτουργίες δημιουργίας - αφαίρεσης κορυφών, ακμών και επιφανειών (λειτουργίες Euler).

Σε παθολογικές περιπτώσεις, για παράδειγμα όταν ο άξονας περιστροφής τέμνει την γεννήτρια καμπύλη, υπάρχει σημείο ανωμαλίας.

2.1.5 ΑΠΟΣΥΝΘΕΣΗ

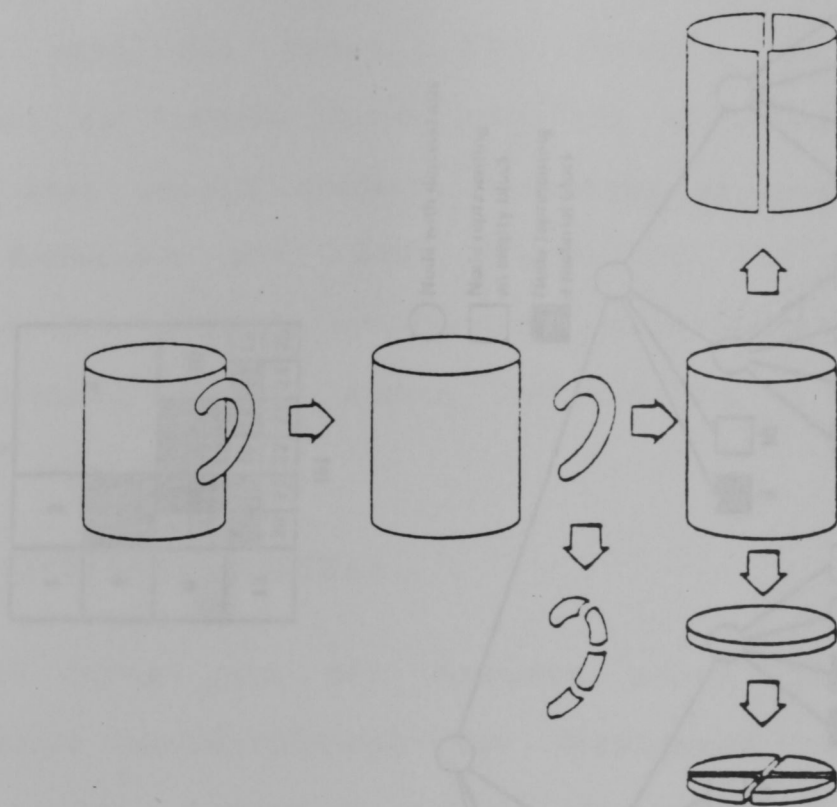
Στα μοντέλα της αποσύνθεσης παίρνουμε το στερεό σώμα και το αποσυνθέτουμε σε μικρότερα κομμάτια, για τα οποία μπορούμε εύκολα να κάνουμε την περιγραφή τους.

Ας πάρουμε για παράδειγμα το φλυτζάνι του σχήματος 2.10, όπου βλέπουμε ένα τρόπο, για το πώς μπορούμε να το αποσυνθέσουμε σε απλούστερα και εύκολα περιγραφόμενα κομμάτια.

Η λειτουργία αυτή αποτελεί το πρώτο και βασικότερο βήμα της διαδικασίας ανάλυσης με πεπερασμένα στοιχεία.

Μια αξιοσημείωτη ειδική περίπτωση αυτής της μεθόδου, είναι εκείνη όπου τα στοιχεία είναι κυβικά και τοποθετημένα σε ένα δίκτυο. Καθώς μειώνεται το μέγεθος των κυβικών στοιχείων, η μέθοδος μετατρέπεται σε ένα σύνολο συνεχών στοιχείων.





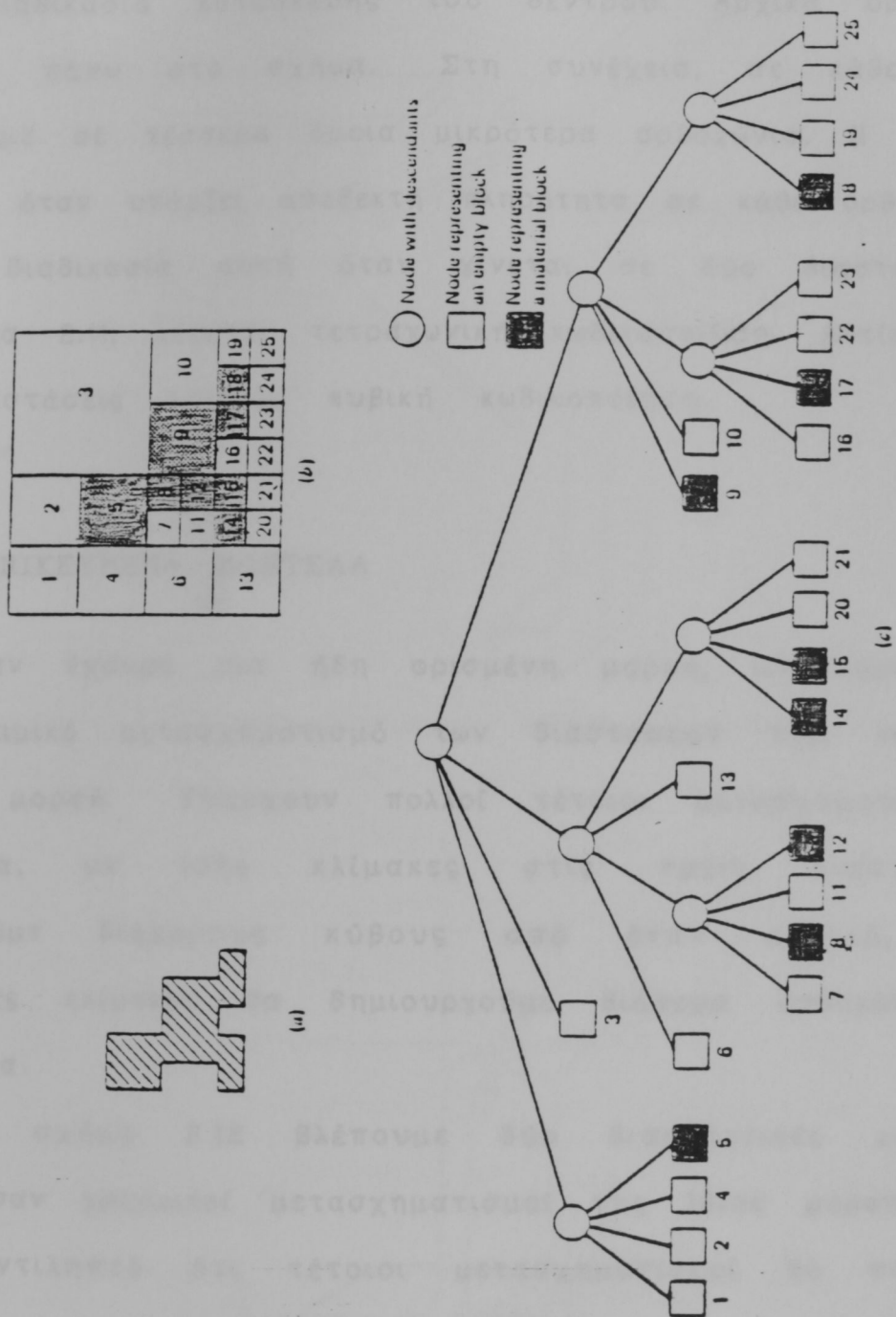
Σχήμα 2.10 Αποσύνθεση ενός φλυτζανιού.

Η μέθοδος αυτή έχει δύο πλεονεκτήματα:

- Είναι εύκολο να βρούμε τη θέση ενός σημείου.
- Εξασφαλίζεται η μοναδικότητα στο χώρο.

Έχει όμως και σημαντικά μειονεκτήματα:

- Απαιτεί πολύ μνήμη.
- Δεν υπάρχει ξεκάθαρη και άμεση σχέση ανάμεσα στα τμήματα ενός αντικειμένου.



Σχήμα 2.11 Παράσταση με δέντρο.

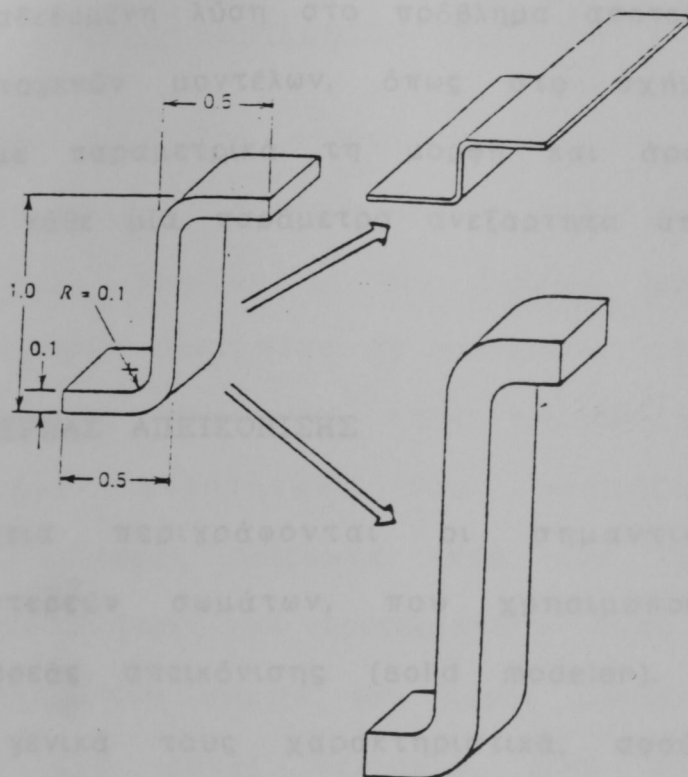
Η παράσταση με δέντρο δίνεται στο σχήμα 2.11 όπου φαίνεται και η διαδικασία κατασκευής του δέντρου. Αρχικά ορίζουμε ένα ορθογώνιο πάνω στο σχήμα. Στη συνέχεια, σε κάθε βήμα το υποδιαιρούμε σε τέσσερα όμοια μικρότερα ορθογώνια. Η υποδιαίρεση σταματάει όταν υπάρξει αποδεκτή πληρότητα σε κάθε ορθογώνιο.

Η διαδικασία αυτή όταν γίνεται σε δύο διαστάσεις, όπως στο σχήμα 2.11, λέγεται τετραγωνική κωδικοποίηση. Αντίστοιχα στις τρεις διαστάσεις λέγεται κυβική κωδικοποίηση.

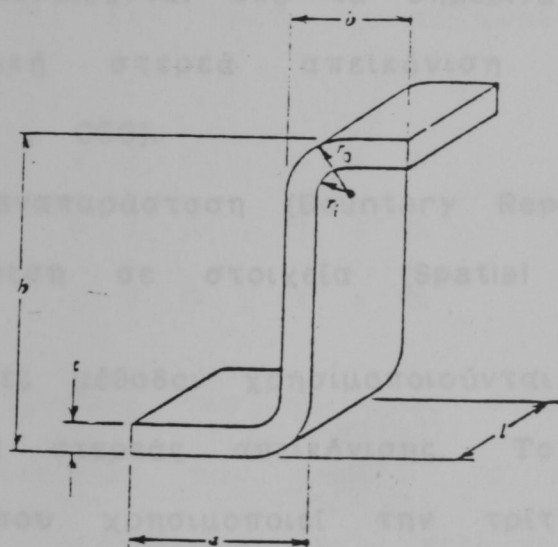
2.1.6 GENIKEYMENA MONTELA

Όταν έχουμε μια ήδη ορισμένη μορφή, μπορούμε με έναν απλό γραμμικό μετασχηματισμό των διαστάσεών της, να ορίσουμε μια νέα μορφή. Υπάρχουν πολλοί τέτοιοι μετασχηματισμοί. Για παράδειγμα, με ίσες κλίμακες στις τρεις διαστάσεις θα δημιουργούμε διάφορους κύβους από έναν αρχικό, ενώ με διαφορετικές κλίμακες θα δημιουργούμε διάφορα ορθογώνια παραλληλεπίπεδα.

Στο σχήμα 2.12 βλέπουμε δύο διαφορετικές μορφές που προήλθαν σαν γραμμικοί μετασχηματισμοί της ίδιας μορφής. Εύκολα γίνεται αντιληπτό ότι τέτοιοι μετασχηματισμοί θα παρουσιάζουν αρκετά προβλήματα, αφού όλες οι διαστάσεις σε μια κατεύθυνση θα μεταβληθούν με την ίδια κλίμακα. Έτσι για παράδειγμα τα τετράγωνα θα γίνουν ορθογώνια και οι κύκλοι ελλείψεις.



Σχήμα 2.12 Μετασχηματισμοί σύνθετου αντικειμένου.



Σχήμα 2.13 Γενικευμένο πρωτογενές μοντέλο.

Την πιο διαδεδομένη λύση στο πρόβλημα αποτελεί η χρήση των γενικευμένων πρωτογενών μοντέλων, όπως στο σχήμα 2.13, με τα οποία περιγράφουμε παραμετρικά τη μορφή και άρα μας επιτρέπει να αλλάζουμε την κάθε μία παράμετρο ανεξάρτητα από τις υπόλοιπες.

2.2 ΜΕΘΟΔΟΙ ΣΤΕΡΕΑΣ ΑΠΕΙΚΟΝΙΣΗΣ

Στη συνέχεια περιγράφονται οι σημαντικότεροι τρόποι αναπαράστασης στερεών σωμάτων, που χρησιμοποιούνται σε ένα σύστημα CAD στερεάς απεικόνισης (solid modeler). Βέβαια εδώ θα περιγραφούν τα γενικά τους χαρακτηριστικά, αφού η δομή της εσωτερικής αναπαράστασης ενός μοντέλου είναι τελείως διαφορετική για διαφορετικά συστήματα CAD στερεάς απεικόνισης.

Υπάρχουν τρεις διαφορετικές μέθοδοι αναπαράστασης ενός σώματος, που χρησιμοποιούνται από τα σημερινά συστήματα:

- Συνδυετική στερεά απεικόνιση (Constructive Solid Geometry , CSG).
- Οριακή αναπαράσταση (Boundary Representation , B-rep).
- Υποδιαίρεση σε στοιχεία (Spatial Subdivision).

Οι δύο πρώτες μέθοδοι χρησιμοποιούνται σχεδόν σε όλα τα σημερινά συστήματα στερεάς απεικόνισης. Το πιο γνωστό, μέχρι σήμερα, σύστημα που χρησιμοποιεί την τρίτη μέθοδο, είναι το INSIGHT της εταιρίας Phoenix Data Systems.

2.2.1 ΣΥΝΔΕΤΙΚΗ ΣΤΕΡΕΑ ΑΠΕΙΚΟΝΙΣΗ (CSG)

Με τη μέθοδο αυτή ένα στερεό παριστάνεται στον υπολογιστή σαν ένας συνδιασμός στοιχειωδών στερεών αντικειμένων. Βασικό εργαλείο στην δημιουργία του τελικού μοντέλου είναι η χρήση των τριών σχέσεων Bool πάνω σε πρωτογενή σώματα.

Αρκετές φορές τα πρωτογενή αυτά σώματα είναι συνδυασμοί ακόμη απλούστερων οντοτήτων που ονομάζονται ημίχωροι (half-spaces). Οι ημίχωροι ορίζονται από μια επιφάνεια άπειρης έκτασης που χωρίζει πλήρως τον τρισδιάστατο χώρο σε μια στερεή και μια κενή περιοχή. Ένα σημείο του χώρου μπορεί να ανήκει είτε σε μια από τις δύο αυτές περιοχές, είτε στην διαχωριστική επιφάνεια.

Πολλοί τέτοιοι ημίχωροι μπορούν να συνδιαστούν για την παραγωγή στερεών σωμάτων ή κενών χώρων. Για παράδειγμα ένας κύβος ακμής m ορίζεται από την τομή έξι ημίχωρων, οι οποίοι ορίζονται από τις ανισότητες:

$$x \geq 0, \quad x \leq m$$

$$y \geq 0, \quad y \leq m$$

$$z \geq 0, \quad z \leq m$$

Ένας άλλος ορισμός του ημίχωρου δίνεται από τον ορισμό μιας αναλυτικής επιφάνειας που χωρίζει το χώρο σε στερεά και κενή περιοχή. Για παράδειγμα ένας κυκλικός δίσκος ακτίνας r ορίζεται από την ανισότητα:

$$x^2 + y^2 \leq r^2$$

Για τον προσδιορισμό των κορυφών, ακμών και επιφανειών ενός στερεού που παριστάνεται με τη μέθοδο της συνδετικής στερεάς απεικόνισης, απαιτείται η ανάπτυξη του δυαδικού δέντρου. Η διαδικασία αυτή, ιδιαίτερα όταν το σώμα είναι πολυσύνθετο, είναι χρονοβόρα, τόσο στην ανάπτυξη του δέντρου, όσο και σε υπολογιστικό χρόνο.

2.2.2 ΟΡΙΑΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ (B-rep)

Με τη μέθοδο της οριακής αναπαράστασης το στερεό σώμα παριστάνεται με έναν γράφο (βλέπε παράγραφο 2.1.2). Ο γράφος περιέχει τα τοπολογικά στοιχεία (κορυφές, ακμές, πλευρές) και τον τρόπο σύνδεσής τους. Για να δημιουργηθεί το μοντέλο μετατρέπουμε τα τοπολογικά στοιχεία σε γεωμετρικά (σημεία, γραμμές, επιφάνειες). Η αντιστοιχεία μεταξύ αυτών των στοιχείων είναι η ακόλουθη:

κορυφή - σημείο

ακμή - γραμμή (καμπύλη)

πλευρά - επιφάνεια

Στη βάση δεδομένων ενός B-rep μοντέλου, σχεδόν πάντα, επαναλαμβάνονται δύο ή και περισσότερες φορές ορισμένα δεδομένα. Αυτό από την μια καθιστά την αναζήτησή τους πολύ πιο γρήγορη. Από την άλλη, εκτός από το ότι χρησιμοποιεί περισσότερη μνήμη, έχει το μειονέκτημα, ότι σε διάφορες αλλαγές της γεωμετρίας του

μοντέλου, είναι πιθανό να δημιουργηθούν αριθμητικά σφάλματα από τις πράξεις που κάνει ο υπολογιστής. Το πιθανότερο αποτέλεσμα είναι να εισαχθούν κάποιες ασυνέχειες και, για παράδειγμα, μια κορυφή να μην υπάρχει ακριβώς και μόνο στις πλευρές που ανήκει.

Σε ένα μοντέλο οριακής αναπαράστασης μια πλευρά είναι μια πεπερασμένη περιοχή μιας επιφάνειας. Σε μερικά μοντέλα μια καμπύλη επιφάνεια μπορεί να προσεγγιστεί από μια σειρά επίπεδων πεπερασμένων επιφανειών. Η μέθοδος αυτή είναι πολύ απλή υπολογιστικά, αλλά μπορεί να δώσει ανακριβή αποτελέσματα.

Γενικά η δημιουργία ή η διόρθωση ενός στερεού μοντέλου με τη μέθοδο της οριακής αναπαράστασης γίνεται με τη βοήθεια:

- Των σχέσεων Bool.
- Της πρόσθεσης-αφαίρεσης κορυφών, ακμών και πλευρών.
- Της σάρωσης.
- Λειτουργιών μικρών ρυθμίσεων (φιλλέτο, απότμηση κλπ)

2.2.3 ΥΠΟΔΙΑΙΡΕΣΗ ΣΕ ΣΤΟΙΧΕΙΑ

Με τη μέθοδο της υποδιαίρεσης σε στοιχεία, διαιρούμε το στερεό σώμα σε μικρά τρισδιάστατα στοιχεία. Το καθένα από αυτά τα στοιχεία στα αγγλικά λέγεται "voxel" (το αντίστοιχο του "pixel", αλλά στις τρεις διαστάσεις).

Αν χρησιμοποιήσουμε αυτή τη διαδικασία για τη δημιουργία ενός μοντέλου, η μνήμη που χρειάζεται είναι πολύ μεγάλη. Για παράδειγμα, αν διαθέσουμε μόνο ένα bit για κάθε voxel και

προσπαθήσουμε να περιγράψουμε ένα τμήμα ενός στερεού που αποτελείται από μόνο 1000 voxel, η μνήμη που θα απαιτηθεί ξεπερνάει τα 100 MBytes.

Για την ελάττωση της απαιτούμενης μνήμης και τη βέλτιστη διαχείρισή της, αναπτύσσονται πάρα πολλές τεχνικές τόσο στο λογισμικό όσο και στην αρχιτεκτονική των ηλεκτρονικών υπολογιστών.

Το σύστημα INSIGHT της Phoenix Data Systems που λειτουργεί με τη μέθοδο της υποδιαίρεσης σε στοιχεία, δεν χρησιμοποιείται για συνηθισμένες μηχανολογικές, ηλεκτρονικές ή άλλου τέτοιου είδους εφαρμογές. Οι δυνατότητές του αξιοποιούνται για την δημιουργία στερεών μοντέλων που έχουν πολύπλοκη μορφή, όπως μοντέλα που περιγράφουν τμήματα του ανθρώπινου σώματος με εφαρμογές στην ιατρική.

2.3 ΣΥΓΚΡΙΣΗ ΤΩΝ ΜΕΘΟΔΩΝ ΣΥΝΔΕΤΙΚΗΣ ΣΤΕΡΕΑΣ

ΑΠΕΙΚΟΝΙΣΗΣ ΚΑΙ ΟΡΙΑΚΗΣ ΑΝΑΠΑΡΑΣΤΑΣΗΣ

Διεξάγεται μια μεγάλη συζήτηση στις μέρες μας μεταξύ διαφόρων ερευνητών και κατασκευαστών γύρω από τα πλεονεκτήματα και τα μειονεκτήματα των δύο παραπάνω μεθόδων που μπορεί να χρησιμοποιεί ένα σύστημα στερεάς απεικόνισης.

Γενικά 'σήμερα:

Τα συστήματα συνδετικής στερεάς απεικόνισης (CSG):

- Είναι πιο ικανά στην δημιουργία περίπλοκων επιφανειών.
- Έχουν περισσότερες δυνατότητες και ευκολίες για να μοντελοποιήσουν ένα σώμα παραμετρικά. Για παράδειγμα το μήκος μιας πλευράς ενός κομματιού ή η διάμετρος μιας οπής, μπορούν πολύ εύκολα να τροποποιηθούν σε ένα CSG μοντέλο.
- Πετυχαίνουμε καλύτερη διαχείριση μνήμης. Όσο όμως το σώμα γίνεται πιο πολύπλοκο, η ποσότητα της μνήμης που απαιτείται, αυξάνεται γεωμετρικά.

Τα συστήματα οριακής αναπαράστασης (B-rep):

- Είναι πιο γρήγορα και πιο ικανά στις περισσότερες εφαρμογές στερεάς μοντελοποίησης.
- Οι πληροφορίες που συνοδεύουν μια επιφάνεια ή μια ακμή (όπως φιλέτο, απότμηση κλπ), μπορούν κατευθείαν να περιγραφούν σε ένα μοντέλο B-rep. Κάτι τέτοιο δεν είναι δυνατό να γίνει σε ένα μοντέλο CSG επειδή οι οριακές επιφάνειες προσδιορίζονται όταν γίνεται η εκτίμηση του μοντέλου.

Δύο γνωστά συστήματα που δημιουργούν μοντέλα στερεάς απεικόνισης είναι:

- EUCLIDIS της εταιρίας Matra Datavision (M-DTV), που βρίσκεται στα εργαστήρια του Πολυτεχνείου Κρήτης.
- CADDS4X της εταιρίας Prime-Computervision που βρίσκεται στο Κέντρο Καινοτομιών Αθήνας (ΚΕΚΑ) Α.Ε.

Και τα δύο αυτά συστήματα, όπως και σχεδόν όλα τα σημερινά συστήματα στερεάς απεικόνισης, κάνουν χρήση και των δύο μεθόδων που αναπτύχθηκαν παραπάνω. Για παράδειγμα συστήματα που δημιουργούν το στερεό με την μέθοδο B-rep, χρησιμοποιούν την μέθοδο CSG για λειτουργίες όπως η απομάκρυνση των κρυφών γραμμών. Ή συστήματα βασισμένα στην μέθοδο CSG, χρησιμοποιούν την μέθοδο B-rep για να δημιουργήσουν ένα αρχείο το οποίο μπορεί να χρησιμοποιηθεί στην επικοινωνία μεταξύ διαφόρων εκδόσεων συστημάτων στερεάς απεικόνισης.

Τα συστήματα που συνδιάζουν και τις δύο αυτές μεθόδους αποκαλούνται υβριδικά.

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

3.1 ΓΕΝΙΚΑ

ΣΧΕΣΕΙΣ BOOL ΣΤΟ AutoCAD

3.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό θα γίνει μια αναλυτική παρουσίαση της διαδικασίας, με την οποία έγινε εισαγωγή στο σχεδιαστικό πακέτο AutoCAD των σχέσεων Bool, με τη μορφή εντολής. Καθώς επίσης και πώς μπορεί αυτή η εντολή να ενσωματωθεί στα μενού του AutoCAD.

Πριν συζητήσουμε όμως για την νέα εντολή, είναι αναγκαίο να δούμε ορισμένα βασικά ζητήματα, για το ίδιο το AutoCAD και την βάση δεδομένων του, όπως και για την γλώσσα AutoLISP.

3.2 ΤΟ AutoCAD

Το AutoCAD είναι σήμερα ένα από τα γνωστά και διαδεδομένα σχεδιαστικά πακέτα. Απευθύνεται σε οποιονδήποτε ασχολείται με το σχέδιο. Έχει κατασκευαστεί από την εταιρία Autodesk. Είναι ένα πακέτο γενικής σχεδίασης και μπορεί να προσαρμοστεί σε αρκετές εξειδικευμένες σχεδιαστικές ανάγκες. Επίσης διάφορες άλλες εταιρίες, έχουν δημιουργήσει μια μεγάλη ποικιλία βοηθητικών προγραμμάτων, που τρέχουν σε περιβάλλον AutoCAD.

Το ίδιο το AutoCAD έχει τη δυνατότητα να τρέχει σε έναν σημαντικό αριθμό διαφορετικών υπολογιστικών συστημάτων. Λειτουργεί σε περιβάλλον MS-DOS (ή PC-DOS), OS/2, UNIX καθώς και Macintosh II.

Η έκδοση 10 (Release 10) του πακέτου κυκλοφορεί σε δύο μορφές:

- Για υπολογιστές με επεξεργαστή τον 8086 (ή 8088) της Intel και πάνω, με απαραίτητο τον αντίστοιχο μαθηματικό συνεπεξεργαστή (δηλαδή τον 8087 και πάνω).
- Για υπολογιστές με επεξεργαστή τον 80386 και πάνω, με απαραίτητο τον αντίστοιχο μαθηματικό συνεπεξεργαστή.

Τέλος στην εργασία αυτή όταν αναφερόμαστε στο AutoCAD, θα εννοούμε πάντα την έκδοση 10 (AutoCAD Release 10).

3.3 Η AutoLISP

Η AutoLISP είναι μία έκδοση της γλώσσας LISP, που δημιουργήθηκε από την εταιρία Autodesk ειδικά για το AutoCAD.

Η LISP είναι μία από τις πρώτες γλώσσες προγραμματισμού υψηλού επιπέδου που αναπτύχθηκαν και συνεχίζει να αναπτύσσεται. Σήμερα βρίσκει πολλές εφαρμογές κύρια σε προγράμματα εμπειρων συστημάτων (Expert Systems) και τεχνητής νοημοσύνης (Artificial Intelligence, AI).

Εκτός από τις δυνατότητες της AutoLISP που έχει σαν γλώσσα προγραμματισμού, δίνει στο χρήστη του AutoCAD τρία ακόμα πλεονεκτήματα:

- Δημιουργία νέων εντολών για το AutoCAD.
- Ομαδοποίηση σχεδιαστικών εντολών και δημιουργία σχεδίων με ή χωρίς διάλογο.
- Επέμβαση στη βάση δεδομένων του AutoCAD.

Η AutoLISP υποστηρίζει εννέα τύπους δεδομένων:

Τύπος	Παράδειγμα
ακέραιοι (integers)	3 1
πραγματικοί αριθμοί (real numbers)	1.3 8
αλφαριθμητικά (strings)	"Chania"
λίστες (lists)	(0.0 2.72 1)
δείκτες αρχείων (file descriptors)	<File: 6620>
ονόματα ενοτήτων (entity names)	<Entity name: 60000056>
ομάδες επιλογών (selection sets)	<Selection set: 3>

σύμβολα (symbols)

mypoint

συναρτήσεις (subrs)

while

Οι ακέραιοι έχουν μέγεθος 16-bit (τόσο σε 16-bit συστήματα όσο και σε 32-bit) και εύρος από -32768 μέχρι 32767.

Οι πραγματικοί αριθμοί είναι πάντα διπλής ακρίβειας κινητής υποδιαστολής αριθμοί (double precision floating-point), χρησιμοποιούν 64-bit, ο μέγιστος είναι ο $1.7e+308$ και ο κοντινότερος στο μηδέν θετικός, είναι ο $1.7e-307$.

Τα αλφαριθμητικά μπορεί να έχουν οποιοδήποτε μέγεθος. Εντούτοις οι αλφαριθμητικές σταθερές δεν μπορούν να ξεπεράσουν σε μήκος τους 100 χαρακτήρες.

Οι λίστες είναι ο βασικότερος τύπος δεδομένων της AutoLISP. Μια λίστα μπορεί να αποτελείται από οποιονδήποτε αριθμό ακεραίων, πραγματικών, αλφαριθμητικών ή κάποιον άλλο από τους εννέα τύπους δεδομένων της AutoLISP. Αρκεί να βρίσκονται μέσα σε παρενθέσεις. Για παράδειγμα η παράσταση:

(1.23 2.34 3.3)

είναι μία λίστα με τρία στοιχεία - μπορεί να είναι οι συντεταγμένες ενός σημείου - .

Η παράσταση:

(repeat 4 (setq a (+ a 10)))

είναι επίσης μια λίστα που αποτελείται από δύο στοιχεία. Το δεύτερο στοιχείο της είναι και αυτό μια λίστα με τρία στοιχεία

και το τρίτο στοιχείο αυτής της λίστας είναι μία λίστα με τρία στοιχεία. Η λίστα αυτή μέσα σε ένα πρόγραμμα της AutoLISP σημαίνει:

(+ a 10)

Πρόσθεσε στην τιμή του a

τον ακέραιο 10 .

(setq a (+ a 10))

Θέσε στο a το αποτέλεσμα

της πρόσθεσης.

(repeat 4 (setq a (+ a 10)))

Επανάλαβε το προηγούμενα

τέσσερεις φορές.

Με τους δείκτες αρχείων η AutoLISP μας επιτρέπει να διαχειριζόμαστε (ανάγνωση, εγγραφή, δημιουργία) ASCII αρχεία δίσκου.

Το όνομα ενότητας είναι ένας αλφαριθμητικός κωδικός που αποκτάει το κάθε σχεδιαστικό αντικείμενο όταν δημιουργείται με το AutoCAD. Για παράδειγμα η δομή:

<Entity name: 60000c72>

είναι ένα όνομα ενότητας. Το αριθμητικό μέρος της δομής αυτής είναι μια δεκαεξαδική σταθερά που ανήκει αποκλειστικά και μόνο σε ένα σχεδιαστικό αντικείμενο. Όταν θέλουμε να βρούμε τη βάση δεδομένων ενός σχεδιαστικού αντικειμένου, την αναζητούμε με τη βοήθεια αυτής της δομής.

Με την ομάδα επιλογής μπορούμε να επιλέξουμε περισσότερα από ένα σχεδιαστικά αντικείμενα που θα αποδοθούν σε αυτήν την ομάδα επιλογής. Σε αυτήν θα υπάρχουν όλα τα ονόματα ενοτήτων

των σχεδιαστικών αντικειμένων που έχουν επιλεγεί. Για παράδειγμα η δομή:

<Selection set: 3>

σημαίνει ότι είναι η τρίτη ομάδα επιλογής (έχουν δημιουργηθεί άλλες δύο). Με την βοήθεια της AutoLISP, μπορούμε να βρούμε πόσα και ποια ακριβώς είναι τα ονόματα ενοτήτων που επιλέχτηκαν με αυτήν την ομάδα επιλογής.

Τα σύμβολα είναι όλες οι λέξεις (ονόματα μεταβλητών, σταθερών, συναρτήσεων κλπ) που δηλώνονται από το χρήστη.

Οι συναρτήσεις είναι το σύνολο όλων των ειδικών λέξεων-συναρτήσεων που προσφέρονται από την AutoLISP.

Το μικρότερο πρόγραμμα που μπορεί να γραφτεί σε AutoLISP είναι η κενή λίστα, δηλαδή:

()

το οποίο όταν τρέξει δεν κάνει απολύτως τίποτα. Συνήθως μια εφαρμογή σε AutoLISP αποτελείται από συναρτήσεις δηλωμένες από το χρήστη. Αυτές περιέχουν άλλες συναρτήσεις τόσο του χρήστη, όσο και της AutoLISP και εκτελούνται ανάλογα με τις εντολές ελέγχου που έχει θέσει ο προγραμματιστής. Οι εντολές ελέγχου τοποθετούνται είτε μέσα στο σώμα των ίδιων των συναρτήσεων, είτε σε συναρτήσεις που αναλαμβάνουν το ρόλο ελέγχου.

Η γενική μορφή δήλωσης μιας συνάρτησης είναι:

```
(defun <myfunc> (<ar1 ...> / <lv1 ...>)
  (<body>))
```

defun είναι συνάρτηση της AutoLISP που ορίζει την συνάρτηση χρήστη με το όνομα <myfunc>.

<ar1 ...> είναι η λίστα των ορισμάτων της συνάρτησης.

<lv1 ...> είναι η λίστα των τοπικών μεταβλητών της συνάρτησης.

<body> είναι η λίστα εντολών που εκτελείται όταν καλείται συνάρτηση.

Είναι πιθανό η συνάρτηση να μην έχει ορίσματα, όπως και να μην έχει τοπικές μεταβλητές. Τότε η γενική μορφή της συνάρτησης γίνεται:

```
(defun <myfunc> ()
  (<body>))
```

και όλες οι μεταβλητές που δηλώνονται σε αυτήν είναι καθολικές.

Η AutoLISP επιτρέπει στο σώμα μιας συνάρτησης χρήστη, να δηλωθούν και άλλες συναρτήσεις χρήστη. Επίση παρέχει το δικαίωμα της αναδρομικότητας, δηλαδή η συνάρτηση μπορεί να καλεί τον εαυτό της. Γενικά είναι μία γλώσσα με πλήρη δόμηση κατά λειτουργικές μονάδες. Άλλο ένα χαρακτηριστικό της, είναι η προθεματική μορφή δήλωσης (prefix notation) του τελεστή όλων των συναρτήσεων. Έτσι η διαίρεση 10 διά 2.5 δηλώνεται (/ 10 2.5).

Το AutoCAD έχει ενσωματωμένο τον ερμηνευτή (interpreter) της AutoLISP. Ένα πρόγραμμα σε AutoLISP μπορεί να πληκτρολογηθεί κατευθείαν από την γραμμή διαταχών του AutoCAD. Όμως θα χαθεί μόλις βγούμε από των διορθωτή σχεδίων (drawing editor) του AutoCAD. Σχεδόν πάντα γράφουμε τον πηγαίο κώδικα του προγράμματος σε κάποιον ASCII-editor και το σώζουμε σε δίσκο με την κατάληξη .LSP. Για να φορτώσουμε ένα πρόγραμμα, για παράδειγμα το MYPROG.LSP, δίνουμε στην γραμμή διαταχών του AutoCAD:

```
(load "myprog")
```

Ο ερμηνευτής της AutoLISP δεν κάνει διάκριση κεφαλαίων και μικρών γραμμάτων. Αν υπάρχουν συντακτικά λάθη στο πρόγραμμα, θα ενημερώσει κατά την διάρκεια της λειτουργίας load που γίνεται η ερμηνεία (μετάφραση) του προγράμματος.

3.4 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΤΟΥ AutoCAD

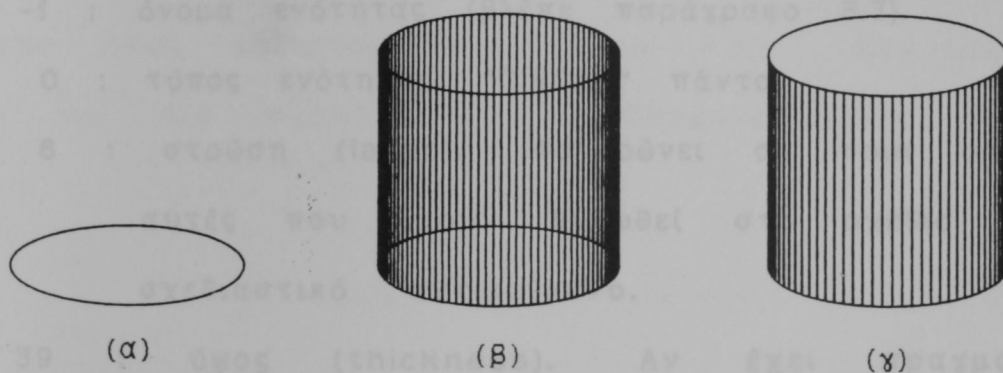
Για να αντιληφθούμε την λειτουργία της εντολής Bool, θα κάνουμε δύο συγκεκριμένα παραδείγματα, για να δούμε ποια είναι τα στοιχεία που δημιουργούνται στην βάση δεδομένων του AutoCAD, όταν τα σχεδιαστικά μας αντικείμενα είναι:

- Κύλινδρος (που δημιουργείται με την εντολή Circle).
- Παραλληλεπίπεδο (που δημιουργείται με την εντολή Pline).

Τα δύο σώματα με τα οποία θα λειτουργεί η εντολή Bool.

3.4.1 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΕΝΑΝ ΚΥΛΙΝΔΡΟ

Η πιο συνηθισμένη μέθοδος για την κατασκευή ενός κυλίνδρου στο AutoCAD είναι η εξής (βλέπε σχήμα 3.1): Να σχεδιάσουμε έναν κύκλο (σχήμα 3.1.α) ο οποίος θα σαρωθεί παράλληλα μεταξύ δύο παράλληλων επιπέδων (χρησιμοποιώντας κατάλληλα τις εντολές Elev και Circle του AutoCAD). Θα δώσει τον κύλινδρο του σχήματος 3.1.β. Στο σχήμα 3.1.γ βλέπουμε τον ίδιο κύλινδρο με απομακρυσμένες τις κρυφές γραμμές.



Σχήμα 3.1 Κύλινδρος στο AutoCAD.

Με τη βοήθεια της AutoLISP επιλέγεται το σχεδιαστικό αντικείμενο που θέλουμε, ο κύλινδρος στην περίπτωση μας, και παίρνουμε τις εξής πληροφορίες από τη βάση δεδομένων του AutoCAD:

```
( (-1 . <Entity name: 60000018>)
  (0 . "CIRCLE")
  (8 . "0")
  (39 . 4.0)
  (10 . 5.0 2.0 0.0)
  (40 . 1.5)
  (210 . 0.0 0.0 1.0)
)
```

Η παραπάνω δομή είναι μία λίστα της AutoLISP. Κάθε στοιχείο αυτής της λίστας έχει δύο τμήματα που χωρίζονται με μια τελεία. Το πρώτο τμήμα είναι ένας κωδικός αριθμός που αναφέρεται τη λειτουργία αυτού του στοιχείου για το συγκεκριμένο τύπο ενότητας. Στην παραπάνω περίπτωση οι κωδικοί αριθμοί δηλώνουν τα εξής:

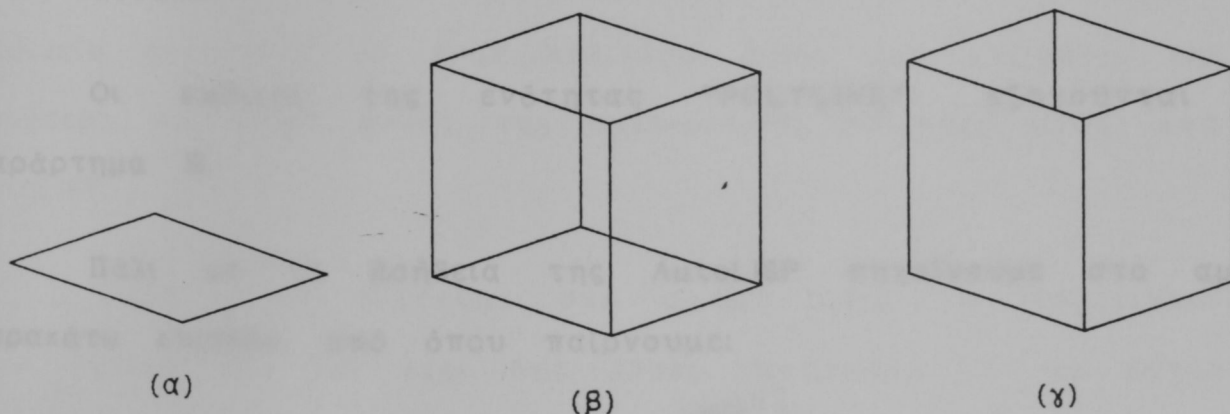
- 1 : όνομα ενότητας (βλέπε παράγραφο 3.3).
- 0 : τύπος ενότητας. "CIRCLE" πάντα.
- 8 : στρώση (layer). Φανερώνει σε ποια στρώση από αυτές που έχουν δηλωθεί στο σχέδιο ανήκει το σχεδιαστικό αντικείμενο.
- 39 : ύψος (thickness). Αν έχει πραγματοποιηθεί παράλληλη σάρωση, φανερώνει το μέγεθος της παράλληλης μετατόπισης.
- 10 : συντεταγμένες του κέντρου του κύκλου.
- 40 : ακτίνα του κύκλου.
- 210 : διάνυσμα διεύθυνσης. Φανερώνει σε ποιο σύστημα συντεταγμένων δημιουργήθηκε το σχεδιαστικό αντικείμενο.

Σχήμα 3.2 Παράλληλεπένδυση στο AutoCAD.

Αυτά τα στοιχεία αποτελούν το σύνολο των δεδομένων που μπορούμε να πάρουμε για έναν κύλινδρο.

3.4.2 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΕΝΑ ΠΑΡΑΛΛΗΛΕΠΙΠΔΟ

Η πιο συνηθισμένη μέθοδος για να κατασκευάσουμε ένα παραλληλεπίπεδο στο AutoCAD, είναι παρόμοια με εκείνη που κατασκευάζουμε έναν κύλινδρο. Δηλαδή (βλέπε σχήμα 3.2), σχεδιάζουμε ένα ορθογώνιο (σχήμα 3.2.α), το οποίο σαρώνεται παράλληλα μεταξύ δύο παράλληλων επιπέδων (χρησιμοποιώντας κατάλληλα τις εντολές Elev και Pline του AutoCAD). Δίνει το παραλ/πεδο όπως φαίνεται στο σχήμα 3.2.β. Στο σχήμα 3.2.γ βλέπουμε το ίδιο παραλ/πεδο με απομακρυσμένες τις κρυφές γραμμές.



Σχήμα 3.2 Παραλληλεπίπεδο στο AutoCAD.

Με τη βοήθεια της AutoLISP παίρνουμε τα εξής στοιχεία από τη βάση δεδομένων:

```
( (-1 . <Entity name: 600000030>)
  (0 . "POLYLINE")
  (8 . "0")
  (66 . 1)
  (10 . 0.0 0.0 0.0)
  (39 . 3.5)
  (70 . 1)
  (40 . 0.0)
  (41 . 0.0)
  (210 . 0.0 0.0 1.0)
  (71 . 0)
  (72 . 0)
  (73 . 0)
  (74 . 0)
  (75 . 0)
)
```

Βλέπουμε το όνομα της ενότητας (-1 .), τον τύπο της ενότητας (0 .), τη στρώση που βρίσκεται (8 .), το ύψος (39 .), δεν παίρνουμε όμως πληροφορίες για τις κορυφές. Αυτό συμβαίνει γιατί η βάση δεδομένων ενός POLYLINE βρίσκεται σε περισσότερα από ένα επίπεδα.

Οι κωδικοί της ενότητας "POLYLINE" εξηγούνται στο Παράρτημα Β.

Πάλι με τη βοήθεια της AutoLISP πηγαίνουμε στο αμέσως παρακάτω επίπεδο, από όπου παίρνουμε:

```
( (-1 . <Entity name: 60000048>)
  (0 . "VERTEX")
  (8 . "0")
  (39 . 3.5)
  (10 . 2.0 2.0 0.0)
  (40 . 0.0)
  (41 . 0.0)
  (42 . 0.0)
  (70 . 0)
  (50 . 0.0)
)
```

Βλέπουμε ότι ο τύπος της ενότητας είναι "VERTEX" (κορυφή). Ο κωδικός 10 ακολουθείται από τις συντεταγμένες αυτής της κορυφής.

Συνεχίζοντας τη διαδικασία αυτή παίρνουμε κάθε φορά τα δεδομένα της επόμενης κορυφής.

Όλοι οι κωδικοί της ενότητας "VERTEX" εξηγούνται στο Παράρτημα Β.

Είναι φανερό πως η βάση δεδομένων για ένα παραλληλεπίπεδο είναι πολύ πιο δομημένη και μπορούμε να πάρουμε πολύ περισσότερα στοιχεία σε σχέση με έναν κύλινδρο. Αυτό μας επιτρέπει και την καλύτερη διαχείριση αυτής της σχεδιαστικής ενότητας μέσα από την AutoLISP.

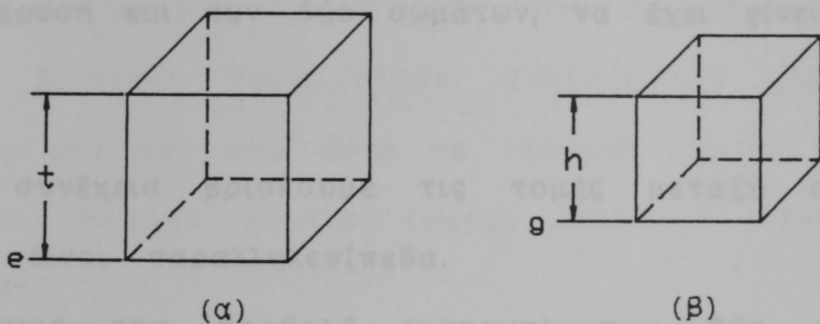
Όμως, όπως βλέπουμε στο σχήμα 3.2.γ, το παραλληλεπίπεδο που σχεδιάσαμε, δεν έχει δύο έδρες. Μπορούμε με τις κατάλληλες εντολές του AutoCAD να σχεδιάσουμε τις δύο επιφάνειες. Όμως τότε το αντικείμενό μας θα αποτελείται από τρεις σχεδιαστικές ενότητες

και πραχτικά γίνεται αδύνατη η διαχείρισή του, γιατί απαιτείται η επιλογή τριών σχεδιαστικών αντικειμένων για την επιλογή ενός παραλληλεπιπέδου.

3.5 Η ΕΥΡΕΣΗ ΤΟΜΩΝ ΣΤΗΝ ΕΝΤΟΛΗ BOOL

Για να βρούμε τις τομές μεταξύ πλευρών και ακμών εργαζόμαστε ως εξής:

Πρώτα ελέγχουμε αν υπάρχει κοινός χώρος μεταξύ των δύο παράλληλων επιπέδων που έχει σαρωθεί (βλέπε παράγραφο 2.1.4) το πρώτο σώμα, με τα δύο παράλληλα επίπεδα που έχει σαρωθεί το δεύτερο σώμα.



Σχήμα 3.3 Σώματα στο AutoCAD.

Έστω ότι είναι (βλέπε σχήμα 3.3.α):

- e το υψόμετρο που βρίσκεται η βάση του πρώτου σώματος,
- $e + t$ το υψόμετρο μέχρι το οποίο έχει σαρωθεί η βάση,
- t το ύψος του σώματος.

Όμοια έστω ότι είναι (βλέπε σχήμα 3.3.β):

g το υψόμετρο που βρίσκεται η βάση του δεύτερου σώματος,

$g + h$ το υψόμετρο μέχρι το οποίο έχει σαρωθεί η βάση,

h το ύψος του σώματος.

Για να υπάρχει λοιπόν κοινός χώρος μεταξύ: των δύο παράλληλων επιπέδων που έχει σαρωθεί το πρώτο σώμα, με τα δύο παράλληλα επίπεδα που έχει σαρωθεί το δεύτερο σώμα, πρέπει να ισχύει τουλάχιστον μία από τις δύο σχέσεις:

$$e < g < e + t$$

$$e < g + h < e + t$$

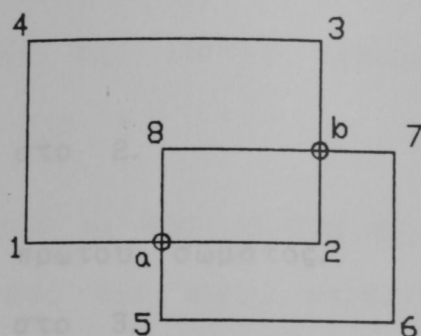
Αν δεν ισχύει καμία από τις σχέσεις αυτές, τότε τα σώματα αποκλείεται να τέμνονται.

Για να δώσουν σωστό αποτέλεσμα οι παραπάνω σχέσεις πρέπει η παράλληλη σάρωση και των δύο σωμάτων, να έχει γίνει κατά τον ίδιο άξονα.

Στην συνέχεια βρίσκουμε τις τομές μεταξύ ακμών όταν τα δύο σώματα είναι παραλληλεπίπεδα.

Παίρνουμε την προβολή (κάτοψη) των δύο σωμάτων (βλέπε σχήμα 3.4), στο επίπεδο που ορίζεται από τους δύο άξονες που είναι κάθετοι στον άξονα που πραγματοποιήθηκε η σάρωση. Για παράδειγμα, αν η σάρωση έγινε κατά τον z -άξονα, προβάλουμε τα σώματα στο xy -επίπεδο.

Τα σημεία τομής βρίσκονται, λύνοντας τα συστήματα εξισώσεων για κάθε συνδιασμό μιας ακμής του πρώτου παραλ/πέδου, με κάθε μία ακμή του δεύτερου παραλ/πέδου (δεκαέξι συστήματα,



Σχήμα 3.4 Κάτοψη παραλληλεπιπέδων.

δύο εξισώσεων, δύο αγνώστων). Για παράδειγμα το σημείο τομής b στο σχήμα 3.4, θα βρεθεί από την λύση του συστήματος των δύο εξισώσεων που: η μία ορίζεται από τα σημεία 2 και 3 της κάτοψης του πρώτου παραλ/πέδου και η άλλη από τα σημεία 7 και 8 της κάτοψης του δεύτερου παραλ/πέδου, αναζητώντας βέβαια λύση μέσα στα όρια που μπαίνουν από αυτά τα τέσσερα σημεία.

Αν δεν βρεθούν σημεία τομής, τότε τα σώματα αποκλείεται να τέμνονται.

Για να μπορεί να δώσει σωστά αποτελέσματα η παραπάνω διαδικασία, πρέπει ο άξονας κατά τον οποίο γίνεται η σάρωση, να είναι κάθετος στους άλλους δύο άξονες.

3.6 ΒΑΣΙΚΗ ΔΟΜΗ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΕΝΤΟΛΗΣ BOOL

Η βασική ροή του προγράμματος Bool είναι η ακόλουθη:

1. Αρχή.
Πήγαινε στο 2.
2. Επιλογή πρώτου σώματος.
Πήγαινε στο 3.
3. Έλεγχος αν το σώμα είναι
παραλ/πεδο ή κύλινδρος.
Αν είναι πήγαινε στο 4.
Αν δεν είναι πήγαινε στο 5.
4. Επέμβαση στη βάση δεδομένων
του πρώτου σώματος.
Δημιουργία της λίστας δεδομένων του.
Πήγαινε στο 5.
5. Επιλογή δεύτερου σώματος.
Πήγαινε στο 6.
6. Έλεγχος αν το σώμα είναι
παραλ/πεδο ή κύλινδρος.
Αν είναι πήγαινε στο 7.
Αν δεν είναι πήγαινε στο 8.

7. Επέμβαση στη βάση δεδομένων
του δεύτερου σώματος.
Δημιουργία της λίστας δεδομένων του.
Πήγαινε στο 8.
8. Αν ένα από τα δύο σώματα ή
και τα δύο δεν είναι παραλ/πεδο
ή κύλινδρος, γράψε το κατάλληλο
μήνυμα στην οθόνη και
πήγαινε στο 18 (τέλος).
Αλλιώς πήγαινε στο 9.
9. Εύρεση τομών μεταξύ ακμών και
επιφανειών των δύο σωμάτων.
Πήγαινε στο 10.
10. Επιλογή σχέσης Bool.
Ένωση, πήγαινε στο 11.
Τομή, πήγαινε στο 13.
Αφαίρεση, πήγαινε στο 16.
11. Αν τα σώματα είναι ίδια (οι λίστες
δεδομένων τους είναι ίσες) ή
αν δεν τέμνονται
πήγαινε στο 18 (τέλος).
Αλλιώς πήγαινε στο 12.

12. Σχεδίασε την ένωση (με βάση τις
λίστες δεδομένων των δύο σωμάτων
και της λίστας τομών).
Πήγαινε στο 18 (τέλος).

13. Αν τα σώματα είναι ίδια (οι λίστες
δεδομένων τους είναι ίσες),
πήγαινε στο 18 (τέλος).

Αν τα σώματα δεν τέμνονται

πήγαινε στο 14.

Αλλιώς πήγαινε στο 15.

14. Διέγραψε τα δύο σώματα (σβήστα).

Πήγαινε στο 18 (τέλος).

15. Σχεδίασε την τομή (με βάση τις
λίστες δεδομένων των δύο σωμάτων
και της λίστας τομών).

Πήγαινε στο 18 (τέλος).

16. Αν τα σώματα είναι ίδια (οι λίστες
δεδομένων τους είναι ίσες),

πήγαινε στο 14.

Αν τα σώματα δεν τέμνονται

πήγαινε στο 18 (τέλος).

Αλλιώς πήγαινε στο 17.

17. Σχεδίασε την αφαίρεση (με βάση τις

λίστες δεδομένων των δύο σωμάτων

και την λίστα τομών).

Πήγαινε στο 18.

18. Τέλος.

Η εντολή θα δουλέψει σε οποιοδήποτε σύστημα αναφοράς δηλωμένο από το χρήστη (τοπικό σύστημα συνταταχμένων) αρκεί:

- Η παράλληλη σάρωση των δύο σωμάτων να γίνει ως προς τον ίδιο άξονα.
- Ο άξονας κατά τον οποίο γίνεται η σάρωση να είναι κάθετος στους άλλους δύο άξονες.
- Τα δύο σώματα να σχεδιαστούν στο ίδιο σύστημα αναφοράς.

Στις αμέσως επόμενες παραγράφους θα μιλήσουμε για την εντολή Bool αναλυτικά. Με τη βοήθεια σχημάτων θα δούμε πώς λειτουργεί και τι συγκεκριμένες δυνατότητες έχει, για κάθε μια από τις τρεις σχέσεις Bool (ένωση, τομή, αφαίρεση) και για κάθε συνδιασμό μεταξύ των δύο σωμάτων (παράλληλεπίπεδο, κύλινδρος).

3.7 ΕΝΩΣΗ

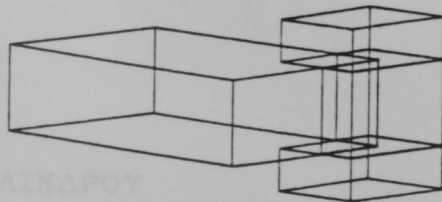
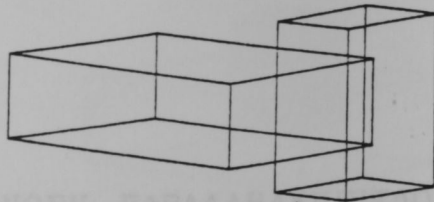
3.7.1 ΕΝΩΣΗ ΔΥΟ ΠΑΡΑΛΛΗΛΕΠΙΠΕΔΩΝ

Στα σχήματα 3.5 α και β βλέπουμε ορισμένες περιπτώσεις ένωσης μεταξύ δύο παραλληλεπιπέδων (στα σχήματα 3.5 γ και δ βλέπουμε τις ίδιες περιπτώσεις με απομακρυσμένες τις κρυφές γραμμές).

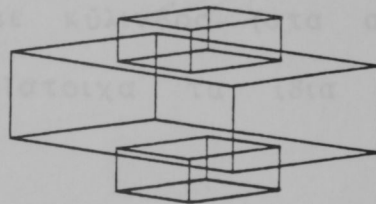
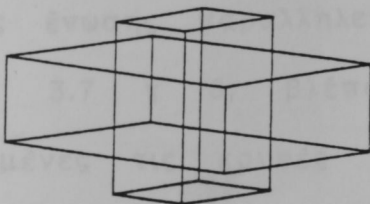
Στην αριστερή πλευρά του σχήματος βλέπουμε τα σώματα που έρχονται σε σχέση και στη δεξιά πλευρά, το αποτέλεσμα της σχέσης. Αριστερά στο σχήμα 3.5.α, βλέπουμε ένα παραλληλεπίπεδο που η μία του γωνία μπαίνει μέσα σε ένα άλλο παραλ/πεδο. Επιλέγοντας τα δύο σώματα, η εντολή Bool θα σχεδιάσει το σώμα που φαίνεται δεξιά στο σχήμα 3.5.α. Αυτή είναι η ένωση των δύο σωμάτων (στο σχήμα 3.5.γ είναι ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Αντίστοιχα αριστερά στο σχήμα 3.5.β βλέπουμε ένα παραλ/πεδο που μέσα του περνάει ένα άλλο παραλ/πεδο. Δεξιά στο ίδιο σχήμα φαίνεται η ένωσή τους που δίνεται με την εντολή Bool (στο σχήμα 3.5.δ βλέπουμε ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Οι τομές μεταξύ επιφανειών και ακμών βρίσκονται όπως περιγράφηκε στην παράγραφο 3.5.

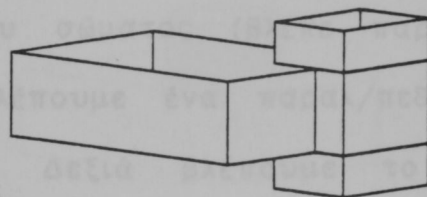
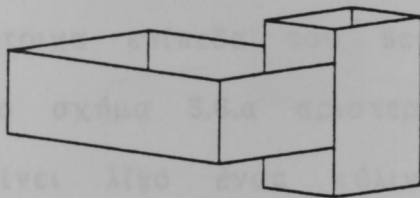
Το βασικότερο πρόβλημα που δημιουργείται είναι ότι αδυνατεί το AutoCAD να εκφράσει σαν ένα σχεδιαστικό αντικείμενο σύνθετα σώματα, όπως αυτά που φαίνονται δεξιά στο σχήμα 3.5. Έτσι ενώ η ένωση δύο σωμάτων μας δίνει ένα σώμα, αναγκαζόμαστε τις



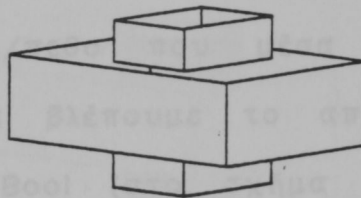
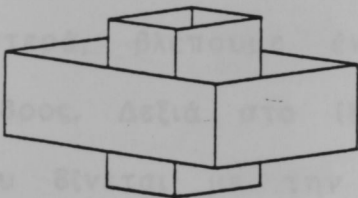
(α)



(β)



(γ)



(δ)

Σχήμα 3.5 Ένωση παραλληλεπιπέδων

περισσότερες φορές, το ένα σώμα να το εκφράσουμε με δύο τρία ή και περισσότερα σχεδιαστικά αντικείμενα στη βάση δεδομένων του AutoCAD.



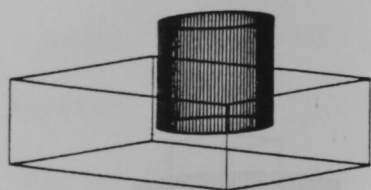
3.7.2 ΕΝΩΣΗ ΠΑΡΑΛΛΗΛΕΠΙΠΕΔΟΥ ΚΥΛΙΝΔΡΟΥ

Στα σχήματα 3.6 α β και 3.7 α β βλέπουμε ορισμένες περιπτώσεις ένωσης παραλληλεπιπέδου με κύλινδρο (στα σχήματα 3.6 γ δ και 3.7 γ δ, βλέπουμε αντίστοιχα τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

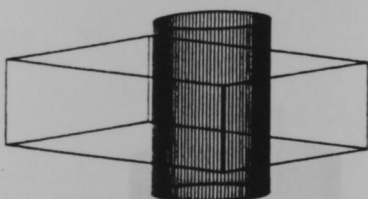
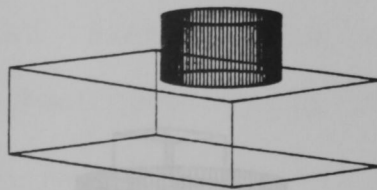
Εδώ βρίσκουμε μόνο τις επιφάνειες τομής κατά τον άξονα που έχινε η σάρωση, ελέγχοντας αν υπάρχει κοινός χώρος μεταξύ των δύο παράλληλων επιπέδων που έχει δημιουργηθεί το πρώτο σώμα, με τα αντίστοιχα επίπεδα του δεύτερου σώματος (βλέπε παράγραφο 3.5).

Στο σχήμα 3.6.α αριστερά, βλέπουμε ένα παραλ/πεδο που μέσα του μπαίνει λίγο ένας κύλινδρος. Δεξιά βλέπουμε το αποτέλεσμα της ένωσης που δίνει η εντολή Bool (στο σχήμα 3.6.γ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.6.β αριστερά, βλέπουμε ένα παραλ/πεδο που μέσα του περνάει ένας κύλινδρος. Δεξιά στο ίδιο σχήμα βλέπουμε το αποτέλεσμα της ένωσης που δίνεται με την εντολή Bool (στο σχήμα 3.6.δ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

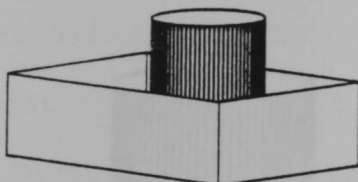
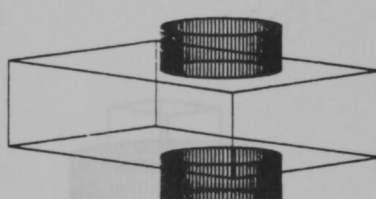
Στο σχήμα 3.7.α αριστερά, βλέπουμε έναν κύλινδρο που μέσα του περνάει ένα παραλ/πεδο. Δεξιά βλέπουμε το αποτέλεσμα της ένωσης που δίνει η εντολή Bool (στο σχήμα 3.7.γ ακριβώς τα ίδια



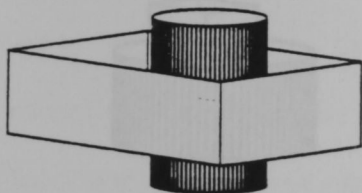
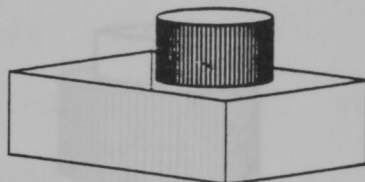
(α)



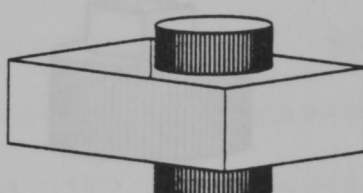
(β)



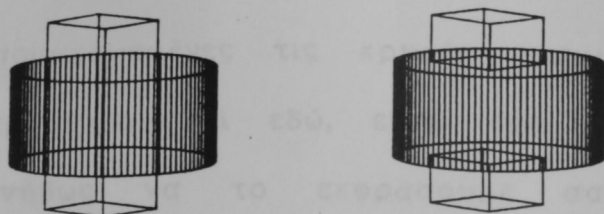
(γ)



(δ)



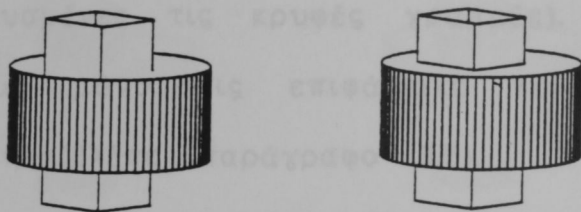
Σχήμα 3.6 Ένωση παραλληλεπιπέδου κυλίνδρου.



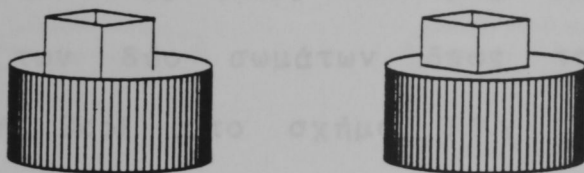
(α)



(β)



(γ)



(δ)

Σχήμα 3.7 Ένωση παραλληλεπιπέδου κυλίνδρου.

σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.7.β αριστερά, βλέπουμε έναν κύλινδρο που μέσα του μπαίνει λίγο ένα παραλ/πεδο. Δεξιά στο ίδιο σχήμα βλέπουμε το αποτέλεσμα της ένωσης που δίνεται με την εντολή Bool (στο σχήμα 3.7.δ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

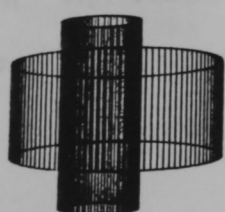
Το βασικό πρόβλημα, και εδώ, είναι πως το παραχόμενο σώμα, δεν μπορούμε συνήθως να το εκφράσουμε σαν ένα σχεδιαστικό αντικείμενο με το AutoCAD.

3.7.3 ΕΝΩΣΗ ΔΥΟ ΚΥΛΙΝΔΡΩΝ

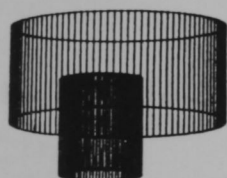
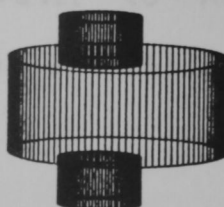
Στα σχήματα 3.8 α και β, βλέπουμε ορισμένες περιπτώσεις ένωσης δύο κυλίνδρων (στα σχήματα 3.8 γ και δ φαίνονται τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Εδώ βρίσκουμε μόνο τις επιφάνειες τομής κατά τον άξονα που έγινε η σάρωση (βλέπε παράγραφο 3.5).

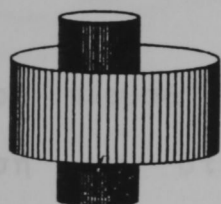
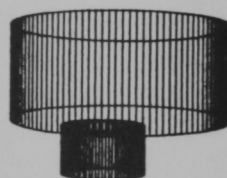
Αριστερά στο σχήμα 3.8.α βλέπουμε έναν κύλινδρο που από μέσα του περνάει ένας άλλος κύλινδρος. Δεξιά στο ίδιο σχήμα δίνεται η ένωση των δύο σωμάτων όπως την βρίσκει και την σχεδιάζει η εντολή Bool (στο σχήμα 3.8.γ φαίνονται ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.8.β αριστερά βλέπουμε έναν κύλινδρο που στην βάση του έχει μπει λίγο μέσα ένας άλλος κύλινδρος. Δεξιά στο ίδιο σχήμα βλέπουμε την ένωση των δύο σωμάτων όπως δίνεται από την εντολή



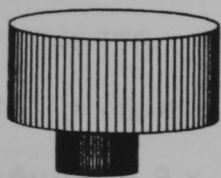
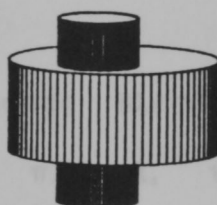
(α)



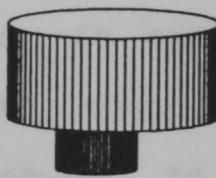
(β)



(γ)



(δ)



Σχήμα 3.8 Ένωση κυλίνδρων.

Bool (στο σχήμα 3.8.8 φαίνονται τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

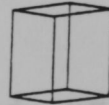
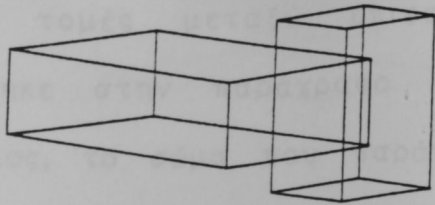
Και σε αυτή τη διαδικασία αδυνατεί το AutoCAD, σχεδόν σε όλες τις περιπτώσεις, να εκφράσει το αποτέλεσμα της πράξης με ένα σχεδιαστικό αντικείμενο.

3.8 ΤΟΜΗ

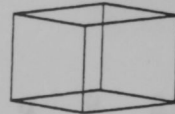
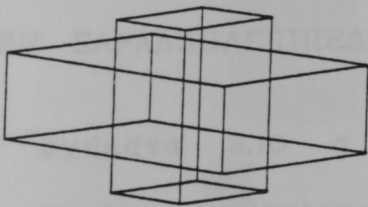
3.8.1 ΤΟΜΗ ΔΥΟ ΠΑΡΑΛΛΗΛΕΠΙΠΕΔΩΝ

Στα σχήματα 3.9 α και β βλέπουμε ορισμένες περιπτώσεις τομής μεταξύ δύο παραλληλεπιπέδων (στα σχήματα 3.9 γ και δ βλέπουμε τις ίδιες περιπτώσεις με απομακρυσμένες τις κρυφές γραμμές).

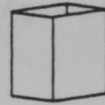
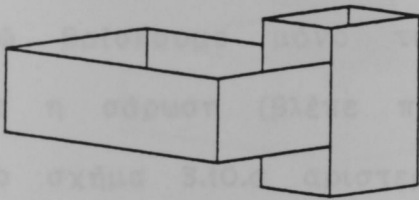
Στην αριστερή πλευρά του σχήματος βλέπουμε τα σώματα που έρχονται σε σχέση και στη δεξιά πλευρά, το αποτέλεσμα της σχέσης. Αριστερά στο σχήμα 3.9.α, βλέπουμε ένα παραλληλεπίπεδο που η μία του γωνία μπαίνει μέσα σε ένα άλλο παραλ/πεδο. Επιλέγοντας τα δύο σώματα, η εντολή Bool θα σχεδιάσει το σώμα που φαίνεται δεξιά στο σχήμα 3.9.α. Αυτή είναι η τομή των δύο σωμάτων (στο σχήμα 3.9.γ είναι ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Αντίστοιχα αριστερά στο σχήμα 3.9.β βλέπουμε ένα παραλ/πεδο που μέσα του περνάει ένα άλλο παλ/πεδο. Δεξιά στο ίδιο σχήμα φαίνεται η τομή τους που δίνεται



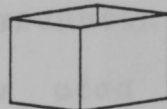
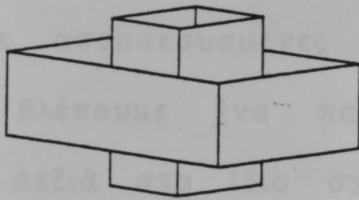
(α)



(β)



(γ)



(δ)

Σχήμα 3.9 Τομή παραλληλεπιπέδων.

με την εντολή Bool (στο σχήμα 3.9.8 βλέπουμε ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Οι τομές μεταξύ επιφανειών και ακμών βρίσκονται όπως περιγράφηκε στην παράγραφο 3.5.

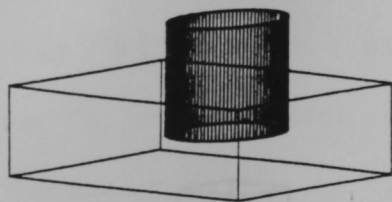
Τέλος, το σώμα που παράγεται κατά την τομή, μπορούμε να το εκφράσουμε με ένα σχεδιαστικό αντικείμενο στο AutoCAD.

3.8.2 ΤΟΜΗ ΠΑΡΑΛΛΗΛΕΠΙΠΕΔΟΥ ΚΥΛΙΝΔΡΟΥ

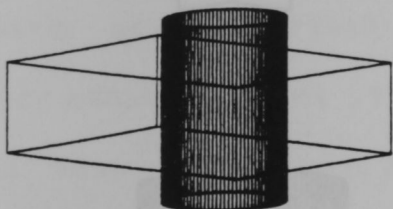
Στα σχήματα 3.10 α β και 3.11 α β βλέπουμε ορισμένες περιπτώσεις τομής παραλληλεπιπέδου με κύλινδρο (στα σχήματα 3.10 γ δ και 3.11 γ δ, βλέπουμε αντίστοιχα τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Εδώ βρίσκουμε μόνο τις επιφάνειες τομής κατά τον άξονα που έγινε η σάρωση (βλέπε παράγραφο 3.5).

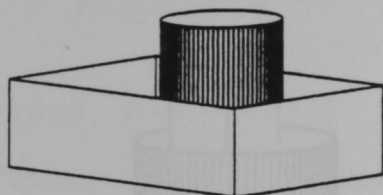
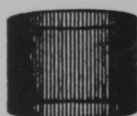
Στο σχήμα 3.10.α αριστερά, βλέπουμε ένα παραλ/πεδο που μέσα του μπαίνει λίγο ένας κύλινδρος. Δεξιά βλέπουμε το αποτέλεσμα της τομής που δίνει η εντολή Bool (στο σχήμα 3.10.γ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.10.β αριστερά, βλέπουμε ένα παραλ/πεδο που μέσα του περνάει ένας κύλινδρος. Δεξιά στο ίδιο σχήμα βλέπουμε το αποτέλεσμα της τομής που δίνεται με την εντολή Bool (στο σχήμα 3.10.δ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).



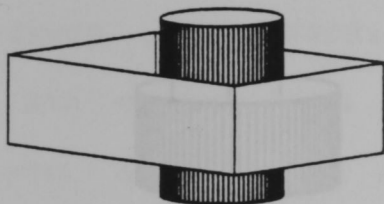
(α)



(β)



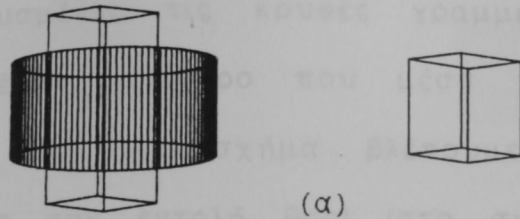
(γ)



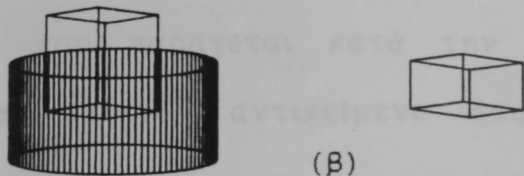
(δ)



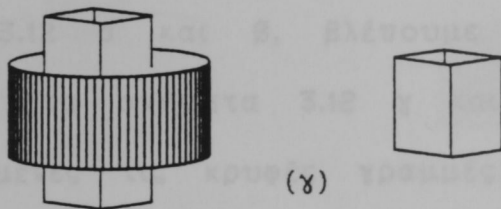
Σχήμα 3.10 Τομή παραλληλεπιπέδου κυλίνδρου



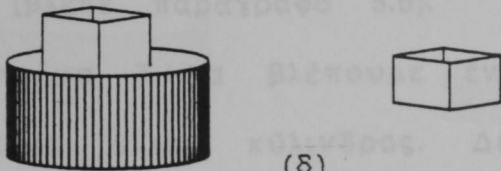
(α)



(β)



(γ)



(δ)

Σχήμα 3.11 Τομή παραλληλεπιπέδου κυλίνδρου.

Στο σχήμα 3.11.α αριστερά, βλέπουμε έναν κύλινδρο που μέσα του περνάει ένα παραλ/πεδο. Δεξιά βλέπουμε το αποτέλεσμα της τομής που δίνει η εντολή Bool (στο σχήμα 3.11.γ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.11.β αριστερά, βλέπουμε έναν κύλινδρο που μέσα του μπαίνει λίγο ένα παραλ/πεδο. Δεξιά στο ίδιο σχήμα βλέπουμε το αποτέλεσμα της τομής που δίνεται με την εντολή Bool (στο σχήμα 3.11.δ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

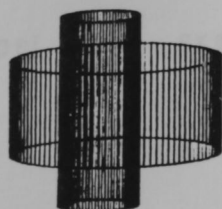
Τέλος το σώμα που παράχεται κατά την τομή μπορούμε να το εκφράσουμε με ένα σχεδιαστικό αντικείμενο στο AutoCAD.

3.8.3 ΤΟΜΗ ΔΥΟ ΚΥΛΙΝΔΡΩΝ

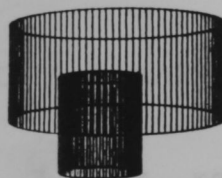
Στα σχήματα 3.12 α και β, βλέπουμε ορισμένες περιπτώσεις τομής δύο κυλίνδρων (στα σχήματα 3.12 γ και δ φαίνονται τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Εδώ βρίσκουμε μόνο τις επιφάνειες τομής κατά τον άξονα που έχινε η σάρωση (βλέπε παράγραφο 3.5).

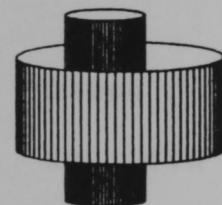
Αριστερά στο σχήμα 3.12.α βλέπουμε έναν κύλινδρο που από μέσα του περνάει ένας άλλος κύλινδρος. Δεξιά στο ίδιο σχήμα δίνεται η τομή των δύο σωμάτων όπως την βρίσκει και την σχεδιάζει η εντολή Bool (στο σχήμα 3.12.γ φαίνονται ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.12.β αριστερά βλέπουμε έναν κύλινδρο που στην βάση του έχει μπει λίγο μέσα ένας άλλος κύλινδρος. Δεξιά στο ίδιο σχήμα βλέπουμε την τομή των δύο



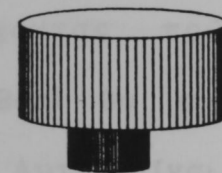
(α)



(β)



(γ)



(δ)



Σχήμα 3.12 Τομή κυλίνδρων.

σωμάτων όπως δίνεται από την εντολή Bool (στο σχήμα 3.12.8 φαίνονται τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

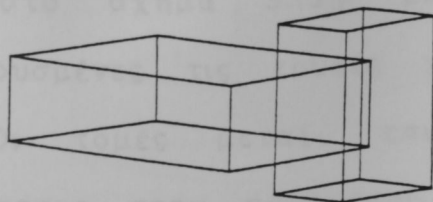
Τέλος το αποτέλεσμα της πράξης δίνει ένα σώμα, το οποίο στο AutoCAD μπορεί να εκφραστεί επίσης με ένα σχεδιαστικό αντικείμενο.

3.9 ΑΦΑΙΡΕΣΗ

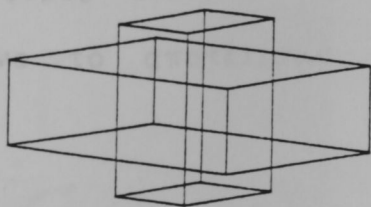
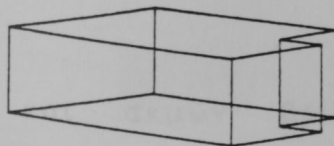
3.9.1 ΑΦΑΙΡΕΣΗ ΔΥΟ ΠΑΡΑΛΛΗΛΕΠΙΠΕΔΩΝ

Στα σχήματα 3.13 α και β βλέπουμε ορισμένες περιπτώσεις αφαίρεσης μεταξύ δύο παραλληλεπιπέδων (στα σχήματα 3.13 γ και δ βλέπουμε τις ίδιες περιπτώσεις με απομακρυσμένες τις κρυφές γραμμές).

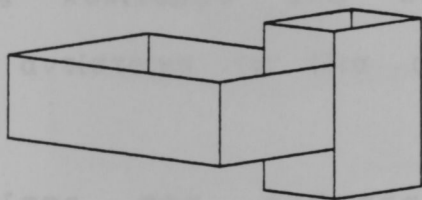
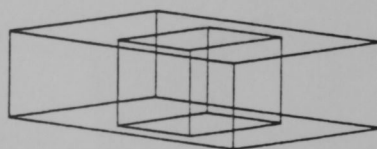
Στην αριστερή πλευρά του σχήματος βλέπουμε τα σώματα που έρχονται σε σχέση και στη δεξιά πλευρά, το αποτέλεσμα της σχέσης. Αριστερά στο σχήμα 3.13.α, βλέπουμε το πρώτο παραλληλεπίπεδο που η μία του γωνία μπαίνει μέσα σε ένα δεύτερο παραλ/πεδο. Επιλέγοντας πρώτα το πρώτο σώμα και μετά το δεύτερο, η εντολή Bool θα σχεδιάσει το σώμα που φαίνεται δεξιά στο σχήμα 3.13.α. Αυτή είναι η αφαίρεση του δεύτερου σώματος από το πρώτο (στο σχήμα 3.13.γ είναι ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Αντίστοιχα αριστερά στο σχήμα 3.13.β βλέπουμε ένα παραλ/πεδο που μέσα του περνάει ένα δεύτερο παραλ/πεδο. Δεξιά στο ίδιο σχήμα φαίνεται η



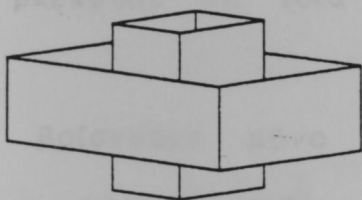
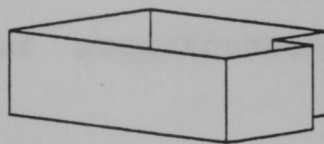
(α)



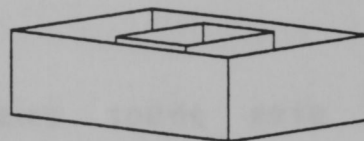
(β)



(γ)



(δ)



Σχήμα 3.13 Αφαίρεση παραλληλεπιπέδων.

αφαίρεση του δεύτερου από το πρώτο όπως δίνεται με την εντολή Bool (στο σχήμα 3.13.8 βλέπουμε ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Οι τομές μεταξύ επιφανειών και ακμών βρίσκονται όπως περιγράφηκε στην παράγραφο 3.5.

Το βασικότερο πρόβλημα που δημιουργείται είναι ότι αρκετές φορές αδυνατεί το AutoCAD να εκφράσει σαν ένα σχεδιαστικό αντικείμενο το αποτέλεσμα της πράξης.

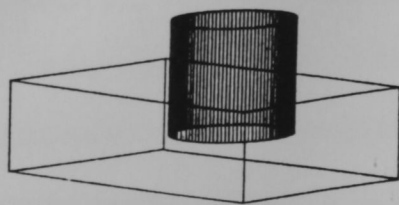
3.9.2 ΑΦΑΙΡΕΣΗ ΠΑΡΑΛΛΗΛΕΠΙΠΕΔΟΥ ΚΥΛΙΝΔΡΟΥ

Στα σχήματα 3.14 α β βλέπουμε ορισμένες περιπτώσεις αφαίρεσης κυλίνδρου από παραλλ/πεδο (στα σχήματα 3.14 γ δ βλέπουμε αντίστοιχα τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

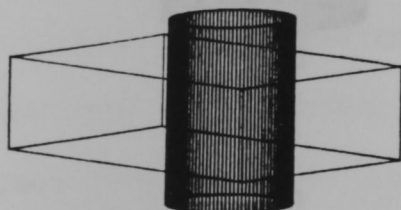
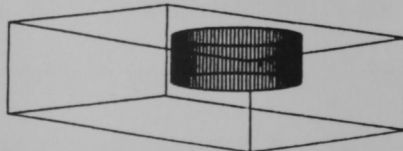
Επίσης στα σχήματα 3.15 α β βλέπουμε ορισμένες περιπτώσεις αφαίρεσης παραλ/πέδου από κύλινδρο (στα σχήματα 3.15 γ δ βλέπουμε τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Εδώ βρίσκουμε μόνο τις επιφάνειες τομής κατά τον άξονα που έγινε η σάρωση (βλέπε παράγραφο 3.5).

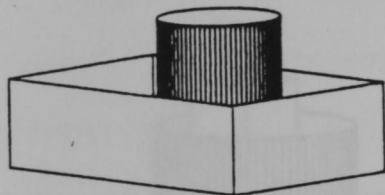
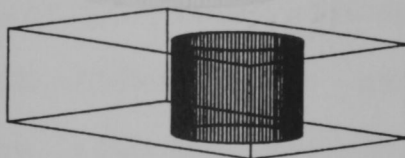
Στο σχήμα 3.14.α αριστερά, βλέπουμε ένα παραλ/πεδο που μέσα του μπαίνει λίγο ένας κύλινδρος. Δεξιά βλέπουμε το αποτέλεσμα της αφαίρεσης του κυλίνδρου από το παραλ/πεδο που δίνει η εντολή Bool



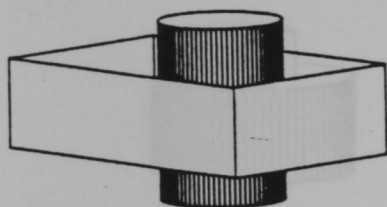
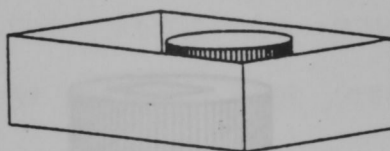
(α)



(β)



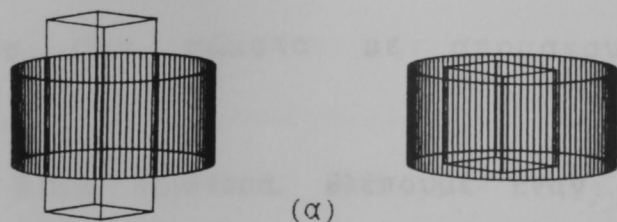
(γ)



(δ)



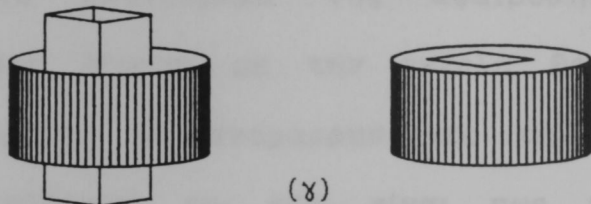
Σχήμα 3.14 Αφαίρεση κυλίνδρου από παραλληλεπίπεδο.



(α)



(β)



(γ)



(δ)

Σχήμα 3.15 Αφαίρεση παραλληλεπιπέδου από κύλινδρο.

(στο σχήμα 3.14.γ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.14.β αριστερά, βλέπουμε ένα παραλ/πεδο που μέσα του περνάει ένας κύλινδρος. Δεξιά στο ίδιο σχήμα βλέπουμε το αποτέλεσμα της αφαίρεσης του κυλίνδρου από το παραλ/πεδο που δίνεται με την εντολή Bool (στο σχήμα 3.14.δ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Στο σχήμα 3.15.α αριστερά, βλέπουμε έναν κύλινδρο που μέσα του περνάει ένα παραλ/πεδο. Δεξιά βλέπουμε το αποτέλεσμα της αφαίρεσης του παραλ/πέδου από τον κύλινδρο που δίνει η εντολή Bool (στο σχήμα 3.15.γ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.15.β αριστερά, βλέπουμε έναν κύλινδρο που μέσα του μπαίνει λίγο ένα παραλ/πεδο. Δεξιά στο ίδιο σχήμα βλέπουμε το αποτέλεσμα της αφαίρεσης του παραλ/πέδου από τον κύλινδρο που δίνεται με την εντολή Bool (στο σχήμα 3.15.δ ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Το βασικό πρόβλημα, και εδώ, είναι πως το παραγόμενο σώμα, δεν μπορούμε να το εκφράσουμε σαν ένα σχεδιαστικό αντικείμενο με το AutoCAD.

3.9.3 ΑΦΑΙΡΕΣΗ ΔΥΟ ΚΥΛΙΝΔΡΩΝ

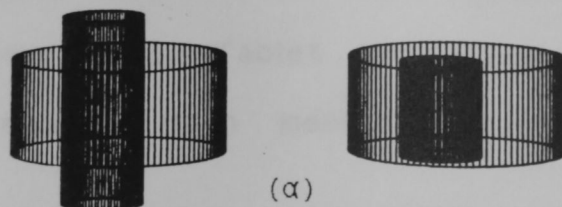
Στα σχήματα 3.16 α και β, βλέπουμε ορισμένες περιπτώσεις αφαίρεσης κυλίνδρου από κύλινδρο (στα σχήματα 3.16 γ και δ φαίνονται τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Εδω βρίσκουμε μόνο τις επιφάνειες τομής κατά τον άξονα που έγινε η σάρωση (βλέπε παράγραφο 3.5).

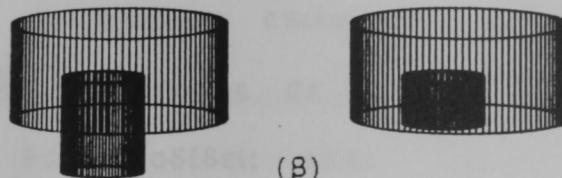
Αριστερά στο σχήμα 3.16.α βλέπουμε έναν κύλινδρο που από μέσα του περνάει ένας δεύτερος κύλινδρος. Δεξιά στο ίδιο σχήμα δίνεται η αφαίρεση του δεύτερου από τον πρώτο όπως την βρίσκει και την σχεδιάζει η εντολή Bool (στο σχήμα 3.16.γ φαίνονται ακριβώς τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές). Στο σχήμα 3.16.β αριστερά βλέπουμε έναν κύλινδρο που στην βάση του έχει μπει λίγο μέσα ένας δεύτερος κύλινδρος. Δεξιά στο ίδιο σχήμα βλέπουμε την αφαίρεση του δεύτερου από τον πρώτο, όπως δίνεται από την εντολή Bool (στο σχήμα 3.16.δ φαίνονται τα ίδια σώματα με απομακρυσμένες τις κρυφές γραμμές).

Και σε αυτή τη διαδικασία αδυνατεί το AutoCAD να εκφράσει το αποτέλεσμα της πράξης με ένα σχεδιαστικό αντικείμενο.

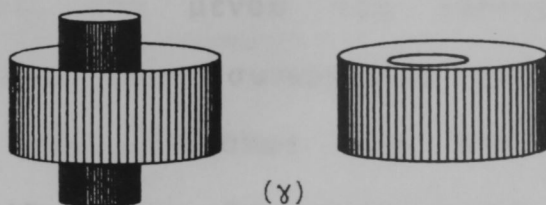
Σχήμα 3.16 Αφαίρεση κυλίνδρων.



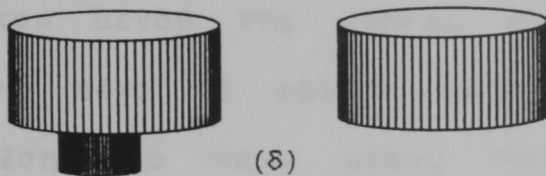
(α)



(β)



(γ)



(δ)

Σχήμα 3.16 Αφαίρεση κυλίνδρων.

3.10 Η ΕΝΤΟΛΗ BOOL ΣΤΑ ΜΕΝΟΥ ΤΟΥ AutoCAD

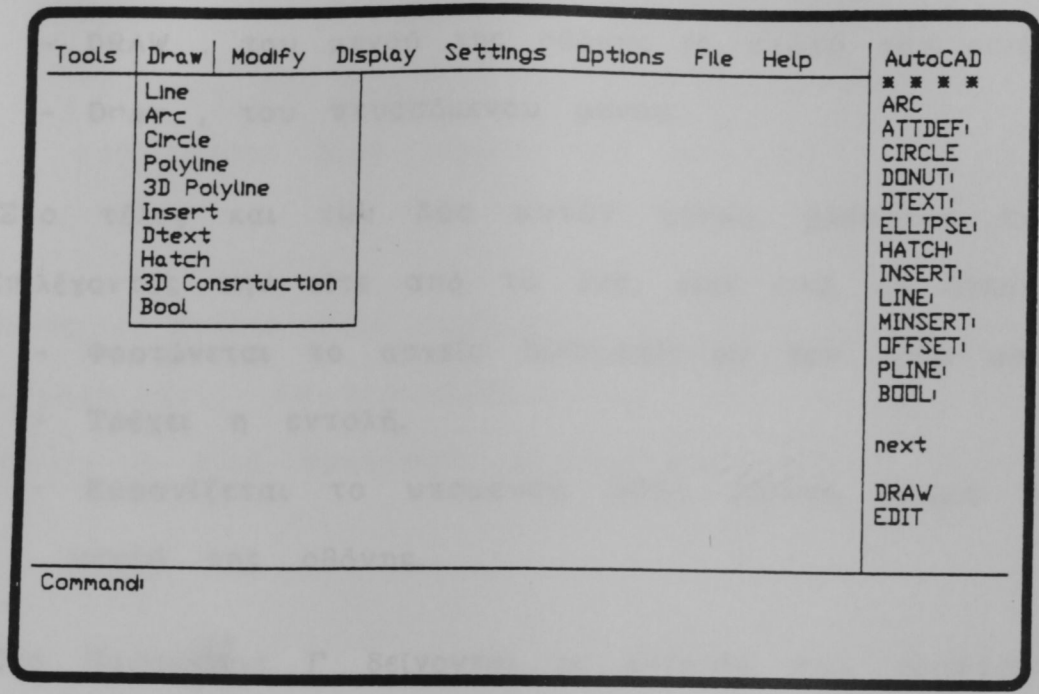
Το AutoCAD υποστηρίζει τέσσερα είδη μενού:

- Μενού οθόνης (Screen menu)
- Πτυσσόμενο μενού (Pull-Down menu)
- Μενού πινακίδας (Tablet menu)
- Μενού εικόνων (Icon menu)

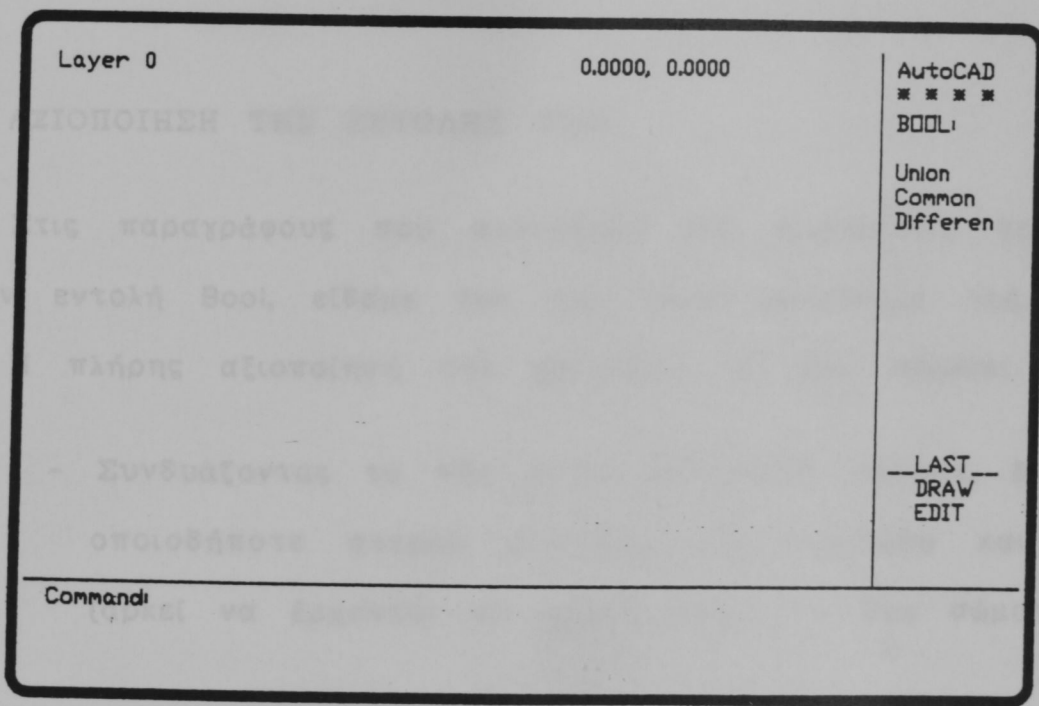
Για την οργάνωση ενός μενού πρέπει να απαντηθούν ερωτήσεις όπως: Σε ποιο ή σε ποια μενού από αυτά που υποστηρίζει το AutoCAD θα μπει η εντολή; θα υπάρχει επικοινωνία μεταξύ των μενού; αν ναι ποια θα είναι αυτή; ο χρήστης σε ποια θέση αν μπει η εντολή θα διευκολύνεται και θα αποδίδει; κλπ.

Στη συνέχεια αυτής της παραγράφου θα δούμε, σύντομα, πώς η εντολή Bool μπαίνει στο μενού της οθόνης και στο πτυσσόμενο μενού, καθώς και μια μορφή συνεργασίας τους.

Στο σχήμα 3.17 βλέπουμε όλη την οθόνη που βλέπει ο χρήστης του AutoCAD με τα δύο αυτά μενού. Συγκεκριμένα δεξιά και εκτός σχεδιαστικής οθόνης, βρίσκεται ένας χώρος που δείχνει κάποιο υπομενού του μενού της οθόνης. Επίσης αν πάμε με το ποντίκι, τη φωτεινή πένα (ή κάποιο άλλο κατάλληλο περιφεριακό) στην κατάλληλη θέση στο πάνω μέρος της σχεδιαστικής οθόνης, μπορούμε να επιλέξουμε πιο υπομενού από το πτυσσόμενο μενού θέλουμε να κατεβάσουμε.



Σχήμα 3.17 Η εντολή BOOL στα μενού του AutoCAD.



Σχήμα 3.18 Το μενού BOOL.

Στο σχήμα 3.17 φαίνονται τα μενού:

- DRAW , του μενού της οθόνης (η πρώτη του σελίδα).
- Draw , του πτυσσόμενου μενού.

Στο τέλος και των δύο αυτών μενού, βλέπουμε την εντολή Bool. Επιλέγοντάς την είτε από το ένα, είτε από το άλλο:

- Φορτώνεται το αρχείο BOOL.LSP αν δεν είναι φορτομένο.
- Τρέχει η εντολή.
- Εμφανίζεται το υπομενού BOOL (βλέπε σχήμα 3.18) στο μενού της οθόνης.

Στο Παράρτημα Γ δίνονται οι εντολές που προστέθηκαν στο αρχείο ACAD.MNU (μενού του AutoCAD) για την δημιουργία της παραπάνω μορφής των μενού.

3.11 ΑΞΙΟΠΟΙΗΣΗ ΤΗΣ ΕΝΤΟΛΗΣ BOOL

Στις παραγράφους που αναλύσαμε τις διαδικασίες που γίνονται με την εντολή Bool, είδαμε την πιο απλή λειτουργία της.

Η πλήρης αξιοποίησή της βρίσκεται σε δύο σημεία:

- Συνδυάζοντας τα δύο αυτά πρωτογενή σώματα, δημιουργούμε οποιοδήποτε στερεό μοντέλο, όσο σύνθετο και αν είναι (αρκεί να έρχονται σε σχέση αυτά τα δύο σώματα).

- Επίσης στη δημιουργία ενός στερεού μοντέλου, είναι προφανές πως μπορούμε να συνδυάσουμε την εντολή Bool με οποιαδήποτε άλλη εντολή του AutoCAD.

1. Νίκος Γ. Σιδηράς.

Τέλος η εντολή Bool παραμένει μια "ανοιχτή εντολή". Μπορεί να εμπλουτιστεί συμπεριλαμβάνοντας κι άλλες περιπτώσεις, συνδυάζοντας κι άλλα πρωτογενή σώματα όπως σφήνες, σφαίρες κλπ.

2. Νίκος Χαϊλαλής.

"Σχεδιασμός και παραγωγή με τη βοήθεια Η/Υ".

σημειώσεις παραδόσεων στο Πολυτεχνείο Κρήτης, Αθήνα 1992.

3. Robert H. Johnson.

"SOLID MODELING: A STATE-OF-THE-ART REPORT", CAD/CIM ALERT, NORTH-HOLLAND, Management Science, Inc., Chestnut Hill, Massachusetts 1985.

4. Martti Mäntylä.

"AN INTRODUCTION TO SOLID MODELING", Computer Science Press, Maryland 1988.

5. George Omura.

"THE ABS's OF AutoLISP", SYBEX INC., Alameda CA 1990.

6. George O. Head, μετάφραση Χρήσιμος Αντωνίου.

"ΜΑΘΕΤΕ ΤΗΝ AutoLISP", με την ελληνική έκδοση Ελκιδόπουλος, Αθήνα 1990.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Ηλίας Γ. Στράγκας,
"ΣΧΕΔΙΑΣΗ ΚΑΙ ΠΑΡΑΓΩΓΗ ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ",
Χανιά 1990.
2. Νίκος Μπιλάλης,
"Σχεδιασμός και παραγωγή με τη βοήθεια Η/Υ",
σημειώσεις παραδόσεων στο Πολυτεχνείο Κρήτης, Χανιά 1989.
3. Robert H. Johnson,
"SOLID MODELING: A STATE-OF-THE-ART REPORT", CAD/CIM ALERT,
NORTH-HOLLAND, Management Roundable, Inc., Chestnut Hil,
Massachusetts 1986.
4. Martti Mantyla,
"AN INTRODUCTION TO SOLID MODELING", Computer Science Press,
Maryland 1988.
5. George Omura,
"THE ABS's OF AutoLISP", SYBEX Inc., Alameda CA 1990.
6. George O. Head, μετάφραση Χρήστος Κωτσοβίλης,
"ΜΑΘΕΤΕ ΤΗΝ AutoLISP", για την ελληνική έκδοση Κλειδάριθμος,
Αθήνα 1990.

7. Autodesk Inc.,

"AutoCAD 10 Reference Manual", Command Technology Corp.,
Oakland CA 1988.

8. Autodesk Inc.,

"AutoLISP 10 Programmer's Reference", Command Technology
Corp., Oakland CA 1988.

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α

Η ΛΙΣΤΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ BOOL.LSP

```

*****
*
*                               BOOL.LSP
*
*       Σπύρος   Ελευσινιώτης   Δεκέμβρης   1990
*
*       Σχέσεις Bool μεταξύ:
*       Παραλληλεπίπεδων - Κυλίνδρων
*
*   Φορτώνεται:
*       Command: (load "bool"),
*               ή από το μενού της οθόνης,
*               ή από το πτυσσόμενο μενού,
*               επιλέγοντας "Bool" από τα υπομενού Draw
*
*   Τρέχει:
*       Command: bool                (ή από τα μενού)
*       Select 1st object:           (επέλεξε ένα)
*       Select 2nd object:           (επέλεξε ένα)
*       Union/Common/Difference:    (επέλεξε σχέση Bool
*                                   πληκτρολογώντας U, C ή D
*                                   ή από το μενού της οθόνης)
*
*   Τα σώματα πρέπει να έχουν σχεδιαστεί με τις εντολές:
*       ELEV, PLINE   για τα παραλληλεπίπεδα
*       ELEV, CIRCLE  για τους κυλίνδρους
*
*****

```

```

Επεξεργασία Βασικών Δεδομένων Του Πρώτου Παραλληλεπίπεδου
-----*****-----
(defun getpol ()
    ; / subent thic1 ver1st vertx1
    ; / pt11 elev1 top1
    (redraw (cdr (assoc -1 fst1st)) 3)
    (setq sub1st (entnext (cdr (assoc -1 fst1st) )))

```

```

(setq thic1 (cdr (assoc 39 (entget sub1st) )))

(repeat 5
  (setq vtx1 (cdr (assoc 10 (entget sub1st) )))
  (setq ver1st (append ver1st (list vtx1) ))
  (setq sub1st (entnext sub1st) )
)
(setq pt11 (car ver1st)
      pt12 (cadr ver1st)
      pt13 (caddr ver1st)
      pt14 (cadddr ver1st)
)
(setq elev1 (caddar ver1st)
      top1 (+ elev1 thic1)
      erpt pt11
)
)
)

; ----- Επεξεργασία Βασικών Δεδομένων Του Δεύτερου Παραλληλεπίπεδου -----
; -----*****-----
(defun get2pol ()
  ; / sub2nd snd1st thic2 ver2nd vtx2
  ; / pt21 pt22 pt23 pt24 elev2 top2 chpt
  (redraw (cdr (assoc -1 snd1st)) 3)
  (setq sub2nd (entnext (cdr (assoc -1 snd1st) )))
  (setq thic2 (cdr (assoc 39 (entget sub2nd) )))
  (repeat 5
    (setq vtx2 (cdr (assoc 10 (entget sub2nd) )))
    (setq ver2nd (append ver2nd (list vtx2) ))
    (setq sub2nd (entnext sub2nd) )
  )
  (setq pt21 (car ver2nd)
        pt22 (cadr ver2nd)
        pt23 (caddr ver2nd)
        pt24 (cadddr ver2nd)
  )
  (setq elev2 (caddr pt21)
        top2 (+ elev2 thic2)
        chpt pt21
  )
)
)

; ----- Επεξεργασία Βασικών Δεδομένων Του Πρώτου Κυλίνδρου -----
; -----*****-----
(defun getcir ()
  (redraw (cdr (assoc -1 fst1st)) 3)
  (setq cpt1 (cdr (assoc 10 fst1st))
        rd1 (cdr (assoc 40 fst1st))
  )
)

```



```

        thic1 (cdr (assoc 39 fst1st))
    )
    (setq elev1 (caddr cpt1)
          top1 (+ elev1 thic1)
    )
    (setq erpt (append erpt
        (list (+ rd1 (car cpt1)))
        (list (cadr cpt1))
        (list elev1)
    )
    )
)
)

; -----Επεξεργασία Βασικών Δεδομένων Του Δεύτερου Κυλίνδρου-----
; -----*****-----
(defun get2cir ()
    ; / cpt2 snd1st rd2 thic2 elev2 top2
    ; / ncp top1 chpt
    (redraw (cdr (assoc -1 snd1st)) 3)
    (setq cpt2 (cdr (assoc 10 snd1st))
          rd2 (cdr (assoc 40 snd1st))
          thic2 (cdr (assoc 39 snd1st))
    )
    (setq elev2 (caddr cpt2)
          top2 (+ elev2 thic2)
    )
    (setq ncp nil)
    (setq ncp (append ncp
        (list (car cpt2))
        (list (cadr cpt2))
        (list top1)
    )
    )
    (setq chpt (append chpt
        (list (+ rd2 (car cpt2)))
        (list (cadr cpt2))
        (list elev2)
    )
    )
)
)

```

```

; -----Εύρεση Των Σημείων Τομής Των Παραλληλεπιπέδων-----
; -----*****-----
(defun polpol ()
    (setq t1 nil t2 nil
          t3 nil t4 nil
          t5 nil t6 nil
          t7 nil t8 nil
          t9 nil t10 nil
    )
)

```



```

        (if (= kwd "Difference")
            (command "erase" pt11 "")
        )
        (varset)
        (setq *error* olderr)
        (princ)
    )

    ( (not (equal fstlst sndlst))
      (if (= kwd "Union") (druni) )
      (if (= kwd "Common") (drcom) )
      (if (= kwd "Difference") (drcut) )
    )
  )
)

; -----
; Σχεδίαση Του Ζου Κάτω Από Το 1ο
; -----
(defun drunder ()
    ; / elev2 elev1 sndlst cpt2 rd2
    ; / pt21 pt22 pt23 pt24
    (command "elev" elev2 (- elev1 elev2))

    (cond ( (= (cdr (assoc 0 sndlst)) "CIRCLE")
            (command "circle" cpt2 rd2)
          )
          )
    (cond ( (= (cdr (assoc 0 sndlst)) "POLYLINE")
            (command "pline" pt21 pt22 pt23 pt24 "c")
          )
          )
)

; -----
; Σχεδίαση Του Ζου Πάνω Από Το 1ο
; -----
(defun drabove ()
    ; / top1 top2 sndlst ncp cpt2 rd2
    ; / npt21 npt22 npt23 npt24
    ; / pt21 pt22 pt23 pt24
    (command "elev" top1 (- top2 top1))

    (cond ( (= (cdr (assoc 0 sndlst)) "CIRCLE")
            (setq ncp nil)
          )
          )
    (defun drd (setq ncp (append ncp (list (car cpt2))
                                         (list (cadr cpt2))
                                         (list top1)
                                         )
              )
          )
    (command "circle" ncp rd2)
)

```



```

;-----
(defun drupin () ; / chpt elev2 top1
  (command "change" chpt ""
    "p" "e" elev2 "t" (- top1 elev2) "")
)

; Σχεδίαση Της Ένωσης του 1 2ου
;-----*****-----
(defun druni () ; / chpt elev1 elev2 top1 top2
  (command "erase" chpt "")

  (cond ( (> elev1 elev2) (drunder))
    )
  (cond ( (< top1 top2) (drabove))
    )
  (if (= p1plt "PLPLT")
    (polout1)
  )
)

; Σχεδίαση Της Ένωσης Δύο Παράλληλες Πεδών
; Όταν Υπάρχει Πλάγια Τομή
;-----*****-----
(defun polout1 ()

  (cond ( (and (< elev2 elev1) (< top1 top2))
    (command "elev" elev1 thic1)
  )
  ( (and (< elev2 elev1) (< top2 top1))
    (command "elev" elev1 (- top2 elev1))
  )
  ( (and (< elev1 elev2) (< top1 top2))
    (command "elev" elev2 (- top1 elev2))
  )
)

;----- Όταν Η Τομή Είναι Σε Μία Πλευρά -----
;-----
(defun (cond ( (and (/= t2 nil) (/= t4 nil))
  (setvar "flatland" 1)
  (command "pline" t4 pt21 pt22 t2 "c")
  (setvar "flatland" 0)
)
  ( (and (/= t5 nil) (/= t7 nil))
    (setvar "flatland" 1)
    (command "pline" t5 pt22 pt23 t7 "c")
    (setvar "flatland" 0)
  )
)

```

```

    ( (and (/= t10 nil) (/= t12 nil))
      (setvar "flatland" 1)
      (command "pline" t12 t10 pt23 pt24 "c")
      (setvar "flatland" 0)
    )
    ( (and (/= t13 nil) (/= t15 nil))
      (setvar "flatland" 1)
      (command "pline" pt21 t13 t15 pt24 "c")
      (setvar "flatland" 0)
    )
  )
)

```

----- Όταν Η Τομή Είναι Σε Δύο Πλευρές (Γωνία) -----

```

(cond ( (and (/= t2 nil) (/= t15 nil))
  (setvar "flatland" 1)
  (command "pline" pt21 pt22 t2 pt11 t15 pt24 "c")
  (setvar "flatland" 0)
)
  ( (and (/= t4 nil) (/= t7 nil))
    (setvar "flatland" 1)
    (command "pline" pt21 pt22 pt23 t7 pt12 t4 "c")
    (setvar "flatland" 0)
  )
  ( (and (/= t5 nil) (/= t12 nil))
    (setvar "flatland" 1)
    (command "pline" t5 pt22 pt23 pt24 t12 pt13 "c")
    (setvar "flatland" 0)
  )
  (cond ( (and (/= t10 nil) (/= t13 nil))
    (setvar "flatland" 1)
    (command "pline" pt21 t13 pt14 t10 pt23 pt24 "c")
    (setvar "flatland" 0)
  )
  )
)
)

```

; Σχεδίαση Της Τομής 1ου & 2ου

;-----*****-----

```

(defun drcom () ; / erpt elev2 elev1 top1 top2
  (cond ( (/= plplt "PLPLT")
    (command "erase" erpt "")
  )
  (cond ( (and (< elev2 elev1) (< top1 top2) )
    (drallin) )
  )
  (cond ( (and (< elev2 elev1) (> top1 top2) )
    (drdwin) )
  )
  )
)

```

```

        (cond ( (and (< top1 top2) (> elev2 elev1) )
                (drupin) )
              )
      )
      ( (= plplt "PLPLT")
        (plplcom)
      )
    )
  )
)

```

```

; Σχεδίαση Της Τομής Δύο Παραλληλεπipedων
; Όταν Υπάρχει Πλάγια Τομή
; -----*****-----

```

```

(defun plplcom ()
  (cond ( (and (< elev2 elev1) (< top1 top2))
          (command "elev" elev1 thic1)
        )
        ( (and (< elev2 elev1) (< top2 top1))
          (command "elev" elev1 (- top2 elev1))
        )
        ( (and (< elev1 elev2) (< top1 top2))
          (command "elev" elev2 (- top1 elev2))
        )
      )
)

```

```

; ----- Όταν Η Τομή Είναι Σε Μία Πλευρά -----

```

```

  (cond ( (and (/= t2 nil) (/= t4 nil))
          (command "erase" pt11 pt21 "")
          (setvar "flatland" 1)
          (command "pline" t4 t2 pt23 pt24 "c")
          (setvar "flatland" 0)
        )
        ( (and (/= t5 nil) (/= t7 nil))
          (command "erase" pt11 pt21 "")
          (setvar "flatland" 1)
          (command "pline" pt21 t5 t7 pt24 "c")
          (setvar "flatland" 0)
        )
        ( (and (/= t10 nil) (/= t12 nil))
          (command "erase" pt11 pt21 "")
          (setvar "flatland" 1)
          (command "pline" pt21 pt22 t10 t12 "c")
          (setvar "flatland" 0)
        )
        ( (and (/= t13 nil) (/= t15 nil))
          (command "erase" pt11 pt21 "")
          (setvar "flatland" 1)
        )
      )
)

```



```

(plpldif)
)
)
)

; Σχεδίαση Της Αφαίρεσης Παραλληλεπιδέδων
; Όταν Υπάρχει Πλάγια Τομή
; -----*****-----
(defun plpldif ()
  (cond ((and (< elev2 elev1) (< top1 top2))
    (command "elev" elev1 thic1)
  )
    ((and (< elev2 elev1) (< top2 top1))
    (command "elev" elev1 (- top2 elev1))
  )
    ((and (< elev1 elev2) (< top1 top2))
    (command "elev" elev2 (- top1 elev2))
  )
  )
)

; ----- Όταν Η Τομή Είναι Σε Μία Πλευρά -----
(defun init ()
  (cond ((and (/= t2 nil) (/= t4 nil))
    (command "erase" pt21 "")
    (setvar "flatland" 1)
    (command "pline" t4 t2 pt23 pt24 "c")
    (setvar "flatland" 0)
  )
    ((and (/= t5 nil) (/= t7 nil))
    (command "erase" pt21 "")
    (setvar "flatland" 1)
    (command "pline" pt21 t5 t7 pt24 "c")
    (setvar "flatland" 0)
  )
    ((and (/= t10 nil) (/= t12 nil))
    (command "erase" pt21 "")
    (setvar "flatland" 1)
    (command "pline" pt21 pt22 t10 t12 "c")
    (setvar "flatland" 0)
  )
    ((and (/= t13 nil) (/= t15 nil))
    (command "erase" pt21 "")
    (setvar "flatland" 1)
    (command "pline" t13 pt22 pt23 t15 "c")
    (setvar "flatland" 0)
  )
  )
)

```

----- Όταν Η Τομή Είναι Σε Δύο Πλευρές -----

```
(cond ((and (/= t2 nil) (/= t15 nil))
      (command "erase" pt11 pt21 "")
      (setvar "flatland" 1)
      (command "pline" t2 pt12 pt13 pt14 t15 pt23 "c")
      (setvar "flatland" 0)
    )
      ((and (/= t4 nil) (/= t7 nil))
      (command "erase" pt11 pt21 "")
      (setvar "flatland" 1)
      (command "pline" pt11 t4 pt24 t7 tp13 pt14 "c")
      (setvar "flatland" 0)
    )
      ((and (/= t5 nil) (/= t12 nil))
      (command "erase" pt11 pt21 "")
      (setvar "flatland" 1)
      (command "pline" pt11 pt12 t5 pt21 t12 pt14 "c")
      (setvar "flatland" 0)
    )
      ((and (/= t10 nil) (/= t13 nil))
      (command "erase" pt11 pt21 "")
      (setvar "flatland" 1)
      (command "pline" pt11 pt12 pt13 t10 pt22 t13 "c")
      (setvar "flatland" 0)
    )
  )
)
(setq var1st nil
      var2nd nil)
```

Τοποθέτηση Αρχικών Τιμών Στις Βασικές Μεταβλητές
Περιγραφή Βασικών Μεταβλητών

-----*****-----

(defun initial ()

----- Για Το Πρώτο Σώμα -----

```
(setq fstlst nil      ; λίστα δεδομένων

      thic1 nil      ; ύψος
      elev1 nil      ; υψόμετρο βάσης
      top1 nil       ; υψόμετρο κορυφής

      sublst nil     ; υπολίστα παραλληλεπίπεδου
      vtx1 nil       ; λίστα κορυφών βάσης παρα/πέδου

      pt11 nil       ; κορυφή βάσης παρα/πέδου
      pt12 nil       ; >>
      pt13 nil       ; >>
      pt14 nil       ; >>)
```

```

      cpt1    nil      ; κέντρο βάσης κυλίνδρου
      rd1     nil      ; ακτίνα βάσης κυλίνδρου
    )

;----- Για Το Δεύτερο Σώμα -----

(setq  sndlst  nil      ; λίστα δεδομένων

      thic2    nil      ; ύψος
      elev2    nil      ; υψόμετρο βάσης
      top2     nil      ; υψόμετρο κορυφής

      sub2nd   nil      ; υπολίστα παραλληλεπίπεδου
      vtx2     nil      ; λίστα κορυφών βάσης παρα/πέδου

      pt21     nil      ; κορυφή βάσης παρα/πέδου
      pt22     nil      ; >>
      pt23     nil      ; >>
      pt24     nil      ; >>

      cpt2     nil      ; κέντρο βάσης κυλίνδρου
      rd2      nil      ; ακτίνα βάσης κυλίνδρου
    )

;----- Βοηθητικές Μεταβλητές -----

(setq  ver1st  nil
      ver2nd  nil

      fstobj  nil
      sndobj  nil

      chpt    nil
      erpt    nil
      ncp     nil

      npt21   nil
      npt22   nil
      npt23   nil
      npt24   nil

      plplt   nil
    )

;-----
*** ΚΕΝΤΡΙΚΟ ΠΡΟΓΡΑΜΜΑ ** ΟΜΟΙΟΥΡΓΙΑ ΤΗΣ ΕΝΤΟΛΗΣ ***
;-----

Αποθήκευση Τιμών
Των Μεταβλητών Του Συστήματος
;-----*****-----

```

```

(defun varget ()
  (setq vosm (getvar "osmode")
    vorth (getvar "orthomode")
    vcmd (getvar "cmdecho")
    vblip (getvar "blipmode")
    velev (getvar "elevation")
    vthick (getvar "thickness")
  )
)

; Επανεναποθέτηση Αρχικών Τιμών
; Των Μεταβλητών Του Συστήματος
; -----*****-----
(defun varset ()
  (setvar "osmode" vosm)
  (setvar "orthomode" vorth)
  (setvar "cmdecho" vcmd)
  (setvar "blipmode" vblip)
  (setvar "elevation" velev)
  (setvar "thickness" vthick)
)

; Έξοδος σε περίπτωση εσωτερικού λάθους
; -----*****-----
(defun boolerr (s)
  (princ (chr 7))
  (if (/= s "Function cancelled")
    (princ (strcat "\nError: " s))
  )
  (if (/= fstlst nil)
    (redraw (cdr (assoc -1 fstlst)))
  )
  (if (/= sndlst nil)
    (redraw (cdr (assoc -1 sndlst)))
  )
  (varset)
  (setq *error* olderr)
  (princ)
)

; *****
; *** ΚΕΝΤΡΙΚΟ ΠΡΟΓΡΑΜΜΑ ** ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΕΝΤΟΛΗΣ ***
; *****
(cond (and (or (fstlst) nil) (or (sndlst) nil))
  (defun C:BOOL ()
    ; / fstlst sndlst
    (cdr (assoc 0 sndlst)) "POLYLINE")
  )
)

```



```

(setq olderr *error*
      *error* boolerr
)
(varget)

(initial)

(mapcar 'setvar '( "osmode"
                  "orthomode"
                  "cmdecho"
                  "blipmode"
                  )
      '( 0 0 0 0 )
)

(setq fstlst (entget (car (entsel
                          "\nSelect 1st object: "))))
)
(cond ( (= (cdr (assoc 0 fstlst)) "CIRCLE")
      (getcir)
      )
      ( (= (cdr (assoc 0 fstlst)) "POLYLINE")
      (getpol)
      )
      ( (/= (cdr (assoc 0 fstlst))
            (or "CIRCLE" "POLYLINE"))
      )
      )
(setq fstobj "First object is not Cylinder or Box")
)
)
(vreset)
(princ)
(setq sndlst (entget (car (entsel
                          "\nSelect 2nd object: "))))
)
(cond ( (= (cdr (assoc 0 sndlst)) "CIRCLE")
      (get2cir)
      )
      ( (= (cdr (assoc 0 sndlst)) "POLYLINE")
      (get2pol)
      )
      ( (/= (cdr (assoc 0 sndlst))
            (or "CIRCLE" "POLYLINE"))
      )
      )
(setq sndobj "Second object is not Cylinder or Box")
)
)

(cond ( (and (= fstobj nil) (= sndobj nil))
      (if (and (= (cdr (assoc 0 fstlst)) "POLYLINE")
                (= (cdr (assoc 0 sndlst)) "POLYLINE"))
          )
      )

```

```

    )
    (polpol)
)

(srela)

)
( (and (/= fstobj nil) (/= sndobj nil))
  (redraw (cdr (assoc -1 fstlst)))
  (redraw (cdr (assoc -1 sndlst)))
  (prompt "\nObjects are not Cylinders or Boxes")
)

( (/= fstobj nil)
  (redraw (cdr (assoc -1 fstlst)))
  (redraw (cdr (assoc -1 sndlst)))
  (prompt "\n")
  (princ fstobj)
)

( (/= sndobj nil)
  (redraw (cdr (assoc -1 fstlst)))
  (redraw (cdr (assoc -1 sndlst)))
  (prompt "\n")
  (princ sndobj)
)

)
(setq *error* olderr)

(varset)
(princ)
)

; ***** Τ Ε Λ Ο Σ Π Ρ Ο Γ Ρ Α Μ Μ Α Τ Ο Σ *****
; *****
;
71 : κωδικός κειμένου (κανονικό ή κατοπτρικό).
72 : κωδικός τοποθέτησης κειμένου.
73 : πυκνότητα κλέγματος στην (M) διεύθυνση.
74 : πυκνότητα κλέγματος στην (N) διεύθυνση.
75 : επιφάνεια προσομοίωσης (τετραγωνική ή κυβική
    B-Spline, Bezier).

```

ΠΑΡΑΡΤΗΜΑ Β

Κωδικοί ενότητας "POLYLINE" :

- 1 : όνομα ενότητας.
- 0 : τύπος ενότητας, "POLYLINE" πάντα.
- 8 : στρώση που σχεδιάστηκε το πολύγραμμο.
- 66 : 1 αν υπάρχουν κορυφές.
- 10 : (0.0 0.0 0.0) πάντα
- 39 : ύψος παράλληλης μετατόπισης, σε περίπτωση παράλληλης σάρωσης.
- 70 : τύπος πολύγραμμου (κλειστό, ανοιχτό κλπ.).
- 40 : αρχικό πάχος πολύγραμμου.
- 41 : τελικό πάχος πολύγραμμου.
- 210 : διάνυσμα διεύθυνσης. Φανερώνει το σύστημα αναφοράς που σχεδιάστηκε το πολύγραμμο.
- 71 : κωδικός κειμένου (κανονικό ή κατοπτρικό).
- 72 : κωδικός τοποθέτησης κειμένου.
- 73 : πυκνότητα πλέγματος στην (M) διεύθυνση.
- 74 : πυκνότητα πλέγματος στην (N) διεύθυνση.
- 75 : επιφάνεια προσαρμογής (τετραγωνική ή κυβική B-Spline, Bezier).

Κωδικοί ενότητας "VERTEX" :

- 1 : όνομα ενότητας.
- 0 : τύπος ενότητας, "VERTEX" πάντα.
- 8 : στρώση που σχεδιάστηκε η κορυφή.
- 39 : ύψος μετατόπισης σε περίπτωση
παράλληλης σάρωσης.
- 10 : συντεταγμένες κορυφής.
- 40 : αρχικό πάχος γραμμής.
- 41 : τελικό πάχος γραμμής.
- 70 : τύπος κορυφής (υπήρχε στο πολύγραμμο,
δημιουργήθηκε μετά κλπ.).
- 50 : γωνία περιστροφής της κορυφής.

Πρώτη σελίδα του υπομνήματος της οθόνης, που
περιέχει την εντολή Bool:

```
**DR 3
(ARC)SS-X SS-180
(ATTDEF)SS-X SS-ATTDEF
(CIRCLE)SS-X SS-CIRCLE
(DONUT)SS-X SS-DONUT
(DTEXT)SS-X SS-DTEXT
(ELLIPSE)SS-X SS-ELLIPSE
(HATCH)SS-X SS-HATCH
(INSERT)SS-X SS-INSERT
(LINE)SS-X SS-LINE
(MINSERT)SS-X SS-MINSERT
```


ΠΑΡΑΡΤΗΜΑ Γ

Υπομενού Draw του πτυσσόμενου μενού, που περιέχει την εντολή Bool:

```
***POP2
[Draw]
[Line]^C^C$S=X $s=line line
[Arc]^C^C$S=X $s=poparc arc
[Circle]^C^C$S=X $s=popcircl circle
[Polyline]^C^C$S=X $s=pline pline
[3D Polyline]^C^C$S=X $S=3dpoly 3dpoly
[Insert]^C^Csetvar attdia 1 $S=X $s=insert insert
[Dtext]^C^C$S=X $s=Dtext Dtext
[Hatch...]^C^C$i=hatch1 $i=*
[3D Construction...]$i=3dobjects $i=*
[Bool]$S=X $S=bool ^C^C^P(if (not c:bool))+
(load "bool")(princ));bool
```

Πρώτη σελίδα του υπομενού DRAW του μενού της οθόνης, που περιέχει την εντολή Bool:

```
**DR 3
[ARC]$S=X $S=ARC
[ATTDEF:]$S=X $S=ATTDEF ^C^CATTDEF
[CIRCLE]$S=X $S=CIRCLE
[DONUT:]^C^CDONUT
[DTEXT:]$S=X $S=DTEXT ^C^CDTEXT
[ELLIPSE:]$S=X $S=ELLIPSE ^C^CELLIPSE
[HATCH:]$S=X $S=HATCH ^C^CHATCH
[INSERT:]$S=X $S=INSERT ^C^CINSERT
[LINE:]$S=X $S=LINE ^C^CLINE
[MINSERT:]$S=X $S=MINSERT ^C^CMINSERT
```

```

[OFFSET:]$S=X $S=OFFSET ^C^COFFSET
[PLINE:]$S=X $S=PLINE ^C^CPLINE
[BOOL:]$S=X $S=BOOL ^C^C^P(if (not c:bool)+
(load "bool")(princ));bool

[next]$S=X $S=DR2

```

ακεραίοι = integers

αλφαριθμητικά = strings

Υπομενού BOOL του μενού της οθόνης:

αριθμητικό έλεγχος = Numerical Control (NC)

```

**BOOL 3
[BOOL:]^C^C^P(if (not c:bool)+
(load "bool")(princ));bool

```

διπλή ακρίβεια = double precision

[Union]U

[Common]C

[Differen]D

floating-point

ειδικός αρχείο = file descriptors

εμπειρικά συστήματα = expert systems

ένωση = union

επεξεργασία παρτίδας = batch processing

ερμηνευτής = interpreter

ημίσφαιρα = half-space

κορυφή = vertex

μενού πικτόνων = icon menu

μενού οθόνης = screen menu

μενού πινακίδας = tablet menu

ομάδες επιλογών = selection sets

ονόματα οντοτήτων = entity names

οριακή αναπαράσταση = Boundary Representation (BR, B-rep)

ΠΑΡΑΡΤΗΜΑ Δ

ΕΛΛΗΝΟ-ΑΓΓΛΙΚΗ ΜΕΤΑΦΡΑΣΗ ΟΡΩΝ


ακέραιοι = integers
αλφαριθμητικά = strings
απότμηση = chamfer
αριθμητικός έλεγχος = Numerical Control (NC)
αφαίρεση, διαφορά = difference
Διπλής ακρίβειας κινητής υποδιαστολής = double precision
floating-point
Δείκτες αρχείων = file descriptors
έμπειρα συστήματα = expert systems
ένωση = union
επεξεργασία δέσμης = batch processing
ερμηνευτής = interpreter
ημίχωρος = half-space
κορυφή = vertex
μενού εικόνων = icon menu
μενού οθόνης = screen menu
μενού πινακίδας = tablet menu
ομάδες επιλογών = selection sets
ονόματα ενοτήτων = entity names
οριακή αναπαράσταση = Boundary Representation (BR, B-rep)

πεπερασμένα στοιχεία = finite elements
πολύγραμμο = polyline
πραγματικοί αριθμοί = real numbers
προθεματικός συμβολισμός = prefix notation
πρωτογενής = primitive
πτυσσόμενο μενού = pull-down menu
σάρωση = sweeping
στερεά απεικόνιση, στερεά μοντέλα = solid modeling
στρώση = layer
σύμβολα = symbols
συναρτήσεις = subrs
συνδυαστική στερεά απεικόνιση = Constructive Solid Geometry (CSG)
συρματοειδή μοντέλα = wire-frame modeling
σύστημα στερεάς απεικόνισης = solid modeler
σχεδιασμός και παραγωγή με τη βοήθεια υπολογιστών = Computer
Aided Design - Computer Aided Manufacturing (CAD-CAM)
σχεδιασμός καταργασιών με τη βοήθεια υπολογιστών = Computer
Aided Process Planing (CAPP)
σχεδιομελέτη (-τής) = design (-er)
τεχνητή νοημοσύνη = Artificial Intelligence (AI)
τομή = common
υποδιαίρεση σε στοιχεία = spatial subdivision
ύψος (πάχος) = thickness
φιλέτο, στρογκύλεμα = fillet

ΠΑΡΑΡΤΗΜΑ Ε

ΚΑΤΑΤΕΘΕΝΤΑ ΣΗΜΑΤΑ

(TRADEMARKS)



AutoCAD, Autodesk, AutoLISP της Autodesk AG
CADD54X της Prime-Computervision
EUCLIDIS, M-DTV της Marta Datavision (Group)
INSIGHT της Phoenix Data Systems
Intel, 8086, 8087, 8088, 80386 της Intel Corporation
Macintosh II της Apple Computer, Inc.
MS-DOS της Microsoft Corporation
PC-DOS, OS/2 της International Business Machines Corporation
UNIX των AT&T Bell Laboratories

