

**UNIVERSITATEA TEHNICĂ DIN TIMIȘOARA
FACULTATEA DE MECANICĂ**

ing. GERALDINE OGÂRCIN

Teză de doctorat

CONDUCĂTOR ȘTIINȚIFIC

prof. dr. ing. GEORGE SAVII

UNIVERSITATEA "POLITEHNICĂ" TIMIȘOARA BIBLIOTECA CENTRALĂ Nr. volum _____ Dulap _____

2001

**UNIVERSITATEA TEHNICĂ DIN TIMIȘOARA
FACULTATEA DE MECANICĂ**

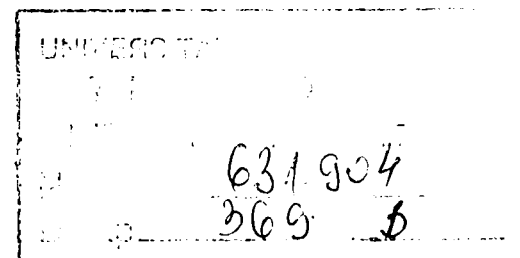
Teză de doctorat

**METODE DE PROIECTARE ORIENTATE PE PROCES
PENTRU SISTEMELE 3D CAD/CIM DE ÎNALTĂ PERFORMANȚĂ**

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA

Autor: ing. Geraldine Ogârcin

Conducător științific: prof. dr. ing. George Savii



Prefața

Scopul tezei de doctorat este de a prezenta un nou mod în care se poate face reprezentarea formelor, cum se pot coda și executa prelucrările mașină folosite în cadrul construcțiilor moderne în lemn.

Elementul de bază în cadrul construcțiilor este bara din lemn de formă paralelipipedică denumită pe parcursul tezei *corp*. Pentru a se construi structura de rezistență a unei clădiri corpurile trebuie modelate astfel încât să se îmbine (munca clasică a dulgherului). În tehnologiile moderne toate modelările efectuate asupra corpurilor sunt codate sub formă de prelucrări mașină pentru a putea fi executate automat de către sisteme automate (robotizate).

Aplicațiile CAD/CAM moderne generează pentru fiecare corp din construcție o secvență de prelucrări mașină. În acest mod un corp complet nefinisat, în urma efectuării de către roboți (sistemele automate de prelucrare cu comandă numerică) a prelucrărilor mașină primește forma dorită de proiectant.

În prima parte se prezintă metode generale de modelare geometrică și vizualizare a obiectelor.

În a doua parte se prezintă codarea prelucrărilor mașină sub formă parametrică și în puncte în scopul transmiterii lor la roboți rapid și cu precizie mare oricât de complicate ar fi ele. Produsul software care realizează aceste coduri este încadrat într-un CAD de proiectare pentru produse din lemn.

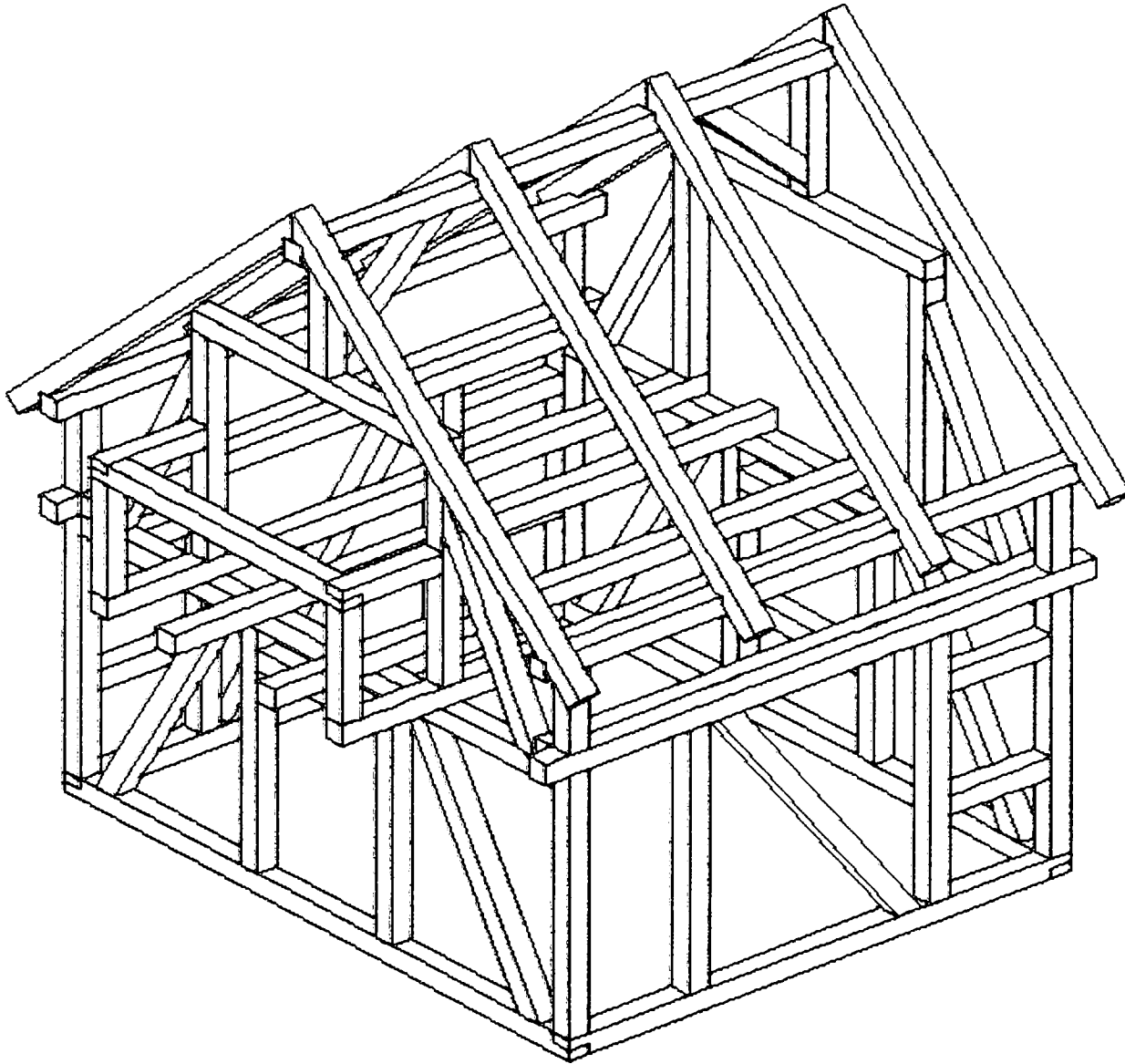
În a treia parte sunt descrise prelucrări specifice de îmbinare a două corpuri împreună cu metodele de generare a codurilor mașină. Descrierea cuprinde și pseudo-codul algorimilor utilizați.

În a patra parte sunt descriși algoritmi pentru efectuarea unei operațiilor booleene, utilizate în efectuarea prelucrărilor, atât pe corpuri convexe cât și pe corpuri concave. Prin noțiunea de *operații booleene* se definesc cele mai elementare operații asupra corpurilor 3D: reuniunea, intersecția și diferența corpurilor. S-au prezentat și funcții matematice utilizate în cadrul operației booleene împreună cu un pseudo-cod al lor.

În a cincea parte se prezintă problemele care au apărut pe parcursul implementării funcțiilor prezentate în părțile anterioare și cum s-a încercat rezolvarea lor.

În anexe sunt prezentate codurile sursă ale unor funcții descrise în cadrul tezei și figuri ale unei construcții în lemn proiectate cu ajutorul mediului CAD în care sunt încadrați algoritmi prezentați.

În figura următoare se prezintă o construcție tipică din lemn care demonstrează utilitatea efectuării prelucrărilor prezentate pe parcursul tezei.



Prezenta lucrare nu ar fi putut elaborată fără sprijinul întregului colectiv din care fac parte, întemeiat și condus mulți ani cu profesionalism și competență de regretatul prof. dr. ing. Ioan Mureșan. Multe din ideile asupra cărora conținutul tezei s-a axat se datorează sugestiilor dumnealui în perioada de pregătire a examenelor și a referatelor.

Deasemenea, mulțumesc prof. dr. ing. Toma L. Dragomir pentru observațiile pertinente și sfaturile date referitoare la prezentarea prezentei teze.

Cuprins

1	MODELARE ȘI VIZUALIZARE	1
1.1	Modelare geometrică	1
1.1.1	Descrierea obiectelor și proceselor	1
1.1.2	Modelarea - principala componentă a CAD	2
1.1.3	Modelul produsului material	3
1.1.4	Metode și mijloace pentru realizarea formelor complexe	4
1.1.4.1	Introducere	4
1.1.4.2	Tehnici de descriere a modelelor geometrice 3D	5
1.2	Vizualizarea modelelor	13
1.2.1	Introducere	13
1.2.2	Linii ascunse și suprafețe ascunse	16
1.3	Concluzii. Contribuții personale	21
2	CODAREA PRELUCRĂRILOR INDIVIDUALE	22
2.1	Introducere. Elemente de bază	23
2.1.1	Forma în puncte	26
2.1.2	Forma parametrică	27
2.1.3	Menținerea muchiilor și fețelor	28
2.1.4	Funcții necesare efectuării prelucrărilor	28
2.2	Prezentarea prelucrărilor individuale	29
2.1.1	Grupa prelucrărilor 4001	31
2.1.1.1	4001, 1 - Tăiere transversală	31
2.1.1.2	4001, 2 – Tăiere longitudinală	33
2.1.2	Grupa prelucrărilor 4002	35
2.1.2.1	4002, 1 – Chertare simplă dreaptă	35
2.1.1.2	4002, 2 – Chertare simplă strâmbă	37
2.1.1.3	4002, 3 – Chertare simplă de capăt	39
2.1.1.4	4002, 4 – Chertare simplă în lungime	40
2.1.1.5	4002, 5 – Tăiere longitudinală dublă concavă	41
2.1.1.6	4002, 6 – Tăiere longitudinală dublă convexă	42
2.1.1.7	4002, 7 – Tăiere în V	44
2.1.3	Grupa prelucrărilor 4003	47
2.1.1.1	4003, 1 – Eclisare standard dreaptă	48
2.1.1.2	4003, 2 – Eclisare standard strâmbă	49

2.1.1.3	4003, 3 – Eclisare de-a lungul corpului	51
2.1.4	Grupa prelucrărilor 4004	53
2.1.1.1	4004, 1 – Chertare colț oarecare	53
2.1.1.2	4004, 2 – Chertare colț dreaptă	54
2.1.5	Grupa prelucrărilor 4005	55
2.1.1.1	4005, 1 - Chertare simplă de-a lungul corpului limitată	56
2.1.1.2	4005, 2 – Gaură de cep deschisă	56
2.1.6	Grupa prelucrărilor 4006	59
2.1.6.1	4006, 1 – Gaură de cep	60
2.1.1.2	4006, 2 – Canal limitat	61
2.1.1.3	4006, 3 – Gaură de cep oarecare	63
2.1.1.4	4006, 4- Gaură de cep coadă de rândunică de capăt	65
2.1.1.5	4006, 5 – Gaură de cep coadă de rândunică laterală	66
2.1.7	Grupa prelucrărilor 4007	68
2.1.7.1	4007, 1 – Cep drept	68
2.1.1.2	4007, 2 – Cep cu deplasamente	70
2.1.1.3	4007, 3 – Cep rotit	72
2.1.1.4	4007, 4 – Cep clasic	72
2.1.1.5	4007, 5 – Cep coadă de rândunică	74
2.1.8	Grupa prelucrărilor 4008	77
2.1.1.1	4008, 1 – Chertare dublă la capăt	77
2.1.1.2	4008, 2 – Chertare dublă pe față laterală	78
2.1.1.3	4008, 3 – Chertare “tăbăcită”	79
2.1.1.4	4008, 4 – Chertare tip sapă	80
2.1.1.5	4008, 5 – Ornament capăt concav	80
2.1.1.6	4008, 6 – Ornament capăt convex	82
2.1.1.7	4008, 7 – Prelucrare tip arc de cerc	83
2.1.1.8	4008, 8 – Profil rotunjit	84
2.1.9	Grupa prelucrărilor 4009	86
2.1.9.1	4009, 1 – Prelucrare efectuată cu fierăstrău cu lanț	86
2.1.10	Grupa prelucrărilor 4101	88
2.1.1.1	4101, 0- Gaură de burghiu oarecare	89
2.1.1.2	4101, 1- Gaură de burghiu perpendiculară	90
2.1.11	Grupa prelucrărilor 4201	91
2.1.11.1	4201, 0 - Marcajul	91
2.1.12	Grupa prelucrărilor 4202	93
2.1.12.1	4202, 0 – Rindeluirea	93
2.3	Concluzii. Contribuții personale	94
3	Codarea prelucrărilor folosind combinația dintre două corpuri	96

3.1	Generarea cepului drept și a găurii de cep aferente	97
3.1.1	Generarea prelucrării pe o față laterală a corpului moale	98
3.1.1.1	Cep care străbate corpul pe secțiunea ei (4007, 1)	101
3.1.1.2	Cep clasic (4007, 4)	102
3.1.1.3	Cep rotit (4007, 3)	102
3.1.1.4	Cep cu deplasamente (4007, 2)	103
3.1.1.5	Ce cod va fi generat și de ce	103
3.1.1.6	Generarea prelucrării pe un capăt al corpului moale	105
3.1.2	Generarea prelucrării pe baza orientării găurii de cep	108
3.2	Generarea cepului coadă de rândunică	112
3.2.1	Generarea prelucrării pe față laterală a corpului moale	112
3.2.1.1	Cep coadă de rândunică ce străpunge corpul pe secțiunea lui	114
3.2.1.2	Cep coadă de rândunică limitat	114
3.2.2	Generarea prelucrării pe un capăt al corpului moale	117
3.2.3	Generarea prelucrării pe baza orientării găurii de cep	118
3.3	Generarea prelucrării eclisare	120
3.3.1	Noțiuni generale despre efectuarea prelucrării de tip eclisare	120
3.3.1.1	Numărul fețelor de contact = 1	120
3.3.1.2	Numărul fețelor de contact = 2	121
3.3.1.3	Numărul fețelor de contact = 3	122
3.3.1.4	Numărul fețelor de contact > 3	122
3.3.1.5	Ce tip de prelucrare se va genera în corpul 1	123
3.3.1.6	Ce tip de prelucrare se va genera în corpul 2	123
3.3.2	Eclisare standard	125
3.3.3	Eclisare de capăt	129
3.3.4	Eclisare frontală	132
3.4	Generarea prelucrării tip chertare simplă	135
3.5	Generarea prelucrării tip chertare dublă	137
3.6	Generarea prelucrării chertare de tip sapă	140
3.7	Generarea prelucrării tip chertare “tăbăcită”	142
3.8	Generarea prelucrării tip decupare în V	143
3.9	Gaură de cep oarecare	145
3.10	Concluzii. Contribuții personale	147
4	ALGORITMI PENTRU OPERAȚII BOOLEENE GENERALE	149
4.1	Pașii parcurși în efectuarea unei operații de intersecție, diferență și reuniune	149

4.2 Algoritmi matematici utilizați în operații booleene	156
4.2.1 Intersecția unei drepte cu un contur poligonal închis	156
4.2.2 Test de interioritate a unui punct într-un contur poligonal închis	158
4.2.3 Calculul normalei unui contur poligonal închis indiferent de forma lui	159
4.2.4 Intersecția a două segmente	161
4.3 Concluzii. Contribuții personale	163
5 PROBLEME SPECIALE. REZOLVĂRI	165
5.1 Rezolvarea problemelor de precizie	165
5.1.1 Efectuarea operațiilor de intersecție o singură dată	165
5.1.2 Verificarea valorilor între ele și față de valorile limită	166
5.1.3 Preciziile diferă în funcție de operație	166
5.1.4 Rotunjirea la valori limită	167
5.2 Rezolvarea problemelor de viteză	168
5.3 Rezolvarea problemelor legate de memorie	171
5.3.1 Memorii tampon reutilizabile	172
5.3.2 Fragmentarea memoriei	173
5.4 Concluzii. Contribuții personale	175
Concluzii	177
Contribuții personale	178
Anexa 1	181
Anexa 2	198
Anexa 3	201
Anexa 4	202
BIBLIOGRAFIA	207

1 MODELARE ȘI VIZUALIZARE

1.1 Modelare geometrică

În acest prim capitol se prezintă bazele teoretice ale metodelor de modelare geometrică necesare prelucrării corpurilor (Savii, 1997). Pentru a vedea efectul acestor prelucrări pe corpuri este necesară vizualizarea rezultatului. În acest scop se prezintă în subcapitolul 2 metode de vizualizare a obiectelor. Astfel utilizatorul poate vedea și corecta, la nevoie, rezultatul operațiilor înainte de a trimite comenzile de efectuare a prelucrărilor la robot. Astfel scad rebuturile și munca devine mai eficientă.

Principalul domeniu în care modelarea geometrică este intens utilizată este producția de bunuri materiale. Legătura strânsă între compartimentele implicate în procesele industriale cere o documentație precisă cu toate informațiile necesare, într-o formă pe care toți participanții o pot înțelege. În cazul automatizării, trebuie să fie posibilă și interpretarea directă de către mașini. Din punct de vedere informațional, aceasta este o problemă clasică de descriere, chiar dacă într-un sistem neobișnuit de dimensiuni. Un rol fundamental joacă aici normativele și limbajele comune.

1.1.1 Descrierea obiectelor și proceselor

Metodele de descriere pentru obiectele și procesele tehnice sunt preluate, în principal, prin dezvoltarea geometriei descriptive. Există astăzi reguli foarte detaliate, prevăzute chiar în norme naționale și internaționale (ISO, DIN, ANSI etc.), care precizează reprezentarea obiectelor și proceselor tehnice pe medii convenționale (desene, liste, hârtie/folie).

Apariția tehnicilor informatizate aduce noi posibilități, fundamentale, de descriere a produselor și proceselor de producție.

Metodele convenționale descriu obiectele prin desene bidimensionale și prin elemente unidimensionale: texte, tabele, liste de repere, planuri de operații etc., pe suport folie, hârtie, film etc. Uneori, se apelează la modele fizice tridimensionale din lemn sau ipsos. Unele părți din aceste descrieri sunt, în metodele convenționale, vagi, imprecise, incomplete, dar pot fi folosite din cauza contribuției subconștientului operatorului uman (prin încadrare în context). Introducerea metodelor informatice moderne (de tip algoritmic)

pretinde o descriere completă și precisă a obiectelor și proceselor de prelucrat, dar și a metodelor de descriere.

1.1.2 Modelarea - principala componentă a CAD

Rezolvarea problemelor modelării este un domeniu principal al muncii de cercetare și proiectare informatică, în conformitate cu cerințele din domeniile de aplicații CAD, CAM și CIM în general. Rezultate foarte importante sunt deja disponibile (de exemplu modelarea solidelor), la rezolvarea altora lucrându-se intensiv (de exemplu bănci de date pentru aplicații tehnice, metode noi de descriere precum cele orientate pe obiecte, metode din inteligența artificială: sisteme expert, rețele neuronale, tehnici *fuzzy*, metodică proiectării etc.). De un deosebit interes sunt, în context, sistemele orientate pe conținutul aplicației (semantice), chiar nenumerate (manipularea simbolurilor, semantica, determinarea și reprezentarea structurii, reprezentabilitatea informațiilor funcționale, geometrice, constructive, manipularea volumelor mari de informații structurate etc.).

Sistemele CAD livrează următoarele tipuri de descrieri de produse:

- *imagini* ca pur ajutor pentru oameni (tip fotografii);
- *desene* (vederi bidimensionale), *tabele*, *liste*, parțial pe baza unor norme; puternică orientare pe imagini, parțial amestecate cu notații tehnice, accesibile pe părți și din alte aplicații software (sistem de desenare);
- *model* de produs reprezentat intern (în calculator), cu părți de informații semantice despre produs, interpretabile și de către alte sisteme, din care se pot obține și desene sau alte elemente într-o formă interpretabilă de către om (sistem de modelare).

În practică se întâlnesc tipuri combinate din cele de mai sus.

Modelele de produse din calculator deschid posibilități foarte mari și importante pentru:

- *dezvoltarea produselor*: verificarea timpurie a soluțiilor de proiectare prin simulare, testare și calculare;
- *fabricație*: pregătirea fabricației, simulare, programarea comenzilor numerice și a roboților, și sunt premise hotărâtoare pentru integrare, automatizare și flexibilizare.

Informația tehnică are semnificație strategică. Sistemele CAD pot prelua, prelucra, transmite, prezenta, reprezenta și arhiva părți importante din această informație. CAD este o tehnologie specială în *informatica aplicată*.

1.1.3 Modelul produsului material

Modelul unui produs material va cuprinde multiplele aspecte necesare producerii și utilizării acestuia: funcția, configurația, gabaritul, comportarea în timp, informații pentru calculare, simulare, testare, descrieri ale structurii, elemente pentru prezentare, însoțitoare la beneficiar, informații tehnologice, comerciale, pentru documentare și arhivare etc. (Gardan 1991).

Funcția

Funcția produsului este considerată independent de realizarea fizică, de exemplu: scheme electrice, electronice, hidraulice, pneumatice, cinematice, de trasee de conducte, fluxul de energie/putere.

Stabilirea configurației, gabaritului și timpului

Sunt implicate două aspecte:

- *geometric*: topologia, metrica, distanțele;
- *cinematic*: legi de mișcare.

Calculare, simulare, testare

Acest grup de aspecte se referă la:

- proprietăți fizice, materiale, suprafețe;
- rezistență, capacitate portantă, testare;
- calculare, cotare, toleranțe;
- analiză/verificare cu metode de element finit (FEM/FEA);
- simulare;
- construirea de modele simplificate.

1.1.4 Metode și mijloace pentru realizarea formelor complexe

1.1.4.1 Introducere

Funcție de structurile de date și de informațiile de model stocate, sistemele 3D sunt clasificate în:

- orientate pe *muchii* (*wireframe*)
- orientate pe *suprafețe* (*surface model*)
- orientate pe *volum* (*solid model*).

Dintre acestea, modelele solide înmagazinează cea mai completă descriere geometrică. Cu toate acestea, modelatoarele de solide sunt încă destul de complicat de utilizat, în general, și sunt limitate atât în complexitatea cât și în acuratețea geometriei. Acestea sunt probabil motivele pentru care modelarea de solide nu este încă tehnica exclusivă în sistemele CAD industriale.

Modelele solide pot fi obținute prin următoarele tehnici mai frecvente (Encarnacao, 1983, Gardan, 1991, Taylor, 1992, Zeid, 1991):

- *geometria constructivă a solidelor* (CSG)
- prin *frontiere* (B-rep)
- prin *enumerare spațială* (octree...)
- prin *baleiere* (sweeping etc.)
- *analitic* (ASM - Analytic Solid Model)
- *parametric/procedural*

La conceperea lor, formele modelelor solide pot fi gândite în termenii unor forme geometrice elementare (primitive de tipul prismă, cilindru, sferă, con etc.), sau în termenii unor forme elementare din punct de vedere al generării tehnologice (de rotație, extrudate etc.). Formele geometrice elementare necesare construirii unor obiecte complexe pot fi obținute utilizând accesul la forme *primitive standard* puse la dispoziție de sistemul de proiectare (modelator). Aceste forme primitive standard trebuie să fie parametrizate, în sensul că utilizatorul poate modifica una sau mai multe dimensiuni.

Utilizarea doar a formelor primitive standard poate limita aplicabilitatea sistemului. De aceea, sistemele CAD oferă utilizatorului, de regulă, posibilitatea de a defini după necesități entități geometrice elementare. Se pot, astfel, crea suprafețe pornind de la curbe, sau volume pornind de la suprafețe. Cea mai simplă metodă este atribuirea de grosimi formelor plane (echivalent extrudării).

O problemă în metoda de generare utilizând primitive este numărul mare de operații booleene necesare. Primitivele sunt construite, modificate și combinate utilizând operații booleene standard. Acest lucru este deseori considerat un dezavantaj, din cauza complexității și a costului ridicat al operațiilor booleene.

Pe de altă parte, operația manuală de grupare a unor fețe orientate coerent pentru a obține un solid este laborioasă și împotriva spiritului general al CAD, astfel încât s-au elaborat *operatori speciali* pentru a asista crearea modelelor solide.

Un astfel de operator special, aplicat unor suprafețe convenabil create, este cel de *îngroșare (thicken)*. Acest operator ia o suprafață de bază, definește o suprafață *offset* a ei și apoi închide spațiul dintre ele cu o bandă laterală. Acest operator este potrivit pentru obiectele care au suprafețele superioară și inferioară (suprafețele legate cu o bandă) similare.

Suprafața care închide corpul (banda), poate fi riglată sau poate fi mai complexă, de tip *loft-at*, ce ține cont și de direcțiile normalelor celor două suprafețe inițiale (bază și *offset*), pentru asigurarea continuității (evitarea muchiilor ascuțite).

1.1.4.2 Tehnici de descriere a modelelor geometrice 3D

Modelele de solide trebuie să satisfacă un grup de cerințe (Encarnação și Schlechtendahl, 1983, Gardan, 1991, Taylor, 1992, Zeid, 1991):

- *validitatea*: să existe un obiect real corespunzător oricărui model posibil de creat;
- *universalitatea*: să se poată crea un model pentru orice obiect 3D;
- *unicitatea interpretării*: unei reprezentări concrete îi corespunde doar un singur obiect;
- *unicitatea expresiei (neambiguitatea)*: unui obiect 3D îi corespunde doar un singur model;
- *caracterul complet*: toate operațiile suportate să fie aplicabile tuturor reprezentărilor (să genereze în toate cazurile entități valide ca operanzi pentru toate operațiile);
- *conciziunea*: informația conținută în model să fie minimă.

Modelarea prin frontiere (B-rep)

Modelele 3D construite prin enumerarea fețelor ce separă obiectele de restul "lunii" (obiecte definite prin "coaja" lor) se numesc *modele prin frontiere - B-rep*. Modelul memorează și poziția interiorului obiectului față de fețe. Tehnica este avantajoasă pentru vizualizări, dar nu este bine adaptată pentru unele operații analitice, precum calculul centrului de greutate sau al momentelor de inerție. În figura 1.1.b este reprezentat schematic modelul B-rep al obiectului din figura 1.1.a.

Modelele prin frontiere au dificultăți în respectarea cerințelor pentru modele solide valide. De exemplu, o problemă o constituie asigurarea *continuității* geometrice, mai ales la aplicarea unor modificări modelului. Pot să apară fețe flotante, cu muchii ce aparțin doar unei fețe. Într-un model corect, o muchie se află la întâlnirea a două și numai două fețe. O altă problemă este *compatibilitatea*: suprafața generată trebuie să fie fizic posibilă. Sunt excluse auto-intersecțiile și ambiguitățile în orientare. Normalele celor două fețe ce se întâlnesc într-o muchie trebuie să fie orientate compatibil. De exemplu, un corp de tip bandă Moebius are probleme de acest tip. Din cauza acestor probleme potențiale, utilizatorul va avea acces doar la nivelul unei interfețe tip CSG.

Modelarea prin geometria constructivă a solidelor (CSG)

Modelele complexe se pot crea grupând componente mai simple, denumite *sub-obiecte* sau *primitive*. Exemple de sub-obiecte frecvente sunt: cuburi, conuri, cilindri, sfere. În tehnica prin *geometria constructivă a solidelor - CSG*, modelul este reprezentat printr-un arbore (sau graf, pentru modele complexe). La capetele ramurilor arborelui se află obiecte primitive parametrabile, iar în noduri - operatori pe mulțimi. Fiecărui nod îi corespunde un sub-obiect (chiar dacă nu este calculat). Tehnica CSG este avantajoasă în cazul modificării geometriei, dar nu este adaptată vizualizării. De aceea, uneori se păstrează în memorie două versiuni de model, una CSG și una *B-rep*, pentru vizualizare. În figura 1.1.c este prezentat modelul CSG al obiectului din figura 1.1.a. Modelul rezultă printr-o reuniune și o diferență aplicate pornind de la forme geometrice simple.

Principalele avantaje al modelării CSG sunt validitatea inerentă, unicitatea interpretării, caracterul complet, consecvența și conciziunea.

Principalul dezavantaj este complexitatea calculelor pentru evaluarea arborelui CSG în vederea vizualizării. Acest lucru face ca editarea interactivă să fie rar întâlnită în

sistemele CSG. Un alt dezavantaj este imposibilitatea reprezentării tuturor obiectelor fizice, de exemplu a celor cu suprafețe curbe cu formă liberă.

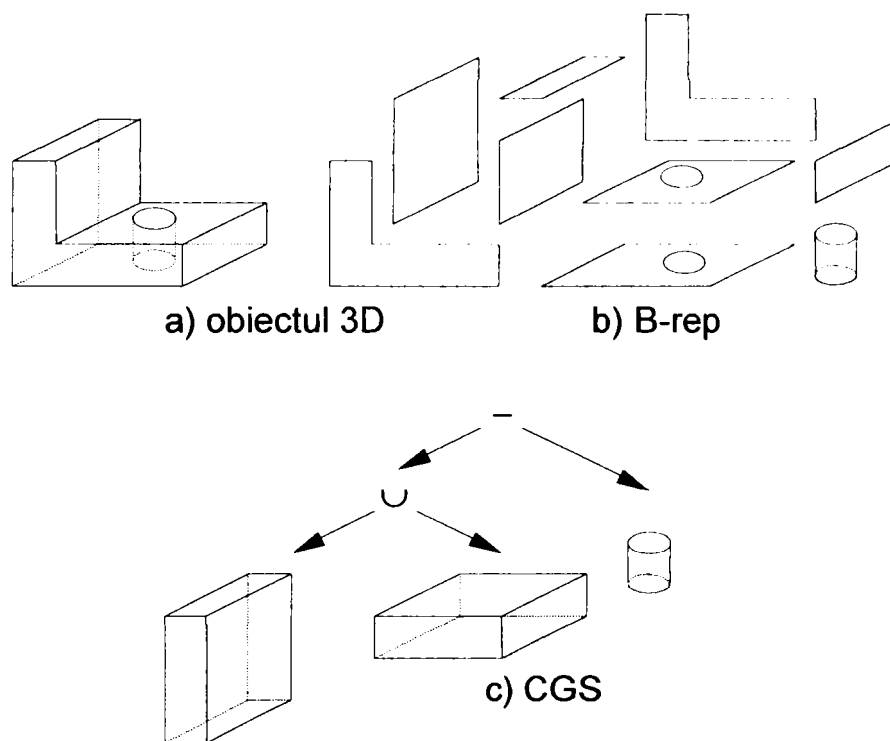


Figura 1.1: Tehnici de modelare 3D

Modelarea prin enumerare spațială

Spațiul modelului este divizat în elemente "atomice" (unitare) volumice, asemănător cazului imaginilor 2D, descrise prin mulțimea de pixeli constituenți (*bit-mapped*). Elementul volumic unitar echivalent unui pixel este denumit *voxel* (*volume element*). Un obiect solid este descris prin fanioane care indică elementele volumice ocupate. Deoarece enumerarea tuturor punctelor volumice ar consuma memorie enormă, se utilizează obiecte unitare speciale. S-a dezvoltat o metodă de divizare recursivă, cunoscută sub numele de *quadtree* în 2D și *octree* în 3D. Un mare avantaj al acestei metode este posibilitatea de descriere a obiectelor complexe neregulate, de tipul celor existente în natură.

Modelarea prin baleiere

Este o tehnică utilă, în primul rând, pentru obiecte 2-dimensionale: solide cu grosime constantă sau cu simetrie axială. Solidele din prima clasă rezultă prin baleiere

rectilinie (tip extrudare), iar cele din a doua clasă - prin baleiere circulară (corpuri de revoluție).

Baleierea constă în deplasarea unui punct, a unui segment sau a unei suprafețe, pe o anumită traiectorie. Pentru forme complexe, traiectoria este o curbă complexă.

Metoda este utilizată mai ales pentru obținerea unor elemente ce urmează a fi incluse în scheme CSG sau *B-rep*. Nu există modelatoare pure prin baleiere, din cauza limitelor metodei.

Modelarea analitică (ASM)

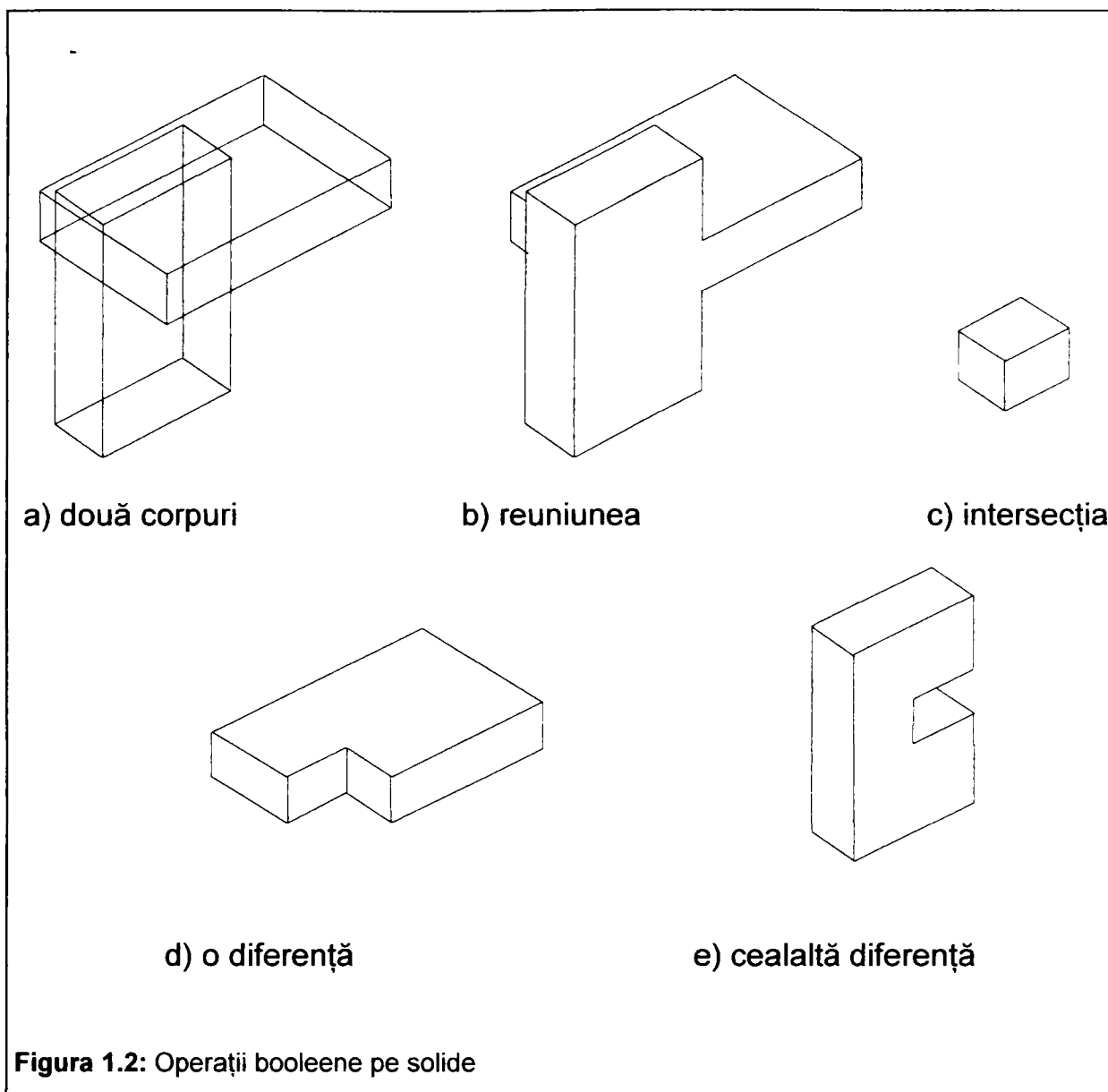
Tehnica ASM este asemănătoare formulărilor izoparametrice tridimensionale din analiza cu elemente finite (FEA). Este adecvată descrierii modelelor cu forme complexe și calculelor proprietăților volumetrice. Exemple ar fi hiperpeticele parametrice tricubice (generalizare a peticelor bidimensionale - bicubice). Un hiperpetic este o aplicație a unui domeniu cubic unitar parametric descris de u , v și w , într-un solid descris de x , y și z în spațiul modelului (cartezian). Orice obiect va putea fi reprezentat ca un ansamblu de hiperpetice cvasi-disjuncte, prin divizarea sa convenabilă.

Modelarea prin instanțierea primitivelor

Metoda prin instanțierea primitivelor se bazează pe reprezentarea parametrică a obiectelor singulare. Un obiect va fi descris de modelul generator al întregii clase (familii) și de valorile parametrilor ce definesc obiectul particular (instanța). Pentru prelucrări mecanice sau operații analitice, metoda este avantajoasă pentru descrierea porțiunilor din obiecte, dar interfața cu utilizatorul este complexă.

Metoda booleană de creare a modelelor solide utilizând forme geometrice elementare

Metoda dominantă pentru crearea de modele solide se bazează pe operații *booleene*. Utilizarea operațiilor booleene este o metodă generală și nu presupune o structură internă particulară a modelatorului de solide ca geometria constructivă a solidelor sau reprezentarea prin frontiere. Figura 1.2 prezintă operațiile booleene fundamentale exemplificate pe un set de două corpuri prismatice. În figura 1.3 este



prezentat un exemplu de creare a unui corp prin această metodă, comparativ cu metoda orientată pe prelucrare (tehnologie).

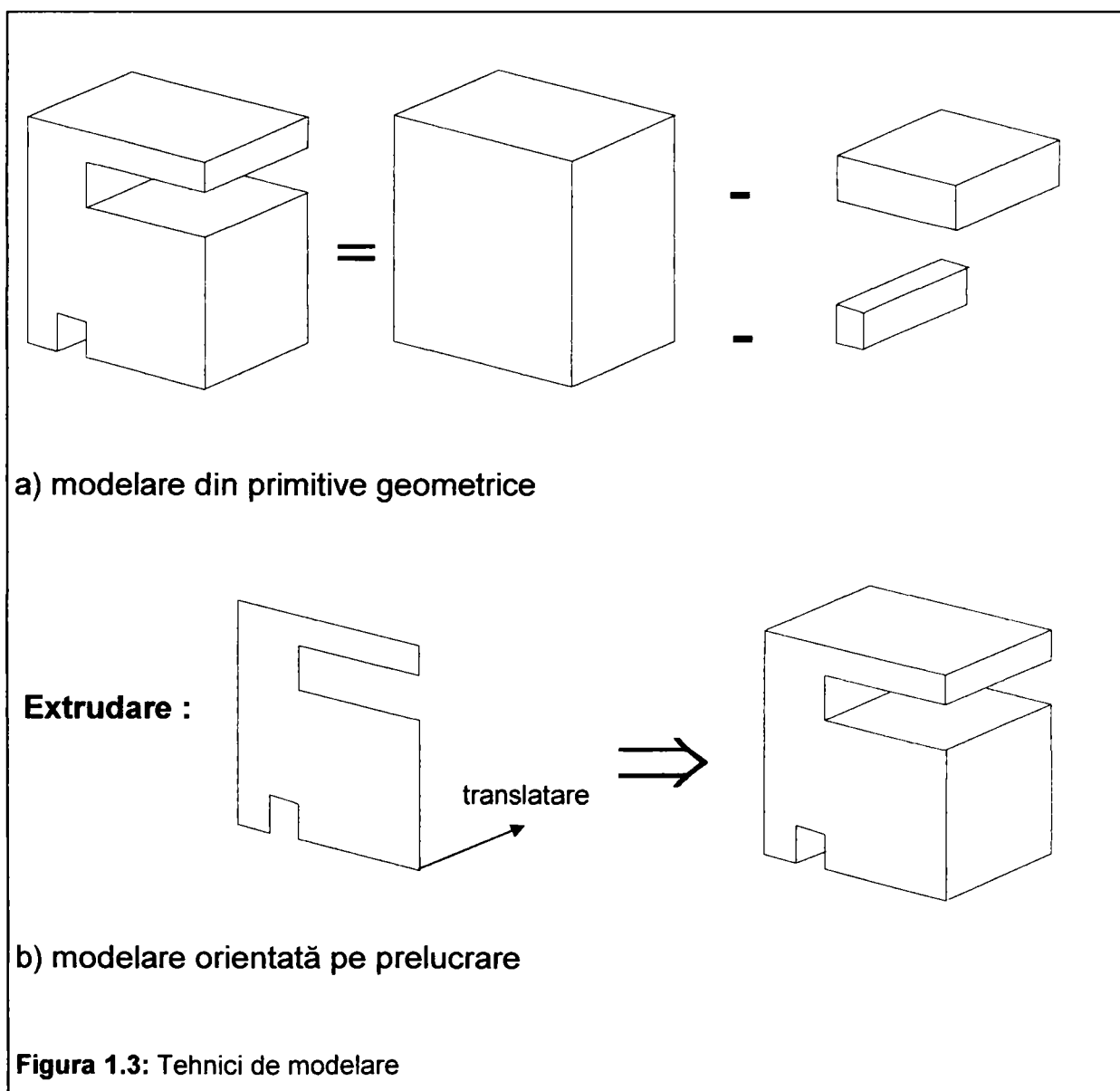
Această tehnică de construire forțează utilizatorul să gândească în termenii *teoriei mulțimilor* din matematică în sisteme de coordonate tridimensionale cu obiecte primitive. Crearea unei fante într-un bloc, de exemplu, cere poziționarea relativă a două prisme în mod convenabil în spațiul 3D și aplicarea funcției diferență pentru extragerea unei părți din blocul de bază. Asemănător, se pot obține obiecte complexe prin operații de tip reuniune sau intersecție.

Operațiile booleene se folosesc și pentru operații utilitare precum verificarea coliziunilor (prin testarea intersecțiilor) și generarea secțiunilor (prin intersecții și diferențe).

Metoda de creare a modelelor solide orientată pe prelucrare

În opoziție cu modelarea prin operații booleene cu primitive geometrice, metoda orientată pe prelucrare este bazată pe comenzi de nivel înalt orientate către procesul de fabricație, precum comenzile echivalente procedurilor *tehnologice* de: extrudare, strunjire, frezare, găurire sau perforare. Această metodă este mai aproape de modul de gândire al proiectantului. Principiul este utilizat de sistemele de modelare prin caracteristici (*feature-based modelling*, Gardan, 1991).

În figura 1.3.b este prezentat același obiect ca în figura 1.3.a, dar exemplificând obținerea lui prin metoda orientată pe prelucrare. În general, principiul este de a utiliza rotirea și translatarea profilurilor plane pentru crearea și modificarea modelelor. Profilul poate fi văzut ca geometria sculei sau, mai exact, ca secțiune prin sculă. Translatarea sau rotirea unui profil este interpretată ca o operație de modelare ce are efecte similare



asupra formei obiectului cu o operație de prelucrare corespunzătoare ce mișcă scula față de semifabricat (sau invers).

De notat că extrudarea, strunjirea, frezarea etc., ca și comenzi de proiectare, nu este necesar să determine tehnologia de fabricație utilizată în realitate.

Pentru a face mai naturală utilizarea acestei metode, s-a introdus conceptul de plan de lucru (*workplane*). Planul de lucru este un plan în spațiul 3D, care este utilizat pentru construirea profilurilor. Selectarea unui plan de lucru produce automat o transformare a vederii pentru a putea utiliza comenzi normale 2D pentru construirea de profiluri în spațiul 3D. Transformarea are loc astfel încât planul de lucru să devină perpendicular pe direcția de privire. Pentru obținerea unor piese complexe, cum ar fi piulițe, roți dințate, arcuri elicoidale etc., se utilizează transformări suprapuse (simultane), combinate cu metodele obișnuite de modelare.

Etapele principale sunt:

1. definirea secțiunii P a sculei prin metode simple 2D, utilizând conceptul de plan de lucru;
2. alegerea funcției de comandă, prelucrarea asociată care definește mișcarea sculei; mișcări simple pot fi suprapuse pentru a defini transformări complexe;
3. se aplică, apoi, metodele de modelare cunoscute (operații booleene, *sweeping*, *lofting* etc.), pentru a realiza intern operațiile tip prelucrare.

În urma studierii tuturor metodelor de descriere a solidelor în cadrul lucrării orice corp va fi modelat printr-o mulțime de suprafețe plane care la rândul lor sunt o mulțime de muchii consecutive și fiecare muchie are un vârf de început și unul de sfârșit. Pentru o descriere completă, care să conțină și informații topologice au fost necesară reținerea următoarelor informații:

- începutul unei muchii coincide întotdeauna cu sfârșitul muchiei anterioare.
- contururile suprafețelor sunt închise și pentru fiecare se reține ecuația planului din care face parte
- fiecare suprafață este parcursă, pe muchii, cu materialul la stânga
- fiecare muchie a unei suprafețe își cunoaște suprafața vecină.

Astfel se pot descrie și corpuri multiplu conexe, adică într-un vârf se pot întâlni mai mult de 3 muchii (figura 1.4) sau pe o muchie se pot întâlni mai mult de 2 suprafețe (figura 1.5).

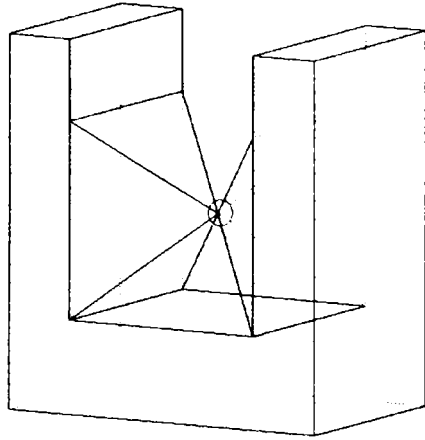


Figura 1.4: Corp cu vârf multiplu conex (marcat cu cerc)

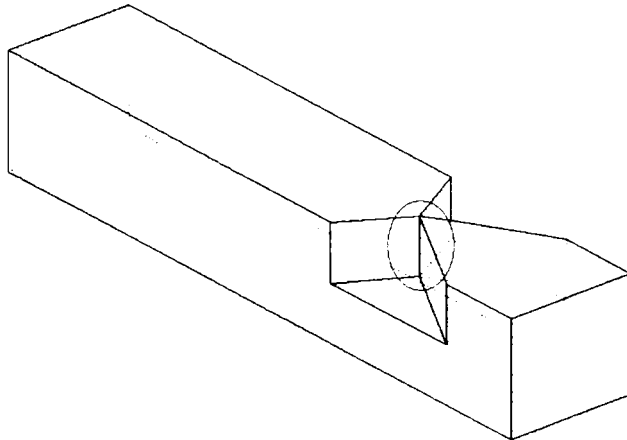


Figura 1.5: Corp cu muchie multiplu conexă (marcată cu cerc)

Rezultatele obținute în urma operațiilor booleene sunt descrise la fel.

Operațiile booleene în cadrul tezei au fost implementate folosind metoda de creare a modelelor solide utilizând forme geometrice elementare.

1.2 Vizualizarea modelelor

1.2.1 Introducere

Reprezentările simple, *wireframe*, au fost și sunt des utilizate, fiind suficient de sugestive. Mai mult, algoritmi corespunzători fiind simpli, conferă rutinelor grafice avantajul de a produce imagini de corpuri 3D cu suficientă rapiditate pentru a putea fi incluse în programe de proiectare asistată, chiar pe calculatoarele cu performanțe relativ modeste în ceea ce privește viteza de prelucrare a datelor (rutinele grafice se pot realiza în timp absolut acceptabil pe microcalculatoare personale sau personal - profesionale). Sub aspectul realismului imaginii însă, rezultatele obținute lasă de dorit.

Dezavantajele reprezentărilor de tip *wireframe* sunt legate de faptul că ele nu oferă privirii o imagine asemănătoare celor cu care ochiul este obișnuit din viața de toate zilele. O fațetă poligonală sau o suprafață oarecare a unui corp real nu poate fi străpunsă cu privirea (cu excepția celor transparente, care însă au o pondere mică); ea este opacă, reflectă spre ochi razele de lumină care permit formarea pe retină a imaginii corpului respectiv, cu zone de umbră și iluminare în variate nuanțe de culori. Pe suprafețele reale putem observa detalii, asperități, o textură care ne face să deosebim din privire, aproape fără greșală, metalul de lemn, de exemplu. Tocmai aceste elemente, pe care le cunoaștem prin experiența zilnică, reproduse pe ecran ar permite crearea iluziei de substanță, producerea de imagini realiste.

Reprezentări realiste ale corpurilor 3D se pot obține prin combinarea a patru tehnici:

- eliminarea tuturor elementelor și părților corpului care, privite în condițiile date, sunt mascate (eliminarea liniilor și suprafețelor ascunse)
- simularea iluminării corpului respectiv cu un sistem de surse de lumină bine precizat (punctiforme, distribuite, lumină paralelă, lumină ambiantă)
- reconstituirea detaliilor existente pe suprafață (a texturii materialului)
- reconstituirea culorii și nuanțelor în care obiectul (corpul) original ar apărea, iluminat fiind în condițiile precizate.

Prin prelucrarea reprezentărilor simple se mărește cantitatea de informație disponibilă pe ecran pentru a permite utilizatorului să înțeleagă relațiile 3D existente între mai multe corpuri sau între componentele unui corp. Se poate adăuga imaginii un

minimum de informație necesar atingerii acestui scop, fără a tinde spre un "realism absolut", realizându-se programe mai simple care să ofere rezultatul în timp rezonabil.

Asemenea rutine grafice sunt astăzi curente în proiectarea asistată de calculator.

Figura 1.6 exemplifică problema "un minimum de informație".

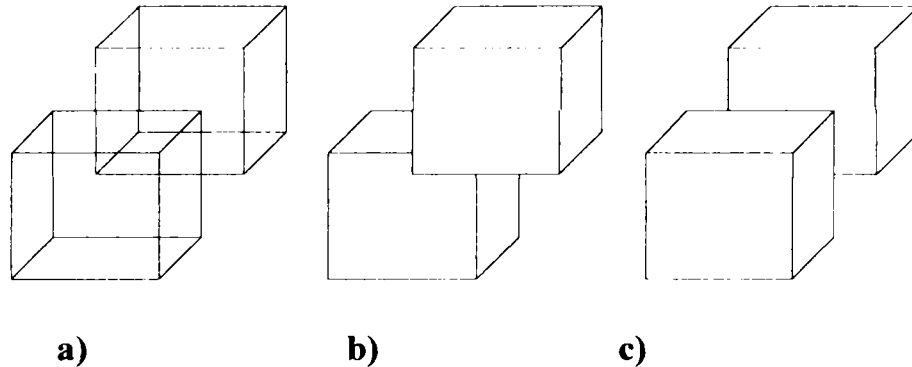


Figura 1.6: Ambiguități privind poziția relativă, în reprezentarea *wireframe*

Figura 1.6-a nu este suficientă pentru ca privitorul să poată preciza poziția relativă a celor două cuburi. Figurile 1.6-b sau c pot lămurii problema. Informațiile cu privire la poziția relativă sunt calitativ complete. Pentru precizarea lor nu a fost nevoie să se reprezinte textura superficială a cuburilor (pentru a recunoaște eventual materialul), culoarea acestora, cum sunt ele umbrite. Imaginile care ar oferi asemenea informații ar avea încorporat cu mult mai mult decît ne trebuie pentru a clarifica problema poziției relative.

O ilustrare clasică este iluzia cuboidului lui Necker (figura 1.7). Nu putem preciza care din reprezentările cu linii ascunse îndepărtate *b - i* redau poziția și forma reale ale cuboidului reprezentat *wireframe* în *a*. Cuboidul poate fi de tip solid plin ("cărămidă", ca în *b* și *c*), tavă (ca în *e* și *h*), sau țevă (ca în *d, f, g* și *i*), poate fi privit de sus (ca în *c, d, e* și *f*), sau de jos (ca în *b, g, h* și *i*) (Savii, 1997).

Și reprezentările simple pot înmagazina informații cu privire la structura 3D a unui corp. Proiecția paralelă poate fi folosită pentru explicitarea 3D a unui corp când se oferă simultan utilizatorului trei proiecții paralele după trei direcții care formează un triedru drept (ca în desene tehnice de execuție).

Proiecțiile în perspectivă oferă posibilitatea de a cunoaște relațiile de profunzime, cu condiția de a ști în prealabil cum arată corpul. Dacă nu se cunoaște forma corpului reprezentat, ambiguitatea persistă. De exemplu, corpul reprezentat în perspectivă în figura 1.7 poate fi un cub sau un trunchi de piramidă.

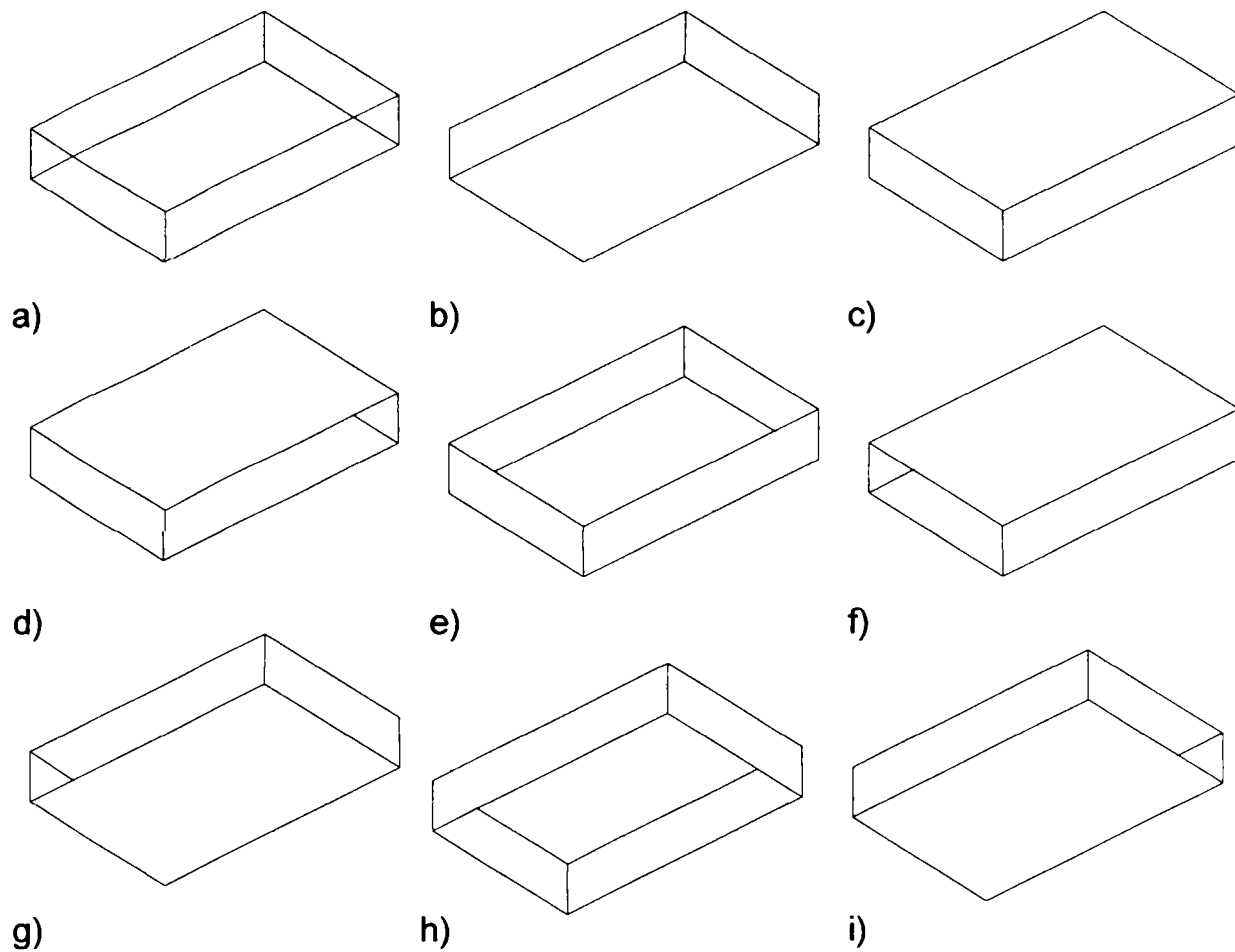


Figura 1.7: Ambiguități la interpretarea formei și orientării

Experiența arată că o proiecție perspectivă este cu atât mai ușor de interpretat cu cât corpurile reprezentate au mai multe linii paralele, deoarece o parte din acestea par să convergă în punctul sau punctele de fugă. Se mai poate folosi și exagerarea artificială a efectului de perspectivă, dar în acest caz, corpurile apar deformate.

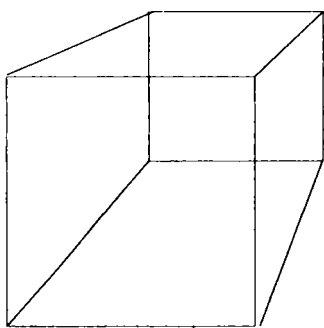


Figura 1.8: Ambiguități privind forma în reprezentarea în perspectivă

O modalitate foarte simplă de realizare a unei iluzii de adâncime este tehnica *depth-cueing* (sau *intensity-cueing*) - modularea intensității în adâncime. Intensitatea luminoasă a liniilor cu care se face desenarea (de regulă de tip *wireframe*) este redusă funcție de distanța muchiei la observator.

Principalele tehnici de vizualizare pentru asistarea procesului cognitiv al proiectantului într-un sistem CAD/CAM, prin mijloace grafice, sunt prezentate în continuare.

1.2.2 Linii ascunse și suprafețe ascunse

Problema liniilor și suprafețelor ascunse este una din cele mai dificile din aplicațiile de grafică pe calculator. Algoritmii pentru linii/suprafețe ascunse încearcă să determine muchiile, fețele sau volumele care sunt vizibile sau invizibile unui observator localizat într-un punct specificat din spațiu.

Complexitatea problemei liniilor/suprafețelor ascunse a condus la un număr mare de soluții diferite. Multe din acestea sunt pentru aplicații specializate. Nu există o soluție perfectă a acestei probleme. Algoritmii rapizi ce pot produce soluțiile la viteza de înmprospătare a imaginii (30...80 Hz) sunt necesari pentru simulări în timp real, de exemplu la simulatoarele de zbor. Algoritmii ce pot produce soluții cu detalii realiste, inclusiv umbre, transparențe, efecte de textură, cu reflexii și refracții într-o multitudine de nuanțe, sunt necesari, de exemplu, pentru simulare sau animație cu calculatorul. Acești algoritmi sunt mai lenți, necesitând uneori chiar ore întregi pentru rezolvare. Din punct de vedere tehnic, transparența, textura, reflexiile nu sunt componente ale problemei liniilor/suprafețelor ascunse. Ele sunt părți ale procesului de redare realistă (*rendering*) a imaginii. În practică trebuie făcut un compromis între viteză și număr de detalii.

Toți algoritmii pentru linii/suprafețe ascunse implică *sortare*. Principala operație de sortare este bazată pe distanța geometrică a unui punct, a unei muchii, fețe sau a unui volum față de punctul de vedere. Ipoteza fundamentală făcută la această sortare este că cu cât un obiect este mai depărtat de punctul de vedere, cu atât este mai probabil ca obiectul respectiv să fie parțial sau total ascuns de altul mai apropiat de privitor. După stabilirea priorităților după distanță, mai rămâne de realizat o sortare pe lateral și pe înălțime pentru a determina dacă, de fapt, un obiect este ascuns de un altul mai apropiat de privitor. Eficiența unui algoritm de linii/suprafețe ascunse depinde semnificativ de eficiența procesului de sortare. *Coerența* (tendința caracteristicilor unei scene de a fi local constante) este utilizată pentru a crește eficiența sortării (Mureșan I., 1996, Mureșan Voichița, 1996).

Algoritmii pentru linii/suprafețe ascunse pot fi clasificați pe baza spațiului sau sistemului de coordonate în care sunt implementați. Algoritmii în spațiul obiect sunt

implementați în sistemul de coordonate fizic în care obiectele sunt descrise. Se pot obține rezultate foarte precise, limitate doar de precizia calculatorului. Prin mărirea, de câteva ori, a acestor imagini rezultate, se obțin imagini de calitate satisfăcătoare. Algoritmii din spațiul obiect sunt în special utili în aplicațiile ingineresti de precizie. Algoritmii în spațiul imagine sunt implementați în sistemul de coordonate al ecranului, în care obiectele sunt vizualizate. Calculările sunt efectuate doar la precizia reprezentării pe ecran (cam până la 1280x1024 puncte). Scenele calculate în spațiul imagine dau rezultate inacceptabile după mărirea.

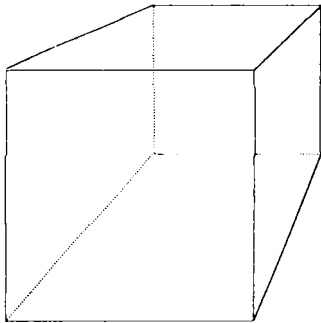


Figura 1.9: Reprezentare cu linii ascunse vizibile

Teoretic, volumul de calcule pentru un algoritm în spațiul obiect, ce compară fiecare obiect din scenă cu fiecare din celelalte, crește proporțional cu pătratul numărului de obiecte din scenă. Similar, volumul de calcule pentru un algoritm în spațiul imagine, care compară fiecare obiect din scenă cu fiecare locație de pixel în coordonatele ecranului, crește proporțional cu numărul de obiecte (volum, fețe, muchii) și cu numărul de pixeli. Algoritmii în spațiul imagine sunt mai eficienți, deoarece este mai ușor de a utiliza avantajele coerenței în spațiul imagine.

Sunt utilizați și algoritmi *hibridi*, ce pot avea viteze superioare.

Pentru ca imaginile afișate să fie mai sugestive, există posibilitatea de a *afișa liniile ascunse* cu un tip special de linie (de exemplu punctată sau cu o anumită nuanță, figura 1.9).

Algoritmii eliminării fețelor orientate spre spate

Este un algoritm utilizat des pentru o primă eliminare a fețelor total invizibile. Se bazează pe faptul că, dacă normala pe una din fețele unui solid este orientată dinspre observator, fața respectivă este invizibilă (presupunând o orientare consecventă a muchiiilor, de exemplu interiorul feței să fie totdeauna la stânga muchiei iar normalele

fețelor să iasă din corp). Orientarea se verifică prin produsul vectorial între versorul normalei pe fața respectivă și versorul direcției de privire.

Acest algoritm dă rezultate corecte (definitive) în cazul afișării corpurilor convexe izolate (Rogers, 1985).

Algoritmul pictorului

Algoritmul are la bază ideea de a considera fiecare față poligonală a solidelor ca o entitate de sine stătătoare. În această ipoteză, se poate urmări analogia cu realizarea unei pânze în ulei de către un pictor. Astfel, pictorul va picta un peisaj începând cu elementele cele mai îndepărtate, care vor fi acoperite, apoi, de către obiecte mai apropiate de ochiul observatorului, pe măsura ce acestea sunt reprezentate pe pânză.

Considerând fiecare față poligonală ca entitate de sine stătătoare, ea va acoperi eventual fețe poligonale aflate la distanță mai mare de ochiul observatorului și va fi eventual acoperită de fețe poligonale aflate la distanță mai mică de ochiul observatorului.

Algoritmul lucrează corect pentru fețe realizate din poligoane convexe ce nu se întrepătrund (Rogers, 1985, Zeid, 1991).

Având în vedere principiile enunțate mai sus, rezultă că rezolvarea problemei eliminării liniilor ascunse impune parcurgerea a doua etape. Prima etapă va fi sortarea fețelor poligonale după adâncime, iar a doua va fi reprezentarea grafică a acestora pe ecran, în ordinea stabilită, însoțită de ștergerea liniilor aflate în interiorul fețelor poligonale reprezentate (prin umplerea interioarelor poligoanelor cu pixeli de o anumită nuanță).

Pentru a realiza doar o ascundere a liniilor, nuanța de umplere a poligoanelor se alege a fi nuanța fondului. Algoritmul poate realiza simultan și o reprezentare nuanțată, în care caz nuanța finală se calculează funcție de unghiul pe care normala feței îl face cu direcția de privire și de nuanța de bază atribuită feței.

Algoritmul lui Roberts

Este primul algoritm cunoscut pentru linii și suprafețe ascunse. Operează în spațiul obiect. Algoritmul elimină întâi muchiile sau planele din fiecare volum care sunt acoperite de volumul respectiv. În continuare, fiecare muchie rămasă a fiecărui volum este comparată cu fiecare din volumele rămase pentru a determina ce porțiuni, dacă există, sunt acoperite de aceste volume. Volumul de calcule este proporțional cu pătratul

numărului de volume. Acesta este unul din motivele (alături de lucrul în spațiul obiect) pentru care algoritmul Roberts nu are extindere pentru dispozitivele raster. Metodele matematice utilizate sunt simple și exacte. Algoritmul a fost îmbunătățit cu o sortare prealabilă după axa z.

Algoritmul Roberts cere ca toate volumele sau obiectele de vizualizat să fie convexe. Volumele concave trebuie divizate în volume componente convexe (Rogers, 1985). Algoritmul consideră că un volum convex cu fețe poligonale plane este reprezentat de o colecție de plane ce se intersectează.

Algoritmi Z-Buffer (tampon de adâncime)

Reprezentativi pentru abordarea reprezentărilor în spațiul imagine sunt algoritmi din familia *tampon de adâncime* sau *z-buffer*. Aceștia sunt relativ simpli, ușor de implementat și performanți, dar reprezintă dezavantajul de a necesita memorie suplimentară pentru păstrarea valorilor cotei z ale unor mulțimi de puncte bine determinate (Moldoveanu Florica, 1996, Rogers, 1985, Zeid, 1991). Cel mai eficient algoritm din această familie (ca viteză de lucru) este algoritmul *screen - z buffer* (tampon de adâncime pentru ecran).

Pentru aplicarea algoritmului, se crează un *masiv bidimensional* (matrice) în care fiecărui pixel de pe ecran să-i corespundă un element din masiv. Legea de corespondență este evidentă: indicii elementului din masiv să reprezinte coordonatele-ecran ale pixelului asociat. Acestui masiv i se datorează, în principal consumul mare de memorie. Masivul poartă numele de *z-buffer* de ecran, sau simplu *z-buffer*. Se inițializează *z-buffer*-ul la o valoare z ce semnifică "foarte îndepărtat", pe care sigur nici unul din punctele modelului de reprezentat nu o va atinge. Pentru reperele de reprezentare în care ecranul este paralel cu planul xy, iar față de privitor axa x este orizontală orientată spre dreapta și axa y este verticală orientată în sus, întâlnim două situații:

- dacă triedrul xyz este drept, atunci *z-buffer*-ul se inițializează la cea mai mică valoare (negativă) reprezentabilă (în continuare se considera acest caz);
- dacă triedrul xyz este stâng, atunci *z-buffer*-ul se inițializează la cea mai mare valoare reprezentabilă.

Se inițializează ecranul (dacă imaginea este creată într-o memorie de reîmprospătare - atunci aceasta din urmă) la o valoare de nuanță corespunzătoare fondului (pentru dispozitivele de afișare care admit mai multe nuanțe pe pixel). La

dispozitivele de afișare cu numai două valori de intensitate, inițializarea se face la "alb" (de exemplu 0) sau "negru" (de exemplu 1).

Pentru fiecare entitate ce se desenează, valoarea coordonatei z se compară cu cea din z -buffer. Dacă este mai mică decât aceasta, punctul respectiv din entitate este invizibil (acoperit). Dacă este mai mare, punctul din entitate este considerat vizibil și nuanța sa este atribuită pixelului corespunzător de pe ecran (Savii, 1997).

Algoritmul descrie un proces în care z -buffer-ul va conține mereu valorile lui z cele mai apropiate de observator, reîmprospătând ecranul cu intensitățile punctelor corespunzătoare.

După terminarea procesului, vom avea transpuse în ecran intensitățile corespunzătoare mulțimii punctelor celor mai apropiate de observator, care reprezintă, de fapt, imaginea fațetelor vizibile ale corpului.

Se remarcă faptul că transpunerea pe ecran se face în ordinea în care se parcurg poligoanele, fără vreo relație cu eșalonarea acestora în adâncime.

Performanțele de viteză ale acestui algoritm tind să fie pe ansamblu constante, de aceea el se recomandă pentru reprezentarea corpurilor sau ansamblurilor de corpuri cu număr mare de fațete sau pentru care numărul de fațete variază în limite largi. Timpul de calcul este aici proporțional cu numărul de pixeli acoperiți de un poligon și cu numărul de poligoane de reprezentat.

Algoritmii din familia z -buffer sunt destinați eliminării suprafețelor ascunse, dar ei pot fi folosiți și ca algoritmi de îndepărtare a liniilor ascunse. Pentru aceasta este suficient să atribuim în mod artificial valori de intensitate pixelilor unui poligon (de exemplu 0 pentru interiorul poligonului și 1 pentru frontieră).

Se mai poate adapta acest algoritm și pentru a reprezenta liniile ascunse punctat.

Algoritmii din familia z -buffer sunt importanți pentru că performanțele sunt constante în raport cu complexitatea corpurilor, iar pentru descrierea corpurilor complexe (cu peste 10000 de fațete) oferă cele mai bune performanțe (în variantă *screen z-buffer*). Acești algoritmi mai oferă avantajul de a putea trata suprafețe strâmbe, fie ele quadrice sau suprafețe parametrice bicubice.

Metoda poate trata fețe reprezentate prin poligoane concave, care se suprapun sau se interpenetrează.

Din cauza avantajelor sale, acest algoritm a fost implementat în hardware (de ex. procesoare grafice, acceleratoare grafice), pentru a permite lucrul "în timp real".

Reprezentarea cu linii ascunse a fost prezentată deoarece pe parcursul lucrării sunt folosite imagini cu corpuri care au fost reprezentate cu ajutorul algoritmilor descriși.

1.3 Concluzii. Contribuții personale

Acest capitol este un studiu bibliografic care are rolul de a crea premisele abordării următoarelor capitole sintetizând informația existentă în literatura de specialitate. El are rolul de a prezenta metodele existente de rezolvare a modelării și vizualizării modelelor. În cadrul aplicației prezentate de teză s-au folosit doar unele din metodele prezentate, cele la care s-a constatat că rezolvă cel mai bine problemele care pot să apară în domeniul construcțiilor în lemn.

În continuare se prezintă contribuțiile personale la acest capitol:

- Sintetizarea rezultatelor cercetărilor pe plan mondial în domeniul abordat de teză, astfel încât să se realizeze o unealtă utilă celor ce doresc să abordeze în continuare acest domeniu.
- Analiza avantajelor și limitelor diferiților algoritmi prezentați în literatura de specialitate, în scopul conceperii unor algoritmi superiori, care să funcționeze corect fără restricții aplicate solidelor operanzi.
- Analiza avantajelor și limitelor diferitelor metode de modelare a solidelor prezentate în literatura de specialitate, din punct de vedere al problemelor de rezolvat.

2 CODAREA PRELUCRĂRILOR INDIVIDUALE

În prima parte a acestui capitol se prezintă elementele de bază utilizate în codarea și conversia codurilor ce descriu prelucrările ce se pot efectua pe o bară paralelipipedică. În partea a doua se prezintă modalitățile de codare a prelucrărilor individuale.

Prelucrarea este o înșiruire de parametri despărțiți prin virgulă care descriu o operație ce trebuie efectuată pe o *bară* pentru a putea fi integrată în cadrul unei construcții (Lignarius, 2000). *Bara* este forma inițială a elementului de construcție. Ea în forma ei inițială este un paralelipiped dreptunghic ca în final, după ce a fost modelată conform prelucrărilor efectuate pe ea, să se îmbine perfect cu toate barele din construcție cu care ea interacționează. Forma inițială se va memora în structura formei finale și ea nu va fi niciodată modificată de prelucrări. Astfel se poate în orice moment reface forma finală aplicând prelucrările pe forma inițială. În continuare, pentru forma finală a barei se va folosi termenul de *corp*. Corpul poate avea dimensiuni mai mici decât bara atât în lungime cât și în secțiune.

Modelarea barei se poate face în două moduri:

- cu ajutorul *prelucrărilor individuale* care pot fi efectuate doar prin specificarea unor parametri relativi la bară fără a fi necesare alte elemente ajutătoare (o altă bară, punct, linie, etc.) (Lignarius, 2001).
- prin îmbinarea barei cu altă bară printr-o anumită prelucrare (Lignarius, 2001)

Primul mod va fi prezentat în acest capitol, iar al doilea în capitolul următor.

2.1 Introducere. Elemente de bază

Paralelipipedul ce circumscrie corpul pe cele trei dimensiuni: lungime, lățime și înălțime va fi denumit în continuare *forma brută* (figura 2.1). Linia paralelă cu cele patru muchii care sunt de-a lungul formeii brute care trece prin centrul de greutate a formeii brute se va denumi în continuare *axa formeii brute*. Ea nu se modifică deci este aceeași și pentru corpul final (Emmer, 1994).

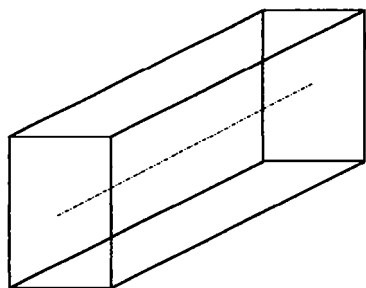


Figura 2.1: Forma brută a corpului

Fețele laterale ale formeii brute sunt paralele cu axa corpului și *fețele de capăt* sunt perpendiculare pe ea (figura 2.2 și 2.3).

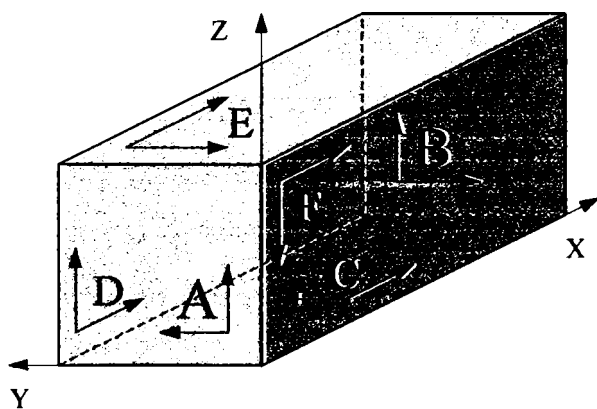


Figura 2.2: Definierea fețelor laterale și cele de capăt ale formeii brute

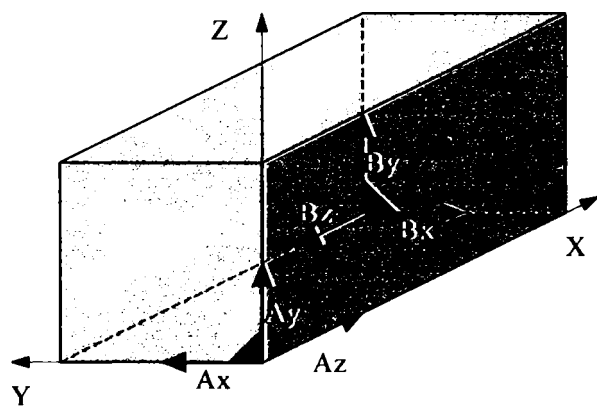


Figura 2.3: Sistemele locale ale fețelor de capăt a formeii brute

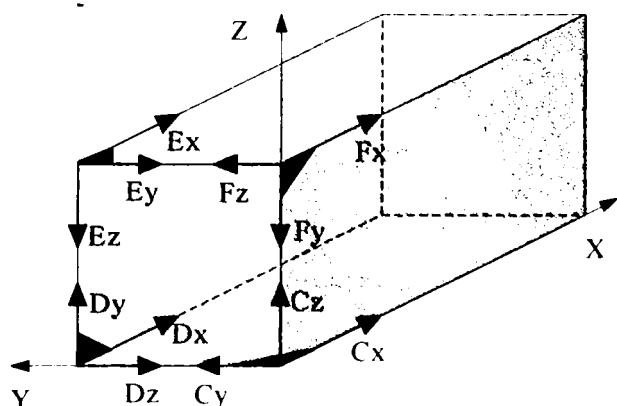


Figura 2.4: Sistemele locale ale fețelor laterale a forme brute

Deci, conform figurilor anterioare *fața de referință* se definește ca fiind una din fețele laterale sau de capăt a forme brute față de care se specifică parametrii.

Fețele de referință se specifică cu ajutorul literelor de la A-F. Ele au o poziție unică în raport cu sistemul local de referință a barei și sunt notate astfel:

- A fața de capăt la începutul barei
- B fața de capăt la sfârșitul barei
- C fața în lungime în planul X-Y
- D fața în lungime paralelă cu F (Z-X)
- E fața în lungime paralelă cu C (X-Y)
- F fața în lungime în planul Z-X

Pe fiecare față se definește un sistem de coordonate ortogonal care are axele x și y în planul feței astfel încât fața să fie complet în cadranul I al sistemului. Axa z are direcția normalei feței și sensul către corp. Poziția sistemului de coordonate de față este unic definită.

Fețele prelucrărilor sunt fețele care rezultă în urma prelucrării corpului. Prelucrările sunt clasificate după anumite proprietăți geometrice și tehnice (Emmer, 1994).

Prelucrarea poate fi sistematizată cu ajutorul noțiunilor comune ale ei. Unele subtipuri, de exemplu găurile perpendiculare, pot fi identificate pe baza geometriei lor, dar trebuie să i se permită utilizatorului introduceri simplificate.

Prelucrările sunt puse sistematic într-o colecție de reprezentări primitive. Ele sunt principial independente și pot să fie grupate. De aceea pot fi definite pe baza asemănărilor geometrice.

Fiecare prelucrare poate fi definită în două moduri.

- unul geometric, intern, denumit *forma în puncte* (Lignarius, 2001).
- descrierea pentru utilizator, denumită *forma parametrică* (Lignarius, 2001).

Forma în puncte este folosită intern la salvarea, prelucrare și predarea din programe CAD spre postprocesor. Ea conține numărul minim de *puncte* ce pot descrie forma geometrică a prelucrării. *Punctul* este specificat prin coordonate 3D, pentru reprezentarea 3D și prin coordonate 2D pentru vizualizări. Pentru introducerea și controlul prelucrărilor se folosește forma parametrică. Ea conține parametrii ca: deplasament, distanțe, unghiuri, etc. Pentru utilizator se va transforma forma în puncte în forma parametrică care poate fi modificată prin intermediul interfețelor ca în final aceasta să se convertească înapoi în forma în puncte.

În funcție de cum modifică sau nu geometria barei, prelucrările pot fi clasificate în:

- prelucrări de capăt care se definesc ca fiind cele care afectează un capăt al barei și pot chiar să-i modifice lungimea. Se execută transversal pe bară.
- prelucrări ce modifică geometria barei dar nu îi afectează nici secțiunea și nici lungimea inițială. Acestea pot fi pe lungimea barei sau pe secțiunea ei
- prelucrări ce nu modifică geometria barei, cum sunt de exemplu marcajele

Prelucrările mai complicate (în general cele care în forma parametrică au trei unghiuri) au nevoie, pentru a putea fi codate, de un sistem de coordonate local al prelucrării. Acesta se definește relativ la prelucrare astfel (Emmer, 1994):

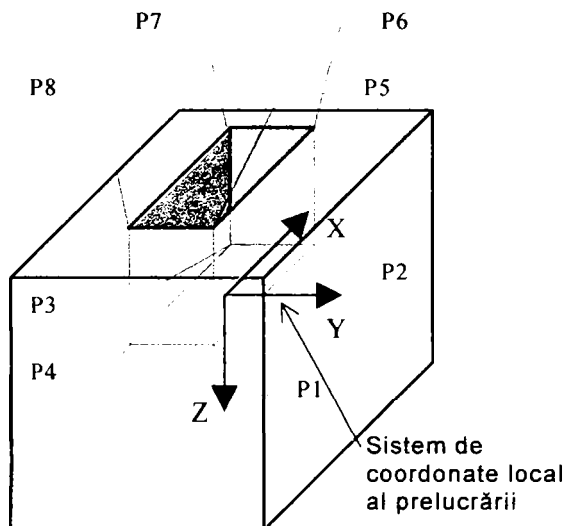


Figura 2.5: Definierea sistemului de coordonate local al prelucrării

Acest sistem este definit relativ la sistemul local al feței de referință. Pentru a ajunge la locul lui, el suferă următoarele rotații și translații (cu EKS este notat sistemul prelucrării și cu BKS sistemul feței de referință):

- a EKS este exact în BKS.
- b EKS este rotit cu unghiul în plan (GW) în jurul lui Z_{EKS} .
- c EKS este rotit cu unghiul de înclinare (NW) în jurul lui Y_{EKS} .
- d EKS este rotit cu unghiul de răsturnare (KW) în jurul X_{EKS} .
- e EKS este deplasat după X, Y, Z.

Cum se pot recalcula unghiurile la trecerea din forma în puncte în forma parametrică:

GW unghiul în planul de bază este întotdeauna între X_{BKS} și proiecția lui X_{EKS} pe XoY_{BKS} .
Se vor trata diferit pentru Z_{EKS} în direcție + sau - Z_{BKS} .

NW unghiul de înclinare este întotdeauna unghiul dintre X_{EKS} și XoY_{BKS} . Se vor trata diferit pentru Z_{EKS} în direcție + sau - Z_{BKS} .

KW unghiul de răsturnare este cel mai greu. Trebuie stabilită din nou situația dinainte de răsturnare (pasul d): dat fiind faptul că în pasul d s-a rotit în jurul axei X_{EKS} , aceasta încă coincide cu X_{EKS} înainte de pasul d. Y_{EKS} înainte de d era așezată în XoY_{BKS} și era perpendiculară pe X_{EKS} . Cu aceste condiții putem determina poziția Y_{EKS} înainte de d (se va face diferența între cazul Z_{EKS} în direcția + sau - Z_{BKS}). unghiul dintre Y_{EKS} înainte și după d este unghiul de răsturnare.

Cu ajutorul acestor unghiuri putem poziționa unic EKS și, dacă se cunoaște un EKS, putem să calculăm unic și unghiurile. Deci dacă se determină forma parametrică din cea în puncte, trebuie să poată fi recunoscut un EKS.

2.1.1 Forma în puncte

Conține: *codul prelucrării* (de forma 4xxx), *funcția* (un număr între 1 și 9), *parametrii suplimentari* (pot fi valori întregi, cu virgulă sau șiruri. Pot să difere de la o prelucrare la alta. Ei sunt despărțiți prin caracterul "~") și *punctele 3D* (date prin coordonatele x, y, z. Descriu geometric prelucrarea) (Lignarius, 2000).

4xxx,y,a~b~c~...,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z,...

Coordonatele punctelor sunt specificate față de sistemul de coordonate local al barei. Punctele au poziții stabilite pe forma brută. Ordinea punctelor unei prelucrări este stabilită și are înțeles funcțional. Normalele fețelor prelucrărilor indică spre exteriorul corpului. Poziția prelucrării pe bara este determinată de geometria ei (Emmer, 1994).

2.1.2 Forma parametrică

Forma parametrică se caracterizează printr-o introducere simplă și repetabilă printr-o interfață de tip cutie de dialog.

- este posibilă folosirea unui număr minim de parametrii, multe însușiri sunt implicite prin specificarea subtipului prelucrării.
- în forma parametrică se folosesc unghiuri. Deplasamentele pe lungime sunt specificate prin deplasament pe x, deplasamentele în secțiune prin deplasament pe y sau z.
- parametrii încearcă să se refere la valori ceea ce duce la o înțelegere mai bună a prelucrării de către utilizator. (de exemplu unghiul unei chertări simple drepte este chiar înclinarea feței acoperișului).
- parametrii sunt dați relativ la sistemul de coordonate local al feței de referință (de exemplu adâncimi).
- pentru fiecare prelucrare se definește o față de referință; determinarea feței de referință nu este întotdeauna univocă (cu toate acestea ea nu se memorează în forma în puncte):
 - utilizatorul poate să aleagă ca față de referință orice față.
 - pentru determinarea automată a feței de referință pe baza geometriei prelucrării se dau reguli specifice fiecărei prelucrări. Dacă determinarea nu este univocă se stabilește o față de referință implicită. Prima față din șirul de fețe posibile care verifică criteriile stabilite se ia ca și față de referință.

Primele trei elemente din forma parametrică sunt aceleași cu primele trei din forma în puncte. Al treilea element conține uneori parametrii suplimentari care nu pot fi determinați din geometrie. Apoi pentru toate prelucrările urmează fața de referință, capătul de referință și deplasamentul pe x. Următorii parametrii diferă de la o prelucrare la alta și descriu numai parametrii care pot fi determinați din geometria prelucrării (Lignarius, 2000).

4xxx,y,a~b~c~...,BS,BE,X,Y,Z,L,B,GW,NW,...

BS	fața de referință (implicit C)
BE	capăt de referință (implicit A)
X	deplasamentul pe lungime a unui punct al prelucrării
Y,Z,L,B...	parametrii specifici fiecărei prelucrări în parte.

2.1.3 Menținerea muchiilor și fețelor

La fiecare prelucrare se generează anumite muchii. Fiecărei muchii i se poate specifica cum să fie prelucrată:

0 trebuie executată ascuțită

+ X muchia nu are voie să fie afectată, se generează o rotunjire cu o rază X

- X muchia poate fi afectată, astfel încât se elimină materialul din apropierea muchiei. X dă raza maximă sau jumătatea lățimii sculei cu care se prelucrează.

Pentru fiecare față definită, a prelucrării se poate specifica subclasificarea care exprimă în ce măsură planul trebuie să rămână neschimbat:

0 valoare standard, trebuie să se mențină complet,

1 poate fi afectată de liniile de tăiere cu alte fețe (de ex.: gaura de cep)

2 poate fi distrusă complet (de ex. fața de capăt în direcția lungimii unei eclisări care este efectuată de-a lungul corpului)

2.1.4 Funcții necesare efectuării prelucrărilor

Generarea unei prelucrări se face parcurgând următorii pași:

- utilizatorul selectează corpul pe care se va memora prelucrarea nouă
- utilizatorul selectează tipul prelucrării noi și specifică prin intermediul cutiei de dialog aferente parametrului prelucrării (care formează forma parametrică)
- intern, în cadrul algoritmului se apelează funcția de conversie din forma parametrică în forma în puncte; se verifică corectitudinea prelucrării și dacă este corectă ea se memorează pe corpul selectat
- se construiește corpul negativ al prelucrării; acest corp se scade din corpul selectat obținându-se corpul rezultat în urma efectuării prelucrării noi

În acest moment utilizatorul poate să modifice (prin modificarea parametrilor acesteia) sau să șteargă oricare prelucrare de pe corp. Înainte de a se deschide cutia de dialog de modificare aferentă unei prelucrări, se realizează conversia din forma în puncte în forma parametrică (pe corp se memorează doar forma în puncte). Se modifică parametrul după care se repetă operațiile prezentate la generarea unei prelucrări noi.

2.2 Prezentarea prelucrărilor individuale

Pentru fiecare prelucrare se va prezenta forma în puncte și cea parametrică cu descrierea paramerilor corespunzători.

Se vor prezenta următoarele prelucrări:

- grupa prelucrărilor 4001
 - 4001, 1 tăiere transversală
 - 4001, 2 tăiere longitudinală
- grupa prelucrărilor 4002
 - 4002, 1 chertare simplă
 - 4002, 2 chertare simplă dreaptă
 - 4002, 3 chertare simplă de capăt
 - 4002, 4 chertare simplă în lungime
 - 4002, 5 tăiere longitudinală dublă concavă
 - 4002, 6 tăiere longitudinală dublă convexă
 - 4002, 7 tăiere în V
- grupa prelucrărilor 4003
 - 4003, 1 eclisare standard dreaptă
 - 4003, 2 eclisare standard strâmbă
 - 4003, 3 eclisare de-a lungul corpului
- grupa prelucrărilor 4004
 - 4004, 1 chertare colț oarecare
 - 4004, 2 chertare colț dreaptă
- grupa prelucrărilor 4005
 - 4005, 1 chertare simplă de-a lungul corpului
 - 4005, 2 gaură de cep deschisă
- grupa prelucrărilor 4006
 - 4006, 1 gaură de cep
 - 4006, 2 canal limitat
 - 4006, 3 gaură de cep oarecare
 - 4006, 4 gaură de cep coadă de rândunică de capăt
 - 4006, 5 gaură de cep coadă de rândunică laterală
- grupa prelucrărilor 4007
 - 4007, 1 cep drept
 - 4007, 2 cep cu deplasamente
 - 4007, 3 cep rotit

- 4007, 4 cep clasic
- 4007, 5 cep coadă de rândunică
- grupa prelucrărilor 4008
 - 4008, 1 chertare dublă la capăt
 - 4008, 2 chertare dublă pe față laterală
 - 4008, 3 chertare "tăbăcită"
 - 4008, 4 chertare tip sapă
 - 4008, 5 ornament capăt concav
 - 4008, 6 ornament capăt convex
 - 4008, 7 prelucrare tip arc de cerc
 - 4008, 8 profil rotunjit
- grupa prelucrărilor 4009
 - 4009, 1 prelucrare efectuată cu firiz rotund
- grupa prelucrărilor 4101
 - 4101, 0 gaură de burghiu oarecare
 - 4101, 1 gaură de burghiu perpendiculară
- grupa prelucrărilor 4201
 - 4201, 0 marcajul
- grupa prelucrărilor 4202
 - 4202, 0 rindeluirea

2.2.1 Grupa prelucrărilor 4001

Geometrie:

- descrie un plan dat prin trei puncte.
- punctele P1, P2 și P3 sunt puncte de tăiere ale planului cu muchiile forme brute

Decrieri tehnice:

- se generează o suprafață a sculei care nu este limitată de nici o față a forme brute
- sculele pot să depășească fețele de limitare a forme brute

Parametrii suplimentari:

- (- valoare pentru suprafața corectă a sculei date)
- (- valoare pentru calitatea prelucrării muchiei sculei date)

2.2.1.1 4001, 1 - Tăiere transversală

- Suprafețele prelucrării nu sunt paralele cu axa forme brute (figura 2.6)

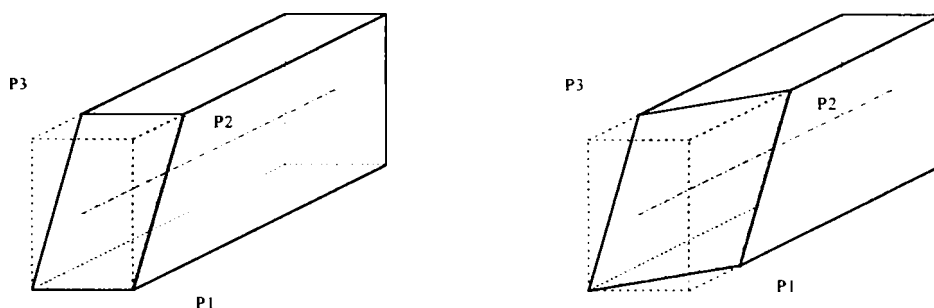


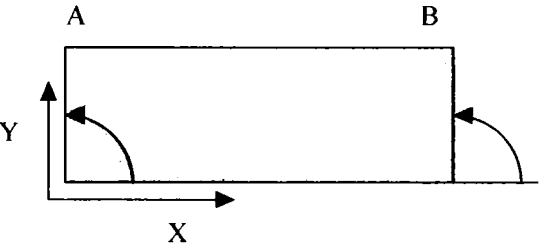
Figura 2.6: Tăiere transversală

Forma în puncte:

4001,1,,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z

- nu are parametri suplimentari

Forma parametrică:**4001,1,,BS,BE,X,GW,NW**

	Parametru	Descriere
BS	Fața de ref.	implicit = C
BE	Capăt de ref.	Capătul spre care este orientată normala planului de tăiere (P1, P2, P3)
X	Deplasare pe x	Se măsoară de la $x=0$ până la punctul dat de tăierea planului de tăiere și axa x a feței de referință
GW	Unghi în planul de bază	<p>Planul de tăiere (P1, P2, P3) și planul de referință se taie de-a lungul unei linii. Unghiul dintre această linie și axa x a feței de referință este GW. De aceea este valabilă condiția : $0^\circ \leq GW < 180^\circ$.</p> <p>Se măsoară de la axa +x a sistemului de referință rotit în sens trigonometric (rotire în jurul lui z). Dacă tăierea este la capătul de început unghiul se măsoară în forma brută, dacă tăierea este la capătul de sfârșit unghiul se măsoară în exteriorul forme brute.</p> 
NW	Unghi de înclinare	Este unghiul diedru dintre planul de tăiere și fața de referință. Unghiul se măsoară de la fața de referință către axa z a feței de referință. Dacă tăierea este la capătul de început unghiul se măsoară în forma brută, dacă tăierea este la capătul de sfârșit unghiul se măsoară în exteriorul forme brute.

2.2.1.2 4001, 2 – Tăiere longitudinală

- Suprafața prelucrării este paralelă cu axa forme brute (figura 2.7)

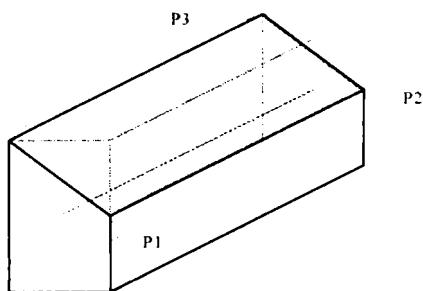


Figura 2.7: Tăiere longitudinală

Forma în puncte:

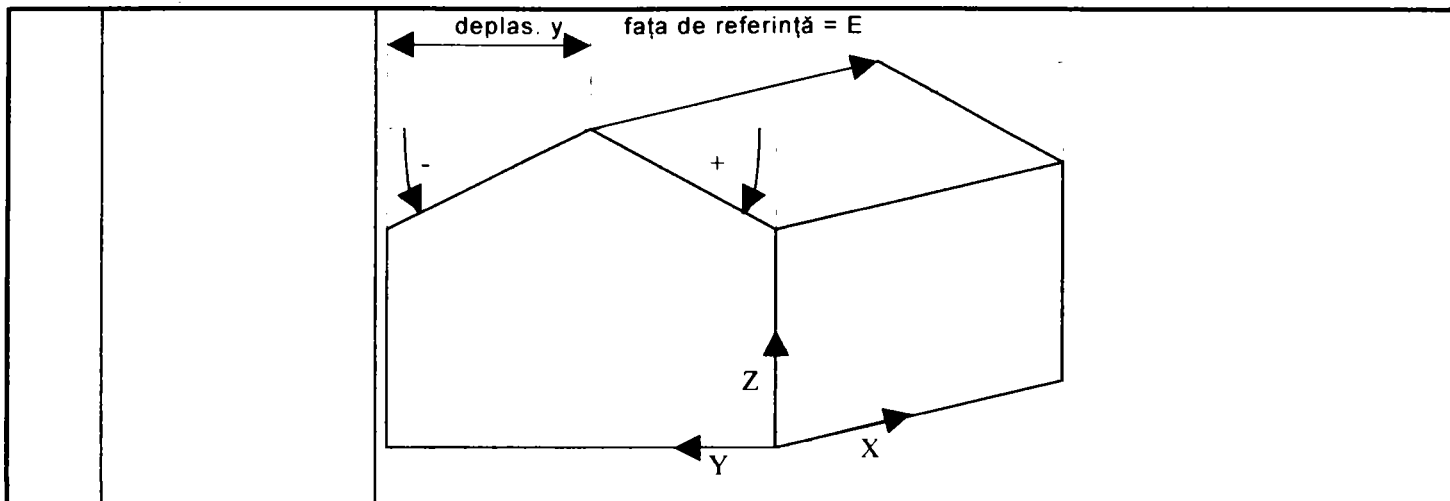
4001,2,,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z

- nu are parametri suplimentari

Forma parametrică:

4001,2,,BS,(BE),(X),Y,NW

	Parametru	Descriere
BS	Fața de ref.	Planul de tăiere taie fața de referință. Se iau în ordinea : E, C, D, F.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	Întotdeauna '0'
Y	Deplasare pe y	Planul de tăiere taie fața de referință după o linie. Y este deplasarea pe y a acestei linii în fața de referință.
NW	Unghi de înclinare	Este unghiul diedru dintre planul de tăiere și fața de referință. Unghiul se măsoară în exteriorul forme brute. Axa de rotație este linia de tăiere cu sensul în sensul axei x a forme brute Dacă se rotește în sens trigonometric unghiul este pozitiv, dacă se rotește în sens orar unghiul este negativ.



2.2.2 Grupa prelucrărilor 4002

Geometrie:

- cu ajutorul a patru puncte se definesc două plane (E1, E2)
- planele sunt mărginite de linia de tăiere a celor două plane
- punctele P1 și P2 sunt punctele de intersecție a liniei de tăiere a planelor cu fețele formei brute
- punctele P3 și P4 sunt intersecția planelor E1 și E2 cu muchiile formei brute
- planul E1 este definit de P1, P2, P3 și E2 de P2, P1, P4

Descrieri tehnice:

- Se generează două plane, care la bază sunt limitate de linia lor de tăiere.
- Sculele nu au voie să taie corpul în două
- Sculele pot să fie direcționate de-a lungul unei direcții fără punct de capăt
- Formarea unei muchii strâmbe nu ridică probleme

Parametrii suplimentari:

- valoare pentru suprafața corectă a suprafețelor sculei date
- valoare pentru calitatea muchiilor sculei date

2.2.2.1 4002, 1 – Chertare simplă dreaptă

- unghiul dintre fețele prelucrării este 90° .
- unghiul dintre muchia (P1,P2) și fețele laterale a formei brute este 90° .
(P1,P2) este paralel cu celelalte fețe laterale deci adâncimea prelucrării este constantă
(figura 2.8)

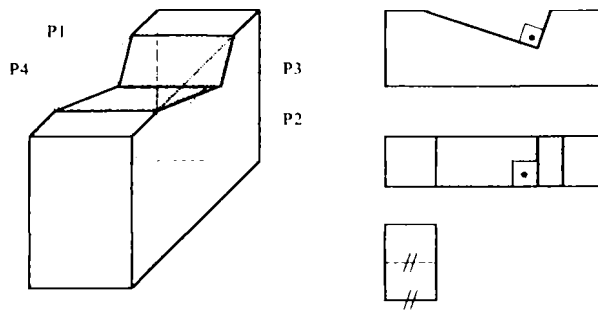


Figura 2.8: Chertare simplă dreaptă

Forma în puncte:

4002,1,A,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z,P4x,P4y,P4z

A gaură burghiu = 0 nu se face nici o gaură
 > 0 diametrul găurii de burghiu

Parametrul are semnificație doar pentru mașină. În mediul CAD nu se generează efectiv o gaură de burghiu.

Forma parametrică:

4002,1,A,BS,BE,X,Z,NW

	Parametru	Descriere
		Muchie chertare: P1 – P2 Punct de poziționare: P1 sau P2 punctul cu cel mai mic Y_{BS} (Y măsurat în fața de referință) Fața de bază: se stabilește direcția de rotire de la punctul de poziționare spre celălalt punct, și se rotește de la fața de bază în sens pozitiv în interiorul chertării
BS	Fața de ref.	C, E, F, D: se duce o suprafață prin muchia chertării cu normala pe direcția bisectoarei unghiului dintre fețele prelucrării. Această față este privită ca și fața de bază la o eclisare strâmbă (4003, 2). Fața de referință este prima față pentru care sunt valabile relațiile $-90^\circ \leq NW \leq +90^\circ$ und $45^\circ \leq SW \leq 135^\circ$.
BE	Fața de bază	Întotdeauna 'A'.

X	Deplasare pe x	X_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
NW	Unghi de înclinare	Este unghiul diedru dintre fața de bază și fața de referință. Unghiul se măsoară în interiorul chertării deci în exteriorul formei brute. Dacă normala feței de bază indică capătul de început al formei brute unghiul este pozitiv, dacă indică capătul de sfârșit unghiul este negativ.

Indicele BS folosit la coordonate are semnificație de sistemul în care este dată coordonata respectivă. (X_{BS} este coordonata X măsurată relativ la sistemul local al feței de referință).

2.2.2.2 4002, 2 – Chertare simplă strâmbă

- unghiul dintre fețele prelucrării este 90° .
- muchia (P1,P2) nu trebuie să fie perpendiculară pe fețele laterale ale formei brute (figura 2.9)

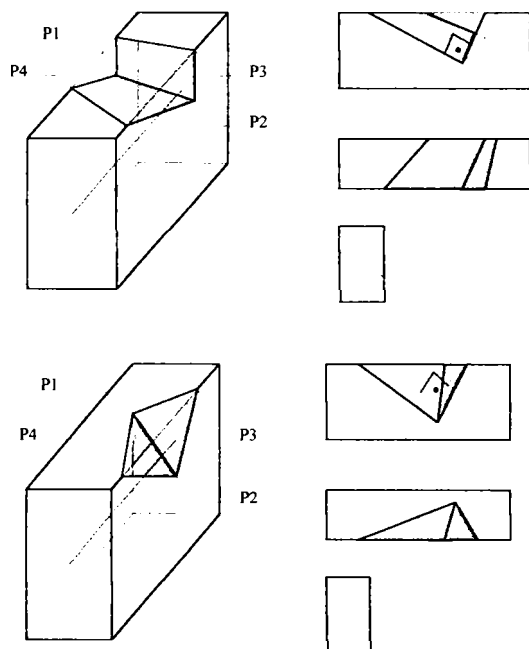


Figura 2.9: Chertare simplă strâmbă

Forma în puncte:

4002,2,A,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z,P4x,P4y,P4z

A gaură burghiu = 0 nu se face nici o gaură
 > 0 diametrul găurii de burghiu

Forma parametrică:

4002,2,A,BS,BE,X,Z,NW,GW,SW

	Parametru	Descriere
		<p>Muchie chertare: P1 - P2</p> <p>Punctul de poziționare: punctul de tăiere dintre linia P1 - P2 cu planul ZoX a feței de referință.</p> <p>Fața de bază: se stabilește direcția de rotire de la punctul de poziționare spre celălalt, deci se rotește fața de bază în interiorul chertării.</p> <p>Fața laterală: Cealaltă față de prelucrare care este perpendiculară pe fața de bază.</p>
BS	Fața de ref.	C, E, F, D: se duce o suprafață prin muchia chertării cu normala pe direcția bisectoarei unghiului dintre fețele prelucrării. Această față este privită ca și fața de bază la o eclisare strâmbă (4003, 2). Fața de referință este prima față pentru care sunt valabile relațiile: $-90^\circ \leq NW \leq +90^\circ$ und $45^\circ \leq SW \leq 135^\circ$.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
GW	Unghi în planul de bază	Este unghiul diedru dintre fața laterală și planul ZoX al sistemului de referință. Se măsoară în interiorul chertării deci în exteriorul forme brute.
NW	Unghi de înclinare	Este unghiul diedru dintre fața de bază și fața de referință (BS). Unghiul se măsoară în interiorul chertării deci în exteriorul forme brute. Dacă normala feței de bază indică spre începutul forme brute unghiul este pozitiv, dacă indică spre capătul de sfârșit al forme brute unghiul este negativ.
SW	Unghiul pantei	Este unghiul diedru dintre fața de bază și planul ZoX al sistemului de referință (BS). Se măsoară în interiorul prelucrării deci în exteriorul forme brute.

2.2.2.3 4002, 3 – Chertare simplă de capăt

- unghiurile între fețele prelucrării sunt 90°.
- unghiul dintre o față a prelucrării și fețele laterale care o mărginesc este 90°. A doua față este paralelă cu această față. De asemenea muchia (P1-P2) este paralelă cu această față laterală
- muchia (P1-P2) nu trebuie să fie perpendiculară pe nici o față laterală (figura 2.10)

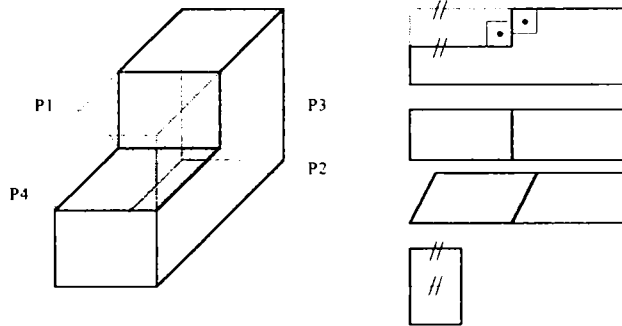


Figura 2.10: Chertare simplă de capăt

Forma în puncte:

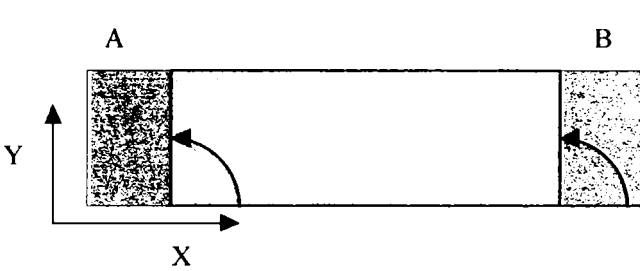
4002,3,,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z,P4x,P4y,P4z

- nu are parametri suplimentari

Forma parametrică:

4002,3,,BS,BE,X,Z,GW

	Parametru	Descriere
		<p>Punct de poziționare: este punctul de tăiere dintre dreapta P1-P2 cu planul ZoX al feței de referință.</p> <p>Fața de bază: suprafața prelucrării care este paralelă cu fața de referință.</p> <p>Fața laterală: a doua suprafață a prelucrării care este perpendiculară pe fața de bază</p>
BS	Fața de ref.	C, E, F, D: Prima față pe care se gasește P3 sau P4 și este paralelă cu P1-P2.

BE	Capăt de ref.	Normala feței laterale este orientată către capătul de referință.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
GW	Unghi în planul de bază	Suprafața prelucrării (P1, P2, P3) se intersectează cu fața de referință de-a lungul unei linii de tăiere. Unghiul dintre această linie și axa x pozitivă a feței de referință este unghiul în planul de bază. Se măsoară întotdeauna de la axa x pozitivă în sens trigonometric: dacă capătul de referință este cel de început (A), unghiul se măsoară în forma brută, dacă este cel de sfârșit (B), unghiul se măsoară în afara forme brute. 

2.2.2.4 4002, 4 – Chertare simplă în lungime

- unghiul dintre suprafețele prelucrării este 90°.
- unghiul dintre suprafețele prelucrării și suprafețele laterale ale forme brute care mărginesc suprafața prelucrării este 90°. A doua suprafață a prelucrării este paralelă cu această suprafață laterală.
- muchia (P1-P2) este paralelă cu axa forme brute (figura 2.11)

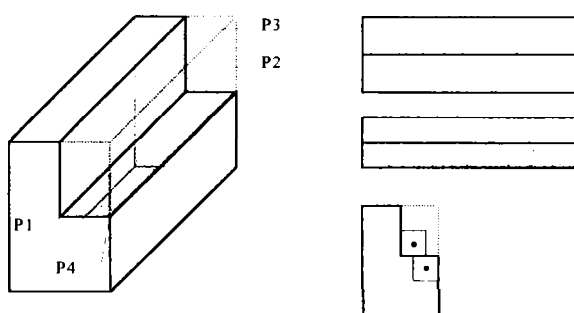


Figura 2.11: Chertare simplă în lungime

Forma în puncte:

4002,4,,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z,P4x,P4y,P4z

- nu are parametri suplimentari

Forma parametrică:

4002,4,,BS,BE,X,Z,B

	Parametru	Descriere
BS	Fața de ref.	C, E, F, D: fața laterală a formei brute care are axa x în interiorul prelucrării.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplas pe x	Întotdeauna '0'
Z	Adâncime	Z_{P1} (= Z_{P2})
B	Lățime	Y_{P1} (= Y_{P2})

2.2.2.5 4002, 5 – Tăiere longitudinală dublă concavă

- muchia (P1-P2) este paralelă cu axa formei brute
- unghiul exterior între suprafețele prelucrării este mai mare decât 180°

Forma în puncte:

4002,5,,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z,P4x,P4y,P4z

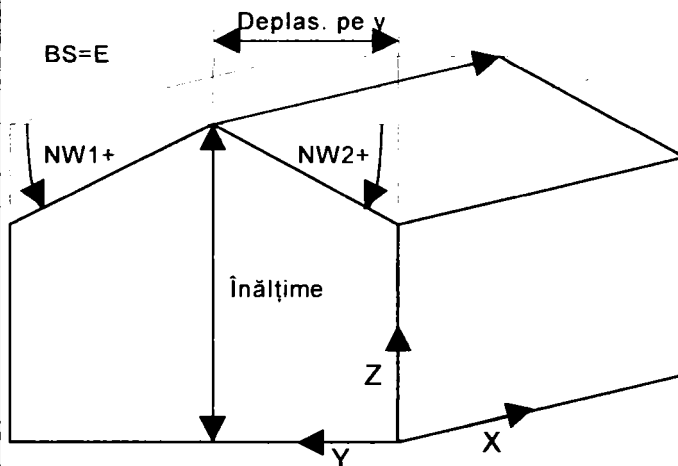
- nu are parametri suplimentari

Forma parametrică:

4002,5,,BS,BE,X,Y,Z,NW1,NW2

	Parametru	Descriere
BS	Fața de ref.	E, C, D, F: prima față laterală a formei brute pe care proiecția unei suprafețe a prelucrării este complet între suprafața laterală anterioară și coama prelucrării iar proiecția celeilalte suprafețe a prelucrării este între coamă și suprafața laterală următoare.

BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	Întotdeauna '0'
Y	Deplas. în secțiune	Suprafețele prelucrării se taie după o linie (coama prelucrării). Deplasarea în secțiune a acestei linii este Y-ul ei în fața de referință.
Z	Înălțimea	Se măsoară de la fața de referință la fața opusă feței de referință în direcția z.
NW1	Unghi de înclinare 1	Este unghiul diedru între fața prelucrării care taie planul XoZ a feței de referință și fața de referință. Unghiul se măsoară în exteriorul forme brute. Axa de rotație este coama prelucrării în direcția pozitivă a axei x a forme brute. Dacă se rotește în sens orar unghiul este pozitiv, dacă se rotește în sens trigonometric unghiul este negativ.
NW2	Unghi de înclinare 2	Este unghiul diedru între suprafața prelucrării care nu taie planul XoZ al feței de referință și fața de referință. Unghiul se măsoară în exteriorul forme brute. Axa de rotație este coama prelucrării cu sensul în sensul axei x a barei. Dacă se rotește în sens trigonometric unghiul este pozitiv, dacă se rotește în sens orar unghiul este negativ.



2.2.2.6 4002, 6 – Tăiere longitudinală dublă convexă

- muchia (P1-P2) este paralelă cu axa forme brute
- unghiul exterior dintre suprafețele prelucrării este mai mic decât 180° (figura 2.12)

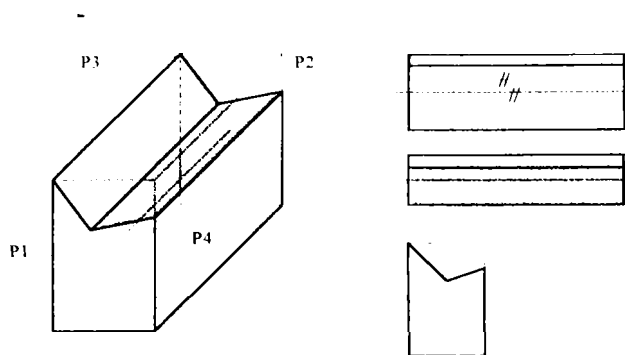


Figura 2.12: Tăiere longitudinală dublă convexă

Forma în puncte:

4002,6,,P1x,P1y,P1z,,...,P4x,P4y,P4z

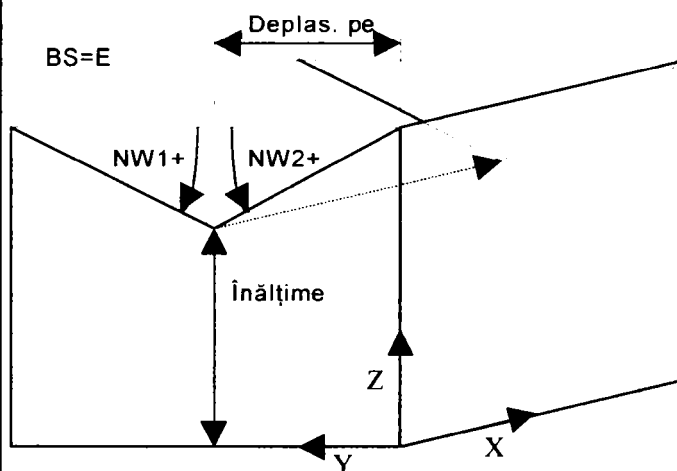
- nu are parametri suplimentari

Forma parametrică:

4002,6,,BS,BE,X,Y,Z,NW1,NW2

	Parametru	Descriere
BS	Fața de ref.	E, C, D, F: este prima față laterală a formei brute pentru care proiecția unei suprafețe de contact (stânga) este complet între suprafața anterioară și muchia prelucrării și cealaltă suprafață a prelucrării între muchia prelucrării și suprafața următoare.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	Întotdeauna '0'
Y	Deplasare pe y	Suprafețele prelucrării se taie după o dreaptă (muchia prelucrării). Deplasarea acestei drepte în secțiune se măsoară de la fața de referință: BS = C sau E: distanța la planul XZ a barei. BS = D sau F: distanța la planul XY a barei.
Z	Înălțimea	Înălțimea muchiei prelucrării se măsoară de la fața de referință la fața opusă în direcția Z.

NW1	Unghi de înclinare 1	Este unghiul diedru dintre suprafața prelucrării care nu taie planul XoZ al planului de referință (stânga) și fața de referință. Unghiul se măsoară în exteriorul forme brute. Axa de rotație este paralelă cu muchia prelucrării cu sensul în direcția x a barei. Dacă se rotește în sens trigonometric unghiul este pozitiv, dacă se rotește în sens orar unghiul este negativ.
NW2	Unghi de înclinare 2	Este unghiul diedru între suprafața prelucrării care nu taie planul XZ al feței de referință (dreapta) și fața de referință. Unghiul se măsoară în exteriorul forme brute. Axa de rotație este paralelă cu muchia prelucrării în sensul axei x a barei. Dacă se rotește în sens orar unghiul este pozitiv, dacă se rotește în sens trigonometric unghiul este negativ.



2.2.2.7 4002, 7 – Tăiere în V

- prelucrarea afectează una din suprafețele de capăt complet sau parțial. Cealaltă suprafață de capăt nu este atinsă.
- cele două suprafețe ale prelucrării formează două linii de tăiere și un punct de tăiere cu fața de referință. Dacă se pune axa barei în acest punct cele două linii trebuie să fie de o parte și alta a liniei (figura 2.13).

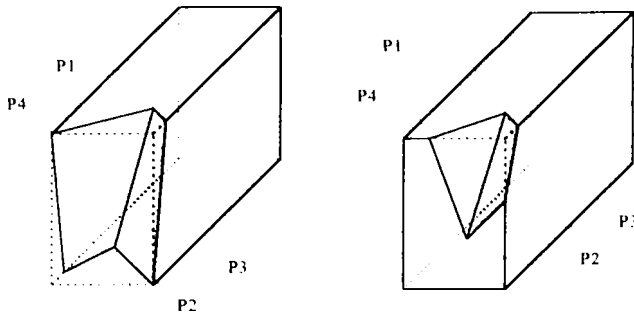


Figura 2.13: Tăiere în V

Forma în puncte:

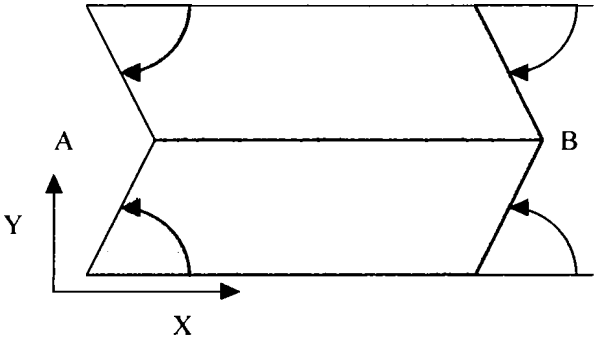
4002,7,,P1x,P1y,P1z,P2x,P2y,P2z,P3x,P3y,P3z,P4x,P4y,P4z

- nu are parametri suplimentari

Forma parametrică:

4002,7,,BS,BE,X,Y,GW1,NW1,GW2,NW2

	Parametru	Descriere
BS	Fața de ref.	C,E,F,D: prima față laterală a formei brute pentru care punctul de tăiere dintre prelungirea liniei de tăiere (P1-P2) și fața de referință este între 0 și Ymax (pentru C, E) respectiv Zmax (pentru D, F).
BE	Capăt de ref.	Fața de capăt care este afectată de prelucrare.
X	Deplasare pe x	Se măsoară până la punctul de tăiere a liniei de tăiere cu fața de referință.
Y	Deplasare pe y	Se măsoară până la punctul de tăiere a liniei de tăiere cu fața de referință
GW1 GW2	Unghi în planul de baza 1+2	Cele două fețe ale prelucrării generează două linii de tăiere (raze) și un punct de tăiere cu suprafața de referință. Dacă se pune axa barei în acest punct cele două raze trebuie să fie în părți diferite ale axei. Raza cu cel mai mare Y sau Z corespunde feței din stânga=1. Acesteia îi corespund GW1 și NW1. Unghiul în planul de bază pentru fața din stânga (GW1) se măsoară de la axa x pozitivă a feței următoare în planul XoY a

		<p>feței de referință în sens orar: Dacă capătul de referință este la început (A), unghiul se măsoară în interiorul forme brute, dacă capătul este cel de sfârșit unghiul se măsoară în exteriorul forme brute.</p> <p>Unghiul pentru fața din dreapta (GW2) este măsurat întotdeauna de la axa x pozitivă în sens trigonometric spre raza din dreapta: dacă capătul de referință este începutul se măsoara în interiorul forme brute, dacă capătul de referință este de sfârșit unghiul se măsoară în exteriorul forme brute.</p> 
<p>NW1</p> <p>NW2</p>	<p>Unghi de înclinare</p> <p>1+2</p>	<p>Sunt unghiurile diedre dintre fețele prelucrării și fața de referință. Unghiul se măsoară întotdeauna de la fața de referință la axa z a feței de referință. Dacă capătul de referință este la început (A), unghiul se măsoara în interiorul forme brute, dacă este la capătul de sfârșit unghiul se măsoară în exteriorul forme brute.</p>

2.2.3 Grupa prelucrărilor 4003

Geometrie:

- se formează trei plane, care se taie după două linii de tăiere mărginite în interiorul corpului.
- prelucrarea se definește prin șase puncte.
- punctele P1, P2, P3 și P4 sunt punctele de intersecție dintre liniile de tăiere și fețele forme brute.
- punctele P5 și P6 sunt punctele de intersecție dintre fețele prelucrării E2 și E3 cu muchiile barei
- planul E1 construiește planul de bază limitat definit de punctele P1, P2, P3; planul E2 este definit de punctele P2, P1, P5; planul E3 este definit de punctele P4, P3, P6
- muchia K1 = P1-P2, K2 = P3-P4 (figura 2.14)

Descrieri tehnice:

- sculele intră pe o parte și trec pe partea cealaltă.
- prelucrarea are două muchii care dacă sunt ascuțite nu ridică probleme

Parametrii suplimentari:

- valoare pentru suprafața corectă a fețelor sculei date
- valoare pentru calitatea muchiilor sculei date

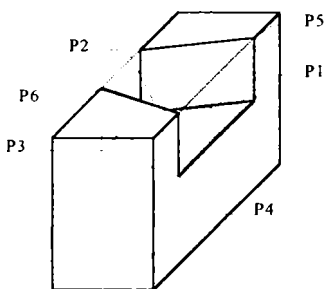


Figura 2.14: Cele șase puncte ce descriu orice tip de eclisare. Poziția lor spațială diferă în funcție de tipul eclisării

2.2.3.1 4003, 1 – Eclisare standard dreaptă

- unghiul dintre fețele prelucrării este 90° .
- fețele laterale ale prelucrării formează cu una din fețele laterale ale formei brute un unghi de 90° . A treia față a prelucrării, fața de bază a prelucrării, este paralelă cu această față laterală. (P1-P2) și (P3-P4) sunt paralele cu această față laterală.
- muchiile (P1-P2) și (P3-P4) sunt paralele între ele dar nu trebuie să fie perpendiculare pe fețele laterale ale formei brute (figura 2.15)

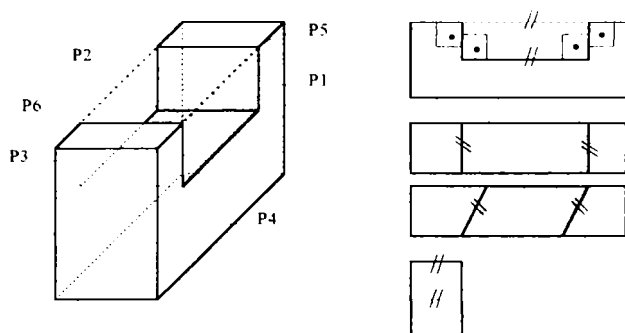


Figura 2.15: Eclisare standard dreaptă (4003, 1)

Forma în puncte:

4003,1,,P1x,P1y,P1z,,...,P6x,P6y,P6z

- nu are parametri suplimentari

Forma parametrică:

4003,1,,BS,BE,X,Z,L,GW

	Parametru	Descriere
		<p>Punct de poziționare: punctul de intersecție dintre linia P1-P2 și P3-P4 cu planul ZoX a feței de referință. Din acestea se alege punctul cu X_{BS} cel mai mic.</p> <p>Fața de bază: fața prelucrării pe care sunt perpendiculare celelalte două fețe.</p> <p>Fața laterală: este perpendiculară pe fața de bază.</p>
BS	Fața de ref.	Fața de bază este paralelă cu această față.

BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
L	Lungime	Distanța dintre fețele laterale
GW	Unghi în planul de bază	Este unghiul diedru dintre fața laterală a prelucrării care trece prin punctul de poziționare și planul ZoX a feței de referință. Se măsoară în interiorul prelucrării deci în exteriorul formei brute.

2.2.3.2 4003, 2 – Eclisare standard strâmbă

- unghiul dintre suprafețele prelucrării este 90° .
- muchiile (P1-P2) și (P3-P4) sunt paralele între ele, dar nu trebuie să fie perpendiculare pe fețele laterale.
- suprafața de bază nu este paralelă cu nici o față laterală a formei brute (figura 2.16)

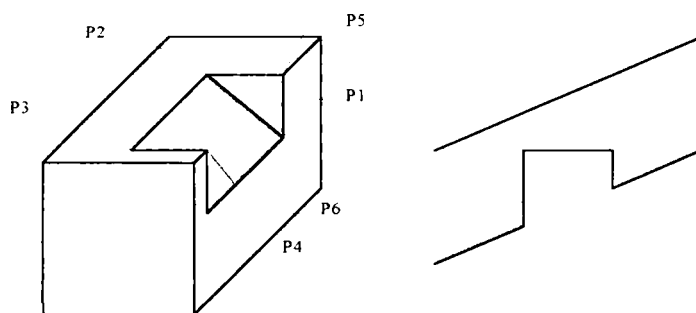


Figura 2.16: Eclisare standard strâmbă (4003, 2)

Se definește un EKS atașat prelucrării ca în figura 2.17.

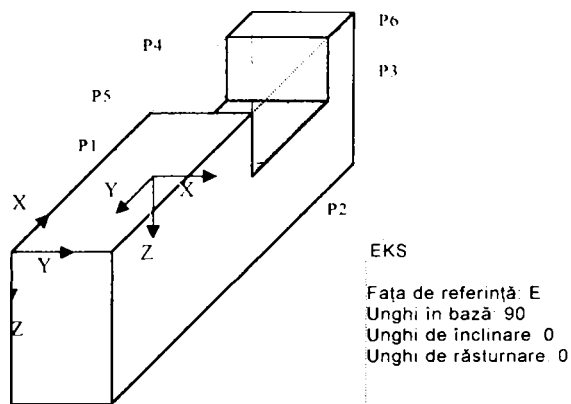


Figura 2.17: Definirea lui EKS în cazul eclisării standard strâmbe

Forma în puncte:

4003,2,,P1x,P1y,P1z,....,P6x,P6y,P6z

- nu are parametri suplimentari

Forma parametrică:

4003,2,,BS,BE,X,Z,L,NW,GW,SW

Parametru	Descriere
	<p>Punct de poziționare: este punctul de intersecție a liniei P1-P2 și P3-P4 cu planul ZoX al feței de referință. Din acestea se alege punctul cu X_{BS} mai mic.</p> <p>Muchia de bază: este muchia interioară a prelucrării care trece prin punctul de poziționare.</p> <p>Fața de bază: pe această față sunt perpendiculare celelalte două.</p> <p>Fața laterală: este perpendiculară pe fața de bază.</p> <p>EKS: axa X trece prin punctul de poziționare și este de-a lungul muchiei de bază. Axa Z este orientată invers față de normala feței de bază.</p>
BS	<p>Fața de ref. C, E, F, D: este prima față din șirul de plane pentru care sunt valabile condițiile:</p> $-90^\circ \leq NW \leq +90^\circ \text{ und } 45^\circ \leq SW \leq 135^\circ.$ <p>(normala feței de bază indică spre această față și unghiul cu muchia de bază este $\leq 45^\circ$. Normala feței de bază este paralelă cu axa X deci intră doar C și F în discuție).</p>

BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
L	Distanța	Distanța dintre fețele laterale.
GW	Unghi în planul de bază	Este unghiul în planul de bază a lui EKS față de BKS.
NW	Unghi de înclinare	Este unghiul de înclinare a lui EKS față de BKS
KW	Unghi de răsturnare	Este unghiul de răsturnare a lui EKS față de BKS

2.2.3.3 4003, 3 – Eclisare de-a lungul corpului

- unghiul dintre fețele prelucrării este 90° .
- muchiile (P1-P2) și (P3-P4) sunt paralele între ele și paralele cu axa
- P1 și P4 sunt pe începutul barei
- fața de bază a prelucrării este paralelă cu axa forme brute

Forma în puncte:

4003,3,A,P1x,P1y,P1z,...,P6x,P6y,P6z

A păstrarea feței de bază 0 = fața de bază poate fi distrusă
1 = (implicit) fața de bază trebuie păstrată

Forma parametrică:

4003,3,A,BS,BE,X,Y,Z,B,KW

	Parametru	Descriere
		<p>EKS: sistem de coordonate al prelucrării: axa X indică direcția P1-P2. axa Z este inversă axei Z a sistemului local al formei brute. Axa Y este în fața de bază. Originea este în punctul de poziționare.</p> <p>Punctul de poziționare: punctul de mijloc între P1 și P4.</p> <p>Fața de bază: pe această față sunt perpendiculare ambele fețe laterale.</p> <p>Fața laterală: este perpendiculară pe fața de bază.</p>
BS	Fața de ref.	C, E, F, D: este prima din șirul de fețe laterale a formei brute în ordinea dată care este tăiată de normala feței de bază dusă în punctul de poziționare.
BE	Capă de ref.	Întotdeauna 'A'.
X	Deplasare pe x	Întotdeauna 0
Y	Deplasare pe y	Y_{BS} al punctului de poziționare
Z	Tiefe	Z_{BS} al punctului de poziționare
B	Lățime	Distanța între fețele laterale ale prelucrării.
KW	Unghi de răsturnare	Unghiul de răsturnare al lui EKS față de sistemul BKS.

2.2.4 Grupa prelucrărilor 4004

Geometrie:

- se generează trei plane, care se taie în trei linii limitate; punctul de tăiere a celor trei plane este în interiorul formei brute.
- prelucrarea se definește prin patru puncte.
- punctele P1, P3 și P4 sunt punctele de tăiere ale liniilor limitate cu fețele formei brute.
- punctul P2 este punctul de intersecție a celor trei plane de prelucrare.
- planul E1 formează fața de bază limitată în două părți; este definit prin punctele P1, P2, P3.
- planul E2 este definit de punctele P2, P1, P4.
- planul E3 este definit de punctele P2, P4, P3.
- muchia K1 = P1-P2, K2 = P2-P3.
- muchia K3 = P2-P4 nu este muchie de limitare a feței de bază.

Descrieri tehnice:

- scula poate să intre pe o anumită față a barei și trebuie să iasă pe aceeași față. Bara nu poate fi străpunsă.
- deseori există o muchie critică, care poate fi rotunjită.
- faptul că muchiile K1 și K2 sunt strâmbe nu ridică probleme.

Parametrii suplimentari:

- valoare pentru suprafața corectă a suprafețelor sculei date
- valoare pentru calitatea muchiilor sculei date

2.2.4.1 4004, 1 – Chertare colț oarecare

- E1 (P1, P2, P3) este planul de bază, celelalte plane sunt perpendiculare pe aceasta.
- planul E1 este perpendicular pe două fețe laterale ale formei brute (figura 2.18)

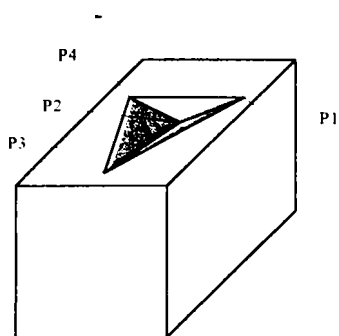


Figura 2.18: Chertare colț

Forma în puncte:

4004,1,,P1x,P1y,P1z,....,P4x,P4y,P4z

Deocamdată are doar descrierea în puncte deoarece nu s-a stabilit o descriere fezabilă pentru forma parametrică.

2.2.4.2 4004, 2 – Chertare colț dreaptă

- fiecare față a prelucrării este perpendiculară pe celelalte două
- fețele prelucrării sunt perpendiculare sau paralele cu fețele laterale ale formei brute (figura 2.19)

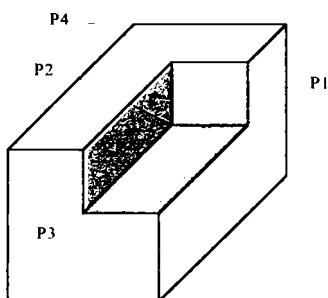


Figura 2.19: Chertare colț dreaptă

Forma în puncte:

4004,2,,P1x,P1y,P1z,....,P4x,P4y,P4z

Deocamdată are doar descrierea în puncte deoarece nu s-a stabilit o descriere fezabilă pentru forma parametrică.

2.2.5 Grupa prelucrărilor 4005

Geometrie:

- se generează patru plane, care se taie de-a lungul a cinci linii în interiorul forme brute; cele două puncte de tăiere comune sunt în forme brute.
- prelucrarea se definește prin șase puncte.
- punctele P1, P4, P5 și P6 sunt punctele de tăiere ale liniilor de tăiere cu fețele forme brute.
- punctele P2 și P3 sunt punctele de tăiere între fețele prelucrării și sunt în interiorul forme brute.
- planul E1 formează fața de bază limitată pe trei părți; este definit de punctele P1, P2, P3 și este mărginită de muchiile K1, K2 și K3.
- planul E2 este deasemenea limitată pe trei părți și este definită de punctele P3, P2, P5.
- planele E3 și E4 sunt limitate în două părți; planul E3 este definit de punctele P5, P2, P1. Planul E4 este definit de punctele P4, P3, P6.
- muchia K1 se construiește din punctele P2 și P3 și se găsește complet în interiorul forme brute; celelalte muchii se definesc astfel:

K2:	P1-P2	K3:	P4-P3	
	K4:	P2-P5	K5:	P3-P6
- muchiile K5 și K6 nu sunt muchii de limită ale feței de bază.

Descrieri tehnice:

- scula poate să intre pe două fețe ale forme brute, dar trebuie să iasă pe aceeași față pe care a intrat. Forma brută nu poate fi străpunsă
- se generează două muchii critice K4 și K5, care pot fi rotunjite
- muchiile ascuțite K1, K2 și K3 nu ridică probleme.

Parametrii suplimentari:

- valoare pentru suprafața corectă a suprafețelor sculei date
- valoare pentru calitatea muchiilor sculei date

2.2.5.1 4005, 1 - Chertare simplă de-a lungul corpului limitată

- toate fețele prelucrării sunt perpendiculare între ele
- fețele prelucrării sunt perpendiculare sau paralele cu fețele laterale ale formei brute (figura 2.20)

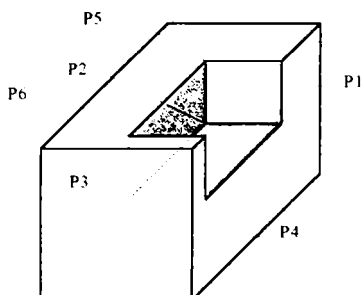


Figura 2.20: Chertare simplă de-a lungul corpului limitată

Forma în puncte:

4005,1,,P1x,P1y,P1z,....,P6x,P6y,P6z

Deocamdată are doar descrierea în puncte deoarece nu s-a stabilit o descriere fezabilă pentru forma parametrică.

2.2.5.2 4005, 2 – Gaură de cep deschisă

- trei din fețele prelucrării mărginesc fața de bază.
- fețele laterale ale prelucrării sunt perpendiculare pe fața de bază și o taie de-a lungul muchiilor (P1-P2) și (P3-P4)
- fețele laterale ale prelucrării sunt paralele între ele.
- fața de capăt taie fața de bază de-a lungul muchiei (P2-P3).
- fața de capăt este perpendiculară pe fața de bază și perpendiculară pe ambele fețe laterale ale prelucrării (figura 2.21)

Prelucrarea se efectuează printr-o diferență deci se pot parcurge următorii pași:

- după diferență rezultă două fețe de prelucrare ca posibile pentru fața de bază (sau fața de capăt) pe muchia de tăiere comună ambele sunt la fel de late. Cea care în cealaltă direcție este mai lungă este fața de bază.
- fața de referință este cea care este tăiată de normala feței de bază (dusă prin mijlocul feței). (sunt posibile și A și B)

- P1 este un punct a feței de bază, dar nu este pe muchia de tăiere dintre fața de capăt și fața de bază și din cele două puncte se alege unul astfel încât (P1, P2, P3) să fie în parea dreaptă.
- deoarece fața de bază nu se poate determina unic din punct de vedere geometric, trebuie determinată mai târziu din poziția punctelor P1, P2, P3, P4. Deci fața de bază se poate determina doar după o tăiere.

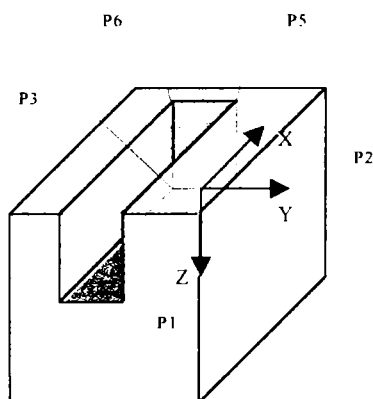


Figura 2.21: Gaură de cep deschisă

Forma în puncte:

4005, 2, A~B, P1x, P1y, P1z, . . . , P6x, P6y, P6z

A păstrarea feței de bază

= 0 fața de bază poate fi distrusă

= 1 (implicit) fața de bază trebuie păstrată

B păstrarea muchiilor (se referă la muchiile perpendiculare pe fața de bază)

= -R valoare negativă, muchiile se vor rotunji cu maximum această rază

= 0 (implicit) muchiile se lasă ascuțite

= +R valoare pozitivă se păstrează și fața de capăt, muchiile se vor rotunji cu această rază

Forma parametrică:

4005, 2, A~B, BS, BE, X, Y, Z, B, L, GW, NW, KW

	Parametru	Descriere
		<p>EKS: axa X indică în direcția P1-P2. Axa Z este inversă axei Z a feței de bază. Axa Y este în fața de bază. Originea este în punctul de poziționare.</p> <p>Punctul de poziționare: punctul de mijloc între P2 și P3.</p> <p>Fața de bază: pe această față de prelucrare sunt celelalte trei perpendiculare.</p>
BS	Fața de ref.	A, B, C, E, F, D: este prima față din șirul de fețe care este tăiată de normala feței de bază în punctul de poziționare.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Y	Deplasare pe y	Y_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
B	Lățime	Distanța dintre fețele laterale ale prelucrării.
GW	Unghi în planul de bază	Este unghiul în planul de bază a lui EKS față de BKS.
NW	Unghi de înclinare	Este unghiul de înclinare a lui EKS față de BKS.
KW	Unghi de răsturnare	Este unghiul de răsturnare a lui EKS față de BKS.

2.2.6 Grupa prelucrărilor 4006

Geometrie:

- se generează cinci plane care se taie de-a lungul a opt linii de tăiere în interiorul formei brute; cele patru puncte comune ale celor cinci plane sunt în interiorul formei brute.
- prelucrarea este definită prin opt puncte.
- punctele P5, P6, P7 și P8 sunt punctele de tăiere dintre liniile de tăiere și fețele formei brute.
- punctele P1, P2, P3 și P4 sunt punctele de tăiere ale planelor prelucrării; sunt în interiorul formei brute.
- planul E1 formează fața de bază limitată pe patru părți; este definit de punctele P1, P2, P3, P4 și limitat de muchiile K1, K2, K3 și K4.
- planele E2, E3, E4 și E5 sunt limitate în trei părți:
E1:P6,P2,P1
E2:P7,P3,P2
E3:P8,P4,P3
E4:P5, P1, P4
- muchiile K1, K2, K3 și K4 sunt complet în interiorul formei brute
K1:P1-P2
K2:P2-P3
K3:P3-P4
K4:P4- P1
- muchiile rămase taie fețele formei brute:
K5:P1,P5
K6:P2,P6
K7:P3,P7
K8:P4,P8
- muchiile K5, K6, K7 și K8 nu sunt muchii de limită ale feței de bază.

Descrieri tehnice:

- scula poate să intre numai pe o față a formei brute. Ea nu poate fi străpunsă.
- se generează cel mult patru muchii critice K5, K6, K7 și K8, care pot fi rotunjite.

- pentru scula "freza cu lanț" trebuie specificat dacă fața de bază poate fi distrusă complet.
- pentru prelucrarea bilaterală limitată în lungime trebuie specificat dacă fața de capăt poate fi distrusă.
- muchiile ascuțite K1, K2 și K3 nu ridică probleme.

Parametrii suplimentari:

- valoare pentru calitatea suprafețelor sculei date
- valoare pentru calitatea muchiilor sculei date

2.2.6.1 4006, 1 – Gaură de cep

- două fețe ale prelucrării sunt paralele: E2 (P7, P3, P2), E4 (P5, P1, P4),
- gaura se construiește cu ajutorul unui cep, E2 și E3 formează fețele laterale ale găurii de cep
- fața de baza E1 este paralelă cu o față laterală a formei brute.
- cele patru fețe laterale sunt perpendiculare pe fața de bază și deci perpendiculare pe fața laterală a formei brute cu care este fața de bază paralelă.
- două din fețele laterale ale prelucrării sunt paralele cu fețele laterale ale formei brute.
- fiecare față laterală este perpendiculară pe cele două fețe laterale care o mărginesc și paralelă cu fața laterală de vizavi (figura 2.22)

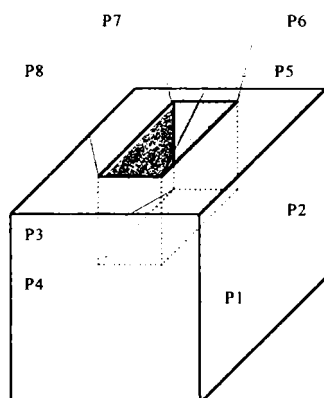


Figura 2.22: Gaură de cep

Forma în puncte:

4006,1,,P1x,P1y,P1z,....,P8x,P8y,P8z

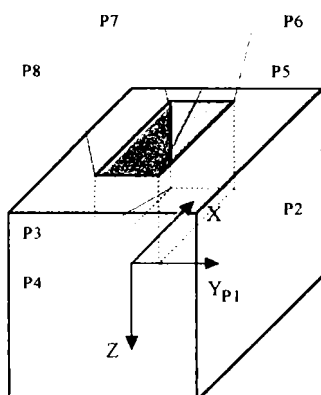
- nu are parametri suplimentari

Forma parametrică:**4006,1,,BS,BE,X,Y,Z,L,B**

	Parametru	Descriere
		Punct de poziționare: P1, P2, P3 sau P4. Din cele două puncte cu Y_{BS} cel mai mic se alege cel cu X_{BS} mai mic.
BS	Fața de ref.	Fața de bază este paralelă cu fața de referință și deschiderea găurii indică către ea. Punctele P5-P8 sunt pe fața de referință.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Y	Deplasare pe y	Este valoarea de mijloc a lui Y_{BS} dintre punctele P1-P4, care au același X_{BS} . Deci este mijlocul găurii de cep.
Z	Adâncime	Z_{BS} al punctului de poziționare
L	Lungime	Distanța pe X de la punctul de poziționare la unul din punctele P1-P4, care nu are același X_{BS} .
B	Lățime	Distanța pe Y de la punctul de poziționare la unul din punctele P1-P4 care nu are același Y_{BS} .

2.2.6.2 4006, 2 – Canal limitat

- fiecare capăt este limitat de fețe ale prelucrării care sunt perpendiculare pe fața de bază și pe fețele laterale ale prelucrării.
- unghiurile dintre fețele prelucrării sunt de 90° .
- (P1-P2) și (P3-P4) sunt paralele între ele și paralele cu axa forme brute
- P1 și P4 sunt mai aproape de capătul de început al forme brute decât P2 și P3
- fața de bază este paralelă cu axa forme brute (figura 2.23)

**Figura 2.23:** Canal limitat

Forma în puncte:

4006, 2, A~B, P1x, P1y, P1z, . . . , P8x, P8y, P8z

A păstrarea feței de bază

= 0 fața de bază poate fi distrusă

= 1 (implicit) fața de bază trebuie păstrată

B păstrarea muchiilor (se referă la muchiile perpendiculare pe fața de bază)

= -R valoare negativă, muchiile se vor rotunji cu maximum aceasta rază

= 0 (implicit) muchiile se lasă ascuțite

= +R valoare pozitivă, muchiile se vor rotunji cu această rază

Forma parametrică:

4006, 2, A~B, BS, BE, X, Y, Z, B, L, KW

	Parametru	Descriere
		EKS: axa X are direcția și sensul lui P1-P2. Axa Z este inversă normalei feței de referință. Originea este în punctul de poziționare. Punctul de poziționare: mijlocul muchiei P1-P4. Fața de bază: pe această față fețele laterale ale prelucrării sunt perpendiculare. Fața laterală: este perpendiculară pe fața de bază și paralelă cu axa formei brute.
BS	Fața de ref.	C, E, F, D: prima față din șir care este tăiată de normala feței de bază dusă în punctul de poziționare.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Y	Deplasare pe y	Y_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
B	Lățime	Distanța dintre fețele laterale.
L	Lungime	Distanța dintre fețele de capăt.

KW	Unghi de răsturnare	Este unghiul de răsturnare EKS față de BKS.
----	---------------------	---

2.2.6.3 4006, 3 – Gaură de cep oarecare

- fața de bază a prelucrării este mărginită de patru fețe, două laterale și două de capăt.
- fețele laterale sunt perpendiculare pe fața de bază și o taie de-a lungul muchiilor (P1-P2) și (P3-P4)
- fețele laterale sunt paralele între ele.
- fețele de capăt taie fața de bază de-a lungul muchiilor (P1-P4) și (P2-P3).
- fețele de capăt sunt perpendiculare pe fața de bază și pe fețele laterale (figura 2.24)

Datorită faptului că prelucrarea se face folosind operația de scădere booleană se poate stabili o anumită ordine pentru punctele prelucrării:

- fața de referință este fața formei brute care este intersectată de normala feței de bază dusă în mijlocul feței. (poate fi și A sau B)
- P1 este punctul din fața de bază cu cel mai mic X_{BS} .
- P2 este punctul legat de P1 pentru care proiecția liniei P1-P2 în fața de bază formează unghiul cel mai mic cu axa X a sistemului feței de referință.
- celelalte puncte rezultă din geometrie.

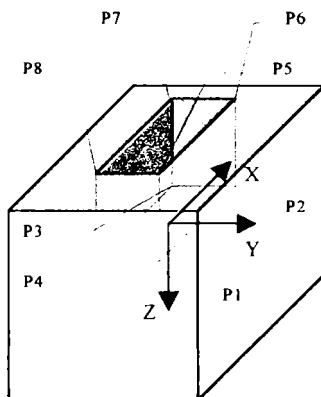


Figura 2.24: Gaură de cep oarecare

Forma în puncte:

4006, 3, A~B, P1x, P1y, P1z, . . . , P8x, P8y, P8z

- A păstrarea feței de bază
 = 0 fața de bază poate fi distrusă
 = 1 (implicit) fața de bază trebuie păstrată
- B păstrarea muchiilor (se referă la muchiile perpendiculare pe fața de bază)
 = -R valoare negativă, muchiile se vor rotunji cu maximum aceasta rază
 = 0 (implicit) muchiile se lasă ascuțite
 = +R valoare pozitivă se păstrează și fața de capăt, muchiile se vor rotunji cu această rază

Forma parametrică:

4006, 3, A~B, BS, BE, X, Y, Z, B, L, GW, NW, KW

	Parametru	Descriere
		EKS: axa X are direcția și sensul lui P1-P2. Axa Z este inversă normalei feței de referință. Axa Y este în planul feței de bază. Originea este în punctul de poziționare. Punctul de poziționare: mijlocul muchiei P1-P3. Fața de bază: pe această față sunt perpendiculare celelalte patru.
BS	Fața de ref.	A, B, C, E, F, D: Prima față din șir care este tăiată de normala feței de bază dusă prin punctul de poziționare.
BE	Capăt de ref.	Întodeauna 'A'.
X	Deplasare pe x	X_{BS} al punctului de poziționare
Y	Deplasare pe y	Y_{BS} al punctului de poziționare
Z	Adâncime	Z_{BS} al punctului de poziționare
B	Breite	Distanța dintre fețele laterale.
L	Lungimea	Distanța dintre fețele de capăt.
GW	Unghi în planul de bază	Este unghiul în planul de bază EKS față de BKS.
NW	Unghi de înclinare	Este unghiul de înclinare EKS față de BKS.
KW	Unghi de răsturnare	Este unghiul de răsturnare EKS față de BKS.

2.2.6.4 4006, 4- Gaură de cep coadă de rândunică de capăt

Din punct de vedere al codării, al sistemului de coordonate al prelucrării este identică cu 4007, 5.

Forma în puncte:

4006, 4, A, P1_x, P1_y, P1_z, ..., P8_x, P8_y, P8_z

A parametru pentru mașină

Forma parametrică:

4006, 4, A, BS, BE, X, Y, Z, B, H, FW, OW, GW, NW, KW

	Parametru	Descriere
		Punctul de poziționare: mijlocul muchiei P5-P8.
BS	Fața de ref.	Bisectoarea unghiului format de muchiile P5-P6 și P7-P8 taie această față în direcția muchiei P7-P8.
BE	Capăt de ref.	Capătul spre care este orientată normala feței de bază.
X	Deplasare pe x	X _{BS} al punctului de poziționare
Y	Deplasare pe y	Y _{BS} al punctului de poziționare
Z	Deplasare pe z	Distanța de la punctul de poziționare la fața de referință măsurată în fața de referință.
B	Lățime	Lățimea cepului
H	Înălțime	Distanța dintre punctele P1 și P5 măsurată perpendicular pe fața de referință.
FW	Unghi de flank	Fetele (P5, P6, P2), (P7, P8, P4), (P8, P5, P1) sunt înclinate cu același unghi față de fața de bază. Acest unghi poate lua valori între 0° și 90°. El se măsoară întodeauna în interiorul formei brute.
OW	Unghi de con	Este unghiul format de muchiile (P5-P6) și (P8-P7). Acest unghi poate lua valori între 0° și 180°.
GW	Unghi în planul de bază	Este unghiul în planul de bază EKS față de BKS.
NW	Unghi de înclinare	Este unghiul de înclinare EKS față de BKS.

KW	Unghi de răsturnare	Este unghiul de răsturnare EKS față de BKS.
----	---------------------	---

2.2.6.5 4006, 5 – Gaură de cep coadă de rândunică laterală

Spre deosebire de prelucrarea anterioară, fața de bază este întotdeauna paralelă cu fața de referință, deci unghiul de înclinare și cel de răsturnare sunt 0°.

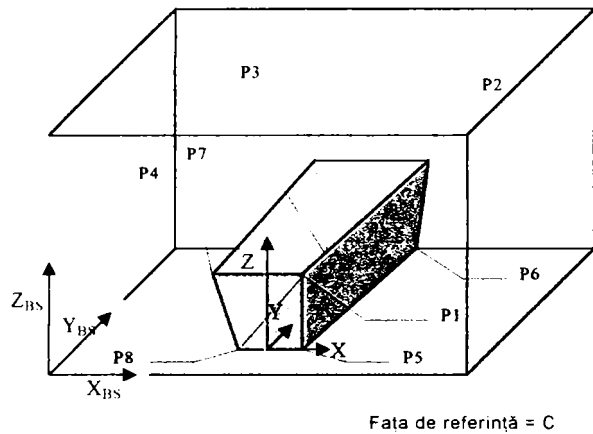


Figura 2.25: Gaură de cep coadă de rândunică laterală

Forma în puncte:

4006, 5, A, P1x, P1y, P1z, ..., P8x, P8y, P8z

A parametru pentru mașină

Forma parametrică:

4006, 5, A, BS, BE, X, Y, B, H, FW, OW, GW

	Parametru	Descriere
		Punct de poziționare: este originea lui EKS, mijlocul muchiei (P5-P8).
BS	Fața de ref.	Punctele P5-P8 sunt în această față.
BE	Capăt de ref.	Întotdeauna A
X	Deplasare pe x	X _{BS} al punctului de poziționare

Y	Deplasare pe y	Y_{BS} al punctului de poziționare
B	Lățime	Lățimea prelucrării
H	Înălțime	Distanța dintre punctele P1 și P5 măsurată perpendicular pe fața de bază.
FW	Unghi de flanc	Fețele (P5, P6, P2), (P7, P8, P4), (P8, P5, P1) sunt înclinate cu același unghi față de fața de bază. Acest unghi poate lua valori între 0° și 90° . El se măsoară întotdeauna în interiorul formei brute.
OW	Unghi de con	Este unghiul format de muchiile P5, P6 și P8, P7. Acest unghi poate lua valori între 0° și 180° .
GW	Unghi în planul de bază	Este unghiul în planul de bază EKS față de BKS.

2.2.7 Grupa prelucrărilor 4007

- se definește cu ajutorul a opt puncte
- P1..P4 sunt într-un plan și formează fața de capăt al cepului
- P5..P8 sunt într-un plan și sunt punctele de intersecție dintre cep și fața de bază
- (P1-P2) definește direcția pe lungimea cepului
- nici un punct al cepului nu este în exteriorul secțiunii forme brute (figura 2.26)

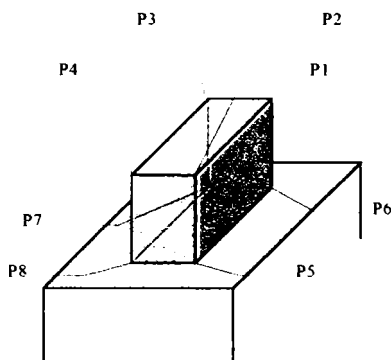
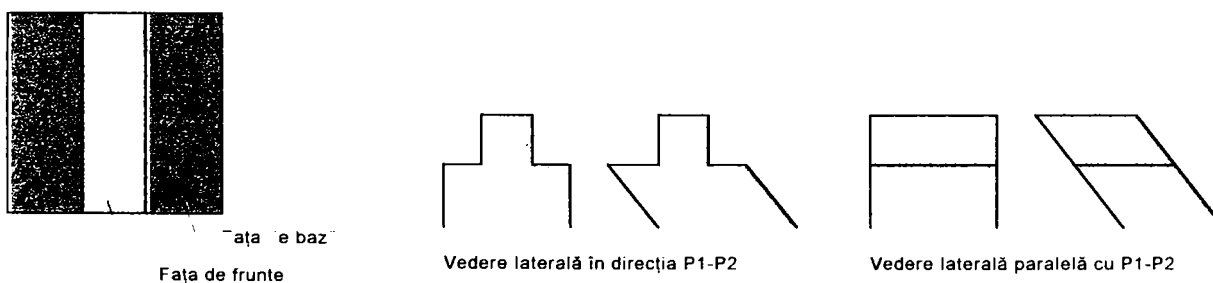


Figura 2.26: Prelucrare tip 4007

2.2.7.1 4007, 1 – Cep drept

- fețele opuse sunt paralele
- fețele de pe lungimea prelucrării sunt perpendiculare pe fața de capăt a cepului
- fețele de pe lungimea prelucrării sunt perpendiculare pe fața de bază
- fața de bază a cepului este perpendiculară pe cel puțin două fețe laterale ale forme brute.
- punctele P1 și P2 sunt de-a lungul cepului deci sunt pe fețe laterale opuse ale forme brute
- cepul se termină pe două fețe laterale ale forme brute, deci P1..P8 sunt în două fețe laterale opuse ale forme brute
- muchia (P1-P4) este perpendiculară pe fața laterală a forme brute (figura 2.27)



Vedere dinspre fața de bază

Figura 2.27: Vederi ale cepului drept din diferite plane

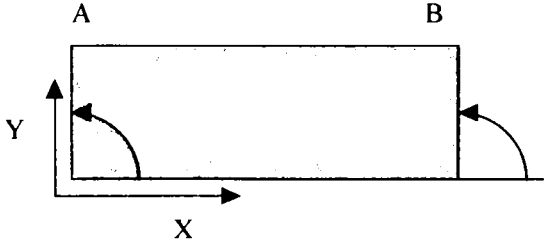
Forma în puncte:

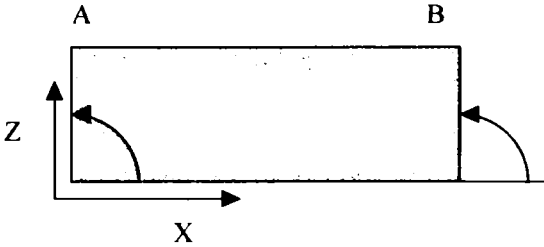
4007,1,,P1x,P1y,P1z,....,P8x,P8y,P8z

- nu are parametri suplimentari

Forma parametrică:

4007,1,,BS,BE,X,Y,Z,B,GW,NW

	Parametru	Descriere
BS	Fața de ref.	C, E, F, D: prima față pe care este P1 sau P2, dar nu amândouă.
BE	Capăt de ref.	Capătul indicat de planul determinat de punctele (P5, P6, P7)
X	Deplasare pe x	Se măsoară de la X=0 până la punctul de intersecție a planului (P5, P6, P7) cu axa X a feței de referință.
Y	Deplasare pe y	Y_{BS} al mijlocului muchiei (P5-P8).
Z	Înălțime	Distanța dintre planele (P1, P2, P3) și (P5, P6, P7).
B	Lățime	Distanța dintre muchiile (P5-P6) și (P7-P8).
GW	Unghi în planul de bază	<p>Planul (P5, P6, P7) se taie de-a lungul unei drepte cu fața de referință. Unghiul dintre această dreaptă și axa x a feței de referință este unghiul în planul de bază:</p> $0^\circ \leq GW < 180^\circ.$ <p>Se măsoară întotdeauna de la axa x pozitivă în sens trigonometric (rotire în jurul lui Z_{BS}): dacă capătul de referință este la început (A), se măsoară în forma brută, dacă capătul de referință este la sfârșit unghiul se măsoară în exteriorul formei brute.</p>
		
NW	Unghi de înclinare	Este unghiul diedru dintre fața prelucrării (P5, P6, P7) și fața de referință. Unghiul se măsoară întotdeauna de la fața de referință

		<p>spre axa z a feței de referință. Dacă capătul de referință este cel de început (A) se măsoară în forma brută, dacă capătul de referință este cel de sfârșit se măsoară în exteriorul forme brute.</p> 
--	--	---

2.2.7.2 4007, 2 – Cep cu deplasamente

- fețele de pe lungimea prelucrării sunt perpendiculare pe fața de frunte a prelucrării
- fețele de pe lungimea prelucrării sunt perpendiculare pe fața de bază
- fața de bază a prelucrării este perpendiculară pe cel puțin două fețe laterale ale forme brute.
- (P1, P2) sunt în lungimea prelucrării
- (P1, P4) este perpendicular pe fețele laterale ale forme brute
- fețele de capăt ale prelucrării sunt ori perpendiculare pe fața de bază ori paralele cu o față laterală a forme brute (figura 2.28)

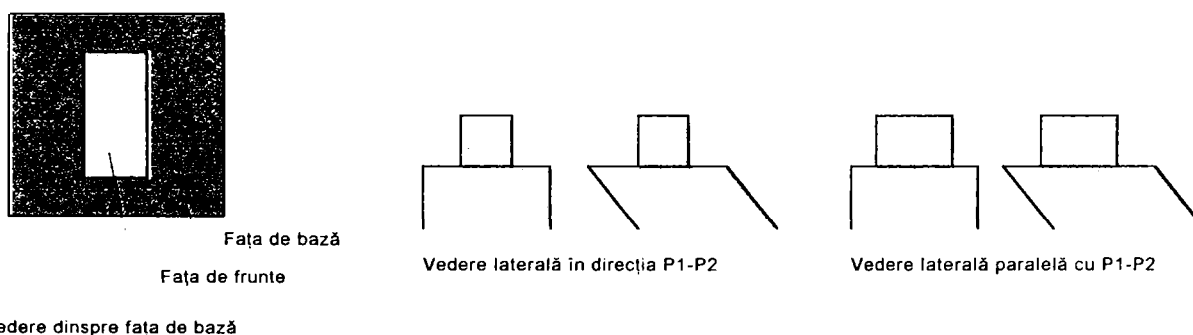


Figura 2.28: Vederi ale cepului cu deplasamente din diferite plane

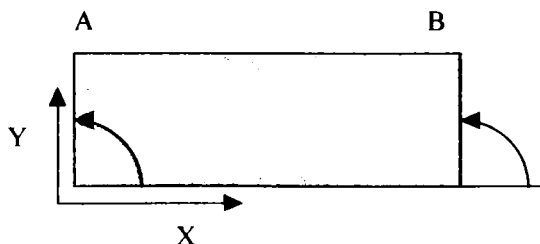
Forma în puncte:

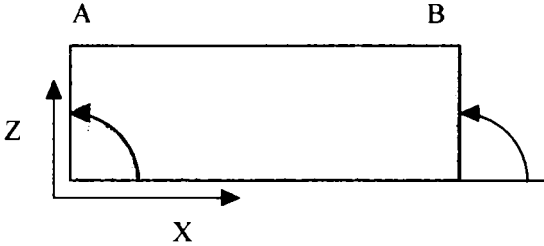
4007,2,,P1x,P1y,P1z,....,P8x,P8y,P8z

- nu are parametri suplimentari

Forma parametrică:**4007,2,,BS,BE,X,Y,Z,B,A1,A2,GW,NW**

	Parametru	Descriere
BS	Fața de ref.	C, E, F, D: prima față pe care se găsește punctul de intersecție a liniei P1-P2 cu fețele laterale ale formei brute dar nu sunt ambele puncte de intersecție.
BE	Capăt de ref.	Capătul indicat de normala planului prelucrării dat de punctele (P5, P6, P7).
X	Deplasare pe x	Se măsoară de la X=0 până la punctul de intersecție a planului prelucrării dat de punctele (P5, P6, P7) cu axa x a feței de referință
Y	Deplas în secțiune	Y_{BS} a mijlocului muchiei (P5-P8).
Z	Înălțime	Este distanța dintre planele (P1, P2, P3) și (P5, P6, P7).
B	Lățime	Este distanța dintre muchiile (P5-P6) și (P7-P8).
A1	Deplasament 1	Este distanța de la punctul P5 la fața laterală măsurată în direcția muchiei (P6-P5): o direcție din P6 spre punctul P5 taie o față laterală a formei brute într-un punct. Distanța spațială de la acest punct la P5 este deplasamentul 1.
A2	Deplasament 2	Este distanța de la punctul P6 la fața laterală măsurată în direcția muchiei (P6-P5).
GW	Unghi în planul de bază	<p>Planul de prelucrare (P5, P6, P7) și fața de referință se taie după o dreaptă. Unghiul dintre această dreaptă și axa x a feței de referință este unghiul în planul de bază: $0^\circ \leq GW < 180^\circ$.</p> <p>Se măsoară întotdeauna de la axa x pozitivă în sens trigonometric (rotire în jurul lui Z_{BS}): dacă capătul de referință este la început (A), unghiul se măsoară în interiorul formei brute, dacă capătul de referință este la sfârșit unghiul se măsoară în exteriorul formei brute.</p>



<p>NW</p>	<p>Unghi de înclinare</p>	<p>Este unghiul diedru dintre planul prelucrării dat de punctele (P5, P6, P7) și fața de referință. Unghiul se măsoară întotdeauna de la fața de referință la axa z a feței de referință. Dacă capătul de referință este capătul de început (A) unghiul se măsoară în interiorul forme brute, dacă capătul de referință este capătul de sfârșit (B) unghiul se măsoară în exteriorul forme brute.</p> 
------------------	---------------------------	--

2.2.7.3 4007, 3 – Cep rotit

- fețele laterale și de capăt ale cepului sunt ori perpendiculare ori paralele cu o față laterală a forme brute.
- fețele laterale și de capăt ale cepului sunt perpendiculare pe fața de bază fețele care se ating sunt perpendiculare și cele care nu se ating sunt paralele (figura 2.29)

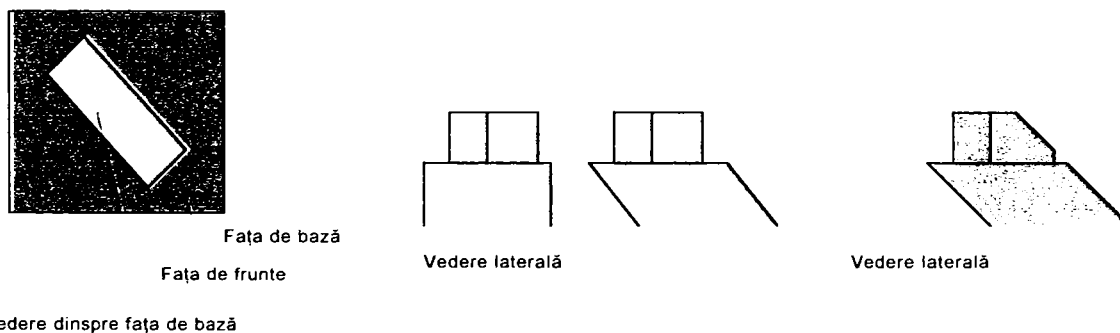


Figura 2.29: Vederi ale cepului rotit din diferite plane

2.2.7.4 4007, 4 – Cep clasic

- fețele de-a lungul prelucrării sunt perpendiculare pe fața de frunte a prelucrării
- fețele de-a lungul prelucrării sunt perpendiculare pe fața de bază
- prelucrarea se termină pe o față laterală a forme brute
- celelalte fețe de capăt ale prelucrării sunt perpendiculare pe fața de bază
- fața de bază este perpendiculară pe cele două fețe laterale ale forme brute pe care și P1-P4 este perpendicular.
- punctele P1 și P2 sunt în lungimea prelucrării

- muchia (P1, P4) este perpendiculară pe două fețe laterale ale formei brute (figura 2.30)

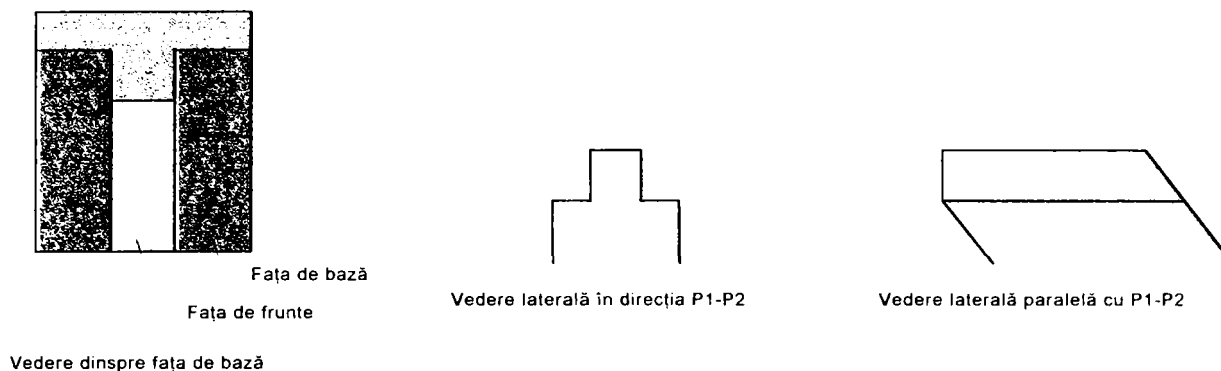


Figura 2.30: Vederi ale cepului clasic din diferite plane

Forma în puncte:

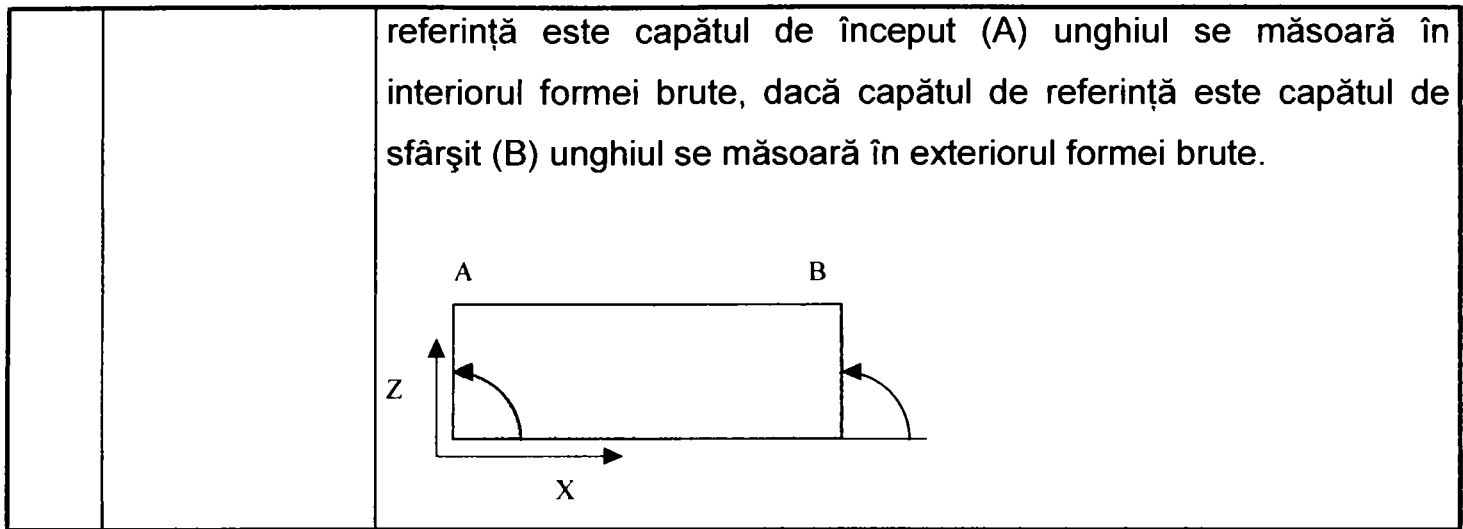
4007,4,,P1x,P1y,P1z,,...,P8x,P8y,P8z

- nu are parametri suplimentari

Forma parametrică:

4007,4,,BS,BE,X,Y,Z,B,NW

	Parametru	Descriere
BS	Fața de ref.	C, E, F, D: prima față pe care este P1 sau P2, dar nu amândouă.
BE	Capăt de ref.	Capătul indicat de planul prelucrării determinat de punctele (P5, P6, P7)
X	Deplasare pe x	Se măsoara de la X=0 până la punctul de intersecție a planului prelucrării determinat de punctele (P5, P6, P7) cu axa x a feței de referință
Y	Deplasare pe y	Y_{BS} a mijlocului muchiei (P5-P8).
Z	Înălțime	Este distanța dintre planele (P1, P2, P3) și (P5, P6, P7).
B	Lățime	Este distanța dintre muchiilor (P5-P6) și (P7-P8).
NW	Unghi de înclinare	Este unghiul diedru dintre planul prelucrării determinat de punctele (P5, P6, P7) și fața de referință. Unghiul se măsoară întotdeauna de la fața de referință la axa z a feței de referință. Dacă capătul de



2.2.7.5 4007, 5 – Cep coadă de rândunică

- fața de frunte este fața formată din punctele: P1..P4
- fața de bază este fața formată din punctele: P5..P8
- fața de bază și fața de frunte sunt paralele
- fețele (P5, P6, P2), (P7, P8, P4) și (P8, P5, P1) sunt înclinate cu același unghi față de fața de bază. Acest unghi se numește unghi de flanc și poate avea valori între 0° și 90° .
- muchiile (P5-P6) și (P8-P7) se intersectează într-un punct care este întotdeauna în direcția P6-P5, deci mai aproape de P5 și P8 decât de P6 și P7.
- unghiul format de muchiile P5-P6 și P8-P7 se numește unghi de con și poate avea valori între 0° și 180° .
- P2, P3, P6 și P7 sunt pe o față laterală a forme brute. Fețele (P5, P6, P2) și (P7, P8, P4) ies în exteriorul feței laterale pe care se află punctele P2, P3, P6 și P7. Vizavi prelucrarea este limitată de fața (P8, P5, P1) a prelucrării (figura 2.31)

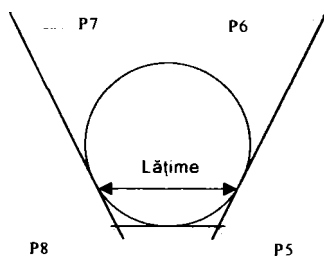


Figura 2.31: Definierea lățimii prelucrării

Fața de bază se construiește astfel:

- valorile care se iau în considerare sunt unghiul de con și lățimea
- se generează două drepte înclinate cu unghiul de con.

- lățimea se măsoară perpendicular pe bisectoarea unghiului format de cele două drepte înclinate.
- prin punctele de intersecție a segmentului de lungime egală cu lățimea, cu cele două drepte înclinate se construiește un cerc tangent la drepte.
- se construiește o dreaptă perpendiculară pe bisectoare și tangentă la cerc. Această dreaptă intersectează dreptele înclinate în punctele P5 și P8 (figura 2.32)

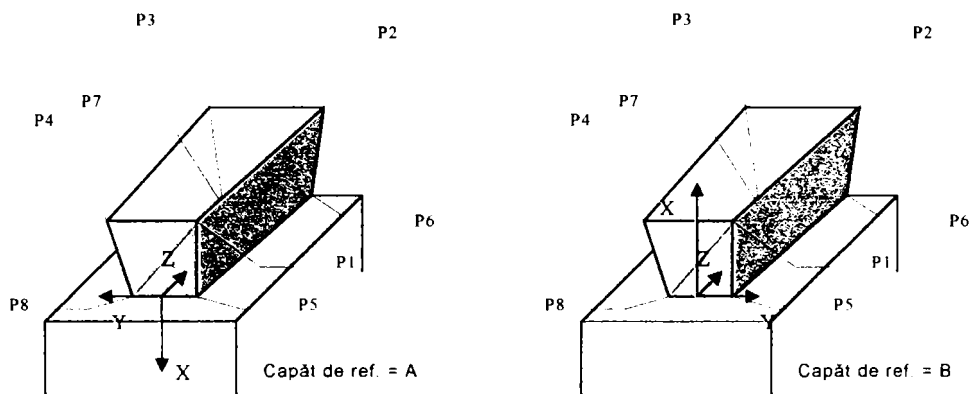


Figura 2.32: Cep coadă de rândunică

Forma în puncte:

4007,5,A,P1x,P1y,P1z,...,P8x,P8y,P8z

A parametru pentru mașină

Forma parametrică:

5000,1,A,BS,BE,X,Y,Z,B,H,FW,OW,GW,NW,KW

Parametru	Descriere
	Punct de poziționare:originea lui EKS. Mijlocul muchiei (P5-P8).
BS	Fața de ref. Bisectoarea unghiului format de muchiile (P5-P6) și (P7-P8) taie această față în direcția P7-P8.
BE	Capăt de ref. Capătul spre care este orientată normala feței de bază.
X	Deplasare pe x X_{BS} al punctului de poziționare
Y	Deplasare pe y Y_{BS} al punctului de poziționare
Z	Deplasament Distanța de la punctul de poziționare la fața de referință măsurată în fața de bază. Unghiul de înclinare trebuie să fie cuprins între valorile -90° și $+90^\circ$.

B	Lățime	Lățimea prelucrării
H	Înălțime	Distanța dintre punctele P1 și P5 măsurată perpendicular pe fața de bază.
FW	Unghi de flanc	Fetele (P5, P6, P2), (P7, P8, P4) și (P8, P5, P1) formează același unghi cu fața de bază. El poate lua valori între 0° și 90°. Se măsoară întotdeauna în afara formei brute.
OW	Unghi de con	Unghiul format de muchiile P5, P6 și P8, P7. El poate lua valori între 0° și 180°.
GW	Unghi în planul de bază	Este unghiul în planul de bază EKS față de BKS.
NW	Unghi de înclinare	Este unghiul de înclinare EKS față de BKS.
KW	Unghi de răsturnare	Este unghiul de răsturnare EKS față de BKS.

2.2.8 Grupa prelucrărilor 4008

- cu ajutorul unui anumit număr de puncte ($P_1..P_{n-1}$) se construiește un poligon
- poligonul este pus pe o față laterală a formei brute
- pentru această față laterală forma brută se pune în stânga și tăierile în dreapta poligonului
- P_n și P_{n-1} reprezintă vectorul cu care trebuie deplasat poligonul față de forma brută
- vectorul de deplasare este întotdeauna perpendicular pe o față laterală a formei brute.

2.2.8.1 4008, 1 – Chertare dublă la capăt

- se specifică cu ajutorul a șase puncte
- P_1 este plasat în vârful chertării deci la unghiul obtuz (figura 2.33)

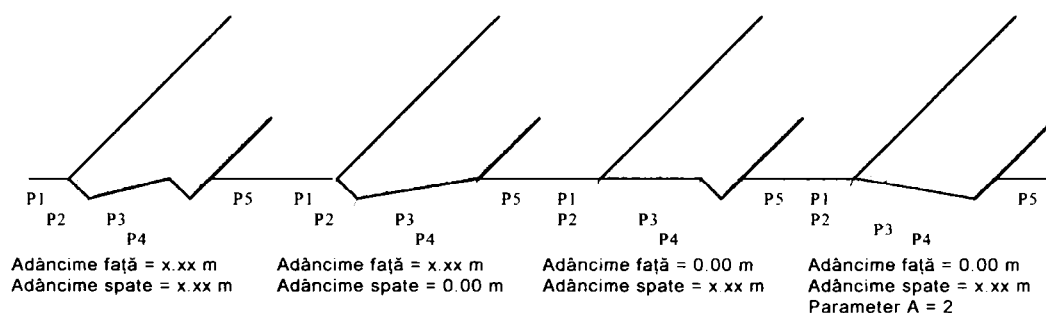


Figura 2.33: Cele patru tipuri de chertare dublă de capăt

Forma în puncte:

4008,1,A,P1x,P1y,P1z,...,P6x,P6y,P6z

- A stabilește tipul chertării
- = 0 chertare dublă normală
 - = 2 chertare dublă la care se ignoră adâncimea față

Forma parametrică:

4008,1,A,BS,BE,X,T1,T2,NW

	Parametru	Descriere
		Fața de bază: este fața formată de punctele (P6, P5, P1).
BS	Fața de ref.	Fața laterală a formei brute pe care sunt punctele P5 și P6
BE	Capăt de ref.	Capătul indicat de normala feței de bază
X	Deplasare pe x	Coordonata X a punctului P1
T1	Adâncime față	Este distanța de la punctul P2 la fața de bază
T2	Adâncime spate	Este distanța de la punctul P4 la fața de bază.
NW	Unghi de înclinare	Este cel mai mic unghi diedru dintre fața de bază și fața de referință.

2.2.8.2 4008, 2 – Chertare dublă pe față laterală

- se specifică cu ajutorul a (5 + 1) puncte
- P1 este plasat în vârful chertării deci la unghiul obtuz

Forma în puncte:

4008,2,A,P1x,P1y,P1z,....,P6x,P6y,P6z

A stabilește tipul chertării

= 0 chertare dublă normală

= 2 chertare dublă la care se ignoră adâncimea față

Forma parametrică:

4008,2,A,BS,BE,X,T1,T2,NW

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a formei brute pe care sunt punctele P5 și P6
BE	Capăt de ref.	Capăt indicat de muchia P5-P1
X	Deplasare pe x	Coordonata X a punctului P1
T1	Adâncime față	Este distanța de la punctul P2 la fața de referință
T2	Adâncime spate	Este distanța de la punctul P4 până la fața de referință

NW	Unghi de înclinare	Este unghiul diedru cel mai mic dintre fața de referință și planul imaginar (P4, P5, P6). Dacă P4 și P5 sunt egale se construiește un punct P2': P2 + vectorul P5-P6. Unghiul cel mai mic dintre fața (P1, P2, P2') și fața de referință este alfa. $NW = (90^\circ - \text{alfa})$.
-----------	--------------------	---

2.2.8.3 4008, 3 – Chertare “tăbăcită”

- se specifică cu ajutorul a cinci puncte
- P1 este în vârful chertării pe o față laterală a formei brute (figura 2.34)

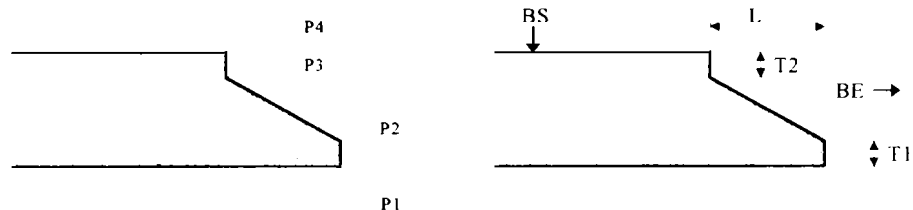


Figura 2.34: Chertare “tăbăcită”

Forma în puncte:

4008,3, ,P1X,P1Y,P1Z,....,P5X,P5Y,P5Z

- nu are parametri suplimentari

Forma parametrică:

4008,3, ,BS,BE,X,L,T1,T2

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a formei brute pe care sunt punctele P4 și P5
BE	Capăt de ref.	Capătul indicat de muchia (P3–P2)
X	(Deplasare pe x)	(X-ul lui P1, se calculează din BE și L)
L	Lungime	Distanța pe x dintre punctele P2 și P3
T1	Adâncime 1	Distanța pe z dintre punctele P1 și P2
T2	Adâncime 2	Distanța pe z dintre punctele P3 și P4

2.2.8.4 4008, 4 – Chertare tip sapă

- se specifică cu ajutorul a șapte puncte
- P1 este în vârful chertării pe o muchie în lungimea forme brute (figura 2.35)

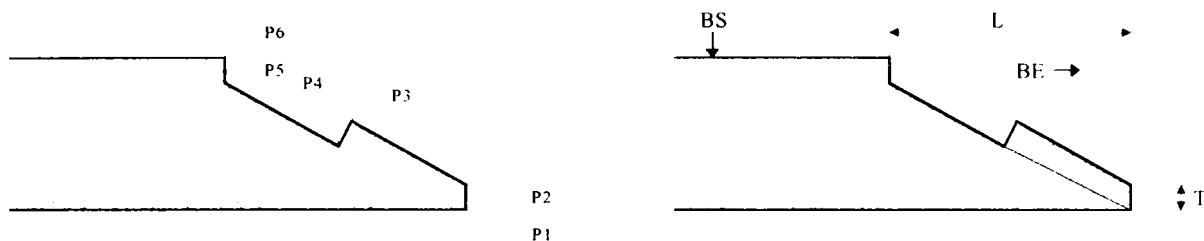


Figura 2.35: Chertare tip sapă

Forma în puncte:

4008,4, ,P1X,P1Y,P1Z,...,P7X,P7Y,P7Z

- nu are parametri suplimentari

Forma parametrică:

4008,4, ,BS,BE,X,L,T

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a forme brute pe care sunt punctele P6 și P7
BE	Capăt de ref.	Capătul indicat de muchia P3–P2
X	(Deplasare pe x)	(X-ul punctului P1, se calculează din BE și L)
L	Lungime	Distanța pe x dintre punctele P2 și P6
T	Adâncime	Distanța pe z dintre punctele P1 și P2

2.2.8.5 4008, 5 – Ornament capăt concav

- se specifică cu ajutorul a nouă puncte
- P1 este în vârful prelucrării pe o muchie în lungimea forme brute
- P4, P5 și P6 subîntind un arc de cerc modelat prin mai multe segmente scurte (figura 2.36)

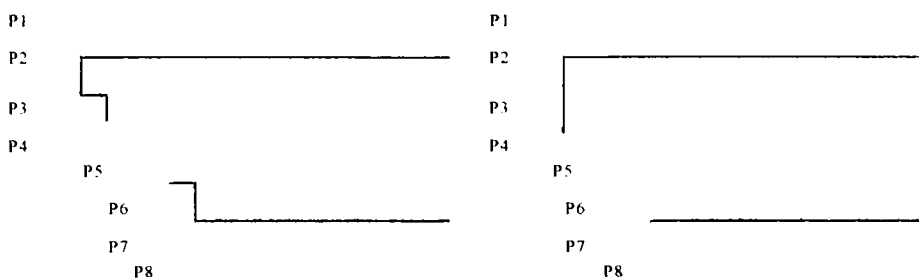


Figura 2.36: Ornament capăt concav

Forma în puncte:

4008,5, ,P1X,P1Y,P1Z,....,P9X,P9Y,P9Z

- nu are parametri suplimentari

Forma parametrică:

4008,5, ,BS,BE,X,L,T,A,R

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a formei brute pe care sunt punctele P8 și P9
BE	Capăt de ref.	Capătul indicat de muchia (P3-P2), sau capătul care conține punctele P2 și P3
X	(Deplasare pe x)	(X-ul punctului P1, se calculează din BS și L)
L	Lungime	Distanța pe x dintre punctele P2 și P3
T	Adâncime	Distanța pe z dintre punctele P1 și P2
A	Deplasament	Distanța pe z dintre punctele P3 și P4 (distanța pe x dintre punctele P6 și P7)
R	Raza	Raza arcului subîntins de punctele P4, P5, P6 (întotdeauna un sfert de cerc).

2.2.8.6 4008, 6 – Ornament capăt convex

- se specifică cu ajutorul a nouă puncte
- P1 este în vârful prelucrării pe o muchie în lungimea forme brute
- P4, P5 și P6 subîntind un arc de cerc modelat prin mai multe segmente scurte (figura 2.37)

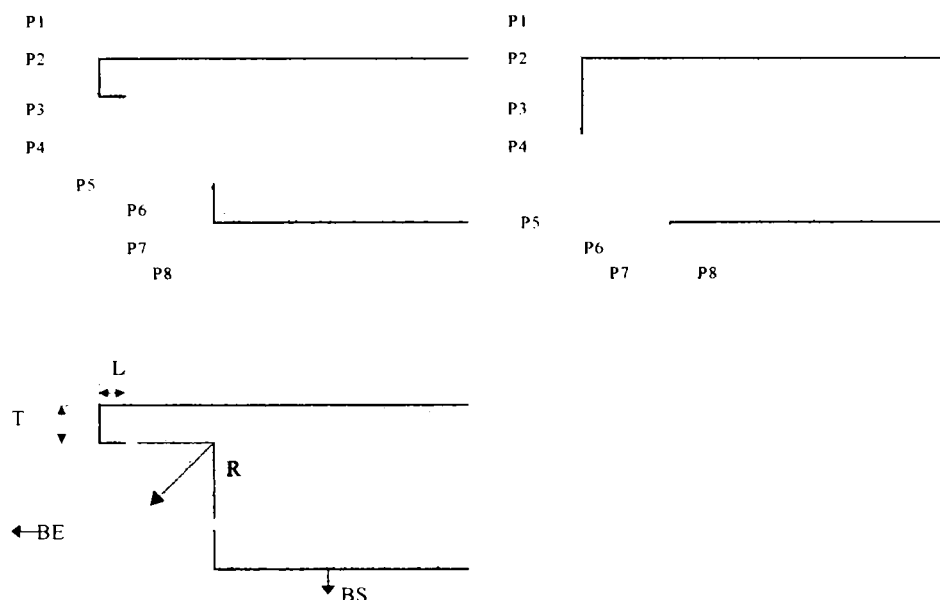


Figura 2.37: Ornament capăt convex

Forma în puncte:

4008,6, ,P1X,P1Y,P1Z,...,P9X,P9Y,P9Z

- nu are parametri suplimentari

Forma parametrică:

4008,6, ,BS,BE,X,L,T,R

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a forme brute pe care sunt punctele P8 și P9
BE	Capăt de ref.	Capătul indicat de muchia (P3-P2), sau capătul care conține punctele P2 și P3
X	(Deplasare pe x)	(X-ul punctului P1, se calculează din BS și L)
L	Lungime	Distanța pe x dintre punctele P2 și P3
T	Adâncime	Distanța pe z dintre punctele P1 și P2

R	Raza	Raza arcului subîntins de punctele P4, P5, P6 (întotdeauna un sfert de cerc)
----------	------	--

2.2.8.7 4008, 7 – Prelucrare tip arc de cerc

- se specifică cu ajutorul a (3 + 1) puncte
- P1 este pe un capăt al arcului de cerc pe o muchie în lungimea forme brute
- P1, P2 și P3 subîntind un arc de cerc. Valoarea maximă pentru unghiul P1P2P3 este 180° (figura 2.38)

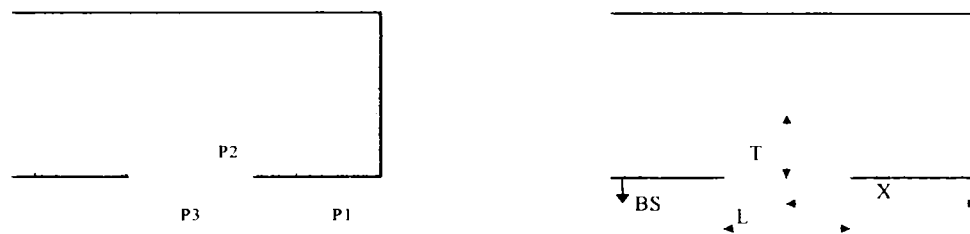


Figura 2.38: Prelucrare tip arc de cerc

Forma în puncte:

4008,7, ,P1X,P1Y,P1Z,....,P4X,P4Y,P4Z

- nu are parametri suplimentari

Forma parametrică:

4008,7, ,BS,BE,X,L,T

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a forme brute pe care sunt punctele P1 și/sau P3
BE	Capăt de ref.	Întotdeauna A
X	Deplasare pe x	X-ul punctului P2
L	Lungime	Distanța pe x dintre punctele P1 și P3
T	Adâncime	Z-ul punctului P2

2.2.8.8 4008, 8 – Profil rotunjit

- se specifică cu ajutorul a (6 + 1) puncte
- P1, P6 și P7 sunt pe o muchie în lungimea forme brute sau pe prelungirea ei.
- P1, P2 (și vectorul P6-P7) definesc un plan.
- P2, P3, P4, P5 descriu o curbă
- P2 și P5 punctele de capăt a curbei
- P3 și P4 sunt punctele de curbură maximă
- curba este de ordinul 3: $ax^3+bx^2+cx+d=0$ (figura 2.39)

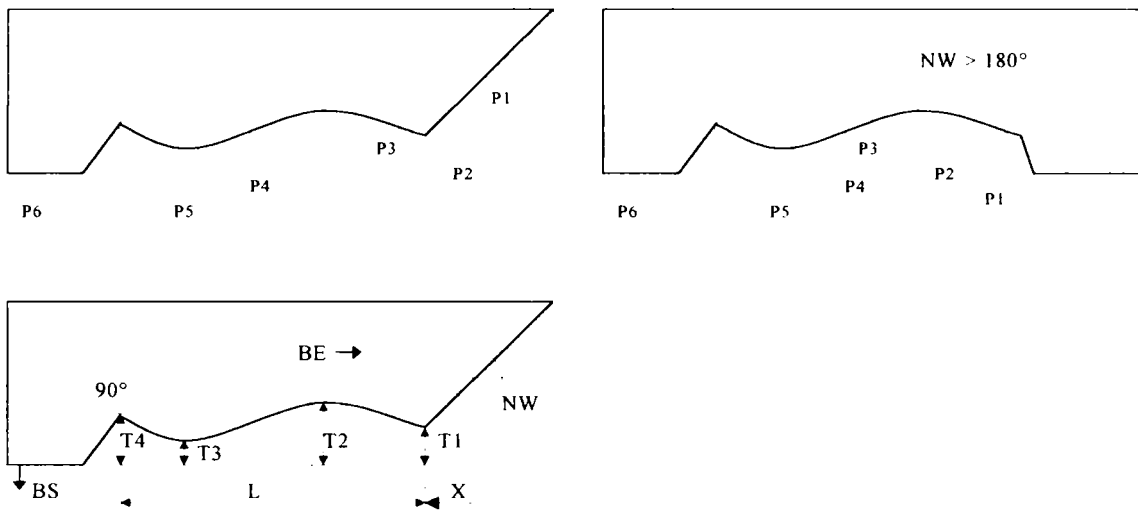


Figura 2.39: Profil rotunjit

Forma în puncte:

4008,8, ,P1X,P1Y,P1Z,....,P7X,P7Y,P7Z

- nu are parametri suplimentari

Forma parametrică:

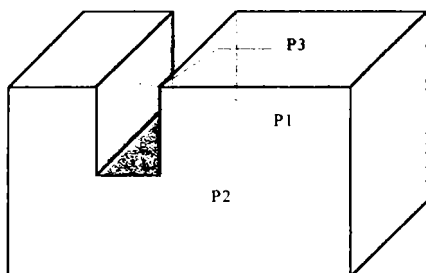
4008,8, ,BS,BE,X,L,T1,T2,T3,T4,NW

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a forme brute pe care sunt punctele P6 și P7
BE	Capăt de ref.	Capătul către care indică muchia (P3-P2)
X	Deplasare pe x	X-ul punctului P2

L	Lungime	Distanța pe x dintre punctele P2 și P5
T1	Adâncime 1	Z-ul punctului P2
T2	Adâncime 2	Z-ul punctului P3
T3	Adâncime 3	Z-ul punctului P4
T4	Adâncime 4	Z-ul punctului P5
NW	Unghi de înclinare	Se definește ca și la tăierea de capăt dusă prin punctele P1, P2. Poate fi mai mare de 180° dar niciodată egal cu 180° .

2.2.9 Grupa prelucrărilor 4009

- lățimea prelucrării este dată de lățimea sculei.
- definit de trei puncte, unde muchia (P1-P2) este muchia de limită
- parametru pentru poziția sculei: în exterior, în interior, la mijloc (=0,1,2). În afară înseamnă că normala feței (P1, P2, P3) este în exteriorul forme brute (figura 2.40)



Poziția sculei:
În exterior = 0

Figura 2.40: Prelucrare tip 4009

2.2.9.1 4009, 1 – Prelucrare efectuată cu fierăstrău cu lanț

- fețele prelucrării nu trebuie să fie paralele cu axa forme brute

Forma în puncte:

4009,1,A~B,P1x,P1y,P1z,...,P3x,P3y,P3z

- A poziția sculei
 = 0 în exterior, în direcția normalei suprafeței
 = 1 în interior, în sens invers normalei suprafeței
 = 0 în mijloc, în direcția normalei sculei

B valoarea maximă a lățimii sculei în [mm]

Forma parametrică:

4009,1,A~B,BS,BE,X,Z,L,NW,GW,SW

	Parametru	Descriere
		<p>Punctul de poziționare: punctul de tăiere a muchiei (P1-P2) cu planul ZoX a feței de referință</p> <p>Muchia de bază:(P1-P2)</p> <p>Fața de bază: este fața care este generată de muchia de bază și normala feței de prelucrare. Normala feței de bază indică spre punctul P3.</p>
BS	Fața de ref.	<p>C, E, F, D: este prima față din șir pentru care sunt valabile condițiile:</p> $0 \leq NW \leq +180^\circ \text{ und } 45^\circ \leq SW \leq 135^\circ.$ <p>Deci normala feței de bază indică spre fața de referință sau este paralelă cu axa formei brute.</p>
BE	Capăt de ref.	Dacă P2 are cel mai mic Y_{BS} , capătul de referință este A, dacă P1 are cel mai mic Y_{BS} , capătul de referință este B
X	Deplasare pe x	Este X_{BS} al punctului de poziționare
Z	Adâncimea	Este Z_{BS} al punctului de poziționare
GW	Unghi în planul de bază	Este unghiul diedru dintre fața de prelucrare și planul ZoX al feței de referință. Se măsoară în interiorul prelucrării, deci în exteriorul formei brute.
NW	Unghi de înclinare	Este unghiul diedru dintre fața de prelucrare și fața de referință. Unghiul se măsoară în interiorul prelucrării deci în afara formei brute. Dacă se rotește în sensul pozitiv al muchiei de bază unghiul este pozitiv.
SW	Unghiul pantei	Este unghiul diedru dintre fața de bază și planul ZoX a feței de referință. Se măsoară în interiorul prelucrării, deci în exteriorul formei brute. Aceasta corespunde unghiului dintre muchia de bază și planul ZoX.

2.2.10 Grupa prelucrărilor 4101

Geometrie:

- prelucrarea se definește prin două puncte.
- P1 este punctul de început și deci punctul de intrare a burghiului.
- P2 este punctul de sfârșit, în condițiile date punctul de ieșire al burghiului.

Descrieri tehnice:

- burghiul intră prin punctul P1 și se oprește în P2 sau iese prin P2.
- calitatea intrării și ieșirii se specifică cu ajutorul unui parametru.

Parametrii suplimentari:

- diametrul găurii
- parametru pentru gaura strapunsă și direcția găurii
- parametru pentru calitatea punctului de intrare și ieșire
- învelitoarea secțiunii. Acest parametru nu are semnificație pentru mașină, dar este important la folosirea introducerii formei parametrice.

A diametrul găurii în mm

B parametru pentru străpungere

= 0 străpunge forma brută (implicit)

= 1 gaură limitată

C parametru pentru sensul găurii

= 0 se poate găuri din oricare punct P1 sau P2 (implicit)

= 1 burghiul trebuie să intre prin punctul P1

D parametru pentru intrarea și ieșirea burghiului

= 0 nici o pretenție (implicit)

= 1 intrarea liberă

= 2 ieșirea liberă

= 3 și intrarea și ieșirea sunt libere

= 4 găurire manuală

- E referința deplasamentului în secțiune. Distanța se măsoară în fețele de referință
 - = 0 din dreapta: se măsoară de la $Y_{BS} = 0$ (implicit)
 - = 1 de la axă: în direcția Y_{BS} pozitiv, invers Y_{BS} negativ
 - = 2 din stânga: invers direcției Y_{BS} pozitiv

2.2.10.1 4101, 0- Gaură de burghiu oarecare

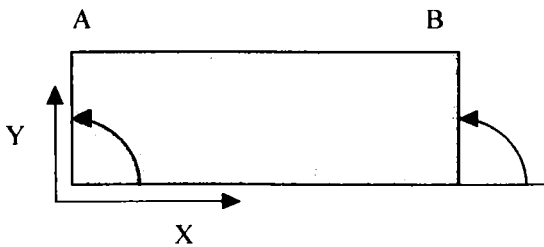
- orientarea găurii este oarecare relativ la sistemul local al formei brute.

Forma în puncte:

4101, 0, A~B~C~D~E, P1X, P1Y, P1Z, P2X, P2Y, P2Z

Forma parametrică:

4101,0,A~B~C~D~E, BS,BE,X,Y,Z,GW,NW

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a formei brute pe care este punctul P1. Poate fi și A sau B
BE	Capăt de ref.	Întotdeauna A
X	Deplasare pe x	X-ul punctului P1
Y	Deplasare pe y	Y-ul punctului P1. Se măsoară de la referință pentru secțiune în fața de referință
Z	Adâncime	Z_{BS} al punctului P2 pentru gaura limitată ($B = 1$)
GW	Unghi în planul de bază	<p>Este unghiul dintre proiecția găurii pe fața de referință și axa x a feței de referință: $0^\circ \leq GW < 180^\circ$.</p> <p>Se măsoară întotdeauna de la axa x pozitivă în sens trigonometric (rotire în jurul lui Z_{BS}).</p> 

NW	Unghi de înclinare	Este unghiul dintre axa găurii și fața de referință. Unghiul se măsoară întotdeauna de la fața de referință la axa z a feței de referință.
-----------	--------------------	--

2.2.10.2 4101, 1- Gaură de burghiu perpendiculară

- gaura este perpendiculară pe o față laterală a formei brute

Forma în puncte:

4101, 1, A~B~C~D~E, P1X, P1Y, P1Z, P2X, P2Y, P2Z

Forma parametrică:

4101,1,A~B~C~D~E, BS,BE,X,Y,Z

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a formei brute pe care este punctul P1. Poate fi și A sau B
BE	Capăt de ref.	Întotdeauna A
X	Deplasare pe x	X-ul punctului P1
Y	Deplasare pe y	Este distanța pe Y a punctului P1 de la referință măsurată în fața de referință
Z	Adâncime	Z_{BS} a punctului P2 pentru gaura limitată ($B = 1$)

2.2.11 Grupa prelucrărilor 4201

Este afectată doar o singură față a formei brute. Geometria corpului nu este afectată.

2.2.11.1 4201, 0 - Marcajul

Geometrie:

- se definește prin trei puncte.
- punctele P1 și P2 definesc direcția liniei de marcaj și reprezintă punctele de capăt.
- punctul P3 definește o anumită parte a marcajului pentru stabilirea poziției sculei; punctul P3 poate să lipsească.

Descrieri tehnice:

Cu ajutorul unui marcator se pot genera diferite marcaje în funcție de posibilitățile mașinii

- se desenează o linie perpendiculară pe muchia barei prin P1
- se desenează o linie prin P1 și P2, chiar dacă linia nu este perpendiculară pe muchia formei brute
- se desenează o linie din P1 până la P2 (punct de sfârșit)
- pe partea lui P3 dat, se mai face un marcaj

Parametrii suplimentari:

- semnul potrivit care trebuie făcut în P3

Forma în puncte:

4201,0,A~B~C~D,P1x,P1y,P1z,....,P3x,P3y,P3z

- A = 0 marcaj normal (contact de bare)
= 1 marcaj de capăt
- B = 0 marcaj simplu
= 1 marcaj dublu
= 2 marcaj dublu numerotat

- C numărul feței din stânga feței de referință la B=2
 D numărul feței din dreapta feței de referință la B=2

Forma parametrică:

4201,0,A~B~C~D,BS,BE,X,L,GW

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a formei brute pe care sunt punctele P1, P2 și P3.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	Se măsoară de la X=0 până la P1 _x sau P2 _x , apoi cel care este cel mai aproape de Y _{BS} =0
L	Lungime	Distanța de la P3 la muchia (P1-P2).
GW	Unghi în planul de bază	Este unghiul dintre muchia (P1-P2) și axa X a feței de referință: 0° ≤ GW < 180°. Pentru marcaje perpendiculare = 90°

2.2.12 Grupa prelucrărilor 4202

2.2.12.1 4202, 0 – Rindeluirea

Geometrie:

- fața de rindeluit se definește prin trei puncte.
- P1 este punctul de început, P2 este punctul de sfârșit al feței rindeluite.
- P3 definește planul.

Descrieri tehnice:

- se specifică doar punctele de început și de sfârșit; fața de rindeluit nu poate fi diferențiată prin regulă; se efectuează operația pe toată lățimea feței de referință (de exemplu poligon precis delimitat).

Forma în puncte:

4202,0,,P1x,P1y,P1z,....,P3x,P3y,P3z

- nu are parametri suplimentari

Forma parametrică:

4202,0,,BS,BE,X,L

	Parametru	Descriere
BS	Fața de ref.	Fața laterală a formei brute pe care sunt punctele P1, P2 și P3.
BE	Capăt de ref.	Întotdeauna 'A'.
X	Deplasare pe x	Se măsoară de la X=0 până la P1 _x sau P2 _x , cel care are cel mai mic X _{BS} .
L	Lungime	Distanța de la P1 la P2.

2.3 Concluzii. Contribuții personale

Prelucrările care modifică geometria formei brute vor fi efectuate cu ajutorul operațiilor booleene descrise în capitolul patru din teză. Ele sunt memorate pe forma brută sub forma unor șiruri ce conțin forma în puncte. Prelucrările sunt realizate în ordinea în care au fost memorate pe forma brută.

Pentru fiecare prelucrare se construiește un corp negativ care se face mai mare decât este nevoie pentru a nu genera erori de precizie în operațiile booleene. La prima prelucrare se ia forma brută ca și corp moale și negativul prelucrării ca și corp tare. Se scade corpul tare din corpul moale și se obține rezultatul primei prelucrări. Cu acest rezultat se continuă procesul de efectuare a următoarei prelucrări. La fiecare prelucrare se ia corpul rezultat din operația anterioară. Astfel, în final corpul va fi modelat conform tuturor prelucrărilor de pe el. Orice modificare a vreunei prelucrări (modificare de parametrii, ștergere prelucrări, inserare prelucrări) duce la o reconstruire a corpului luând procesul de la zero (conform sistemului de modelare CSG).

Această tehnică dă posibilitatea utilizatorului să modeleze elementele de construcție și să le transmită la roboții de prelucrare doar când ele au forma dorită.

Contribuțiile personale în cadrul capitolului actual sunt legate de alegerea claselor potrivite, proiectarea algoritmilor, implementarea și depanarea lor:

- Determinarea și alegerea reprezentării modelelor solidelor celei mai convenabile, din punctul de vedere al prelucrărilor de efectuat, cunoscut fiind faptul că nu orice fel de reprezentare permite orice fel de manipulare a obiectelor
- Analiza datelor necesare algoritmilor și a modalităților de organizare a informației, în scopul asigurării eficienței procesului de prelucrare a lor. Această etapă a procesului de proiectare a fost necesar să fie efectuată de mai multe ori, o dată pentru fiecare mod de abordare a rezolvării problemelor, lucru care a evidențiat dezavantajul utilizării metodei de proiectare structurate comparativ cu cea orientată pe obiect.
- Abordarea sistematică și unitară a căilor de evitare a erorilor datorate preciziei de calcul în cazul reprezentării datelor în virgulă flotantă. Deoarece solidelor rezultate prin operații booleene trebuie să li se poată efectua aceleași prelucrări ca și solidelor inițiale a fost necesar ca în procesul de identificare a diferitelor elemente geometrice să se lucreze cu aceeași precizie în ambele probleme tratate în cadrul tezei.
- Conceperea funcțiilor de conversie din forma parametrică în forma în puncte și invers a prelucrărilor mașină.

- Efectuarea funcțiilor de generare a corpurilor negative, care modelează solidele, pe baza codurilor mașină.
- Implementarea interfețelor prin intermediul cărora se specifică parametrii prelucrărilor individuale
- Efectuarea funcțiilor de modificare a parametrilor prelucrărilor, ștergerea prelucrărilor și inserarea unor prelucrări noi

3 Codarea prelucrărilor folosind combinația dintre două corpuri

În cadrul capitolului anterior s-a prezentat modul în care poate fi prelucrat un singur corp specificând parametrii prelucrărilor. În acest capitol se prezintă modalitatea de efectuare și codare a prelucrărilor automat. De fiecare dată intră în discuție două corpuri, dintre care unul este cel care va fi prelucrat (denumit și corp moale) și cel pe baza căruia e modelat corpul moale (denumit și corp tare). În majoritatea cazurilor sunt necesare și niște informații suplimentare: parametrii, muchii, vârfuri, etc. Acestea vor fi specificate prin intermediul unor cutii de dialog a căror stare se salvează într-un fișier astfel încât la un nou apel al lor valorile să fie preluate din el. Salvarea este utilă în creșterea vitezei de lucru prin faptul că utilizatorul nu trebuie de fiecare dată să introducă valorile doar dacă vrea să le modifice pe cele existente în fișier (Ogârcin Geraldine 32/2001).

Prelucrările care se pot efectua automat între două corpuri sunt (Dietrich's Neues System, no. 4, 2000):

- generarea cepului drept și a găurii de cep aferente
- generarea cepului coadă de rândunică și a găurii de cep aferente
- generarea prelucrării de tip eclisare
- chertare simplă
- chertare dublă
- chertare tip sapă
- chertare "tăbăcită"
- decupare în V
- gaură cep oarecare

3.1 Generarea cepului drept și a găurii de cep aferente

Prima operație care se efectuează este verificarea dacă operația poate să aibă loc analizând poziția celor două corpuri. Corpul 1 se alungește sau scurtează astfel încât să producă prelucrare pe corpul 2. Corpul 2 se alungește doar în cazul în care cele două corpuri sunt în prelungire. Dacă corpul 1 nici prin prelungire nu atinge corpul 2 operația nu poate fi efectuată (Lignarius, 2001).

Corpul 1 se consideră cel tare adică cel care va avea cepul și corpul 2 se consideră cel moale adică el va avea gaura de cep. Pe ambele corpuri se vor pune codurile corespunzătoare. Cepul care se va genera poate fi de patru tipuri: cep care străbate corpul pe toată secțiunea lui (funcția 1), cep care este complet în interiorul secțiunii (funcția 2), cep care poate avea orice poziție față de corpul tare, chiar și rotit (funcția 3) și cep generat dacă corpurile sunt în prelungire (funcția 4).

Există trei modalități de generare a acestei prelucrări:

- cep generat pe o față laterală a corpului moale
- cep generat pe un capăt al corpului moale
- pe baza orientării găurii de cep

3.1.1 Generarea prelucrării pe o față laterală a corpului moale

Pentru a se putea efectua prelucrarea sunt necesare niște informații suplimentare referitoare la parametrii prelucrării care se vor specifica de către utilizator prin intermediul cutiei de dialog din figura 3.1:

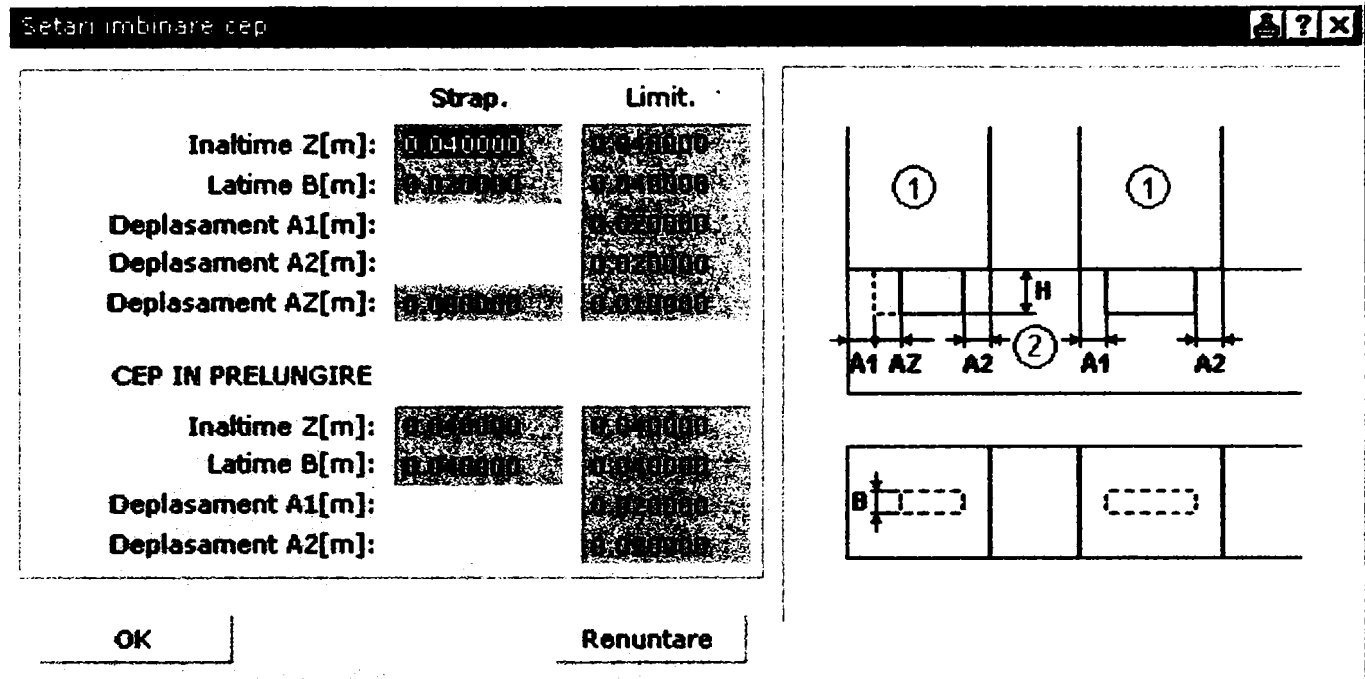


Figura 3.1: Cutia de dialog prin care se specifică valorile implicate ale prelucrării cep de diferite tipuri

Dacă unele din aceste mărimi sunt omise prin necompletarea lor, se deschide o cutie de dialog suplimentară (figura 3.2, 3.3, 3.4, 3.5) forțând utilizatorul să le specifice. Cutiile vor conține mărimile necompletate. În continuare se prezintă cutiile de dialog care se deschid dacă s-au omis toate mărimile.

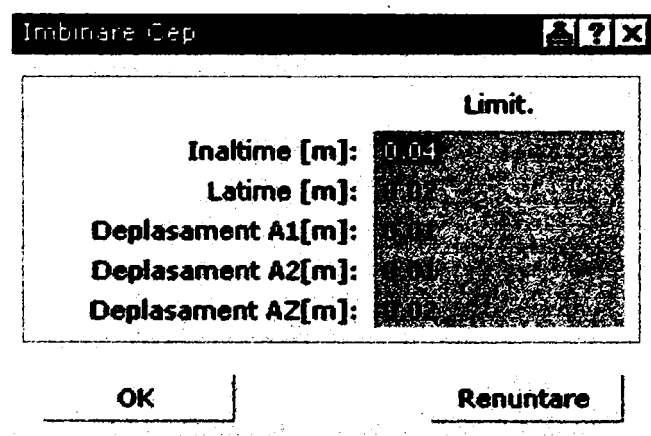


Figura 3.2: Cep limitat când cele două corpuri nu sunt în prelungire

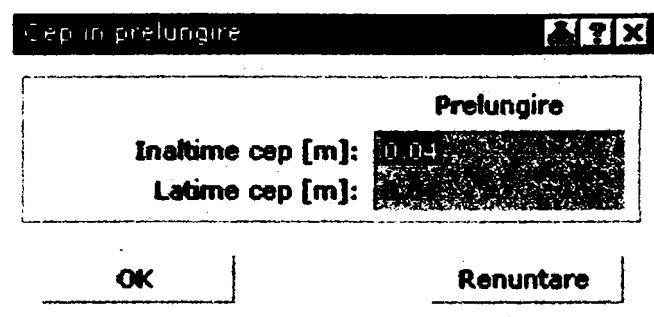


Figura 3.3: Cep limitat când cele două corpuri sunt în prelungire

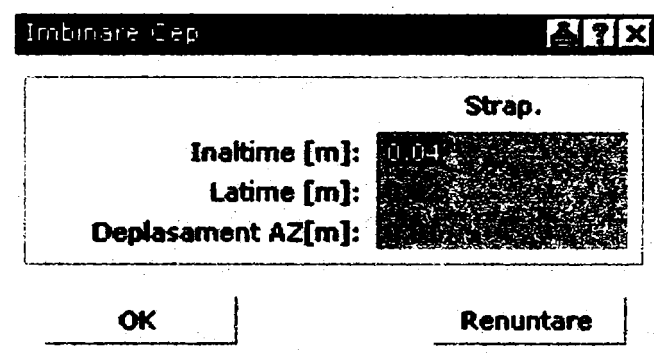


Figura 3.4: Cep străpuns când cele două corpuri nu sunt în prelungire

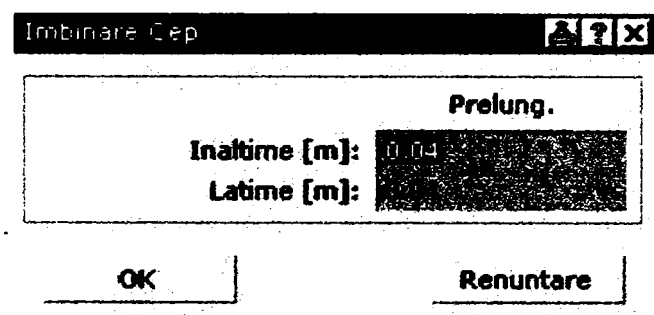


Figura 3.5: Cep străpuns când cele două corpuri sunt în prelungire

După specificarea tuturor parametrilor se încearcă codarea prelucrării analizând poziția celor două corpuri, numărul și forma suprafețelor lor de contact.

Dacă între cele două corpuri există mai multe suprafețe poligonale de contact se propun spre alegere.

Acestea sunt contururi poligonale închise și vor fi denumite în continuare *fețe de contact*. La funcțiile 1, 2 și 3 pot fi doar pe una din fețele laterale ale corpului 2.

Chiar dacă fețele de contact sunt pe două plane laterale alăturate doar una poate fi aleasă. Deci pot să apară doar îmbinările următoare. Fețele corpului 2 care sunt paralele cu axa corpului 1 nu pot fi fețe de contact. În figura 3.6 se pot vedea cazurile posibile.

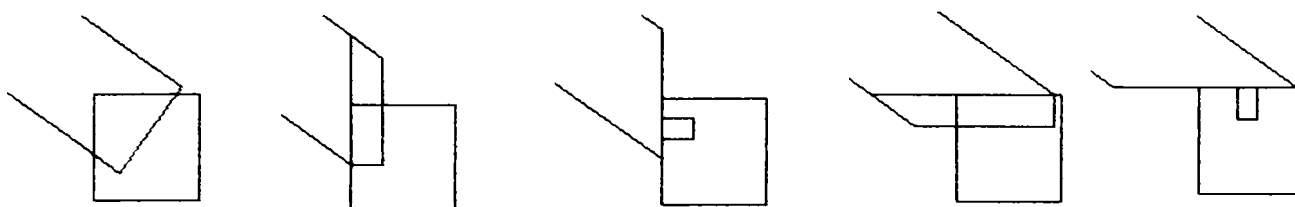


Figura 3.6: Cazuri posibile de obținere a unui cep

Orientarea cepului este întotdeauna după gaura de cep.

Calculul secțiunii cepului nu se bazează pe lățimea sau înălțimea corpului. Pentru secțiune se ia în considerare zona comună dintre cele două corpuri. Secțiunea este în mijlocul zonei comune.

Zona comună ale celor două corpuri nu trebuie să se întindă pe toată secțiunea corpului moale. Un exemplu este prezentat în figura 3.7

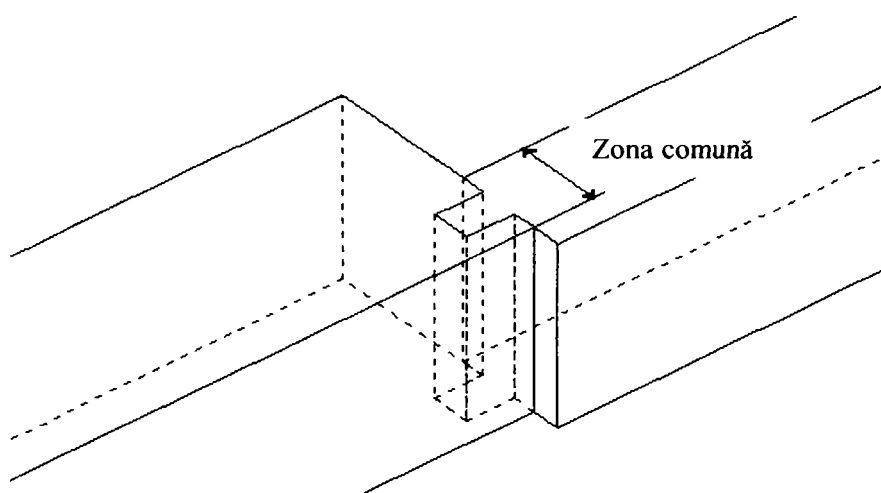


Figura 3.7: Fața de contact a celor două corpuri este zona comună

Dacă corpul tare este poziționat astfel încât gaura în corpul moale ar fi generată la marginea lui se poate introduce un deplasament care se aplică prelucrării, adică ea va fi micșorată cu această valoare.

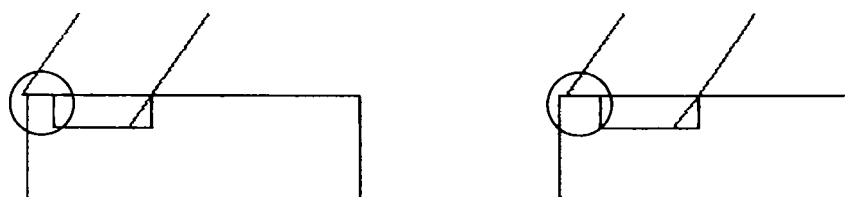


Figura 3.8: Zona de aplicare a unui deplasament suplimentar este evidențiată printr-un cerc.

Cu ajutorul acestei funcții se vor genera prelucrări cu codul 4007, 1, 2, 3, 4 pentru cep.

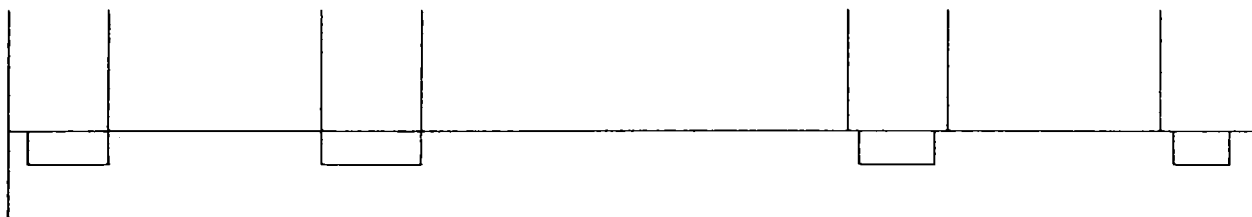


Figura 3.9: Cazurile de cep care se vor genera cu funcțiile care urmează

3.1.1.1 Cep care străbate corpul pe secțiunea ei (4007, 1)

În acest caz nu există deplasamente la prelucrare. În forma parametrică prelucrarea are trei unghiuri, de bază (GW), de înclinare (NW) și de răsturnare (KW) care nu sunt specificate de către utilizator ci rezultă din poziția relativă a corpului tare față de cel moale (Dietrich's Neues System, no. 4, 2000).

Un cep cu unghiul de bază și cel de înclinare egale cu 90° se obține dacă sunt îndeplinite următoarele condiții:

- axa corpului 1 este perpendiculară pe corpul 2
- fețele laterale ale celor două corpuri sunt în plane paralele
- corpul 1 nu este la capătul de început sau de sfârșit al corpului 2, sau corpul 1 este la începutul sau sfârșitul corpului 2 și valoarea pentru deplasament este 0.

Un cep cu unghiul de bază diferit de 90° și unghi de înclinare egal cu 90° se generează dacă sunt îndeplinite condițiile:

- axa corpului 1 este perpendiculară pe axa corpului 2
- fețele laterale ale celor două corpuri nu sunt paralele. Axa corpului 2 este perpendiculară pe două fețe laterale opuse ale corpului 1
- corpul 1 nu este la începutul sau la sfârșitul corpului 2 sau corpul 1 este la începutul sau sfârșitul corpului 2 și valoarea pentru deplasament este 0

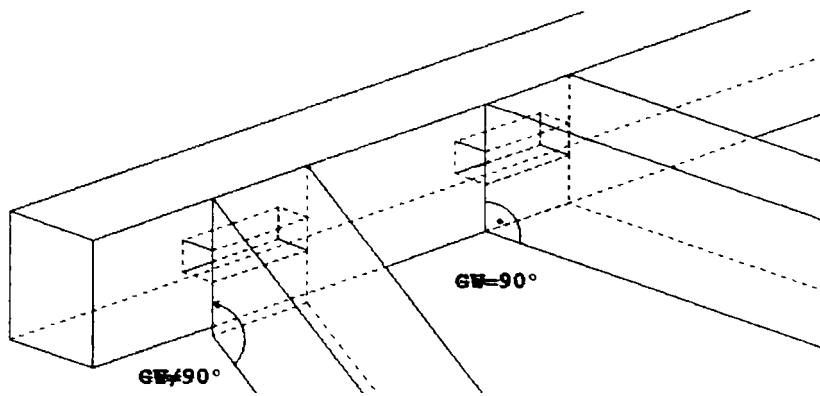


Figura 3.10: Cep ce străbate corpul pe secțiunea lui (4007, 1)

3.1.1.2 Cep clasic (4007, 4)

Este cazul cel mai frecvent întâlnit în practică.

Un cep 4007,4 se generează dacă sunt îndeplinite următoarele condiții:

- axa corpului 1 nu este perpendiculară pe axa corpului 2
- câte două din fețele laterale ale celor două corpuri sunt în plane paralele.
- corpul 1 nu este la începutul sau sfârșitul corpului 2 sau corpul 1 este la începutul și sfârșitul corpului 2 și valoarea pentru deplasament este 0

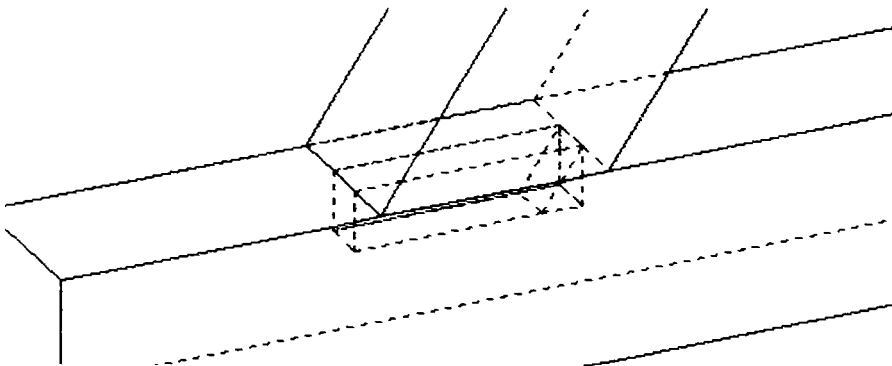


Figura 3.11: Cep ce străbate corpul tare pe secțiunea lui

3.1.1.3 Cep rotit (4007, 3)

Un cep de tip 4007,3 cu unghiul de bază și cel de înclinare de 90° și cel de răsturnare diferit de 90° se generează dacă sunt îndeplinite următoarele condiții:

- axa corpului 1 este perpendiculară pe axa corpului 2.
- fețele laterale ale corpului 1 nu sunt paralele cu fețele laterale ale corpului 2.

Un cep cu toate cele trei unghiuri diferite de 90° se generează dacă este îndeplinită următoarea condiție:

- corpurile sunt astfel orientate încât axa corpului 1 nu este perpendiculară pe nici o față laterală a corpului 2.

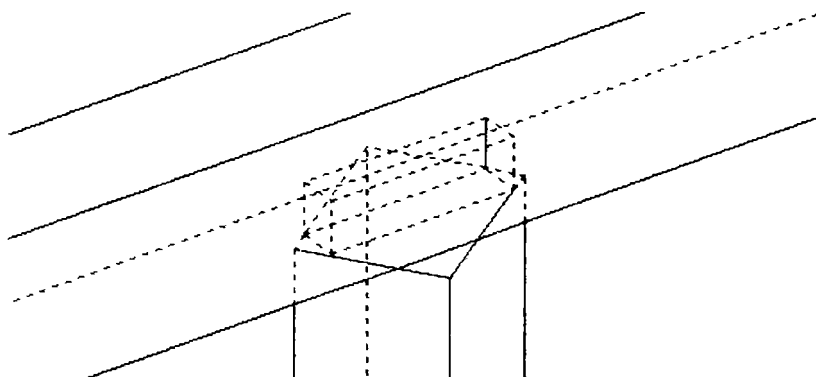


Figura 3.12: Cep rotit (4007, 3)

3.1.1.4 Cep cu deplasamente (4007, 2)

Pentru cepul cu deplasamente (4007,2) cel puțin una din valorile A_1 și A_2 este diferită de 0. Dacă ambele deplasamente sunt 0, atunci se generează un cep care străbate corpul (4007,1), un cep rotit (4007,3) sau un cep trunchiat (4007,4). Valorile pentru deplasarea pe x , deplasarea în secțiune și pentru unghiurile de bază și de înclinare rezultă din construcție și nu se pot modifica ulterior.

Dacă cele două deplasamente sunt diferite, există două posibilități pentru orientarea cepului. În această situație se propune fiecare poziție posibilă pentru cep. Dacă poziția curentă este refuzată se propune următoarea poziție. Ca poziție următoare se propune cea care are ca față de referință fața opusă poziției curente. Deci cepul va fi rotit cu 180° , nu cu 90° .

3.1.1.5 Ce cod va fi generat și de ce

Dacă corpul 1 stă exact pe capătul de început sau sfârșit al corpului 2 și dacă deplasamentul de capăt este diferit de 0, deplasamentul corespunzător capătului în cauză se mărește cu această valoare.

În funcție de poziția celor două corpuri se poate obține și un unghi de răsturnare. În acest caz, nu se generează un cep cu deplasamente (4007,2), ci un cep rotit (4007,3). Lungimea lui va fi mărginită de deplasamente.

Dacă în loc de cep cu deplasamente (4007,2) se generează un cep rotit (4007,3), acesta va fi generat conform regulilor următoare.

Cazul 1: Punctele de tăiere dintre liniile P1-P2 și P3-P4 cu fețele laterale ale corpului 1 stau pe fețe laterale opuse

În această situație deplasamentele se măsoară paralel cu fețele laterale corespunzătoare. Deplasamentul 1 se măsoară paralel cu fața tăiată de linia P1-P2. Următoarea față a corpului din șirul C-D-E-F are același deplasament 1. Fețele laterale opuse corespunzătoare conțin deplasamentul 2. De exemplu: fața D din corpul 1 va fi tăiată de linia P1-P2. Fețelor D și E li se atribuie deplasamentul 1 și fețelor F și C li se atribuie deplasamentul 2 (figura 3.13).

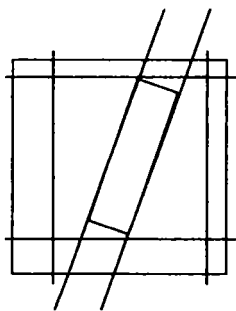


Figura 3.13: Liniile P1-P2 și P3-P4 taie două fețe laterale opuse

Cazul 2: Punctele de tăiere ale liniilor P1-P2 și P3-P4 cu fețele laterale ale corpului 1 sunt pe fețe învecinate

În această situație deplasamentul 1 respectiv deplasamentul 2 se măsoară paralel cu fețele laterale alăturate. Deplasamentul 1 se măsoară paralel cu fețele laterale care sunt tăiate de liniile P1-P2 și P4-P3. Fețelor care sunt tăiate de liniile P2-P1 și P3-P4, li se atribuie deplasamentul 2 (figura 3.14).

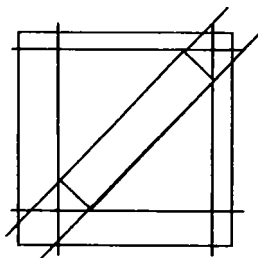


Figura 3.14: Liniile P1-P2 și P3-P4 taie două fețe laterale alăturate

Cazul 3: Ambele linii P1-P2 și P4-P3 taie într-un sens două fețe laterale învecinate ale corpului 1. În celălalt sens taie ambele linii doar o față laterală a corpului 1

În această situație feței laterale din corpul 1 care este tăiată numai de o singură linie i se atribuie deplasamentul 1. Feței care este tăiată de ambele linii sau nu este tăiată deloc i se atribuie deplasamentul 2 (figura 3.15).

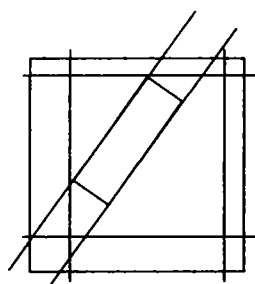


Figura 3.15: Liniile P1-P2 și P3-P4 taie într-un sens două fețe laterale învecinate și în celălalt sens aceeași față laterală

Cazul 4: Punctele de tăiere ale liniilor P1-P2 și P4-P3 cu fețele laterale ale corpului 1 sunt situate pe aceeași față

În această situație fața laterală tăiată de linia P1-P2 cu sensul de la P1 la P2 conține deplasamentul 1. Fața laterală din corpul 1 care este tăiată de linia P1-P2 cu sensul de la P2 la P1 conține deplasamentul 2. Fețele laterale opuse corespunzătoare conțin deplasamentul corespunzător. De exemplu, dacă fața C conține deplasamentul 1, atunci feței E i se atribuie deplasamentul 2 (figura 3.16).

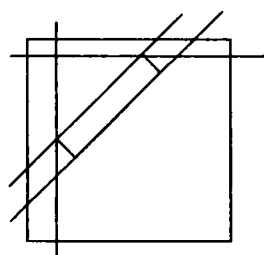


Figura 3.16: Liniile P1-P2 și P3-P4 taie aceleași două fețe laterale alăturate

3.1.1.6 Generarea prelucrării pe un capăt al corpului moale

La funcția cep de capăt între cele două corpuri se generează o îmbinare compusă dintr-un cep normal (4007,1) sau cep rotit (4007,3) și dintr-o gaură de cep de capăt (4003,2) cu o tăiere (4001,1) (figura 3.17). Gaura de cep în corpul 2 are unghiul de

înclinare diferit de 0. Indiferent dacă secțiunea corpului 1 este mai mare sau mai mică decât secțiunea corpului 2 în corpul 2 se generează întotdeauna o gaură de cep de capăt.

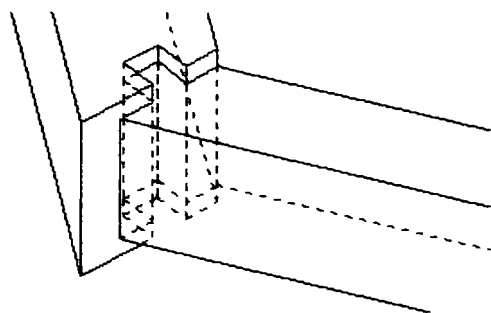


Figura 3.17: Cazul obținerii unei găuri de cep la capăt

Pe baza poziției celor două corpuri se poate ca unghiurile prelucrării să fie diferite de 90° .

Între cele două corpuri există o singură față de contact. Fața de contact este planul bisector (figura 3.18). Ea trece prin punctul de intersecție a celor două axe.

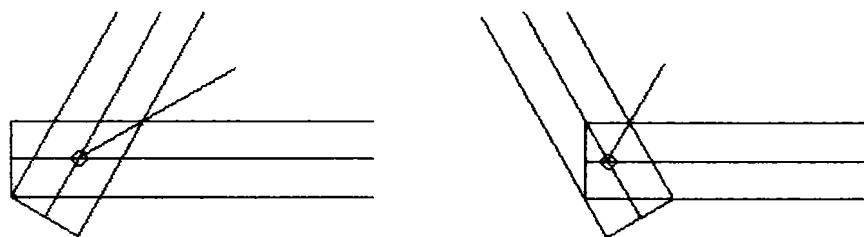


Figura 3.18: Planul ce conține fața de contact este bisector

Dacă axele celor două corpuri sunt paralele și cele patru fețe laterale sunt în plane respectiv paralele (figura 3.19), unghiul de bază și cel de înclinare a cepului sunt 90° (definiția unghiurilor de bază și de înclinare este dată în capitolul 2 subcapitolul 2.2.7.1). Gaura de cep și tăierea se generează astfel încât să se potrivească.

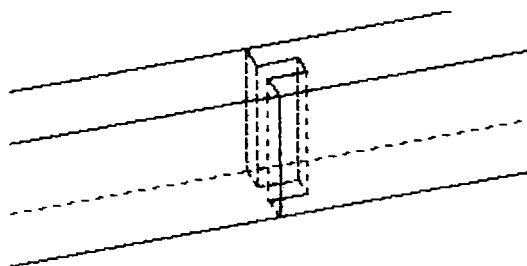


Figura 3.19: Cazul obținerii unei prelucrări cu corpurile cu fețele paralele

Dacă axele celor două corpuri sunt coplanare dar fețele lor laterale nu sunt în plane paralele, doar unul din unghiurile de bază și cel de înclinare este de 90° (figura 3.20). Gaura de cep și tăierea se generează astfel încât să se potrivească.

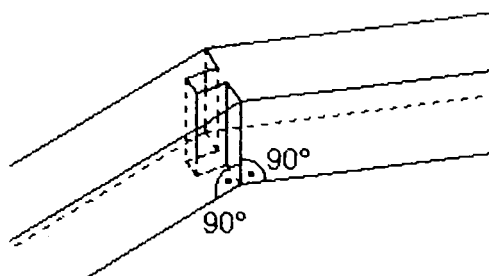


Figura 3.20: Cazul obținerii unei prelucrări cu corpuri cu două fețe laterale neperpendiculare

Dacă nici una din fețele laterale nu sunt în plane paralele ambele unghiuri sunt diferite de 0 (figura 3.21). GW și NW al feței de contact pot fi determinate cum urmează. Din proiecția axei corpului 2 pe planul în care este fața C a corpului 1 se obține valoarea pentru unghiul de bază. Din proiecția axei corpului 2 pe planul în care este fața D a corpului 1, se obține valoarea pentru unghiul de înclinare. Unghiul de răsturnare KW a capului rămâne întotdeauna 0. Gaura de cep și tăierea se vor genera astfel încât să se potrivească.

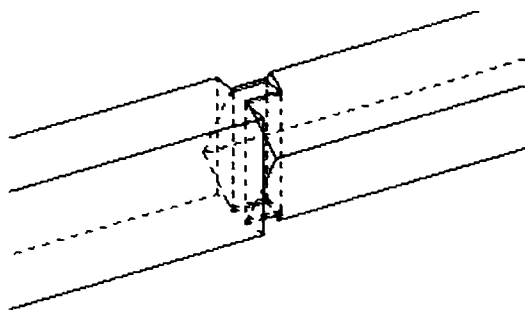


Figura 3.21: Cazul obținerii unei prelucrări cu corpuri cu toate fețele laterale neperpendiculare

Ambele corpuri se vor alungi sau scurta până la planul bisector. Dacă unul din corpuri este mai lung și trece dincolo de celălalt sunt posibile mai multe fețe de contact. În acest caz există și mai multe posibilități de îmbinare a corpurilor. În acest caz se păstrează capătul pe care a fost selectat corpul și acea bucată va rămâne și se va îmbina. Cealaltă parte se elimină.

3.1.2 Generarea prelucrării pe baza orientării găurii de cep

Dacă se dorește generarea prelucrării în funcție de orientarea găurii de cep, se apelează un alt punct de meniu, care propune utilizatorului introducerea parametrilor prin intermediul cutiei de dialog din figura 3.22:

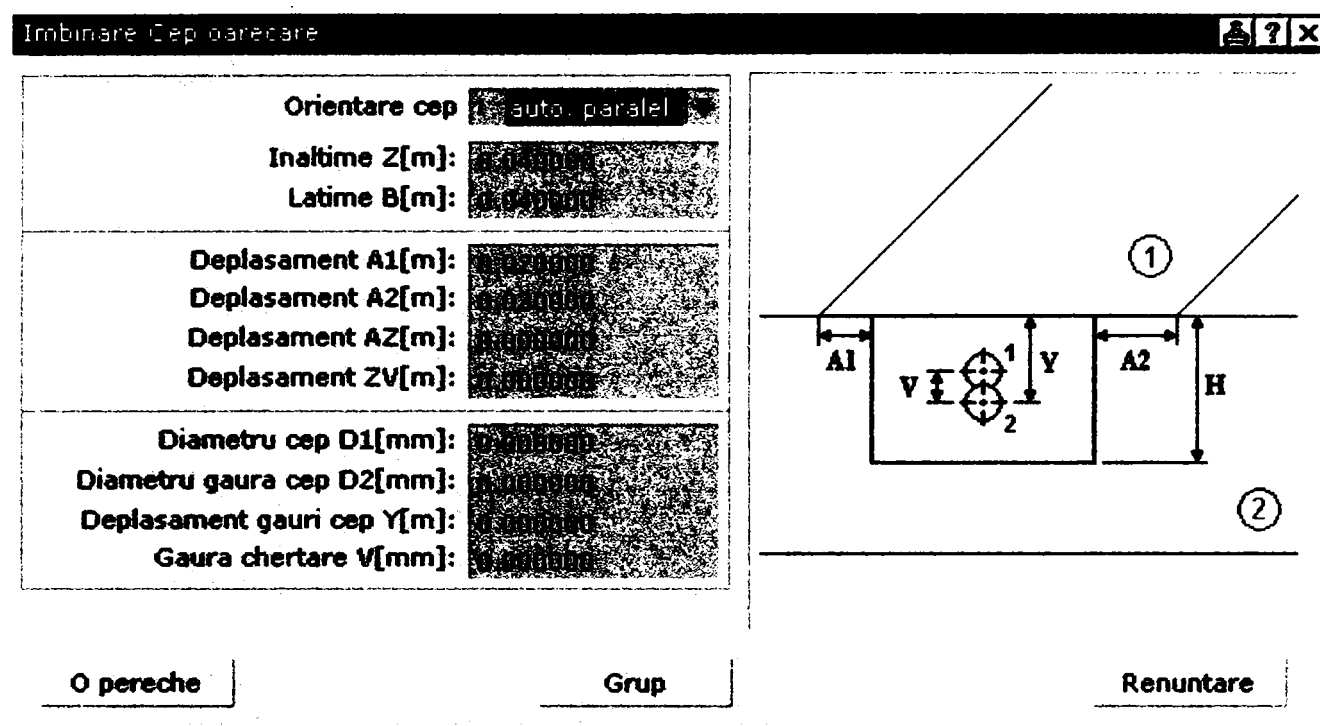


Figura 3.22: Cutia de dialog prin care se specifică parametrii cepului oarecare

În primul câmp se poate specifica orientarea găurii de cep: perpendiculară pe corpul 2, paralelă cu axa corpului 2, selecție sau fețele C, D, E, F.

La setările perpendiculară pe corpul 2 și paralelă cu corpul 2, orientarea cepului se face pe corpul 2. Setările selecție sau "C, D, E, F" se referă la orientarea cepului relativ la corpul 1. Fața selectată este fața de referință a cepului. Lungimea găurii cepului este lungimea liniei P5-P6.

Dacă s-a ales orientarea perpendiculară pe corpul 2, gaura de cep (4006,1) se generează astfel încât lățimea găurii se măsoară pe direcția axei corpului 2 (figura 3.23).

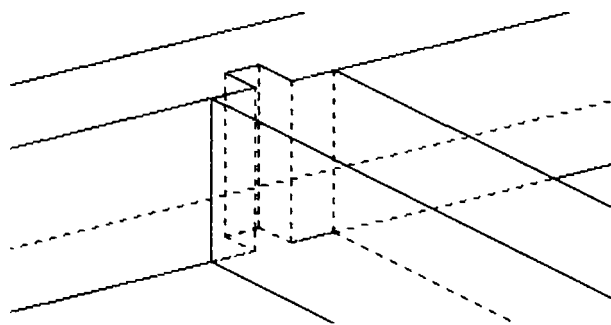


Figura 3.23: Cazul orientării prelucrării perpendicular pe corpul 2

Dacă s-a ales orientarea paralelă cu corpul 2, gaura de cep (4006,1) va fi generată paralelă cu axa corpului 2. Lățimea găurii se măsoară perpendicular pe axa corpului 2.

Dacă s-a ales pentru orientare "selecție", se propune cepul în direcția liniei P1-P2 care este cea mai aproape de axa corpului 2. Utilizatorul poate renunța la poziția propusă. Apoi se propune cepul cu fața de referință egală cu următoarea. Asta înseamnă pentru gaura de cep, că valorile pentru lungime și lățime se inversează.

Dacă s-a setat una din fețe ca față de referință, atunci este decisivă orientarea cepului. Cepul se va genera cu această față ca față de referință cu unghiul de răsturnare egal cu 0. Dacă în corpul 2 s-ar genera o gaură de cep care nu corespunde definiției (paralel cu axa corpului 2), se va genera un alt tip de prelucrare (4006,3).

Dacă valoarea pentru diametru este diferită de 0, se vor genera și găuri împreună cu îmbinarea. În ambele corpuri se pot genera numai găuri perpendiculare (4101,1). Aceste găuri trebuie să fie paralele între ele. Dacă nu se poate, nu se vor genera găuri.

Acest mod de efectuare a prelucrării permite și generarea unui grup de prelucrări, dacă se apasă butonul "Grup". În acest caz se pot alege mai multe corpuri pe post de corpuri tari și mai multe corpuri moi. Programul va încerca să efectueze toate prelucrările posibile tratând fiecare corp tare cu fiecare corp moale. Facilitatea este utilă în cadrul unei construcții complexe unde ar fi incomod pentru utilizator să selecteze corpurile două câte două.

Algoritmul de generare a prelucrării cep și gaură de cep poate fi descris prin următorul pseudo-cod:

```
-se selectează corpul tare și cel moale
// se determină fața laterală a corpului moale pe care se va genera gaura de cep
-se intersectează cele două corpuri
-din corpul rezultat se aleg fețele care nu există în corpul tare inițial
-din aceste fețe se alege cea care este cea mai apropiată de punctul prin care a
    fost selectat corpul tare
-dacă corpul tare nu este suficient de lung el se alungește până la fața de contact
    aleasă
// se determină fața de referință a prelucrării pe corpul moale
-se parcurg planele laterale ale corpului moale și se identifică indicele feței care are
    normala egală cu cea a feței de contact
-se trec punctele feței de contact din sistemul global în sistemul local al corpului
    moale din acesta se trec în sistemul feței de referință dat de indice
-se determină valorile minim și maxim pe x, y; acestea se ajustează cu
    deplasamentele dacă e cazul
// se calculează cele opt puncte ale găurii de cep
p0 = (xmin, (ymin + ymax) / 2 + lățime cep / 2, 0)
p1 = (xmax, (ymin + ymax) / 2 + lățime cep / 2, 0)
p2 = (xmax, (ymin + ymax) / 2 - lățime cep / 2, 0)
p3 = (xmin, (ymin + ymax) / 2 - lățime cep / 2, 0)
p4 = (xmin, (ymin + ymax) / 2 + lățime cep / 2, înălțime cep)
p5 = (xmax, (ymin + ymax) / 2 + lățime cep / 2, înălțime cep)
p6 = (xmax, (ymin + ymax) / 2 - lățime cep / 2, înălțime cep)
p7 = (xmin, (ymin + ymax) / 2 - lățime cep / 2, înălțime cep)
-aceste puncte se trec în sistemul local al volumului moale
-se generează șirul de prelucrări ca:
    4006, 1, , p0, p1, p2, p3, p4, p5, p6, p7
-se inserează în volumul moale
-se inversează punctele între ele
    p0 <-> p1
    p2 <-> p3
    p4 <-> p5
```

p6 <-> p7

- se ajustează în funcție de tăierile existente pe corpul tare
- se generează șirul de prelucrări pentru corpul tare
- se determină capătul corpului tare pe care se va insera prelucrarea
- se elimină prelucrările de la capătul pe care se va insera prelucrarea curentă
- se inserează în clasa de volum a corpului tare prelucrarea
- în final se modelează cele două corpuri conform prelucrărilor existente pe ele folosind operațiile booleene

Datorită faptului că efectuarea prelucrărilor pe corp se face folosind operațiile booleene, se pot genera mai multe prelucrări de tip cep la același capăt. Ca rezultat se obține un corp complex care este foarte des utilizat în construcțiile în lemn din ziua de azi. În figura 3.24 este un astfel de exemplu:

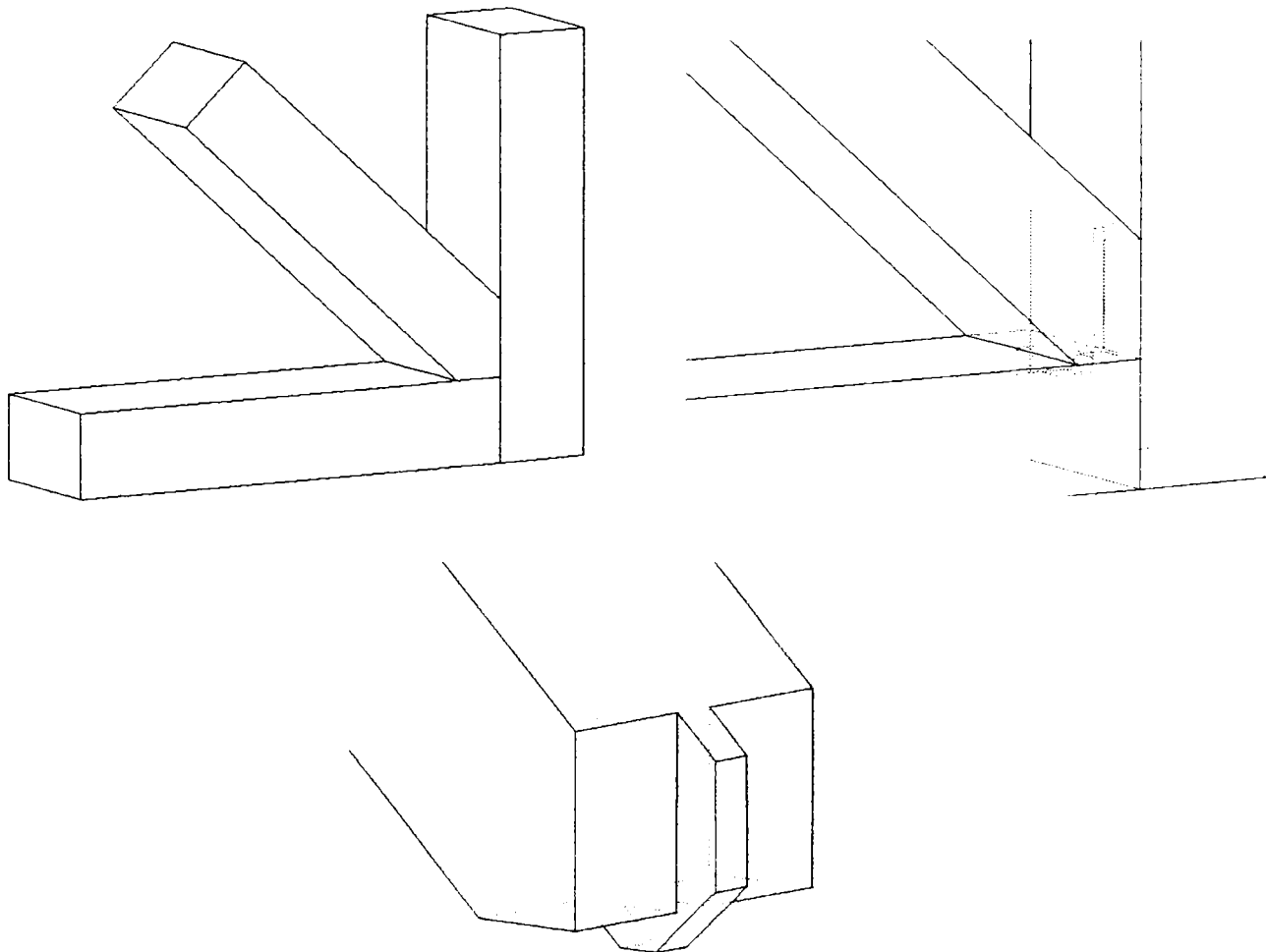


Figura 3.24: Îmbinarea dintre trei corpuri. Corpul oblic va avea două prelucrări cep la capătul îmbinat

3.2 Generarea cepului coadă de rândunică

Toate funcțiile care se vor prezenta în continuare vor genera prelucrări de tip cep coadă de rândunică pe corpul tare (4007,5) și gaura corespunzătoare pe corpul moale (4006, 4) sau (4006,5) în funcție de poziția relativă a prelucrării față de corpul moale adică pe o față laterală sau pe un capăt. Decizia ce fel de prelucrare să se efectueze este luată de către utilizator (Lignarius, 2001).

Ca și în cazul cepului prezentat la subcapitolul anterior există trei modalități de generare:

- cep coadă de rândunică generat pe o față laterală a corpului moale
- cep coadă de rândunică generat pe un capăt al corpului moale
- pe baza orientării găurii de cep

3.2.1 Generarea prelucrării pe față laterală a corpului moale

Specificarea parametrilor se face prin intermediul cutiei de dialog din figura 3.25:

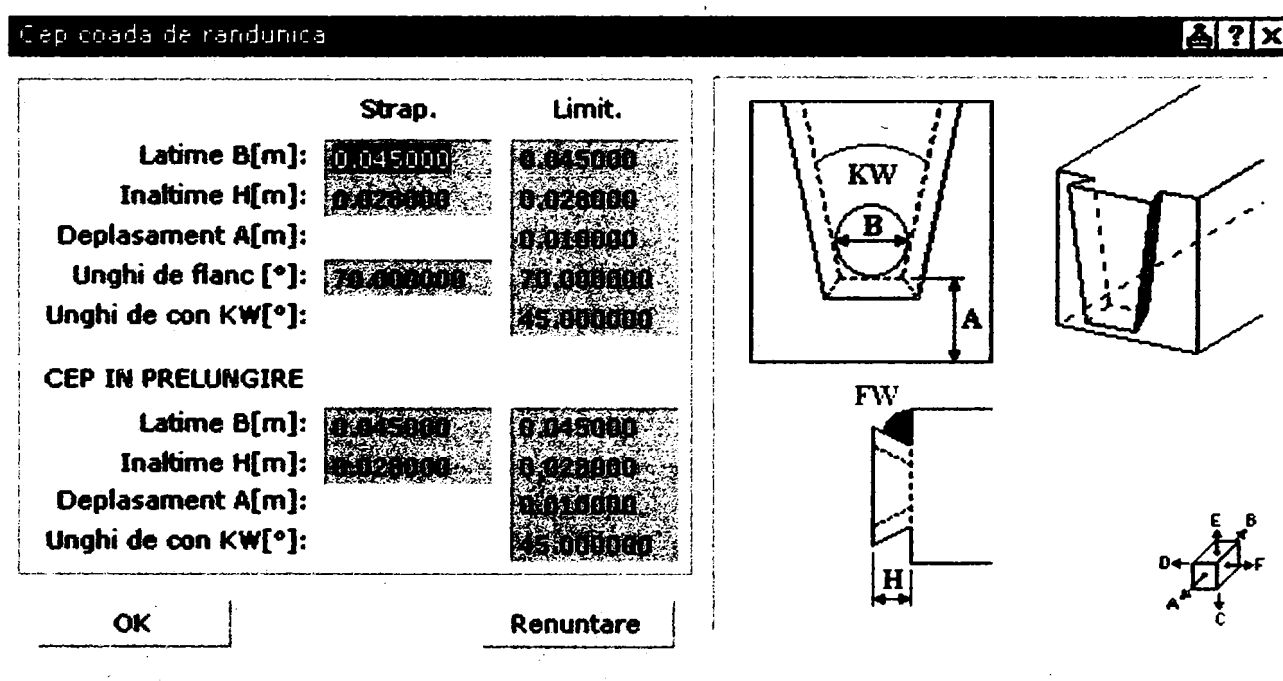


Figura 3.25: Cutia de dialog prin care se specifică valorile implicite ale parametrilor prelucrării

Ca și în cazul prelucrării cep din subcapitolul anterior, dacă nu se introduc toți parametrii, se deschid niște cutii suplimentare care forțează uilizatorul să-i specifice pe cei omiși, pentru a se putea efectua prelucrarea.

Unghiul de bază al găurii cepului coadă de rândunică care este pe o față laterală poate fi 0° sau 180° . În cazul în care prelucrarea este la capăt unghiul de bază poate fi 90° sau -90° .

Corpul tare se alungește până când atinge corpul moale. Dacă nici prin alungire nu se obține nici o față de contact, prelucrarea nu este posibilă.

Dacă între cele două corpuri există mai multe fețe de contact acestea se propun spre alegere. Fețele de contact pot fi doar pe fețele laterale ale corpului moale. (C, D, E, F).

Orientarea prelucrării

Prelucrarea are un sistem propriu de coordonate unde axa Y este de-a lungul prelucrării.

Prelucrarea de pe corpul tare se orientează întotdeauna după gaura corespunzătoare. Gaura se orientează astfel încât axa Y a sistemului atașat prelucrării să fie orientată după axa +Z a sistemului global de coordonate. Dacă axa Y a sistemului atașat prelucrării este paralelă cu planul XoY al sistemului global, prelucrarea se orientează după +Y. Dacă axa Y a sistemului atașat prelucrării este paralelă cu axa X prelucrarea se orientează după +X.

Deplasamentul prelucrării în secțiune

Deplasamentul nu se calculează în funcție de secțiunea vreunui corp ci în funcție de poziția celor două corpuri unul față de celălalt, adică față de zona comună a lor, care este de fapt și fața de contact. Această față nu trebuie să fie pe toată secțiunea vreunui corp (figura 3.26).

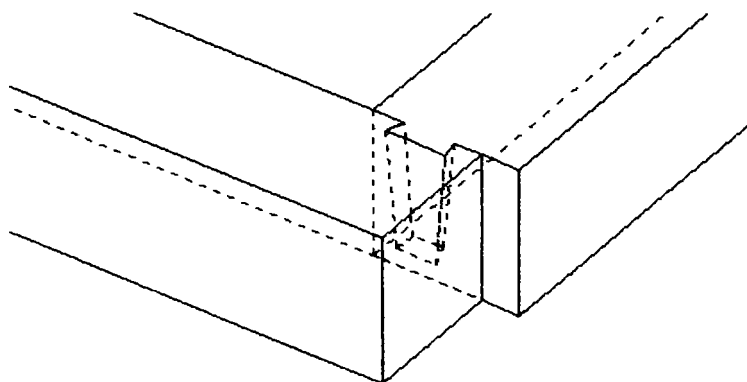


Figura 3.26: Prelucrarea se generează relativ la zona de contact comună

Prelucrările ce se pot genera pot fi de trei tipuri:

- prelucrare ce străpunge corpul pe toată secțiunea (în acest caz parametrii deplasament și unghiul de con sunt 0 și nu pot fi modificați)
- prelucrare limitată
- prelucrare de capăt care poate să străbată corpul moale sau să fie limitată

3.2.1.1 Cep coadă de rândunică ce străpunge corpul pe secțiunea lui

Această prelucrare este o particularizare a prelucrării (4007,5). Ea are proprietatea că parametrii deplasament și unghi de con sunt 0 și ei nu pot fi modificați de către utilizator (figura 3.27). Valorile pentru deplasament pe axa x, deplasament în secțiune, unghi de bază și unghi de înclinare rezultă din poziția celor două corpuri. Unghiul de răsturnare este întodeauna 0 sau 180°. Parametrii care sunt preluați de la utilizator sunt: lățimea cepului, înălțimea cepului și unghiul de flanc.

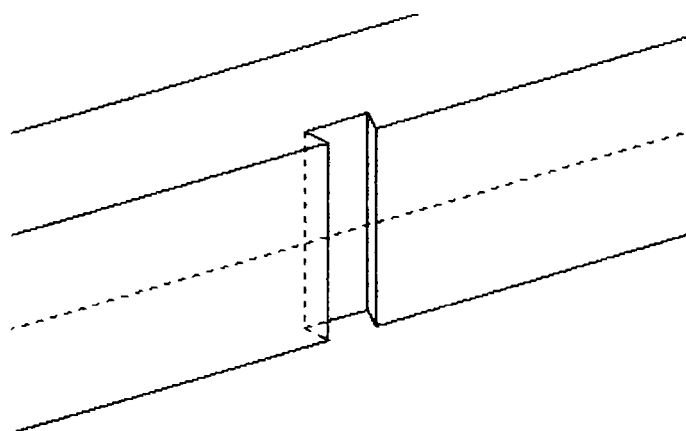


Figura 3.27: Cep coadă de rândunică ce străpunge corpul moale dintr-o parte în alta

3.2.1.2 Cep coadă de rândunică limitat

În acest caz utilizatorul introduce și parametrii pentru deplasament și unghiul de con. Nu se exclud valorile de 0.

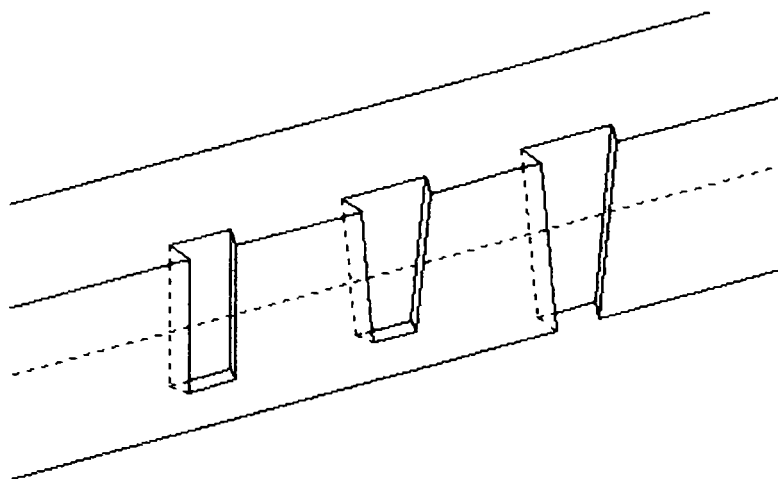


Figura 3.28: Cep coadă de rândunică limitat

În funcție de poziția celor două corpuri se pot obține următoarele cazuri (Lignarius, 2001):

Cazul 1: Fețele laterale ale celor două corpuri care sunt situate în partea negativă a axei Y a sistemului de coordonate a prelucrării sunt în același plan

În acest caz deplasamentul cepului și deplasamentul găurii sunt egale (figura 3.29).

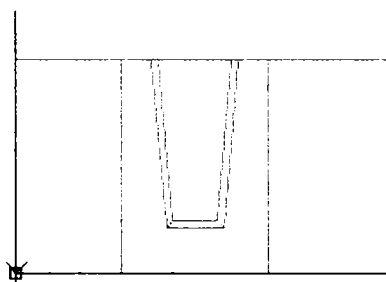


Figura 3.29: Cepul și gaura au același deplasament

Cazul 2: Fața de jos a corpului tare este situată sub fața de jos a corpului moale. Partea mai lată a prelucrării străbate întotdeauna ambele corpuri

În acest caz deplasamentul prelucrării de pe corpul tare se mărește cu valoarea de suprapunere a celor două corpuri (figura 3.30).

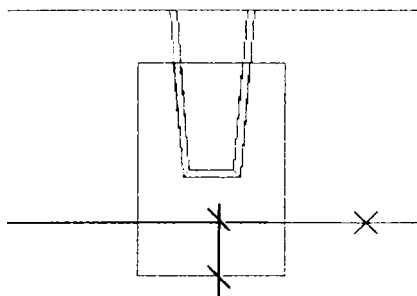


Figura 3.30: Cepul are deplasament mai mare decât gaura de cep

Cazul 3: Fața de jos a corpului tare este situată deasupra feței de jos a corpului moale

În acest caz deplasamentul găurii de cep se va mări cu zona de suprapunere a celor două corpuri (figura 3.31).

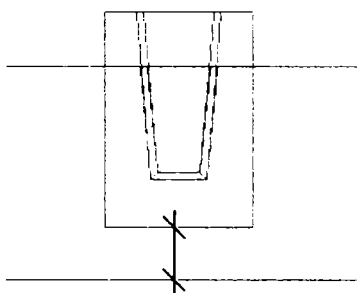
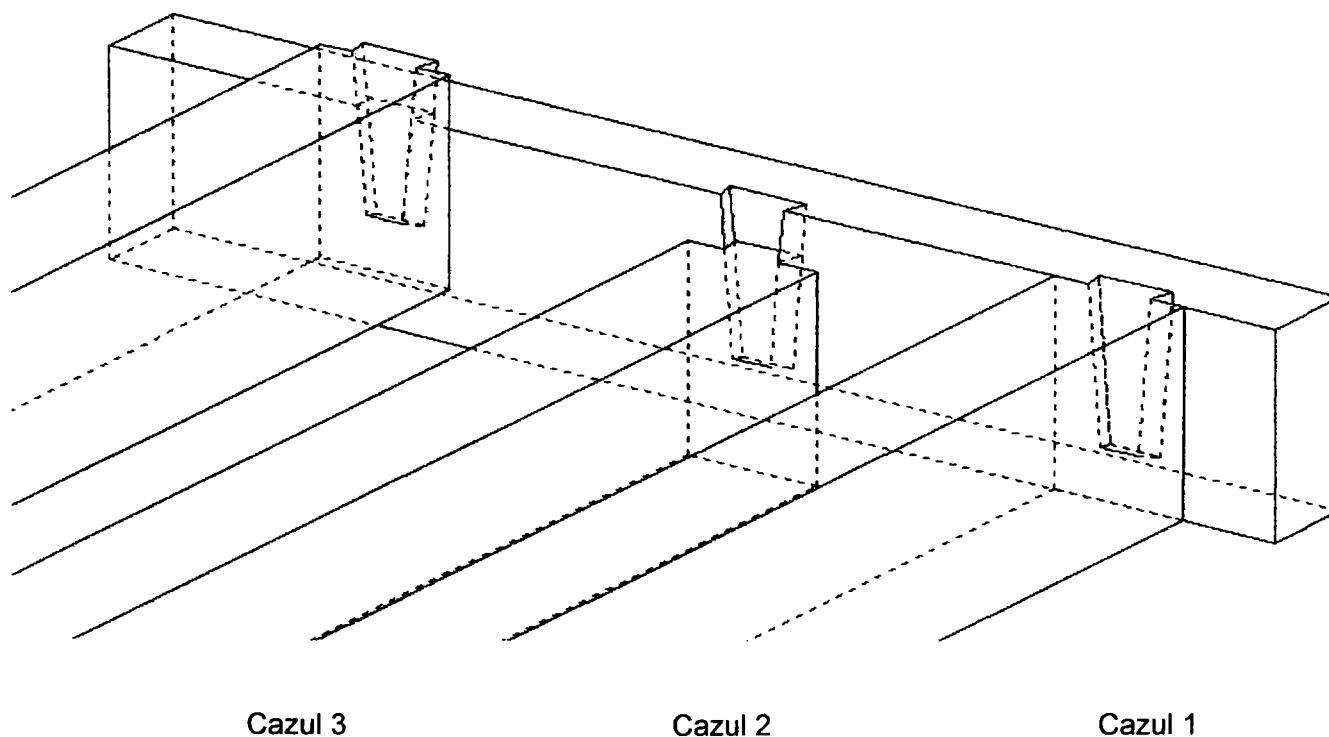


Figura 3.31: Gaura de cep are deplasament mai mare decât cepul



Cazul 3

Cazul 2

Cazul 1

Figura 3.32: Reprezentarea 3D a celor trei cazuri

3.2.2 Generarea prelucrării pe un capăt al corpului moale

Cu ajutorul acestei funcții se generează un cep 4007, 5 și gaura de cep de capăt 4006, 4. Prelucrările se generează în planul bisector al celor două corpuri. Acestea se vor prelungi până la planul lor bisector. Dacă nici prin intermediul prelungirii corpurile nu se ating, prelucrările nu se pot genera. Cele trei unghiuri ale prelucrării: de bază, de înclinare și de răsturnare depind de poziția celor două corpuri unul față de cealălalt. În figura 3.34 se prezintă un cep în prelungire, cu deplasament. Prelucrarea în prelungire este o particularizare a celui cu deplasament prin setarea a doi parametri pe 0: deplasamentul în lungimea prelucrării și unghiul de con (figura 3.33).

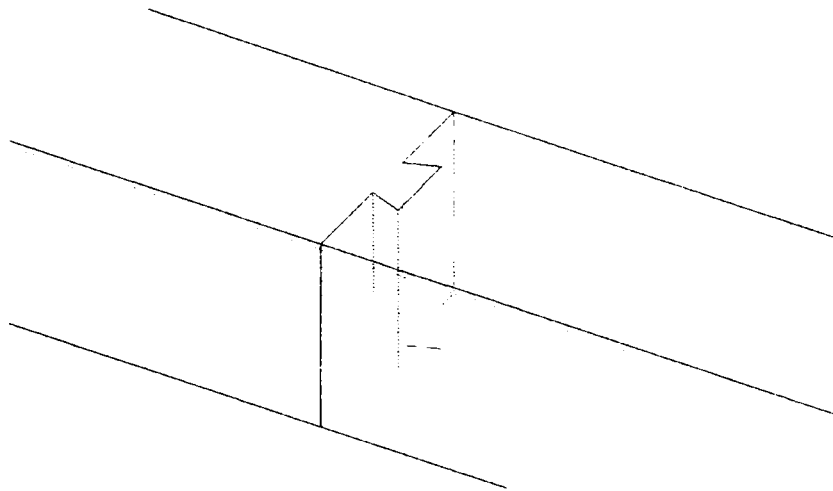


Figura 3.33: Cep generat pe capăt străpuns

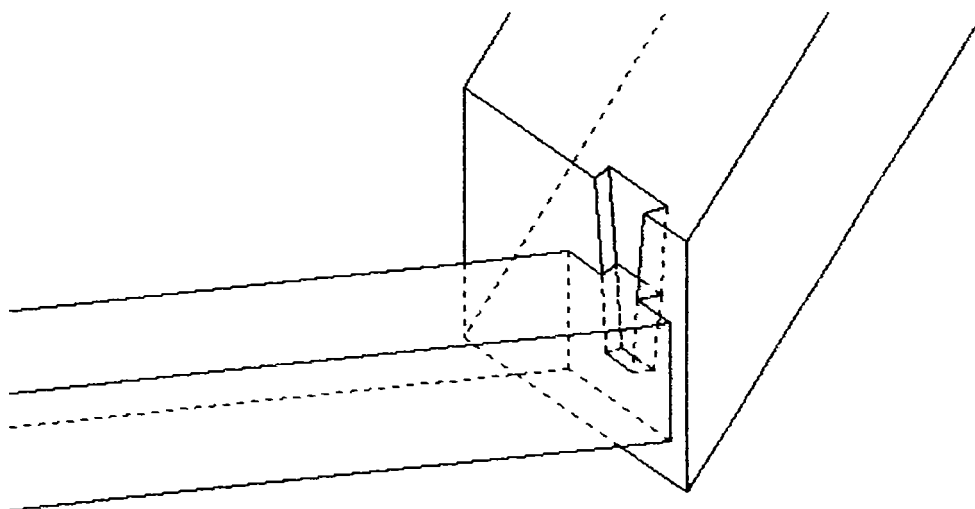


Figura 3.34: Cep generat pe capăt cu deplasament

3.2.3 Generarea prelucrării pe baza orientării găurii de cep

Specificarea parametrilor pentru a efectua prelucrarea ținând cont de orientarea găurii de cep se face prin intermediul cutiei de dialog din figura 3.35:

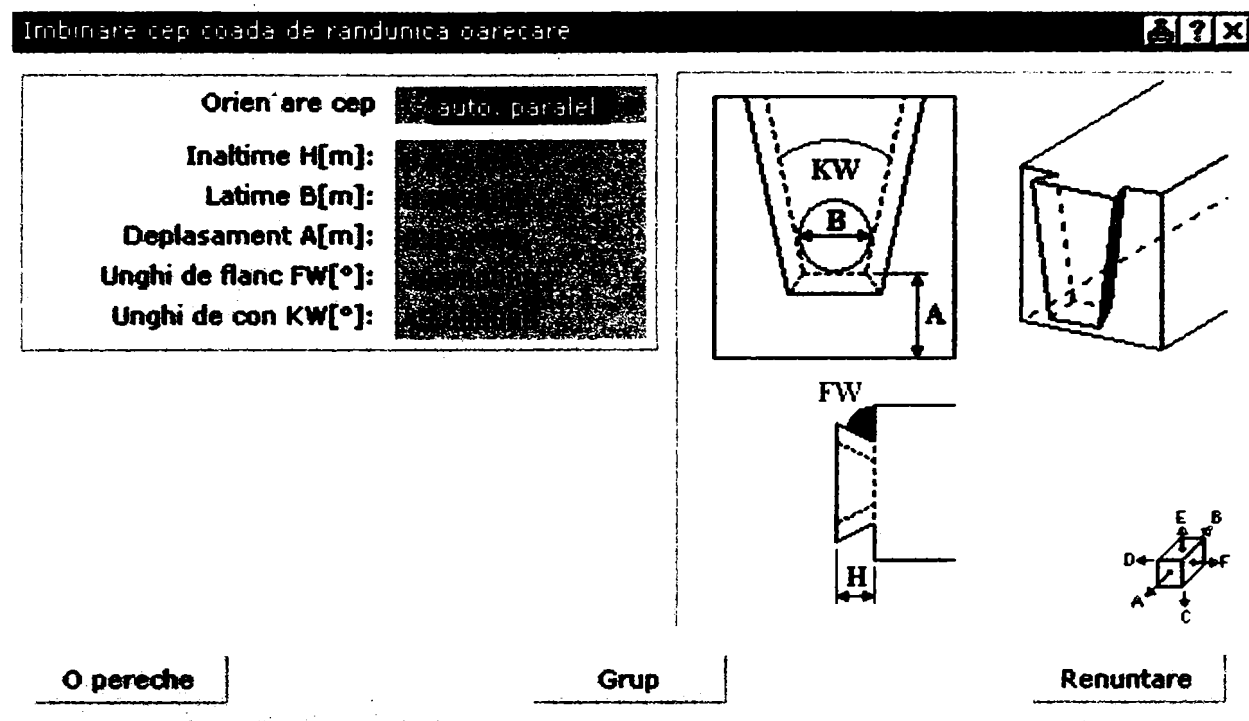


Figura 3.35: Cutia de dialog pentru specificarea parametrilor

Orientarea găurii de cep poate fi:

- automată perpendiculară pe axa corpului moale
- automată paralelă cu axa corpului moale
- la alegere
- cu fața de referință pe corpul tare C, D, E, F

Semnificațiile orientării sunt aceleași ca și la cepul normal. În figura 3.36 este un exemplu de gaură de cep generată de-a lungul axei corpului moale

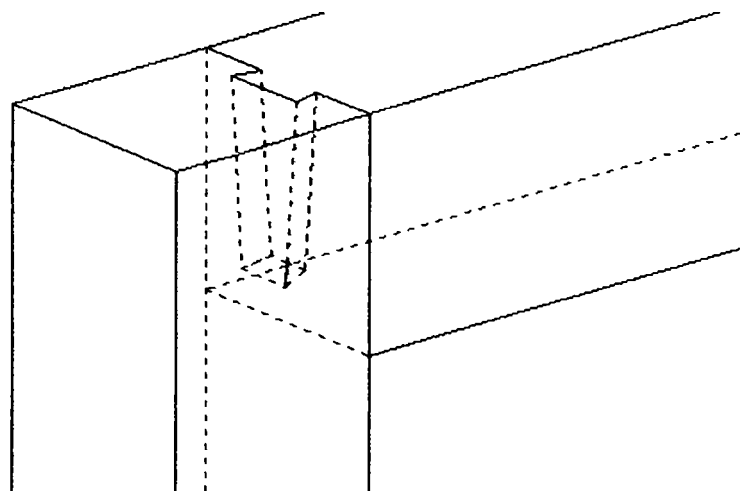


Figura 3.36: Cep oarecare cu deplasament pe lungimea prelucrării

Deplasamentul în secțiune este egal cu mijlocul zonei de suprapunere a celor două corpuri.

Semnificația butonului "Grup" este aceeași cu cea de la cepul drept prezentat în subcapitolul 3.1.

Pseudo-codul pentru acest algoritm diferă de cel de la cepul drept doar prin modul de calcul al punctelor prelucrării.

3.3 Generarea prelucrării eclisare

Există trei metode de generare a prelucrărilor de tip eclisare:

- eclisare standard, caz în care ambele corpuri sunt afectate pe fețele lor laterale
- eclisare de capăt, caz în care ambele corpuri sunt afectate la capetele lor
- eclisare frontală caz în care corpul 1 este afectat la capăt și corpul 2 pe o față laterală

Primul pas în cadrul algoritmului este selectarea celor două corpuri care vor interveni în operație. În ambele corpuri se vor genera prelucrări care pot fi incluse complet sau parțial în interiorul corpurilor sau pot fi la capătul lor.

Tipul prelucrărilor care vor rezulta în urma operației se decide în funcție de poziția relativă a unui corp față de celălalt și de parametrii specificați de către utilizator (Emmer, 1994).

3.3.1 Noțiuni generale despre efectuarea prelucrării de tip eclisare

Pe baza poziției celor două corpuri unul față de celălalt se determină numărul *fețelor de contact*, care se definesc ca fiind poligoanele corpului de intersecție dintre cele două corpuri care există pe corpul 1 și nu există pe corpul 2. Din acestea se alege una care va fi denumită *fața de bază*. Ea are proprietatea că este perpendiculară pe ambele fețe laterale ale prelucrării.

3.3.1.1 Numărul fețelor de contact = 1

Acest caz se poate obține dacă corpul 2 are o secțiune mai mică decât corpul 1. Corpul 2 se sprijină pe o față laterală a corpului 1 și nu îl străbate. Această situație se poate observa în figura 3.37:

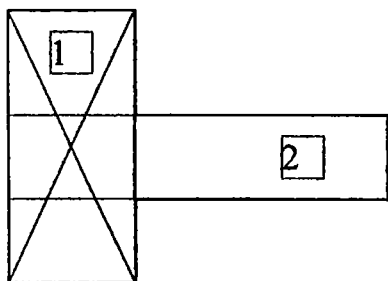


Figura 3.37: Numărul fețelor de contact este 1

În această situație îmbinarea este imposibilă.

3.3.1.2 Numărul fețelor de contact = 2

În acest caz pot să apară două situații (figura 3.38):

- când fețele de contact sunt perpendiculare una pe cealaltă
- când fețele de contact sunt paralele între ele

Fetele de contact sunt perpendiculare

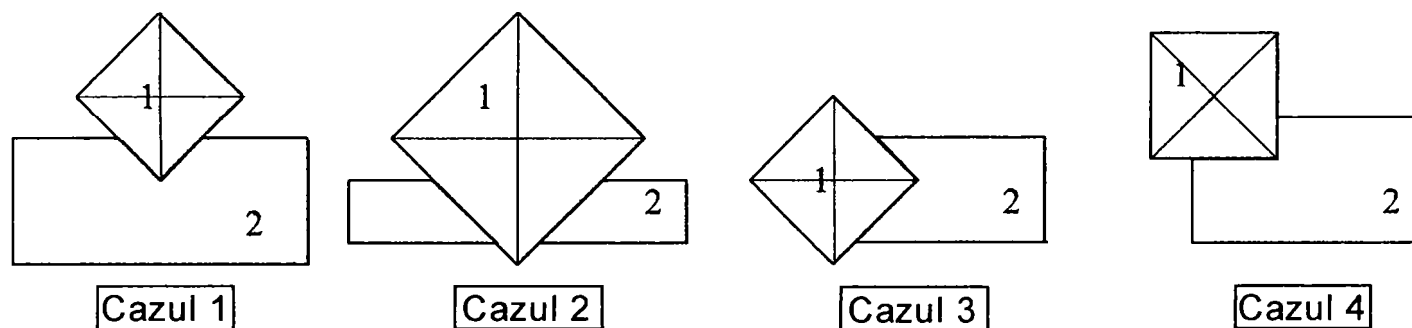


Figura 3.38: Corpul 1 este văzut din capăt și corpul 2 din lateral

Singurul parametru ce poate fi modificat de către utilizator este adâncimea prelucrării în corpul 1. Prelucrarea în corpul 2 se generează astfel încât să se potrivească cu cea din corpul 1.

Fetele de contact sunt paralele

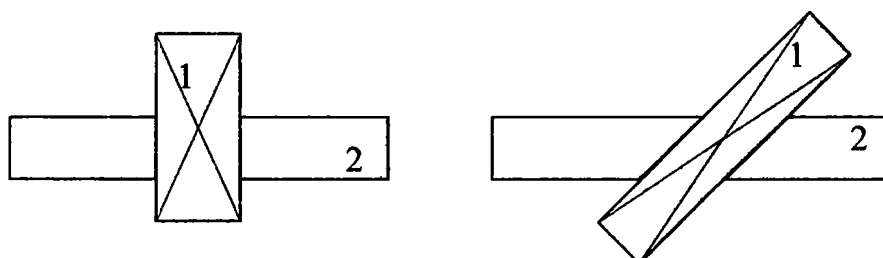


Figura 3.39: Corpul 1 este văzut din capăt și corpul 2 din lateral

În acest caz nu există inițial față de bază. Pentru a genera o față de bază se prelungesc fețele laterale ale corpului 2. Astfel se obțin două sau patru fețe de bază. Acest caz este reprezentat în figura 3.40.

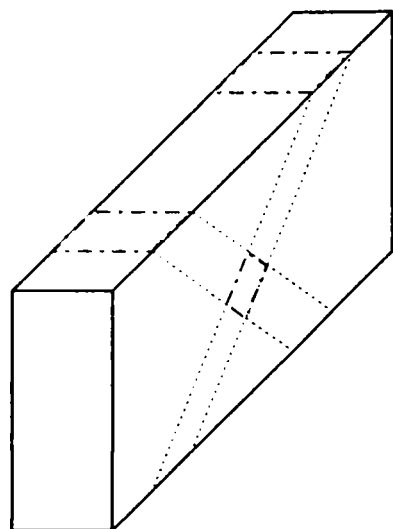


Figura 3.40: Fețele de bază obținute prelungind fețele laterale ale corpului 2
 Fețele de bază obținute se vor propune spre alegere utilizatorului pe rând.

3.3.1.3 Numărul fețelor de contact = 3

Două din fețe sunt perpendiculare pe cea de-a treia. Fața a treia se va alege ca față de bază. Cazurile care pot să apară sunt prezentate în figura 3.41.

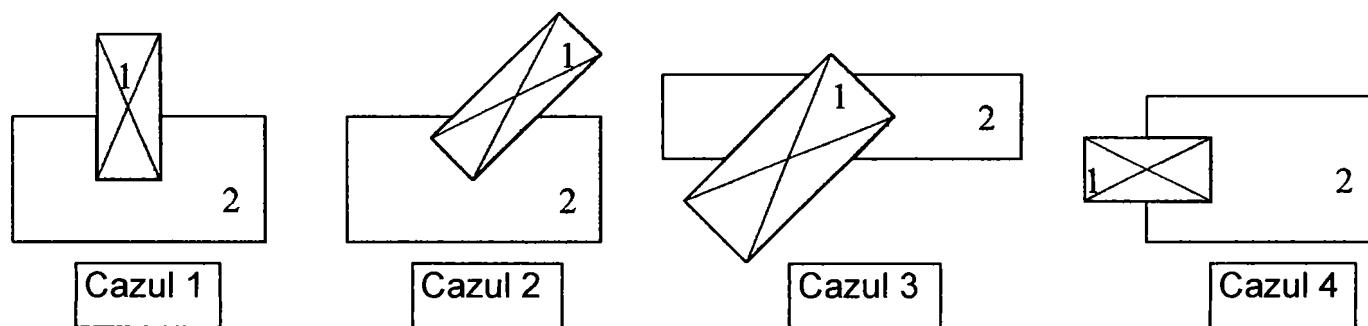


Figura 3.41: Corpul 1 este văzut din capăt și corpul 2 din lateral

3.3.1.4 Numărul fețelor de contact > 3

În acest caz oricare din fețele de contact poate fi fața de bază. În acest scop ele vor fi propuse spre alegere utilizatorului pe rând. Cazurile posibile sunt cele prezentate în figura 3.42.

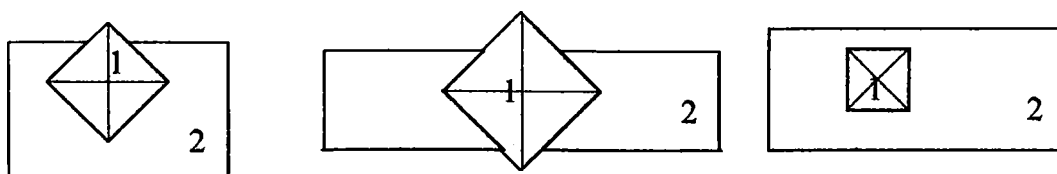


Figura 3.42: Corpul 1 este văzut din capăt și corpul 2 din lateral

După ce s-a ales o față de bază fețele care o mărginesc se prelungesc astfel încât fața paralelă cu fața de bază aleasă se elimină (figura 3.43).

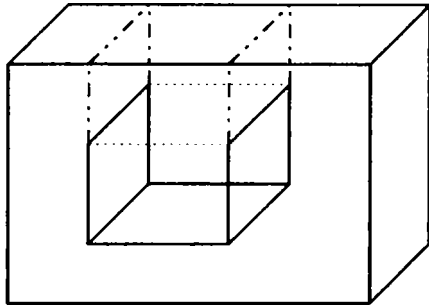


Figura 3.43: Ca față de bază s-a ales fața de jos.

3.3.1.5 Ce tip de prelucrare se va genera în corpul 1

Dacă adâncimea prelucrării în corpul 1 este diferită de 0 se va genera o prelucrare de tip 4002, 3 sau 4003, 1 în corpul 1. Adâncimea este măsurată de la fața laterală a corpului la fața de bază a prelucrării.

Chiar dacă fețele de contact nu străbat corpul 1 prelucrarea ce se va genera trebuie să-l străbată. Problema se rezolvă prelungind muchiile/muchia feței de bază ca în figura 3.44 (Lignarius, 2001) (figura 3.44).

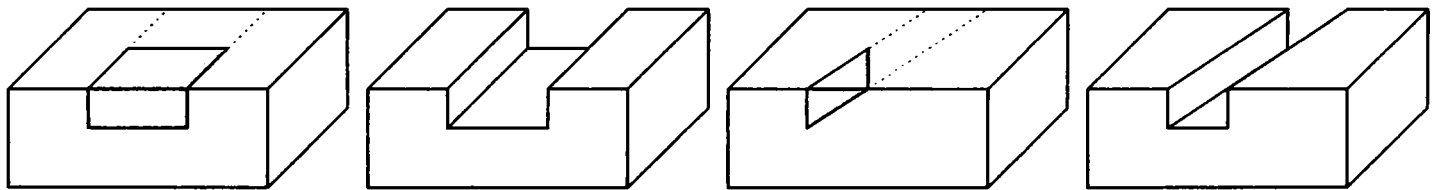


Figura 3.44: Fețele colorate sunt fețele inițiale de contact înainte de a fi prelungite

3.3.1.6 Ce tip de prelucrare se va genera în corpul 2

Dacă adâncimea în corpul 1 se alege egală cu o valoare mai mică decât cea mai mare posibilă, și în corpul 2 se va genera o prelucrare. Forma ei depinde de numărul de fețe noi care iau naștere în urma efectuării operației booleene între cele două corpuri (corpul 1 tare și corpul 2 moale). Astfel pot să existe situațiile:

- dacă cele două corpuri se ating numai într-un vârf sau numai pe o muchie în corpul 2 nu se generează nici o prelucrare
- dacă în urma operației booleene rezultă două fețe noi în corpul 2 se generează o eclisare de capăt (4002, 3), o chertare simplă dreaptă (4002, 1) sau o chertare simplă strâmbă (4002, 2)

- dacă în urma operației booleene rezultă trei fețe noi în corpul 2 se generează o eclisare dreaptă (4003, 1) sau o eclisare strâmbă (4003, 2)
- dacă în urma operației booleene rezultă mai mult de trei fețe noi în corpul 2 se alege ca față de bază cea pe care sunt perpendiculare două fețe ce o mărginesc. Acelea se vor prelungi până străpung corpul după care se continuă algoritmul ca și în cazul trei fețe de contact

Dacă eclisarea este afectată de o tăiere astfel încât o față laterală a prelucrării dispăre, ea se va transforma într-o eclisare de capăt. Două astfel de cazuri se pot vedea în figura 3.45 (Lignarius, 2001).

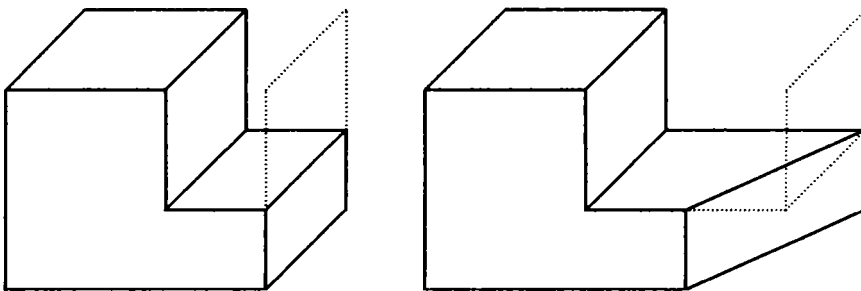


Figura 3.45: Transformarea eclisării standard în eclisare de capăt

3.3.2 Eclisare standard

Algoritmul începe cu selecția celor două corpuri. Primul este cel moale și al doilea cel tare. Se determină poziția celor două corpuri unul față de celălalt și se verifică dacă operația este posibilă. Se stabilește cazul în care se încadrează cele două corpuri și se preiau parametrii prelucrării de la utilizator prin cutia de dialog din figura 3.46:

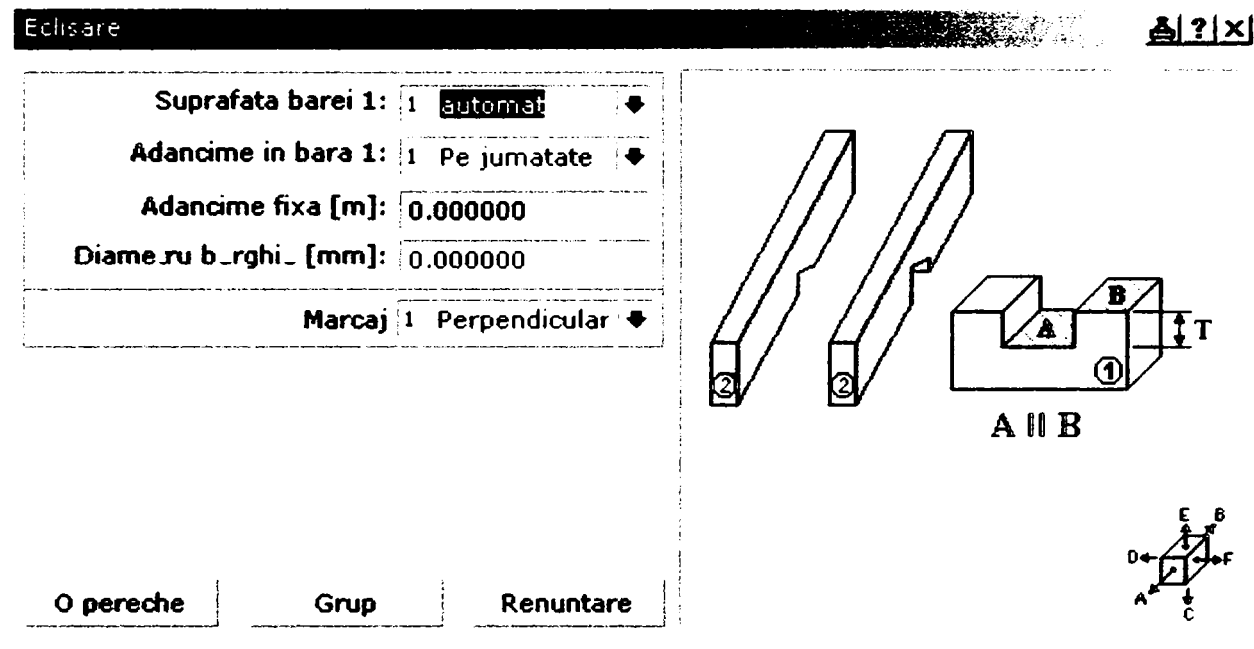


Figura 3.46: Cutia de dialog prin care se specifică parametrii eclisării standard

Utilizatorul poate influența pe ce față a corpului 1 să se efectueze prelucrarea selectând la primul parametru una din opțiunile: automat, selecție, E, C, D, F.

- la opțiunea automat se face prelucrarea pe fața care se potrivește cu corpul 2
- la opțiunea selecție se oferă posibilitatea alegerii oricărei fețe pe cale grafică
- prin E, C, D sau F se indică direct fața pe care se va efectua prelucrarea. Dacă nu se poate face prelucrarea pe fața aleasă de către utilizator, se abandonează operația și corpurile nu-și modifică forma.

Adâncimea prelucrării în corpul 1 poate fi setată pe: jumătate, selecție sau adâncime fixă.

- la opțiunea jumătate se generează automat câte o prelucrare în ambele corpuri, de aceeași adâncime, egală cu jumătate din adâncimea maximă posibilă
- la opțiunea selecție se propune utilizatorului o adâncime egală cu jumătate din valoarea maximă din care s-a scăzut valoarea minimă la care este încă posibilă

prelucrarea (aceasta poate fi și 0). În cutia de dialog de preluare a adâncimii se afișează și valorile maximă și minimă

- la opțiunea adâncime fixă se generează o prelucrare în corpul 1 de adâncime egală cu cea specificată la câmpul “adâncime fixă” și în corpul 2 o prelucrare de adâncime egală cu diferența dintre valoarea maximă posibilă calculată și adâncimea în corpul 1

Dacă la câmpul “diametru burghiu” se introduce o valoare diferită de 0 se generează o prelucrare de tip gaură (4101, 1) pe ambele corpuri. Axa găurii va fi perpendiculară pe fața de bază și va trece prin axele ambelor corpuri.

La opțiunea “marcaj” se specifică tipul de marcaje ce se vor genera pe corpul 2: perpendiculare (figura 3.48 a), de-a lungul prelucrării (figura 3.48 b), sau nici un marcaj (figura 3.47).

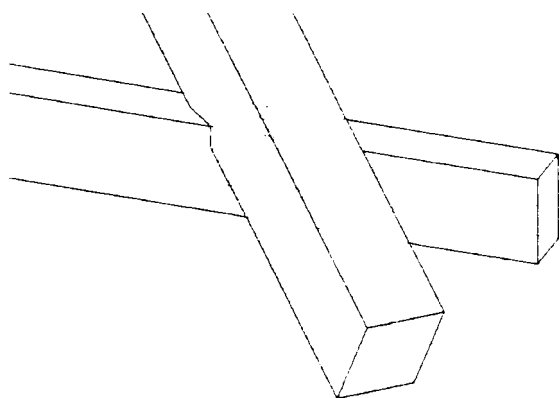


Figura 3.47: Reprezentarea cu linii ascunse a unei eclisări standard

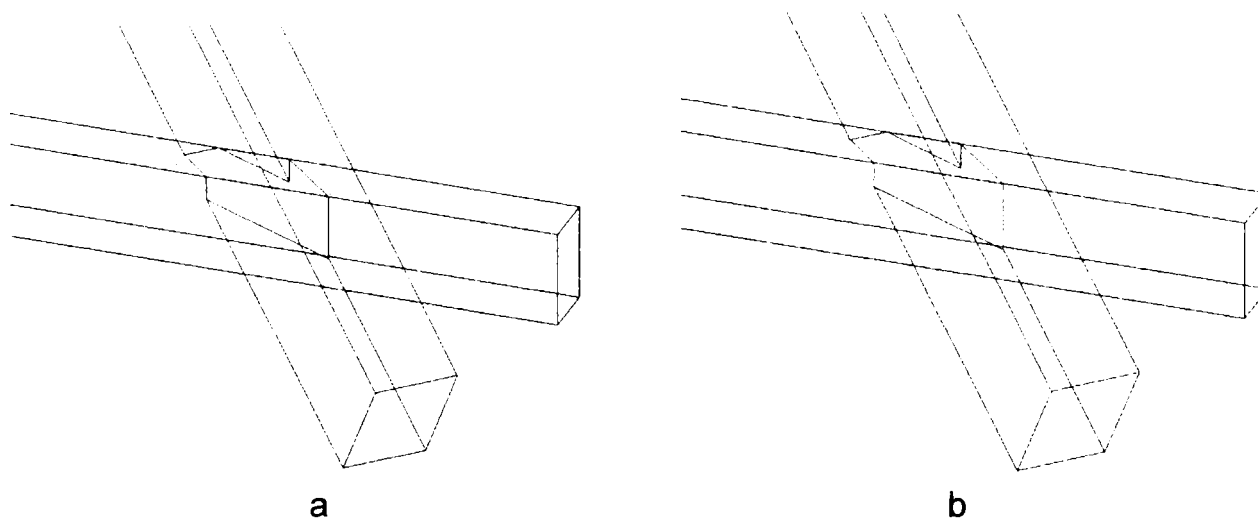


Figura 3.48: Generarea marcajelor perpendiculare (a), sau de-a lungul feței de bază (b)

La selecția butonului “O pereche” se propun utilizatorului spre selecție doar 2 corpuri. La selecția butonului “Grup” se pot selecta mai multe corpuri pe post de corp 1 și mai multe corpuri pe post de corp 2. Algoritmul încearcă să genereze câte o prelucrare între toate combinațiile posibile dintre corpurile selectate.

Algoritmul de generare a prelucrării eclisare standard poate fi descris prin următorul pseudo-cod:

- se selectează cele două corpuri și se generează forma lor brută; acestea se vor lua în calcul de acum înainte în cadrul algoritmului
- se verifică dacă operația este posibilă analizând poziția celor două forme brute; dacă ele sunt paralele sau nu se întâlnesc nici dacă vor fi prelungite operația este imposibilă și se abandonează funcția
- se intersectează cele două forme brute
- se determină numărul fețelor de contact
- se alege fața de bază din fețele de contact
- dacă fața de bază are mai mult de patru muchii se aleg două muchii care sunt paralele și nu sunt de-a lungul corpului 2; acestea se prelungesc până intersectează muchiile care sunt în lungul corpului 2; cele patru puncte de intersecție formează noua față de bază (situația poate fi observată în figura 3.44)

switch(număr fețe de contact)

{

case 2:

- se determină cazul analizând poziția celor două fețe:
 - dacă fețele au o muchie comună pot să apară cazurile 1, 2, 3 sau 4 din figura 3.38
 - se verifică dacă există o muchie dintr-o față care să fie perpendiculară pe cealaltă față.
 - DA – se generează chertare simplă dreaptă sau strâmbă pe corpul 2 și eclisare standard pe corpul 1; adâncimile prelucrărilor depind de adâncimile specificate de către utilizator în interfață
 - NU – se generează chertare simplă dreaptă sau strâmbă pe corpul 2 și nici o prelucrare pe corpul 1
 - dacă fețele sunt paralele pot să apară cazurile din figura 3.39.
 - se construiește automat fața de bază ca fiind fața care leagă cele două fețe paralele
 - se apelează funcția de la “case 3”

case 3:

- se generează prelucrări pe ambele corpuri în funcție de adâncimea specificată

case 4:

- se propune spre alegere fața de bază din cele patru; se elimină fața paralelă cu cea de bază selectată și se apelează funcția de la "case 3"

}

- în toate situațiile, după ce s-au determinat punctele de codare a prelucrărilor se verifică corectitudinea lor: să nu fie două puncte identice, planele care descriu prelucrarea trebuie să fie diferite și în unele cazuri chiar perpendiculare, punctele trebuie să fie situate pe forma brută a corpurilor etc.
- dacă prelucrările sunt greșite se abandonează operația
- dacă prelucrările sunt corecte se analizează celelalte prelucrări de pe corpuri; cele care rămân în afara corpurilor din cauza celor generate acum vor fi șterse
- dacă pe corpul 2 nu s-a generat nici o prelucrare și utilizatorul a setat tipul de marcaj dorit, se generează două marcaje ca în figurile 3.47 sau 3.48

3.3.3 Eclisare de capăt

În acest caz ambele corpuri vor fi afectate la capetele lor. Operația nu poate avea loc dacă corpurile nu au câte două fețe paralele între ele sau dacă corpurile nu se intersectează chiar dacă au fost alungite.

Parametrii prelucrării se stabilesc prin următoarea cutie de dialog:

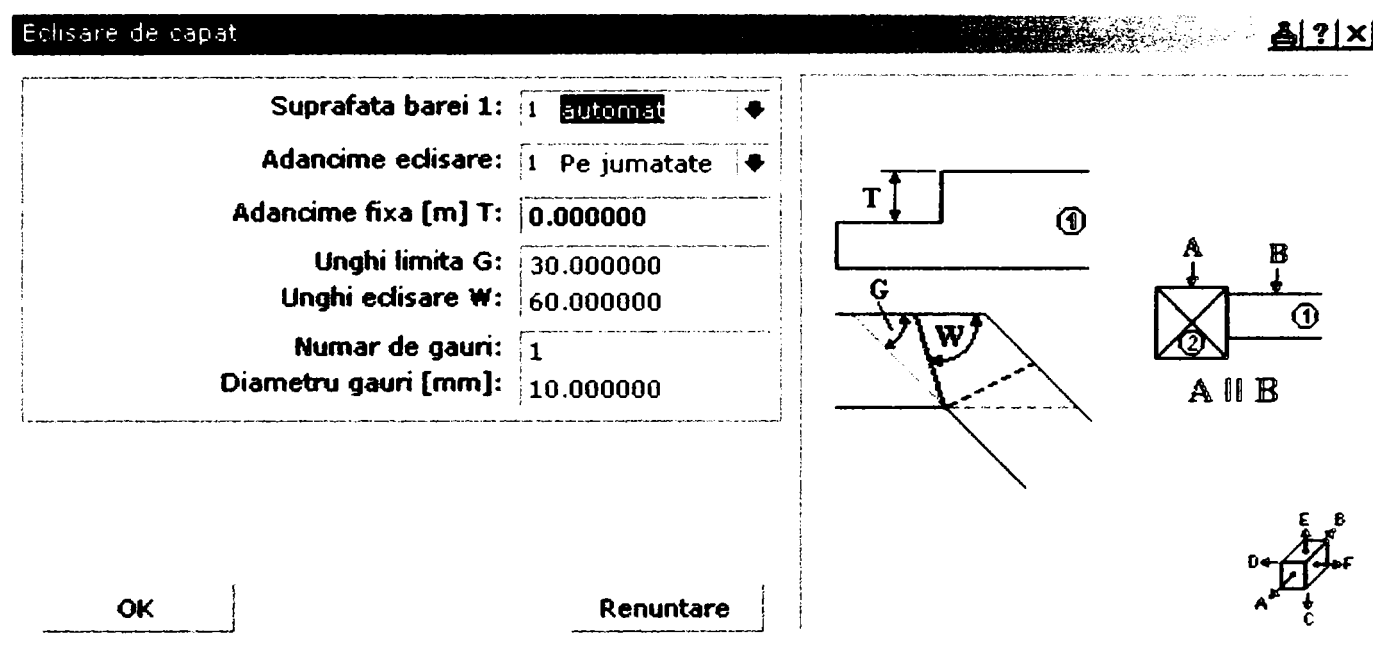


Figura 3.49: Cutia de dialog prin care se specifică parametrii eclisării de capăt

Suprafața eclisării în corpul 1, adâncime eclisare și adâncime fixă au aceeași semnificație ca și în cazul eclisării standard.

Unghiul limită este valoarea minimă acceptată de utilizator pentru unghiul dintre cele două corpuri (figura 3.50). Unghiul dintre două corpuri se definește ca fiind unghiul dintre axele celor două corpuri. Dacă axele nu sunt coplanare, se translatează una din ele în planul celeilalte (pe perpendiculara comună) și apoi se măsoară unghiul.

Dacă unghiul dintre cele două corpuri este mai mare decât 0 și mai mic decât unghiul limită, capetele corpurilor vor fi tăiate cu un plan ce formează un unghi egal cu unghiul de eclisare cu fețele laterale neparalele ale celuilalt corp

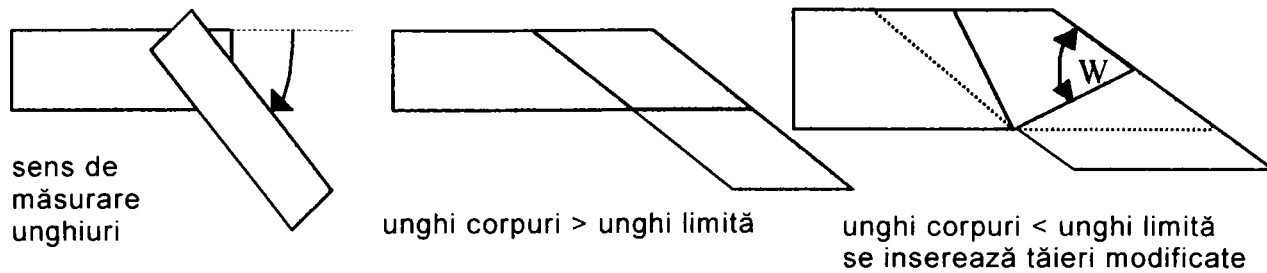


Figura 3.50: Modul de generare a prelucrării în funcție de unghiul dintre corpuri

Dacă la câmpurile “număr de găuri” și “diametru găuri” din cutia de dialog se introduc valori diferite de 0 se vor genera găuri perpendiculare pe fețele paralele ale celor două corpuri. Dacă unghiul dintre cele două corpuri este diferit de 0 se generează o singură gaură la intersecția celor două axe. Dacă unghiul este 0 se generează un număr de găuri egal cu valoarea specificată. Ele se vor repartiza pe lungimea prelucrării la distanțe egale.

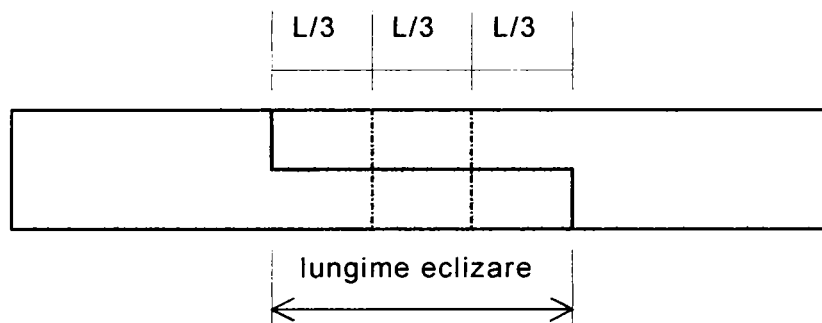


Figura 3.51: Modul de repartizare a 2 găuri pe lungimea prelucrării

Toate prelucrările de la capetele afectate se vor șterge în prealabil.

Algoritmul de generare a prelucrării eclisare de capăt poate fi descris prin următorul pseudo-cod:

- se selectează cele două corpuri și se construiesc formele lor brute; acestea vor fi folosite, în cadrul algoritmului, pentru a genera prelucrările
- se verifică dacă formele brute au câte două fețe laterale paralele și dacă ele se intersectează (chiar și prelungirile lor); dacă ele nu îndeplinesc aceste condiții, se abandonează funcția
- se determină unghiul dintre formele brute
- se intersectează formele brute
- se determină fețele de contact din care se alege fața de bază
- se calculează punctele prelucrărilor plecând de la fața de bază perpendiculară pe ea

- se verifică corectitudinea punctelor de prelucrare
 - DA se șterg prelucrările de la capetele afectate
 - NU se abandonează funcția fără să se genereze nici o prelucrare
- se generează găurile, dacă s-a specificat în cutia de dialog, pe baza unghiului dintre corpuri:
 - unghi = 0 se împarte lungimea eclisării la numărul de găuri și se obține distanța dintre găuri
 - unghi > 0 se generează o singură gaură perpendiculară pe fața de bază prin punctul de intersecție a axelor celor două corpuri.
- dacă unghiul dintre corpuri este > 0 și < unghiul de eclisare, se inserează câte o prelucrare de tip tăiere (4001, 1) la capetele afectate ale celor două corpuri cu unghiul de înclinare egal cu unghiul de eclisare

În figurile 3.52 și 3.53 se observă cum au fost afectate corpurile de prelucrarea eclisare de capăt:

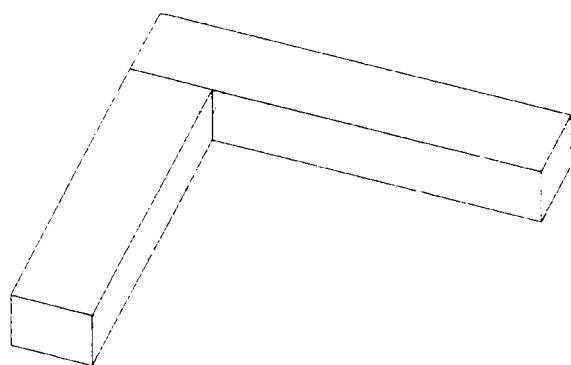


Figura 3.52: Îmbinarea prin prelucrarea de tip eclisare. În figura din dreapta este vizualizată și gaura de burghiu. Unghiul dintre corpuri este diferit de 0.

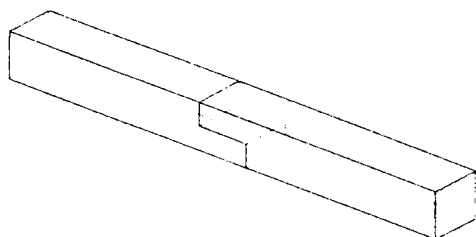


Figura 3.53: Unghiul dintre corpuri este 0. În figura din dreapta sunt vizualizate și 3 găuri de burghiu.

3.3.4 Eclisare frontală

În cadrul acestei prelucrări corpul 1 decupează prelucrarea cu capătul ei pe o față laterală a corpului 2. Parametrii prelucrării se specifică folosind cutia de dialog din figura 3.54:

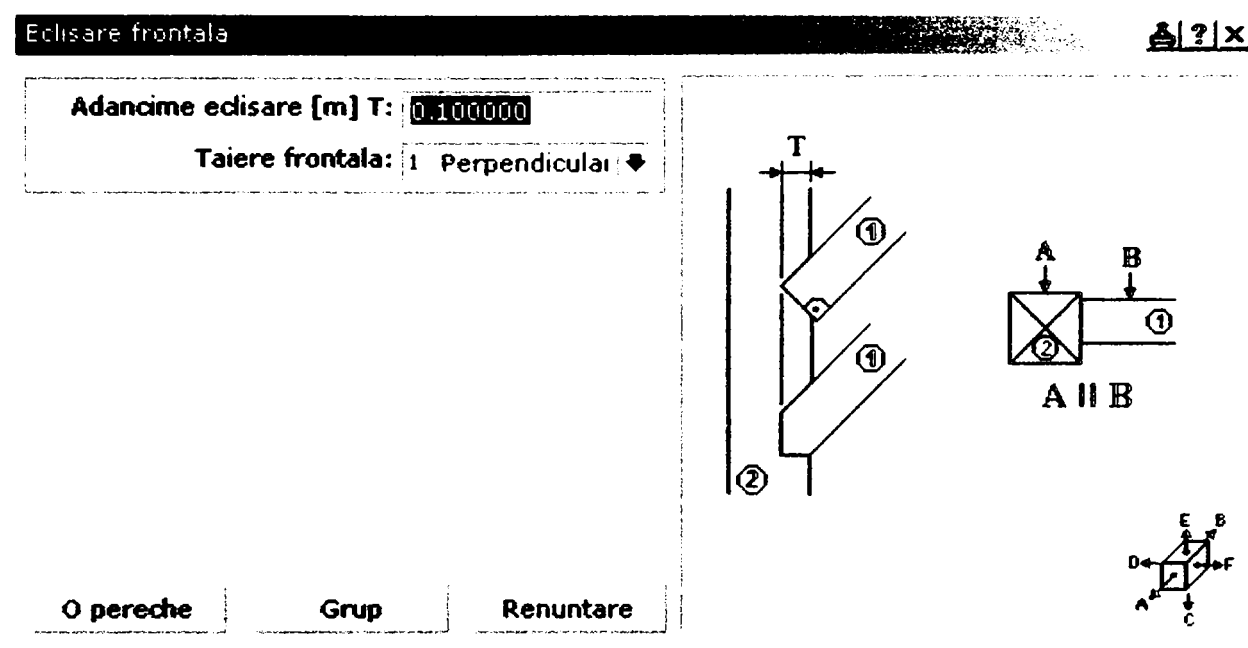


Figura 3.54: Cutia de dialog prin care se specifică parametrii eclisării frontale

Adâncimea în corpul 2 este adâncimea de pătrundere a corpului 1 în corpul 2. Tăierea frontală poate fi selectată: perpendicular sau potrivit. La opțiunea perpendicular se generează o prelucrare de tip tăiere (4001, 1) pe corpul 1 în punctul de adâncime specificată. La opțiunea potrivit tăierea care se generează va avea o înclinare astfel încât planul de tăiere să fie paralel cu fața laterală a corpului 2 care este afectată de prelucrare (figura 3.55).

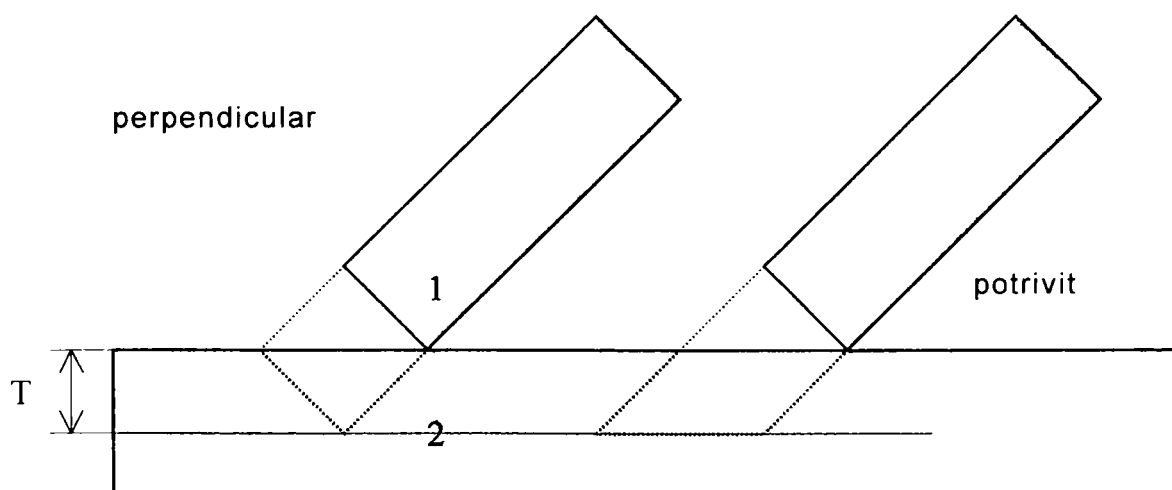


Figura 3.55: Generarea tăierii frontale pe corpul 1

După ce s-au selectat cele două corpuri, se analizează poziția lor relativă una față de cealaltă. Dacă nu au două fețe laterale paralele sau nu se intersectează nici prin prelungire, operația nu se poate efectua.

Deci, în funcție de poziția relativă a celor două corpuri, în corpul 1 se pot genera următoarele prelucrări:

- o singură prelucrare de tip tăiere de capăt (4001, 1) sau
- două prelucrări de tip tăiere de capăt ($2 * 4001, 1$)

În corpul 2 se pot genera prelucrările (figura 3.56):

- chertare simplă (4002, 1)
- eclisare de capăt (4002, 3)
- o eclisare de capăt (4002, 3) și o chertare simplă (4002, 1)
- o eclisare standard (4003, 1)
- o eclisare standard (4003, 1) și o chertare simplă (4002, 1)
- o eclisare strâmbă (4003, 2)

Corpul 2 nu are voie să fie împărțit în două bucăți după ce va fi prelucrat. De aceea, dacă corpurile se întretaie, se propune utilizatorului să selecteze care parte va rămâne după prelucrare. Prelucrările pe corpul 2 se vor genera pe toată lățimea lui indiferent de dimensiunile corpului 1.

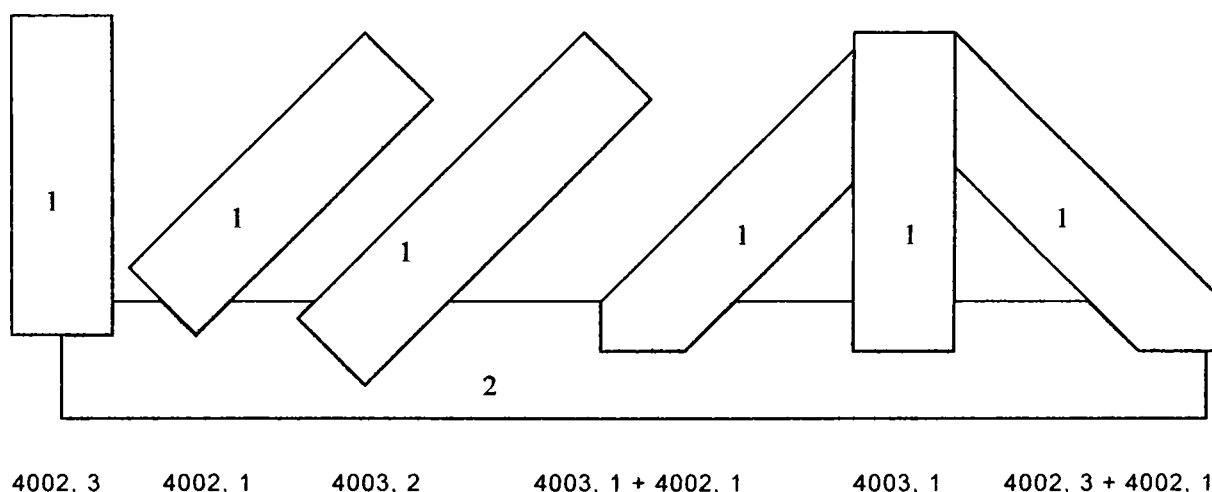


Figura 3.56: Prelucrările ce pot fi generate în corpul 2

Dacă la opțiunea "tăiere frontală" s-a selectat "potrivit", pe corpul 1 se va genera o prelucrare de tip tăiere de capăt (4001, 1) astfel încât corpul 1 să se potrivească perfect în corpul 2. Acest caz este prezentat în figura 3.57:

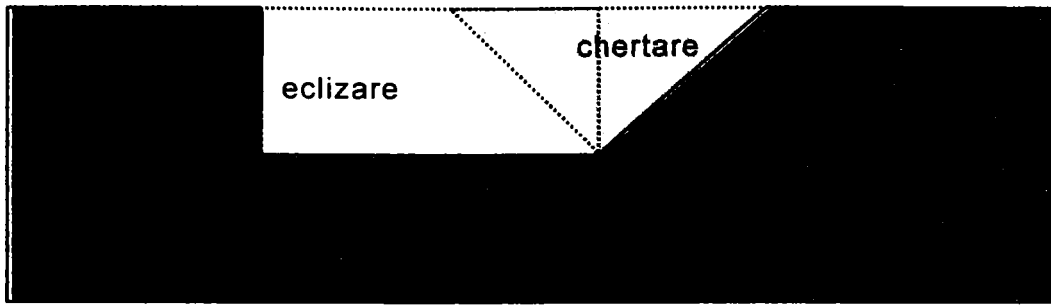


Figura 3.57: Potrivirea corpului 1 în prelucrările corpului 2 (4003, 1 și 4002, 1)

În figura 3.58 și 3.59 se poate observa cum este afectat corpul 1 în cele două cazuri: perpendicular și potrivit:

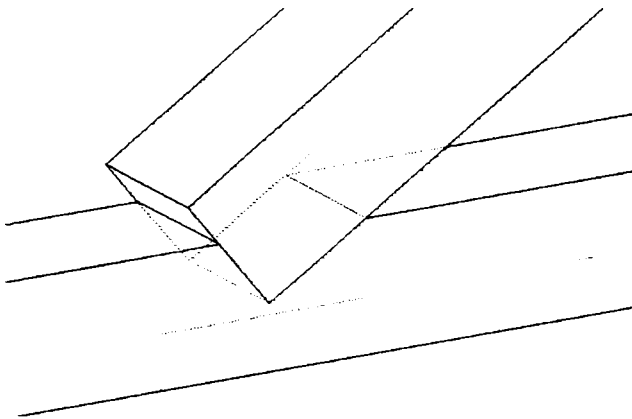


Figura 3.58: Eclisare frontală cu tăierea frontală perpendiculară

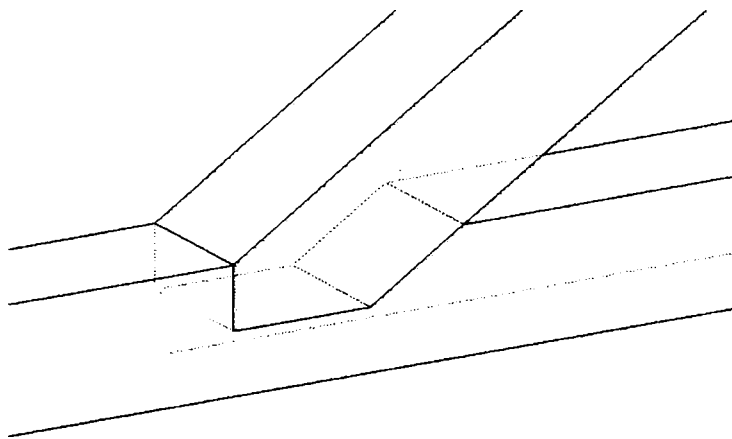


Figura 3.59: Eclisare frontală cu tăierea frontală potrivită

3.4 Generarea prelucrării tip chertare simplă

Această îmbinare este mai simplă din punct de vedere geometric și constructiv decât îmbinările prezentate în subcapitolele anterioare. În operație intră un corp tare, care contribuie la îmbinare cu o muchie, și un corp moale, pe care se va genera prelucrarea (Dietrich's Neues System, 2000).

Algoritmul poate genera două tipuri de prelucrări pe corpul moale:

- chertare simplă dreaptă
- chertare simplă strâmbă

Parametrii prelucrării se specifică prin intermediul cutiei de dialog din figura 3.60:

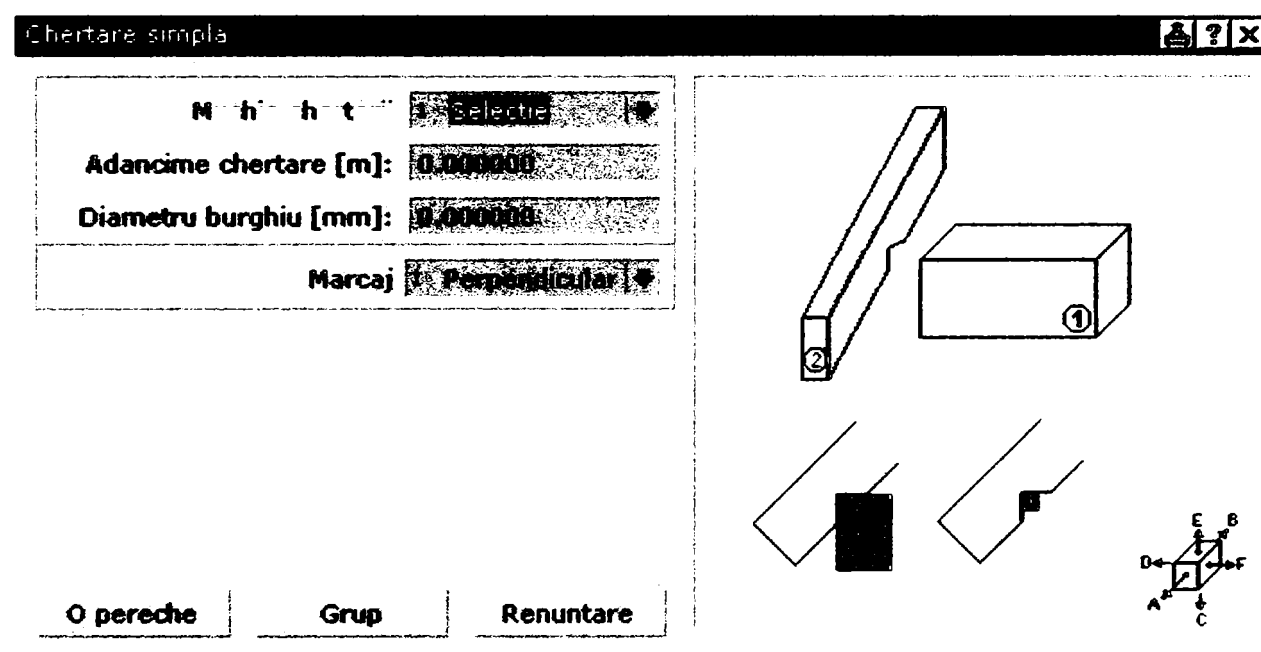


Figura 3.60: Cutia de dialog prin care se specifică parametrii chertării simple

Pentru a genera prelucrarea pe corpul moale este suficientă selecția unei muchii de pe corpul tare. Aceasta se poate face de către utilizator sau automat. Dacă utilizatorul, la opțiunea "Muchia chertării", alege "selecție", se propune utilizatorului să selecteze o muchie de pe corpul 1. Dacă s-a ales opțiunea "automat" se iau toate muchiile care sunt de-a lungul formei brute a corpului tare și se alege muchia care străpunge corpul moale cel mai adânc, adică la distanța cea mai mare față de fața laterală a corpului moale care va fi afectată de prelucrare. Planele prelucrării sunt generate pe baza poligoanelor de pe corpul 1 ce conțin muchia selectată.

La opțiunea "adâncime chertare" se specifică cât de mult trebuie să pătrundă corpul tare în corpul moale de la fața laterală a corpului moale care va fi afectată.

Dacă la “diametru burghiu” se introduce o valoare diferită de 0 se generează o gaură de burghiu pe ambele corpuri prin mijlocul muchiei chertării, având direcția bisectoarei unghiului format de planele prelucrării.

Dacă se dorește se pot genera și niște marcaje pe corpul tare, pentru a marca locul unde trebuie plasat corpul moale în cadrul construcției. Acestea pot fi perpendiculare sau de-a lungul chertării (Lignarius, 2001).

În figura 3.61 se pot observa cazurile cu marcaje de-a lungul feței de bază și marcaje perpendiculare:

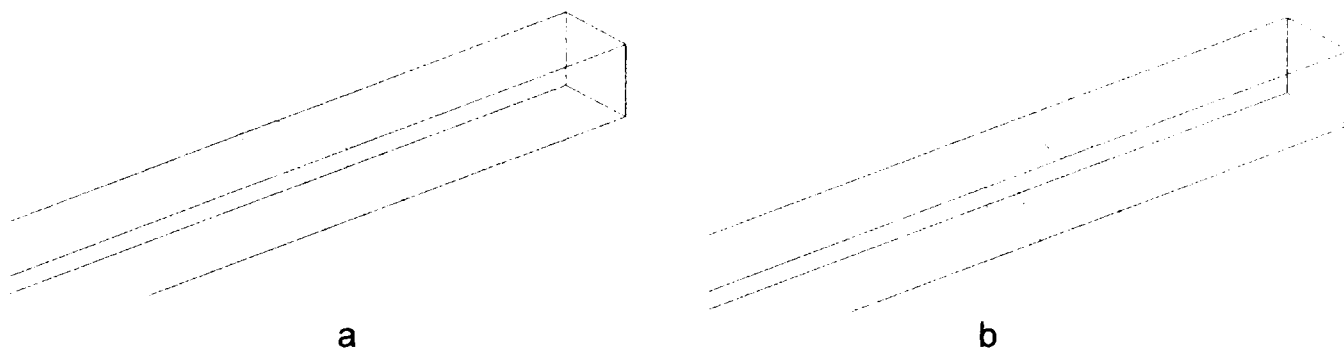


Figura 3.61: Chertare simplă cu marcaje de-a lungul feței de bază (a) și perpendiculare (b)

3.5 Generarea prelucrării tip chertare dublă

În cazul prelucrării tip chertare dublă sunt afectate ambele corpuri. Prima operație care se efectuează este verificarea posibilității efectuării prelucrării. Dacă corpurile nu interacționează nici în prelungire, operația este imposibilă. Dacă poziția relativă a corpurilor unul față de celălalt nu este corectă, adică nu există fețe laterale ale formelor lor brute care să fie paralele între ele, prelucrarea care ar rezulta nu s-ar putea realiza, deci operația este imposibilă.

Singurii parametri care se specifică sunt cele două adâncimi ale chertării prin intermediul cutiei de dialog din figura 3.62:

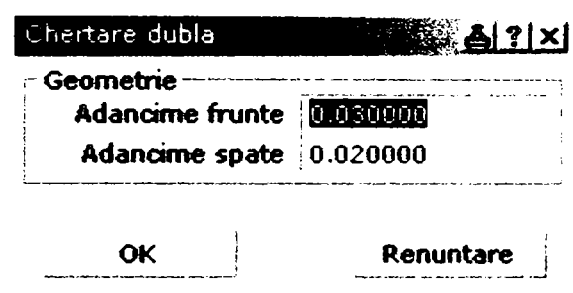


Figura 3.62: Cutia de dialog prin care se specifică parametrii chertării duble

Valorile celor două adâncimi influențează geometria prelucrării.

Dacă corpurile se întretaie astfel încât corpul tare ar fi împărțit în două bucăți distincte în urma operației, se reține partea din corpul tare pe care a fost el selectat cu mouse-ul în procesul de selecție a corpului tare.

Se construiesc formele brute ale celor două corpuri și cu acestea se continuă operația.

În continuare se determină conturul pe care se va poziționa prelucrarea prin intersectarea muchiilor formei brute a corpului tare (care a fost prelungit dacă a fost cazul), cu fața laterală a formei brute a corpului moale care va fi afectată de prelucrare. Acesta este întotdeauna un dreptunghi. Pe el se construiește prelucrarea, determinând punctele ei folosind formulele (notațiile din formule corespund figurii 3.63):

$$X_1 = X_1 \quad Y_1 = 0.0 \quad Z_1 = H_m$$

$$X_2 = X_1 + T_2 / (\text{tg}(90 - W/2)) \quad Y_2 = T_2 \quad Z_2 = H_m$$

$$X_3 = X_4 - T_1 / (\text{tg}(90 - W)) \quad Y_3 = 0.0 \quad Z_3 = H_m$$

$$X_4 = X_5 - T_1 / \operatorname{tg}(w) \quad Y_4 = T_1 \quad Z_4 = H_m$$

$$X_5 = X_1 + H_t / \sin(W) \quad Y_5 = 0.0 \quad Z_5 = H_m$$

Unghiul W este de fapt unghiul dintre cele axe ale celor două forme brute. Coordonatele x , y și z ale punctelor se calculează în sistemul local al formei brute a corpului moale. Înălțimea formei brute a corpului moale este notată cu H_m . Înălțimea formei brute a corpului tare este notată cu H_t .

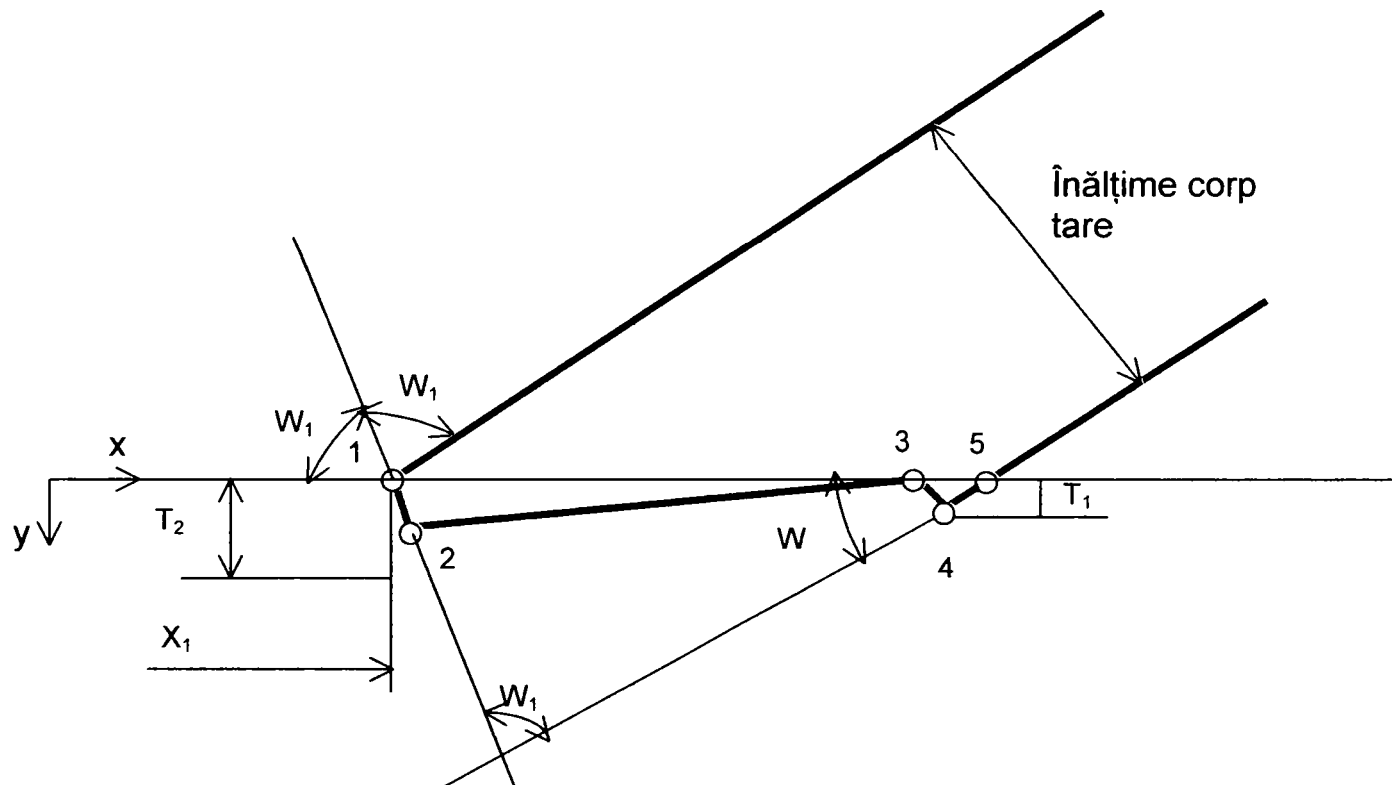


Figura 3.63: Parametrii chertării duble ce intervin în calculul punctelor ce codează prelucrarea

În figura 3.64 se prezintă cum sunt îmbinate două corpuri prin intermediul unei chertări duble.

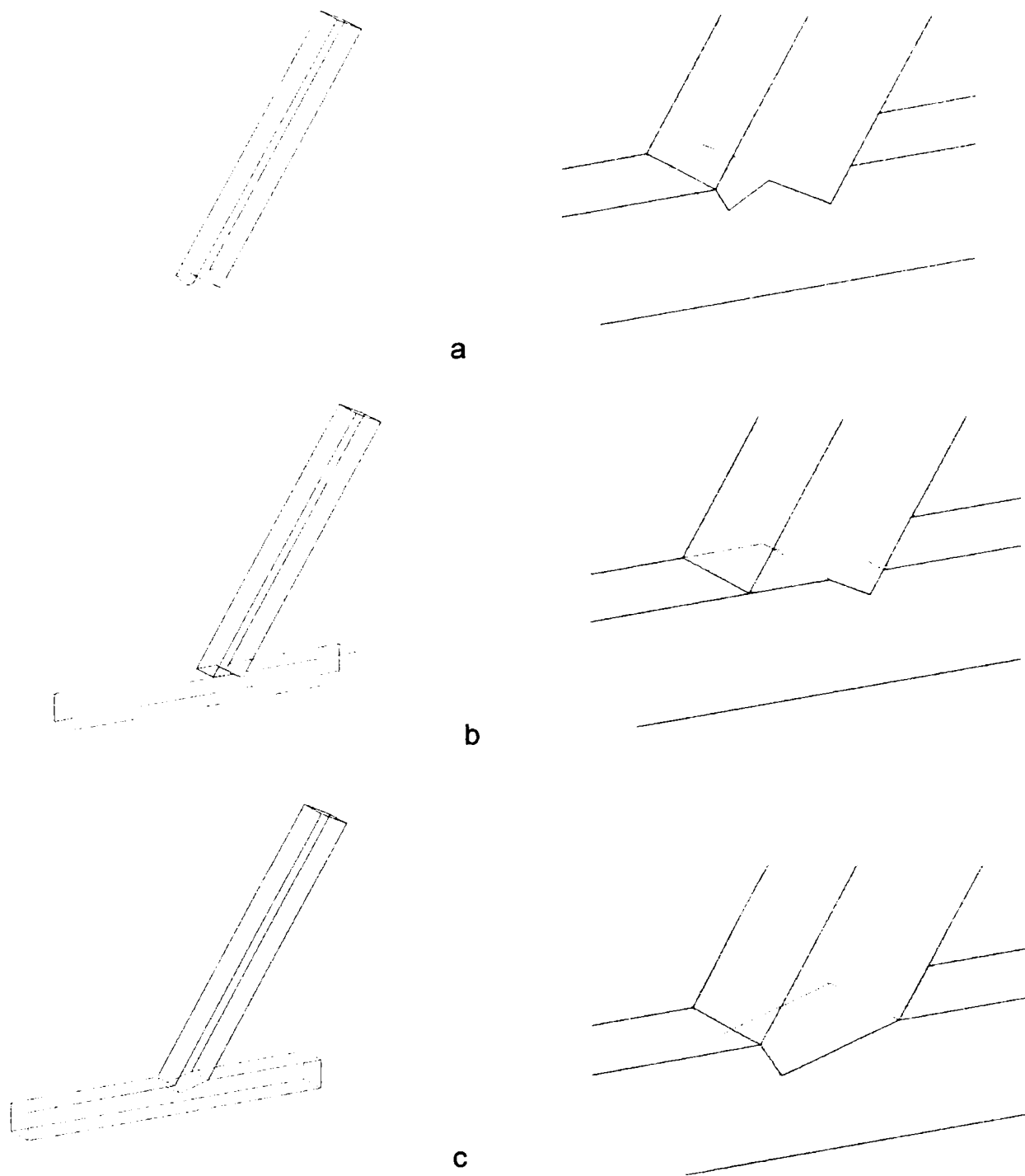


Figura 3.64: Îmbinarea a două corpuri prin intermediul unei chertări duble.

a ambele adâncimi sunt diferite de 0.

b adâncimea de frunte = 0

c adâncimea de spate = 0

3.6 Generarea prelucrării chertare de tip sapă

Prelucrarea poate fi generată numai dacă formele brute ale celor două corpuri sunt în prelungire, sunt identice în secțiune și au fețele laterale paralele. Dacă corpurile nu interacționează nici dacă sunt prelungite, operația este imposibilă.

După ce au fost selectate corpurile, se construiesc formele lor brute cu care se va continua operația. Apoi se specifică parametrii prelucrării prin intermediul cutiei de dialog din figura 3.65.

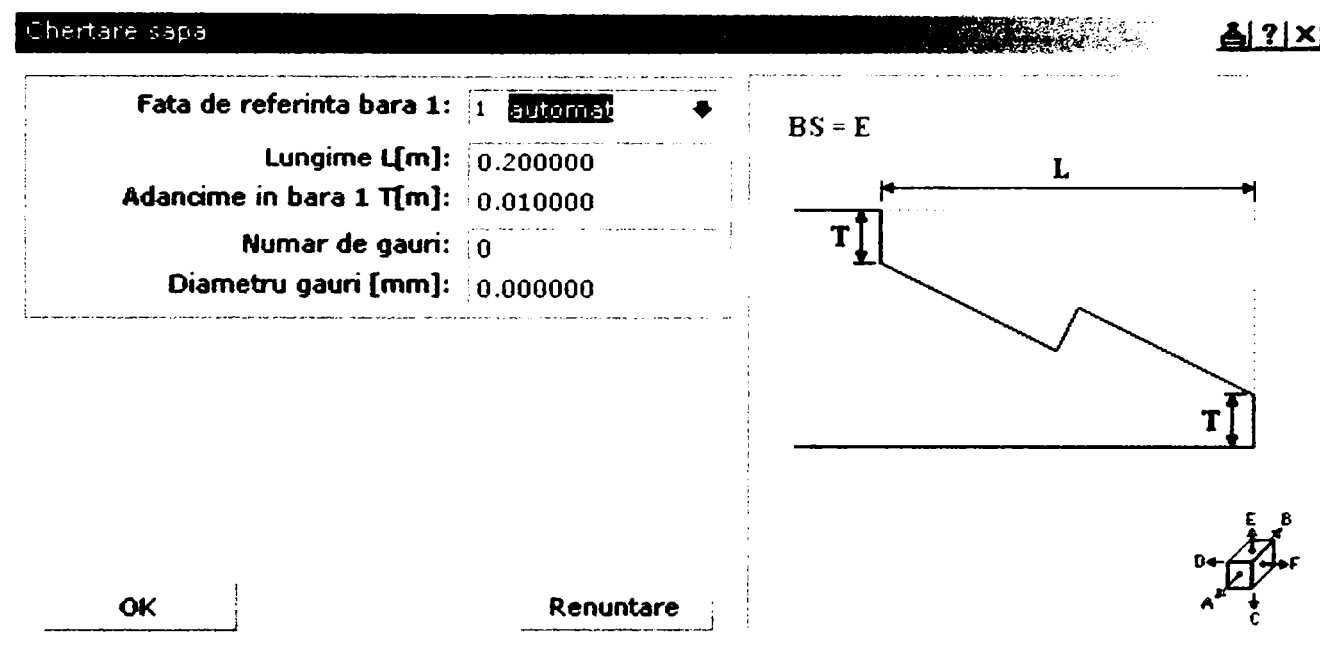


Figura 3.65: Cutia de dialog prin care se specifică parametrii chertării de tip sapă

În continuare se determină capetele celor două forme brute care vor fi afectate de îmbinare. Apoi se alungesc sau scurtează cele două forme brute astfel încât ele să se suprapună la capetele care vor fi afectate pe lungimea egală cu L. Se construiesc cele două forme parametriche ale prelucrărilor direct din parametrii dați de utilizator. Se apelează funcțiile de conversie din forma parametrică în forma în puncte.

Dacă s-au specificat numărul de găuri și diametrul lor, se vor genera pe ambele corpuri găuri perpendiculare distribuite egal pe lungimea L.

În figura 3.66 se poate vedea cum sunt afectate formele brute de îmbinarea prin chertare tip sapă:

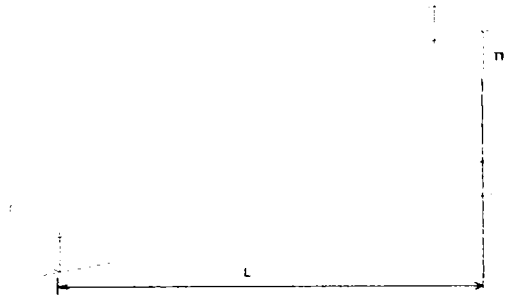


Figura 3.66: Îmbinare prin chertare tip sapă

3.7 Generarea prelucrării tip chertare “tăbăcită”

Din punct de vedere al operațiilor care se efectuează, prelucrarea este identică cu chertarea tip sapă. Cutia de dialog prin care se specifică parametrii este de forma (figura 3.67):

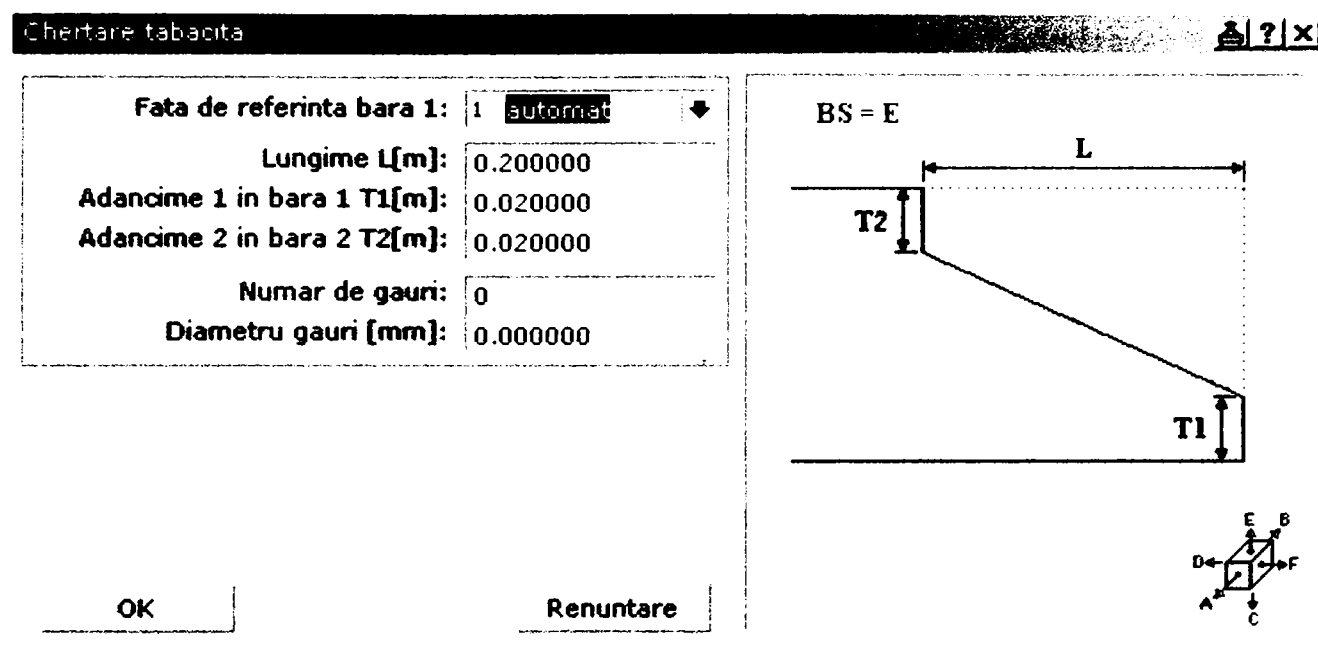


Figura 3.67: Cutia de dialog prin care se specifică parametrii chertării “tăbăcite”

Singura diferență din punct de vedere al parametrilor este existența a două adâncimi care pot să aibă valori diferite.

În figura 3.68 se pot vedea două corpuri îmbinate prin chertare “tăbăcită”.

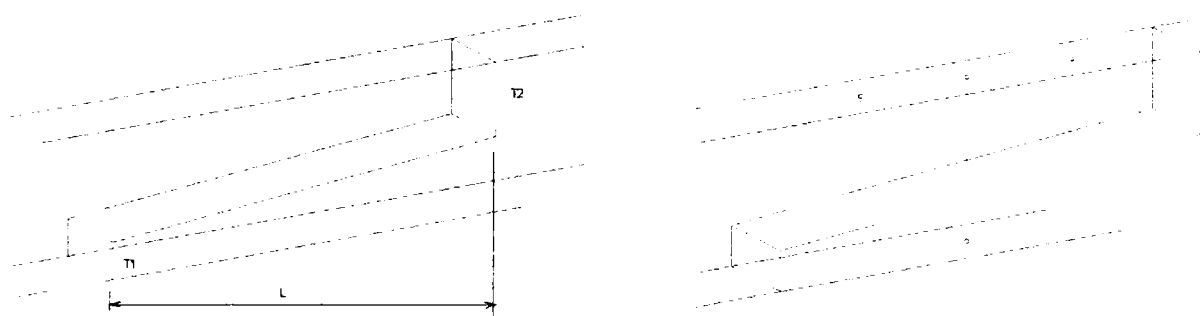


Figura 3.68: Îmbinare prin chertare “tăbăcită”

3.8 Generarea prelucrării tip decupare în V

Prelucrarea se definește prin două plane care se intersectează de-a lungul unei drepte.

Dacă planele sunt paralele sau nu se intersectează, operația nu se poate efectua. Planele pot avea orice poziție față de corp. Dacă au o poziție particulară, adică dreapta de intersecție a celor două plane este paralelă cu axa formei brute, se va genera o tăiere longitudinală dublă convexă (4002, 6), sau dacă poziția corespunde unei chertări simple, se va genera o prelucrare de tip chertare (4002,X) (Lignarius, 2000).

În figura 3.69 se pot vedea diferite tipuri de decupare în V.

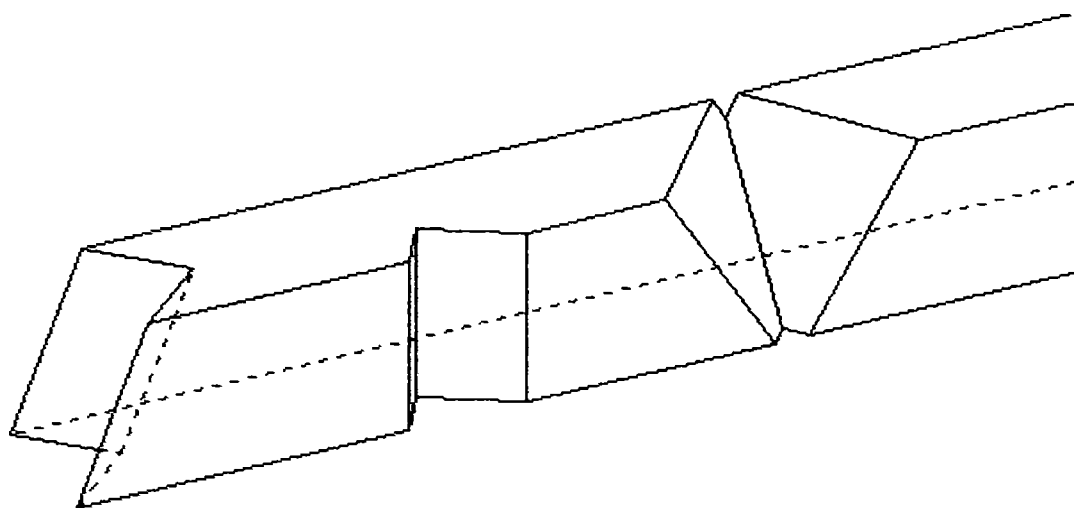


Figura 3.69: Reprezentarea decupării în V

Prelucrarea nu are voie să împartă corpul în două bucăți. Dacă prelucrarea ar afecta forma brută doar cu un singur plan, deci dreapta nu intersectează forma brută, operația nu se poate efectua.

Cele două plane ce definesc prelucrarea se vor alege prin selecția unei muchii a unui alt corp. Planele prelucrării se determină din poligoanele ce mărginesc de o parte și de alta muchia corpului. Acestea nu se iau de pe forma brută ci chiar de pe corp. Nu este nici o problemă dacă unghiul dintre cele două plane nu este 90° , dar trebuie să fie mai mic de 180° .

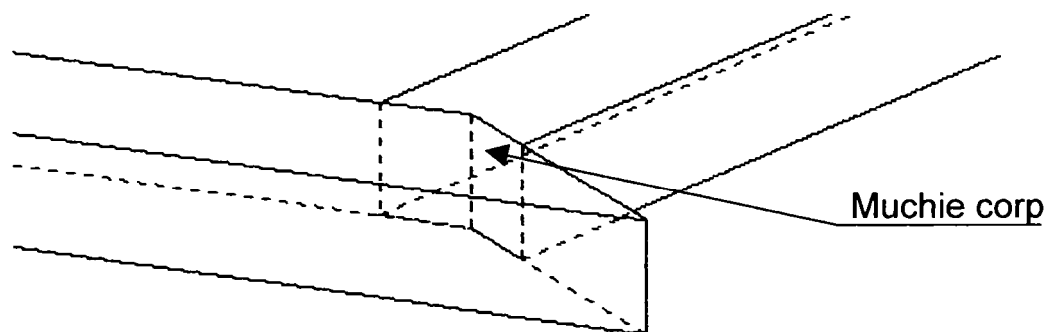


Figura 3.70: Decupare în V pe baza unei muchii a unui corp.

După ce muchia a fost selectată, se construiește forma brută a corpului în care se va genera prelucrarea. Aceasta se va intersecta cu cele două plane și se vor determina punctele prelucrării. Dacă planele au o poziție oarecare, se poate întâmpla ca unele puncte să fie în exteriorul formei brute.

3.9 Gaură de cep oarecare

În operație intervin două corpuri, unul tare și unul moale. Din cel tare se selectează o față pe baza căreia se generează în corpul moale prelucrarea. Corpul tare nu va fi afectat din punct de vedere geometric în urma operației. El intervine în operație doar cu fața care a fost selectată de pe el.

Fața selectată va deveni fața de bază a prelucrării. În funcție de poziția relativă a feței selectate față de corpul moale se vor genera diferite prelucrări: gaură de cep oarecare (4006, 3), gaură cep deschisă (4005, 2), eclisare dreaptă (4003, 1) sau o eclisare strâmbă (4003, 2).

Cutia de dialog prin intermediul căreia se specifică parametrii este de forma (figura 3.71):

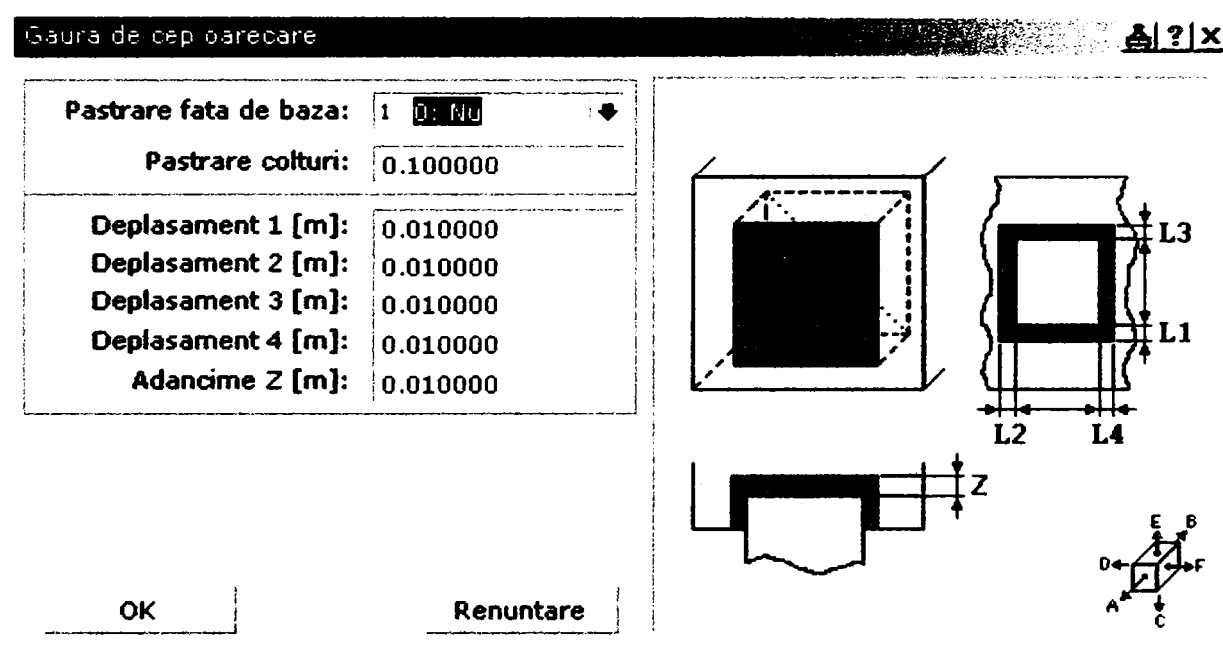


Figura 3.71: Cutia de dialog prin care se specifică parametrii găurii de cep oarecare

Primii doi parametri sunt valabili doar pentru mașină. Ei se referă la cum va fi afectată fața de bază a prelucrării și colțurile prelucrării.

La deplasamentele 1-4 se specifică valorile cu cât vor fi deplasate muchiile feței de bază finale a prelucrării care se va genera față de fața selectată de pe corpul tare (figura 3.72).

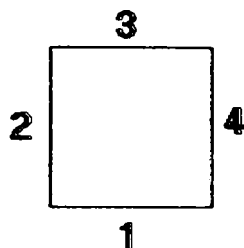


Figura 3.72: Ordinea muchiiilor feței de bază.

Dacă valorile nu sunt egale, se propun spre selecție utilizatorului pozițiile posibile ale feței de bază.

Adâncimea este deplasamentul în direcția normalei feței de bază. Se măsoară perpendicular pe fața de bază.

După ce a fost selectat corpul moale, se cere specificarea unei muchii din corpul tare. Se propun spre alegere pentru fața de bază poligoanele ce mărginesc muchia selectată.

Dacă poligonul selectat ca față de bază nu este un dreptunghi, se calculează dreptunghiul maximal al poligonului ca în figura 3.73.

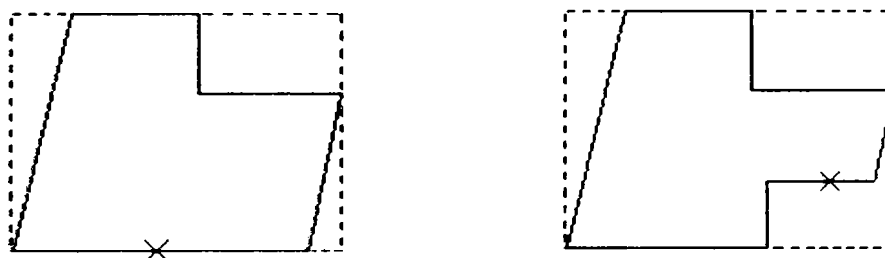


Figura 3.73: Dreptunghiul maximal este reprezentat cu linie punctată. Poligonul este reprezentat cu linie continuă. Cu x este marcată muchia din corpul tare selectată de către utilizator.

Două din laturile dreptunghiului maximal trebuie să fie paralele cu muchia selectată.

În figura 3.74 se poate vedea cum este afectat corpul moale dacă deplasamentele sunt 0 sau diferite de 0.



Figura 3.74: Gaură de cep oarecare. În stânga deplasamentele și adâncimea sunt 0. În dreapta deplasamentele și adâncimea sunt diferite de 0.

3.10 Concluzii. Contribuții personale

Toate îmbinările prezentate în acest capitol au ca scop generarea unor perechi de prelucrări pe elementele de construcție astfel încât ele să se îmbine perfect în cadrul construcției (de ex. a acoperișului).

Fiecare prelucrare, după ce a fost codată, este testată dacă este corectă din punct de vedere al șirului mașină (forma în puncte a prelucrării) și al geomeriei ei.

Înainte de a fi inserată orice prelucrare este testată cum afectează corpul, în sensul că ea poate elimina una sau mai multe prelucrări existente deja pe corp. Ea poate să elimine prelucrări prin faptul că scurtează corpul, sau poate să înghită complet o prelucrare. Testele de înghițire se fac cu ajutorul operațiilor booleene. Se construiesc negativele celor două prelucrări. Se intersectează fiecare cu forma brută. Cele două intersecții se intersectează între ele. Dacă rezultatul intersecției este identic cu corpul rezultat în urma intersecției negativului prelucrării (care se verifică dacă va fi înghițită) cu forma brută rezultă că prelucrarea de pe corp este complet inclusă în prelucrarea care se va insera deci ea va fi eliminată.

Când utilizatorul este mulțumit cu forma obținută, poate să genereze fișierul cu prelucrările mașină, care va fi transmis la roboți. Înainte de a fi scrise în fișier, unele prelucrări suferă niște modificări. Toate punctele prelucrărilor sunt recalulate astfel încât ele să fie poziționate pe forma brută.

Contribuțiile originale prezente în cadrul prezentului capitolului sunt:

- Abordarea sistematică și unitară a codării prelucrărilor mașină, ceea ce a condus inclusiv la o organizare superioară a programelor din punct de vedere structural și al eficienței.
- Implementarea algoritmilor de îmbinare a două solide prin intermediul unor prelucrări mașină. Îmbinarea presupune generarea codurilor și efectuarea lor prin intermediul operațiilor booleene.
- Implementarea interfețelor și a funcțiilor ce asigură funcționarea lor
- Implementarea unui mecanism de reconstruire a corpului pe baza prelucrărilor stocate de el
- Implementarea funcțiilor de testare a corectitudinii prelucrărilor din punct de vedere sintacticii șirului generat și al geometriei punctelor
- Implementarea funcțiilor de calcul al punctelor prelucrărilor pe baza poziției celor două corpuri

- Proiectarea și implementarea funcțiilor de transformare a unor prelucrări în altele. Sunt situațiile în care o prelucrare oarecare are o poziție particulară deci ea poate fi modelată printr-o prelucrare mai simplă. Este preferată prelucrarea mai simplă deoarece ea se va executa mai ușor de către roboți. Prelucrările mai complicate cu poziții oarecare necesită scule speciale.
- Implementarea funcției de recalculare a punctelor prelucrării în vederea transmiterii acestora la roboți.

4 ALGORITMI PENTRU OPERAȚII BOOLEENE GENERALE

4.1 Pașii parcurși în efectuarea unei operații de intersecție, diferență și reuniune

În cadrul acestui subcapitol se vor descrie operațiile ce trebuie efectuate în cazul aplicării unei operații booleene pe două corpuri. Această operație a fost implementată pentru a putea modela corpurile conform prelucrărilor efectuate pe ele. Corpurile pe care au fost generate prelucrări se consideră corpurile moi și negativele prelucrărilor sunt pe corpurile tari. Corpurile tari decupează din corpurile moi cu ajutorul operațiilor booleene. Astfel, la sfârșit corpurile moi vor avea forma finală din construcție. Astfel utilizatorul poate vedea dacă corpurile sunt corecte sau nu. Dacă nu sunt corecte, poate modifica parametrii prelucrărilor care modelează corpul și să refacă corpul. La o remodelare se uită forma veche și se remodelează forma brută conform prelucrărilor modificate.

Forma corpurilor este descrisă printr-o mulțime de suprafețe. Fiecare *suprafață* este dată printr-o mulțime de muchii și fiecare *muchie* are un vârf de început și unul de sfârșit. Suprafețele sunt plane. Aceste suprafețe vor primi pe parcursul algoritmului proprietăți ca INTERIOR și EXTERIOR. Unul din corpuri este considerat cel tare și celălalt moale. Aceasta are importanță în cazul diferenței, unde corpul tare nu suferă modificări, ci el sapă în corpul moale. În funcție de operația selectată corpurile rezultate vor conține toate suprafețele interioare în cazul intersecției, cele exterioare în cazul reuniunii și cele exterioare din corpul moale și interioare din corpul tare în cazul diferenței (Mureșan, 1994, Savii, 1994).

Primul pas din cadrul algoritmului care trebuie parcurs este pregătirea listelor de plane și linii de intersecție a acestor plane, care se vor utiliza pentru efectuarea operației cu un număr minim de erori de precizie. În lista de plane se includ toate suprafețele celor două corpuri. Lista de linii se construiește pe baza muchiilor celor două corpuri și a liniilor de tăiere dintre suprafețele celor două corpuri. Suprafețele sunt descrise printr-o normală și o distanță. Fiecare nod al listei reprezintă o suprafață și memorează toate poligoanele celor două corpuri care sunt în acea suprafață. Liniile sunt descrise printr-un punct și o direcție și ele vor stoca ulterior toate punctele de tăiere dintre muchiile celor două corpuri (Mureșan, 1994).

Al doilea pas este cel de efectuare a tuturor tăierilor dintre muchiile celor două corpuri, fiecare cu fiecare (Ogârcin Geraldine 23/1999). Astfel se completează informațiile în lista de linii. Tot în cadrul acestui pas se determină urmele lăsate de poligoane unul pe celălalt. Poligoanele care se vor intersecta vor fi neapărat din corpuri diferite. Aceste urme se pun în ambele sensuri și pe baza normalelor celor două poligoane se pot da atribute celor două sensuri de INTERIOR și EXTERIOR. Există și cazuri particulare când cele două poligoane care se intersectează fac parte din același plan. În acest caz urmelor li se dă atributul de ATINGERE (Savii, 1994).

Următorul pas este de a închide toate poligoanele mici ce pot fi închise cu algoritmul "*mersul cu prima la stânga*" pe baza muchiilor poligoanelor inițiale și a urmelor care au apărut pe ele care au generat și puncte de tăiere pe muchiile inițiale. Noile poligoane vor putea primi un atribut de INTERIOR sau EXTERIOR dacă conțin cel puțin o urmă de tip INTERIOR respectiv EXTERIOR. Dacă nu, ele vor fi declarate NEUTRE și vor primi un atribut ulterior, prin intermediul unui algoritm de determinare a poligoanelor celor mai interioare (Mureșan, 1994).

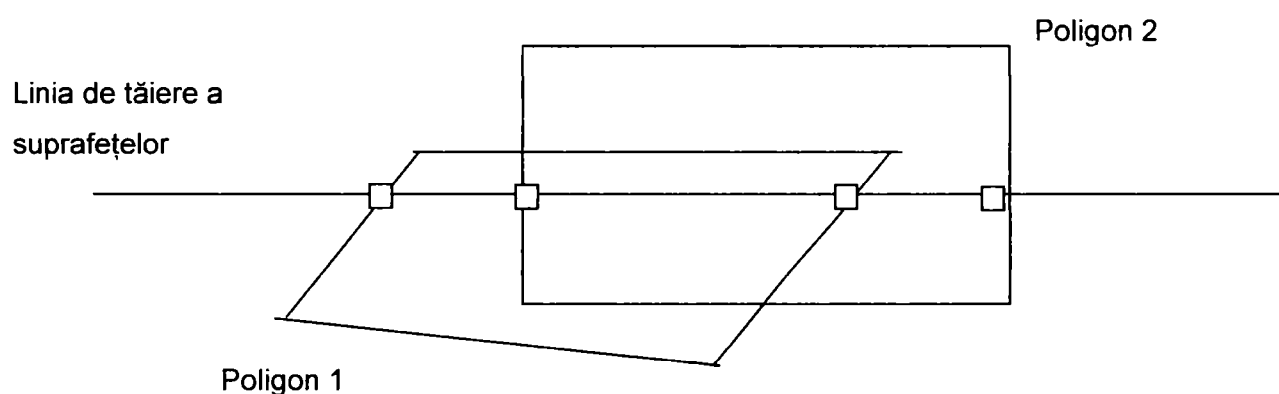


Figura 4.1: Linia și punctele de intersecție a două poligoane care nu sunt în același plan

Punctele marcate printr-un pătrat în figura 4.1 sunt punctele de tăiere dintre linia de intersecție a celor două poligoane și cele două poligoane. Primul și al treilea punct de la stânga la dreapta aparțin poligonului 1 și al doilea și al patrulea poligonului 2. În acest caz, se pun urme pe ambele poligoane între punctele 2 și 3. Urmele pe poligoane trebuie în final să permită închiderea poligoanelor mici (figura 4.2) (Mureșan, 1994).

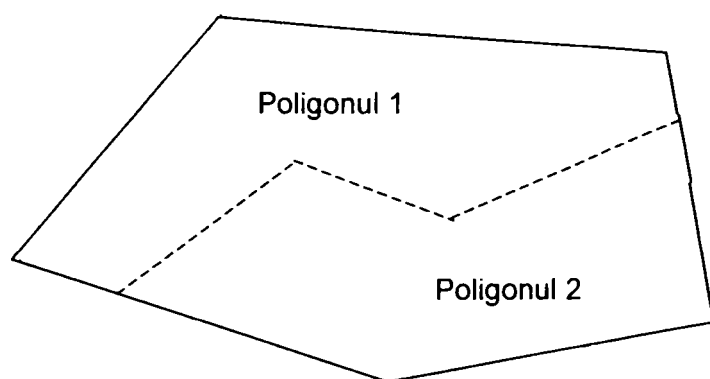


Figura 4.2: Împărțirea unui poligon în două de către urmele lăsate de alte poligoane pe el. Urmele sunt reprezentate punctat.

Urmele nu pot fi de forma celor din figura 4.3. În cadrul poligonului din figura 4.3 nu se pot închide poligoanele mici. A treia urmă poate lipsi din cauza unor erori de precizie. În aceste cazuri nu se poate efectua operația. Cele două corpuri rămân nemodificate (Savii, 1994).

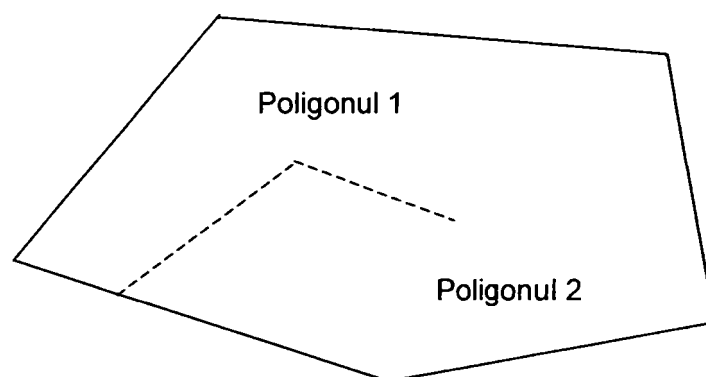


Figura 4.3: Poligon cu urme prea putine.

Un poligon poate avea mai multe urme astfel încât numărul de poligoane mici care rezultă să fie mai mare decât doi, ca în figura 4.4.

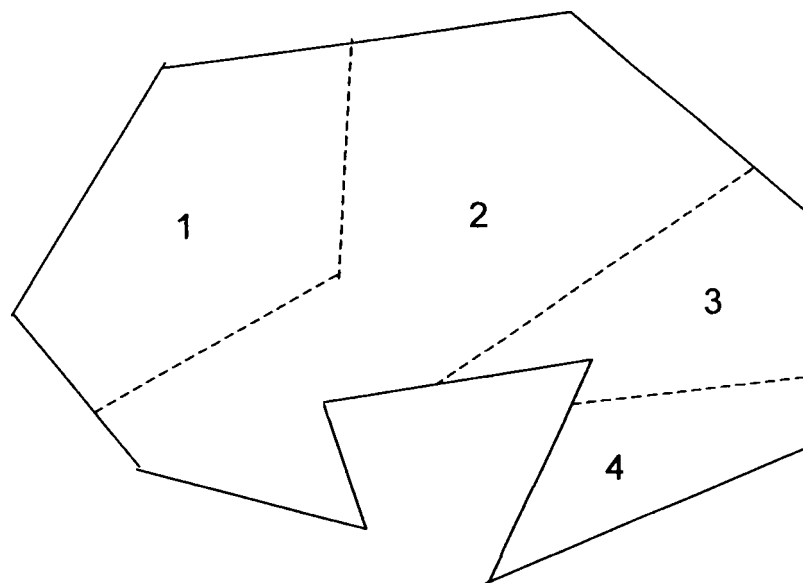


Figura 4.4: Poligon care se împarte în patru poligoane mici. Primul are cinci laturi, al doilea nouă laturi, al treilea șase laturi și al patrulea patru laturi.

După ce au fost închise toate poligoanele mici, se reconstruiește informația de pe liniile de tăiere a suprafețelor cu noile poligoane. Pe baza lor se determina tipul poligoanelor cu atributul actual NEUTRU. Pe o linie pot să existe porțiuni unde se întâlnesc doar două poligoane și altele unde se întâlnesc mai multe (figura 4.5) (Mureșan, 1994).

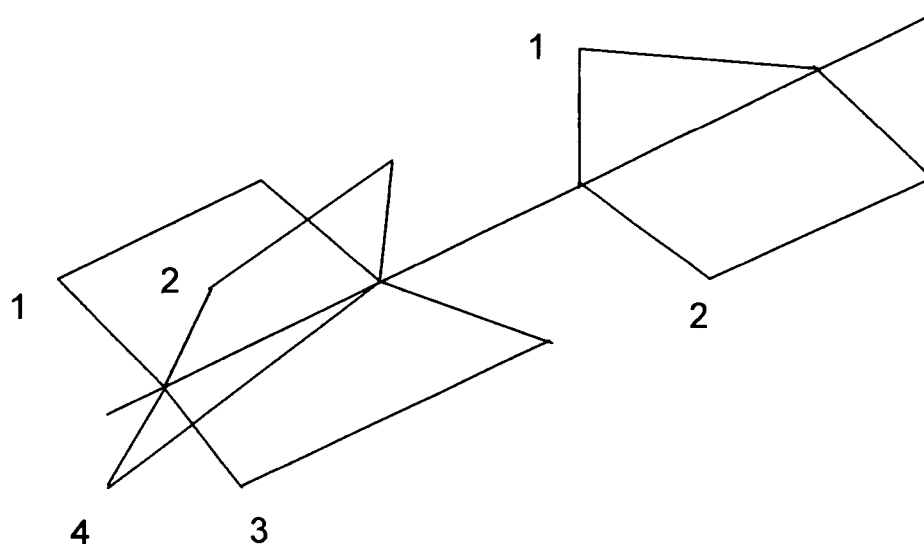


Figura 4.5: Linie de intersecție a două suprafețe cu poligoanele mici care se sprijină pe ea.

Pe baza poziției celor patru poligoane și al informației din ce corp fac parte se poate determina atributul lor (Mureșan, 1994).

În continuare mai există situații în care nu toate poligoanele au un atribut de INTERIOR sau EXTERIOR. Un exemplu este cazul în care unul din corpuri este complet inclus în celălalt, ca în figura 4.6.

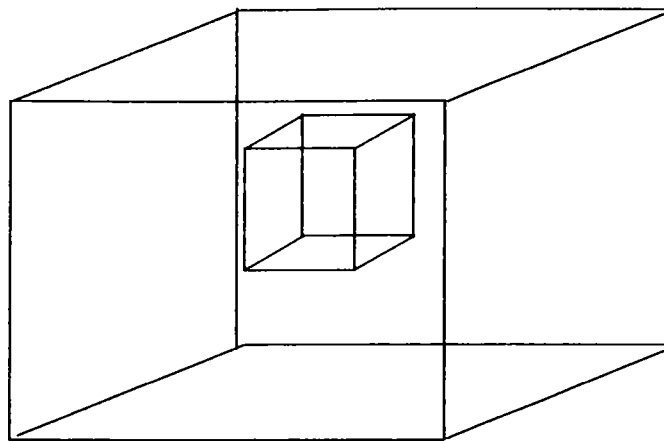


Figura 4.6: Cazul unui corp inclus complet în celălalt.

În acest caz, dacă corpul mic este cel tare, în cazul unei diferențe el produce o cavitate în cel mare. În cazul unei reuniuni se obține corpul mare și în cazul intersecției corpul mic (Mureșan, 1994, Savii, 1994).

Pentru a determina tipul poligoanelor, se apelează algoritmul de determinare a poziției unui punct față de un corp. Acest algoritm va fi descris mai jos.

După ce toate poligoanele mici au atribut, în cazul diferenței se inversează sensul poligoanelor cu atributul INTERIOR care provin din corpul tare. Apoi se aleg poligoanele utile pe baza operației dorite astfel:

INTERSECȚIE: toate poligoanele cu atribut INTERIOR

REUNIUNE: toate poligoanele cu atribut EXTERIOR

DIFERENȚĂ: poligoanele cu atribut INTERIOR din corpul tare și cele cu atribut EXTERIOR din corpul moale (Mureșan, 1994).

În continuare se construiesc liniile de separație dintre cele două corpuri pe baza liniilor de tăiere și atingere. Se alege o linie de tăiere oarecare și se încearcă închiderea spațială a acesteia mergând pe linii de același tip. Pe baza acestor linii se va putea determina numărul final de corpuri.

Următorul pas este comasarea, dacă este posibilă, a poligoanelor care sunt în același plan și au cel puțin o muchie comună. Datorită acestei comasări pot să apară muchii consecutive care să fie în prelungire. În aceste situații se elimină punctele interioare și din două muchii se formează una singură (Mureșan, 1994).

Ultima operație ce trebuie efectuată este construirea noilor corpuri. În acest scop se folosesc liniile de separație determinate la un pas anterior. Propagarea poligoanelor începe de la o asemenea linie de separație. Pentru a determina ce poligon la care corp aparține, se propagă un anumit număr poligoanelor învecinate cu linia și a poligoanelor vecine cu acestea până când propagarea dă numai de poligoane care au deja un număr.

Se mărește numărul cu unu și se continuă propagarea plecând de la o altă linie de separație. Procesul se termină când toate poligoanele au un număr. Poligoanele care au același număr sau numerele lor aparțin aceleiași clase fac parte din același corp. Clasele de echivalență se construiesc pe baza propagării când se învecinează două poligoane cu numere diferite. Atunci cele două numere fac parte din aceeași clasă (Mureșan, 1994, Savii, 1994). Numărul de clase reprezintă numărul de copuri finale. Un exemplu când rezultă mai multe corpuri se prezintă în figura 4.7.

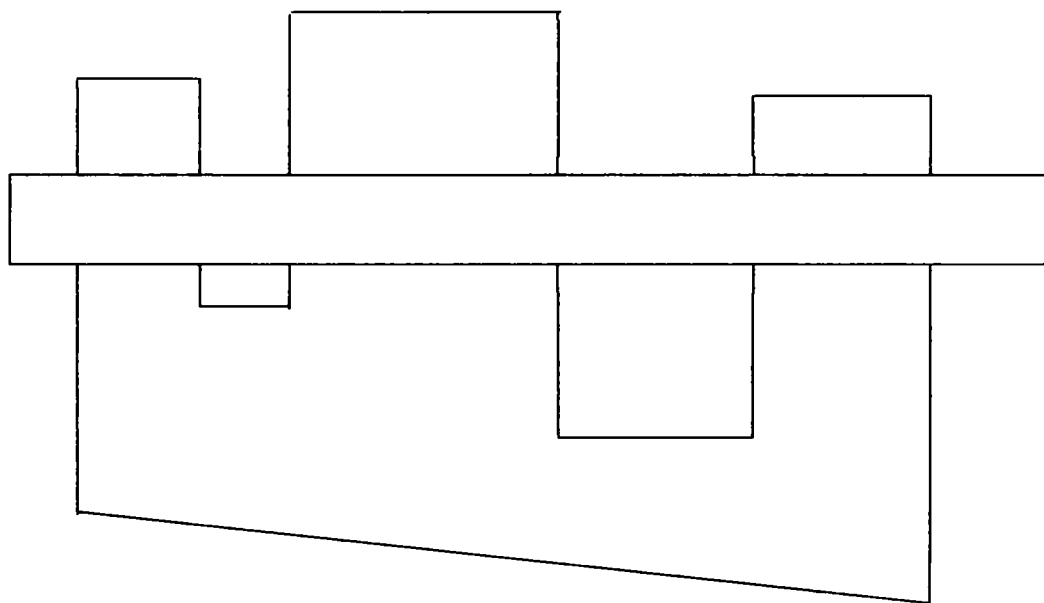


Figura 4.7: Cazul în care rezultă mai multe corpuri.

Corpurile au fost reprezentate în proiecție. Dacă corpul paralelipedic este cel tare și cel în formă de pieptene este cel moale în cazul unei diferențe se vor obține patru corpuri, trei paralelipedice și unul în formă de pieptene. În cazul reuniunii rezultă un singur corp complicat și în cazul intersecției trei corpuri paralelipedice.

În concluzie algoritmul de efectuare a operațiilor booleene se poate descrie cu ajutorul unui pseudo-cod astfel:

- se selectează corpul tare și cel moale
- dacă corpurile nu se intersectează
se părăsește algoritmul
- dacă corpurile au zone comune
{
 - se completează structura de plane ce conțin toate planele existente în cele două corpuri
 - se completează structura de linii obținute din intersecția planelor

- se intersectează toate poligoanele cu liniile
- {
 - se pun linii de tăiere sau atingere pe zonele comune a două poligoane
 - dacă a eșuat închiderea vreunui poligon
 - se părăsește algoritmul}
- se închid poligoanele mici
- se dă tip poligoanelor mici cu atributul NEUTRU
- se aleg poligoanele corecte în funcție de operație
- se determină liniile de separație pe baza liniilor de tăiere și atingere
- se comasează poligoanele care aparțin aceluiaș plan și sunt învecinate
- se elimină vârfurile care sunt mărginite de două muchii în prelungire
- se construiesc noile corpuri pe baza liniilor de tăieri
- }

4.2 Algoritmi matematici utilizați în operații booleene

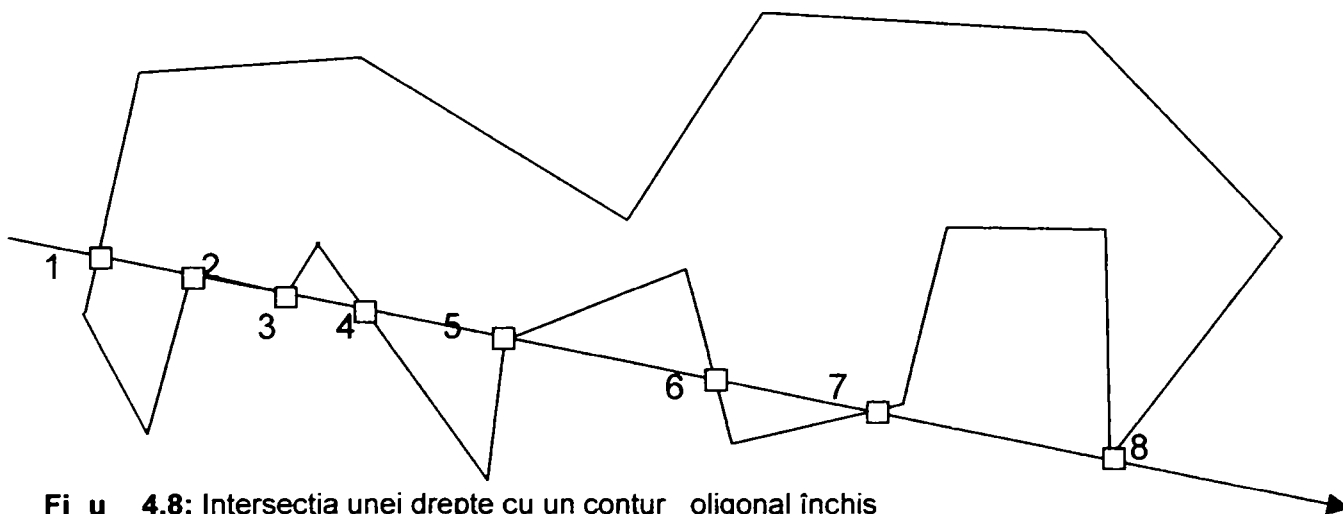
4.2.1 Intersecția unei drepte cu un contur poligonal închis

Dreapta care va tăia conturul are atașat un sens. Conturul poate fi concav.

Prima verificare care trebuie făcută înainte de orice este dacă dreapta aparține planului din care face parte conturul poligonal (Savii, 1994). Pentru a determina toate punctele de tăiere a dreptei cu conturul se va verifica dacă dreapta este suficient de lungă încât să taie întregul contur poligonal închis. Dacă nu este suficient de lungă ea se prelungeste.

Următorul pas este de intersectare a dreptei cu fiecare muchie a conturului. Dacă se constată că dreapta intersectează o muchie chiar printr-un vârf al ei se sare peste muchia următoare. Odată cu determinarea punctului de intersecție se determină și atributul punctului, adică dacă acesta este punct de intrare sau de ieșire (Mureșan, 1994).

În funcție de atributul punctelor se poate determina care sunt segmentele care sunt în interiorul, în exteriorul conturului sau pe contur (Savii, 1994). Punctul are, pe lângă informația că este de intrare sau de ieșire, și atributul de suprapunere. Adică se poate ști dacă este punct de intrare/ieșire propriu-zis sau este intrare/ieșire de suprapunere pe o muchie a conturului. Un exemplu este prezentat în figura 4.8.



Fi_4.8: Intersecția unei drepte cu un contur , oligonal închis

Punctele 1, 4, 6 au atribut de intrare propriu-zisă în contur. Punctul 7 este punct de ieșire propriu-zis. Punctul 2 este o intrare pe o suprapunere și 3 este ieșirea de pe suprapunere. Punctul 5 este ieșire printr-un vârf al poligonului. Punctul 8 atinge exact

vârful conturului deci nu se consideră nici intrare nici ieșire. Pe baza acestor atribute se pot determina segmentele de pe dreaptă care sunt în interiorul conturului: 1-2, 4-5, 6-7.

Algorimul poate fi descris cu ajutorul următorului pseudo-cod:

```

- se determină coordonata de ordonare
  se ia coordonata pe care linia variaza cel mai mult
- se parcurge poligonul muchie cu muchie
{
  - se intersectează muchia curentă cu linia infinită
  - dacă punctul de tăiere este în interiorul muchiei curente
  {
    - se memorează punctul ordonat pe linie împreună cu atributul lui de IN sau
      OUT
    atributul se determină astfel:
    {
      pr_vectorial = vector muchie x vector linie
      pr_mixt = pr_vectorial • normala poligonului
      - dacă pr_mixt = 0
      {
        // punctul poate fi o intrare/ieșire pe/de pe o muchie
        pr_vectorial_1 = vector muchie anterioară x vector linie
        pr_mixt_1 = pr_vectorial_1 • normala poligonului
        - dacă pr_mixt_1 > 0
          punctul este o intrare pe o suprapunere IN, ON
        - dacă pr_mixt_1 < 0
          punctul este o ieșire de pe o suprapunere OUT, ON
        }
      - dacă pr_mixt > 0
        punctul este o intrare
      - dacă pr_mixt < 0
        punctul este o ieșire
    }
  }
}

```

4.2.2 Test de interioritate a unui punct într-un contur poligonal închis

Algoritmul începe cu verificarea dacă punctul este în același plan cu conturul față de care se va verifica poziția lui. Apoi se verifică dacă nu cumva el este identic cu vreun punct al conturului, sau dacă nu cumva este pe o muchie a conturului. Dacă da, se iese din algoritm cu atributul ON. Dacă el nu este pe contur se trasează o semidreaptă oarecare din punct spre orice direcție astfel încât ea să treacă dincolo de cel mai îndepărtat punct al conturului. Se verifică ca nu cumva această direcție să treacă printr-un punct al conturului (Mureșan, 1994, Savii, 1994) (Ogârcin Geraldine 14/1999).

Următorul pas este de a determina numărul de intersecții ale acestei semidrepte cu conturul poligonal. Dacă numărul este impar punctul este în interiorul conturului. Dacă numărul este par, punctul este în exteriorul conturului.

Un exemplu este prezentat în figura 4.9. Numărul de intersecții este de șapte deci impar, în concluzie conform algoritmului punctul este în interiorul conturului.

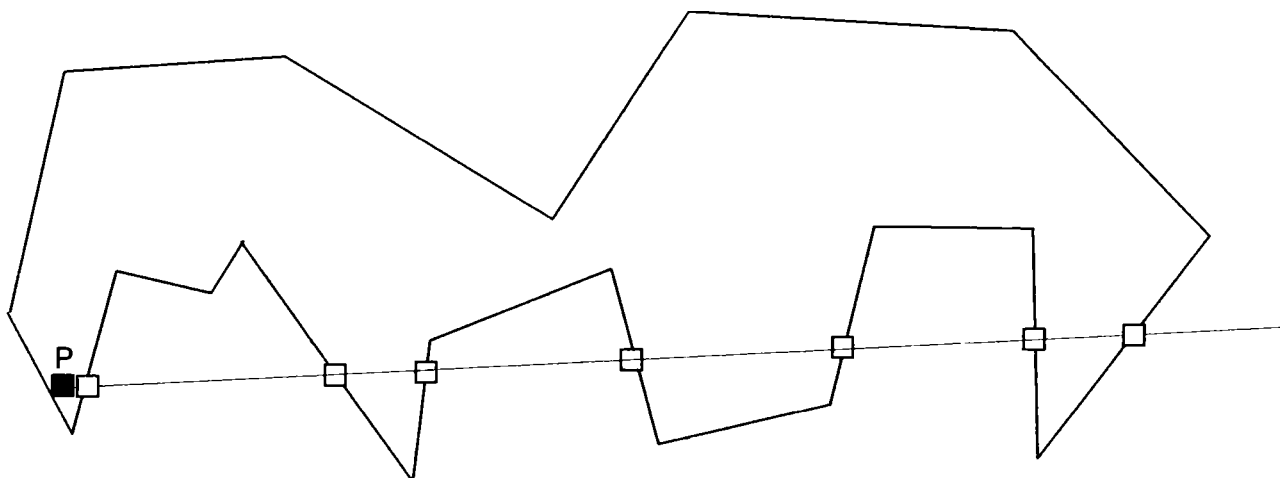


Figura 4.9: Determinarea poziției unui punct P față de un contur poligonal închis

Algoritmul poate fi descris prin următorul pseudo-cod:

```
nr_taieri = 0
```

- se determină muchia poligonului care este cea mai îndepărtată de punct
- se duce o semidreaptă din punct prin mijlocul muchiei determinate anterior
- se parcurge poligonul muchie cu muchie
- {
- dacă punctul este identic cu începutul sau cu sfârșitul muchiei curente
- return ON

```
- dacă punctul este pe muchia curentă
    return ON
- se taie semidreapta cu muchia curentă
- dacă punctul de tăiere este în interiorul muchiei
    nr_taieri ++
}
- dacă nr_taieri este par
    return OUT
- dacă nr_taieri este impar
    return IN
```

4.2.3 Calculul normalei unui contur poligonal închis indiferent de forma lui

Pentru a putea determina sensul normalei unui contur, prima condiție este ca muchiile lui să aibă un sens definit, astfel încât sfârșitul uneia să coincidă cu începutul următoarei muchii. Se ia la întâmplare orice muchie. Se determină mijlocul ei. Se determină direcția normalei conturului prin efectuarea produsului vectorial dintre două muchii consecutive. Se construiește vectorul perpendicular pe muchia aleasă făcând, produsul vectorial dintre vectorul muchiei și direcția normalei. Se calculează versorul acestui vector și se înmulțește cu de zece ori precizia (Mureșan, 1994). Cu ajutorul vectorului obținut se determină un punct P adunându-l la punctul de mijloc al muchiei alese (Savii, 1994). Se determină dacă punctul P este în interiorul sau exteriorul conturului. Pe baza informației de interioritate se determină sensul normalei conturului. Distanța până la origine se determină înlocuind orice punct al conturului în ecuația planului. Un exemplu se prezintă în figura 4.10.

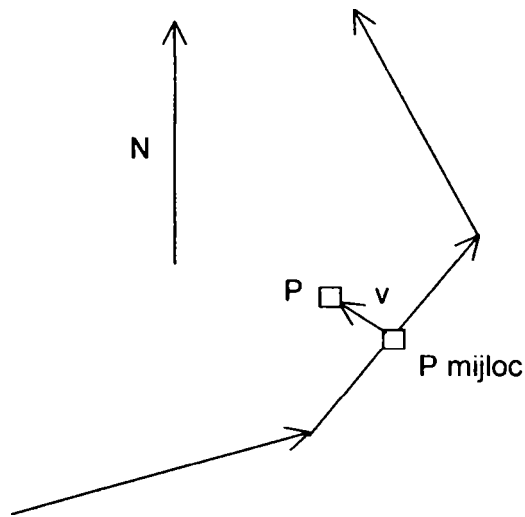


Figura 4.10: Determinarea normalei unui contur poligonal

N este vectorul normal al conturului, v este vectorul perpendicular pe muchia aleasă, P mijloc este punctul de mijloc al muchiei și P este punctul a cărui poziție se determină. Dacă P se află în interiorul conturului atunci sensul dat de produsul vectorial a două muchii ale conturului este același cu al normalei alese la început. Dacă P este în exterior sensul normalei este invers.

Algoritmul poate fi descris prin următorul pseudo-cod:

$N = \text{vector muchia 1 poligon} \times \text{vector muchia 2 poligon}$

- se determină mijlocul muchiei 1 (P mijloc)

$v = N \times \text{vector muchie 1}$

- se calculează versorul lui v

v se înmulțește cu $10 * \text{precizia}$

$P = P_{\text{mijloc}} + v$

- se determină poziția lui P

- dacă P este OUT

- se inversează N

return N

4.2.4 Intersecția a două segmente

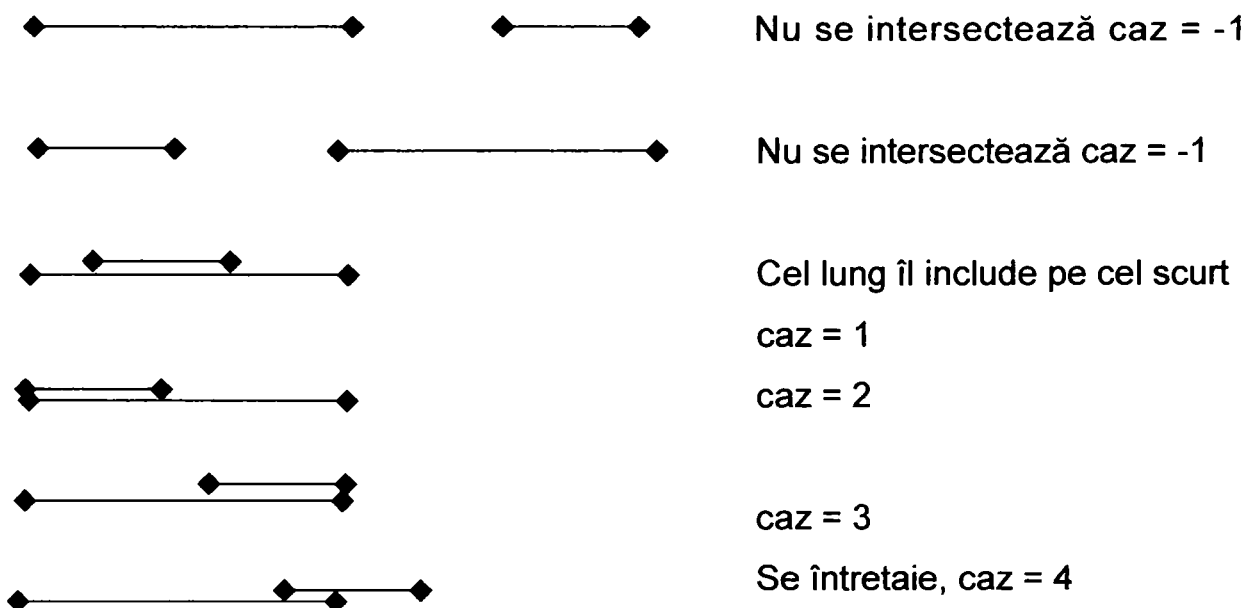
Primul pas în cadrul algoritmului este de a verifica dacă cele două segmente sunt în același plan și dacă nu sunt paralele.

În continuare se calculează distanțele cu semn de la capetele unuia din segmente la suportul celeilalte și invers (Savii, 1994). Dacă vreuna din aceste distanțe are o valoare mai mică decât precizia este clar că unul din segmente se sprijină cu un capăt pe celălalt. Dacă toate patru distanțele sunt sub precizie, înseamnă că cele două segmente sunt în prelungire. În acest caz se determină poziția lor cu ajutorul unui alt algoritm care se va prezenta mai jos.

Dacă segmentele nu sunt în prelungire, se determină raportul în care unul din segmente îl împarte pe celălalt și cu ajutorul acestuia se determină punctul de intersecție. Acesta poate fi în afara unuia sau a ambelor segmente (Mureșan, 1994, Savii, 1994).

Altă metodă ar consta în determinarea în prealabil a poziției celor două segmente unul față de celălalt prin proiectarea lor în cele trei plane: XoY, YoZ, ZoX. Dacă cele două segmente nu se intersectează în nici un plan ele nu se taie în interiorul lor. Această metodă determină doar poziția relativă a celor două segmente, nu și punctul lor de intersecție (Ogârcin Geraldine 18/2000).

Dacă cele două segmente au același suport există următoarele situații:



Pe baza acestor cazuri se pot determina punctele de tăiere care trebuie generate pe muchiile suprapuse (Mureșan, 1994, Savii, 1994).

Algoritmul poate fi descris prin următorul pseudo-cod:

```

- se notează cu v1 vectorul primei muchii și cu v2 vectorul muchiei 2
// se verifică dacă cele două segmente pot forma un plan
pr_vect = v1 x v2
- dacă pr_vect = 0
{
  dist = distanța de la punctul de început al muchiei 1 la suportul muchiei 2
  - dacă dist = 0
  {
    // segmentele au același suport
    - se determină coordonata pe care variază direcția celor
    - se determină distribuția celor patru capete pe coordonata determinată
    - în funcție de distribuție se returnează una din valorile -1, 1, 2, 3 sau 4
  }
  - dacă dist <> 0
    return segmentele sunt paralele
}
// se determină dacă segmentele se taie amândouă pe interior
d1 = distanța de la începutul segmentului 1 la suportul segmentului 2
d2 = distanța de la sfârșitul segmentului 1 la suportul segmentului 2
d3 = distanța de la începutul segmentului 2 la suportul segmentului 1
d4 = distanța de la sfârșitul segmentului 2 la suportul segmentului 1
- dacă semn(d1) <> semn(d2) și semn(d3) <> semn(d4)
{
  // segmentele se taie pe interior
  raport = d1 / (d1 - d2)
  punctul de tăiere = începutul seg 1 + raport * versor seg 1
}
- dacă semn(d1) <> semn(d2) și semn(d3) = semn(d4)
  segmentele se taie pe prelungirea segmentului 2
- dacă semn(d1) = semn(d2) și semn(d3) <> semn(d4)
  segmentele se taie pe prelungirea segmentului 1

```

4.3 Concluzii. Contribuții personale

În concluzie, operațiile booleene sunt foarte utile în realizarea și vizualizarea formei finale a unui corp din construcție. Astfel, crește viteza de proiectare și scad reburile deoarece comenzile de efectuare a prelucrărilor de pe corp vor fi transmise la roboți doar după ce s-a obținut forma dorită.

Contribuțiile personale în cadrul capitolului actual sunt legate de proiectarea algoritmilor prezentați, implementarea și depanarea lor:

- Abordarea sistematică și unitară a metodelor de rezolvare a problemelor geometrice de intersecție dintre diferitele elemente (plane, poligoane, drepte, semidrepte, segmente de dreaptă) în spațiul 3D.
- Analiza unitară a cazurilor particulară de contacte dintre solide în spațiul 3D. A fost necesară tratarea acestor situații cu o atenție deosebită, deoarece abordarea generală nu furniza rezultate corecte.
- Luarea în considerare în toate etapele de proiectare a influenței pe care poate să o aibă existența concavităților sau a găurilor asupra modului de evoluție a algoritmilor utilizați. Admiterea solidelor concave sau cu găuri în lista operanzilor a dus la o complicare importantă a subalgoritmilor, dar a fost necesară pentru a putea prelucra toate tipurile de corpuri reale.
- Conceperea algoritmului de clasificare pentru determinarea fețelor ce vor face parte din corpul sau corpurile rezultate în urma operațiilor booleene. Din mulțimea acestor fețe trebuie să se poată stabili submulțimile corespunzătoare fiecărui corp rezultat.
- Conceperea algoritmului de propagare a proprietății unor fețe de a aparține unui corp rezultat, spre fețele învecinate cu apartenență necunoscută, în scopul clasificării lor relativ la corpurile rezultate.
- Stabilirea strategiei optime pentru modul de alegere a operanzilor în cazul operațiilor booleene care conțin mai mult de doi termeni. Criteriul luat în considerare este cel al timpului de execuție minim.
- Implementarea algoritmilor pentru intersecția diferitelor elemente geometrice în spațiul 3D pentru efectuarea operațiilor booleene dintre solide.
- Implementarea algoritmilor pentru obținerea conturilor de poligoane formate dintr-un set de muchii coplanare, în cazul în care se dorește determinarea poligoanelor cele mai mici sau cele mai mari.
- Implementarea algoritmului pentru asocierea conturilor găurilor la conturul poligonului căruia îi aparțin.

- Implementarea algoritmilor pentru stabilirea poziției relative a unui punct față de un poligon, respectiv relativ la un volum.
- Implementarea algoritmului pentru stabilirea poziției relative a două segmente coliniare date prin capetele lor exprimate prin coordonate în spațiul modelului solidului.

5 PROBLEME SPECIALE. REZOLVĂRI

Probleme speciale

Principalii factori ce generează erori în cadrul unui proiect sunt:

- *Precizia*
- *Viteza*
- *Memoria necesară*

5.1 Rezolvarea problemelor de precizie

Este cea mai mare problemă care poate să apară și cea mai greu corectabilă. Pentru a face ca această problemă să aibă efecte minime există mai multe metode:

1. efectuarea operațiilor de intersecție o singură dată pe parcursul algoritmului
2. valorile apropiate de limitele minime și maxime se verifică cu o anumită precizie
3. precizia în cazul dimensiunilor liniare să difere de cea folosită la produsele scalare, vectoriale sau la arii
4. rotunjirea la valori limită nu este întotdeauna benefică

În cadrul algoritmilor prezențați s-au aplicat toate aceste metode de eliminare a erorilor de precizie după cum se arată în continuare:

5.1.1 Efectuarea operațiilor de intersecție o singură dată

Tăierea tuturor planelor între ele se face o singură dată și rezultatul se salvează în lista de linii. Tot pe aceste linii se stochează și punctele de intersecție dintre muchii. În momentul în care se pun pe suprafețe urmele lăsate de alte suprafețe se caută punctele de continuitate a urmelor între ele (Mureșan, 1995).

Punctele de intersecție se memorează și pe muchii pentru a fi regăsite dacă este cazul. Testul de identitate a două vârfuri se face cu o anumită precizie. În figura 5.1 se prezintă o situație care ar putea să apară dacă intersecțiile se fac de mai multe ori și testele de vârfuri egale se fac cu identitate (perfectă) de coordonate (Steve, 1993).

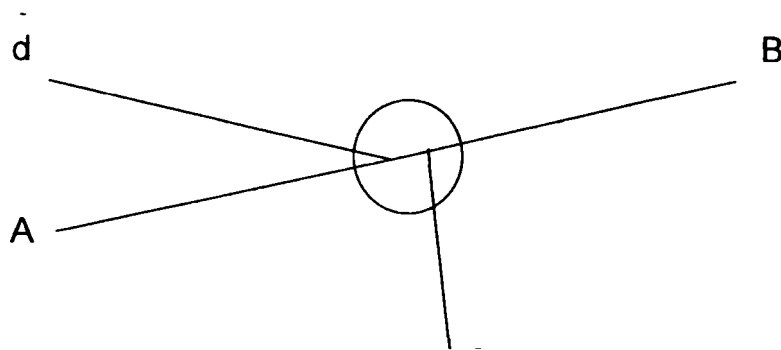


Figura 5.1: Același punct este obținut prin două tăieri ca două puncte distincte

Punctul obținut prin intersecția dintre segmentul AB și dreapta d diferă foarte puțin de punctul obținut prin intersecția segmentului AB cu dreapta d. Dacă cele două puncte nu se memorează pe muchie, sau dacă se memorează pe muchie dar ele se testează dacă sunt perfect identice, se va obține cazul din figură, care duce la generarea unor corpuri eronate, care nu pot fi folosite în continuare în nici o operație (Steve, 1993).

5.1.2 Verificarea valorilor între ele și față de valorile limită

În cazul în care se dorește verificarea dacă coordonatele 3D a două vârfuri sunt egale, se ia fiecare pereche de coordonate și se verifică dacă diferența este mai mică decât o anumită precizie dată. În cadrul proiectului această precizie este de 0.0001. În urma testelor s-a demonstrat că la această precizie numărul de erori care apar este minim.

La fel se procedează în cazul verificării unor valori cu valorile limită pe care poate să le ia. De exemplu, este cazul unui produs vectorial, care dacă este 0 înseamnă că cei doi vectori sunt paraleli sau au același suport. Valoarea produsului vectorial se testează dacă este mai mică decât 0.0001, nu exact 0. Aceeași situație este în cazul produsului scalar dintre doi vectori care, dacă este 0, înseamnă că cei doi vectori sunt perpendiculari. Alt exemplu ar fi valorile funcțiilor trigonometrice în zona valorilor 1 și -1 (Steve, 1993).

5.1.3 Preciziile diferă în funcție de operație

Preciziile care s-au utilizat în cadrul proiectului depind de natura mărimilor care se testează. De exemplu, două coordonate x , y sau z se testează cu precizia 0.0001. Două unghiuri se consideră egale dacă diferența dintre ele este mai mică decât 0.01. Distanța în spațiu dintre două puncte se consideră 0 dacă ea este mai mică decât 0.001. Precizia folosită în cazul rezultatelor funcțiilor trigonometrice este de 0.000001. Folosind aceste

precizii diferențiate pe tipuri de mărimi, s-au obținut mai puține erori datorate preciziilor. Cu toate acestea, numărul lor deși tinde către 0 nu poate fi 0 (Steve, 1993).

5.1.4 Rotunjirea la valori limită

Pentru a evidenția o eroare gravă ce poate să apară dacă unele coordonate se rotunjesc la 0 se consideră următorul exemplu.

Se consideră un corp care are sistemul lui local poziționat exact în sistemul global de axe. Lungimea corpului este de 10m. Sistemul local este dat prin patru puncte, originea și vârfurile celor trei axe situate la o distanță de 1m. (axele trebuie să fie versori). Dacă se rotește corpul în jurul axei z cu 1° rotația la capătul final al corpului este sesizabilă în schimb la nivelul versorilor axelor de coordonate nu. Din acest motiv se poate obține un corp cu sistem local greșit. În figura 5.2 se observă cum sistemul local al corpului este eronat, deoarece el a putut fi rotunjit la 0 pe când capătul corpului nu (Steve, 1992).

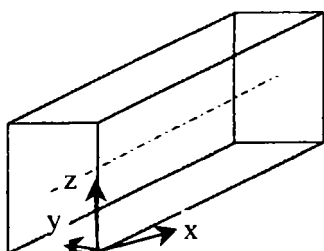


Figura 5.2: Corp de lungime mare rotit în jurul axei z

O situație similară se obține dacă vârful unui corp este foarte aproape de originea sistemului de coordonate global. Dacă se rotunjesc coordonatele la 0 se pot obține suprafețe care nu mai sunt plane, situație ce nu poate fi rezolvată cu ajutorul algoritmului din cadrul proiectului.

Deci, în concluzie, rotunjirea la valorile limită trebuie făcută în funcție de situația dată, pentru a nu genera corpuri eronate (cu suprafețe curbe) care nu mai pot fi folosite în cadrul unor operații ulterioare (Steve, 1992).

5.2 Rezolvarea problemelor de viteză

Viteza este un alt factor important ce intervine în aprecierea unui program. Dacă programul funcționează fără nici o eroare de precizie, dar lent, el nu este acceptabil. Trebuie să se facă un compromis între rezolvarea erorilor de precizie, a celor de viteză și a celor de memorie ocupată. Un utilizator preferă să beneficieze câte puțin din fiecare. Problemele de viteză și de memorie ocupată depind și de resursele de care dispune utilizatorul (Steve, 1993).

Pentru a câștiga în viteză, se pot elimina operațiile inutile, sau se poate renunța la unele funcții care au fost introduse pentru a corecta unele erori de precizie. Dar aceste operații trebuie făcute cu mare grijă, pentru a nu afecta funcționarea corectă a algoritmului. În general, testele și desenările pe ecran pe parcursul algoritmului duc la încetinirea lui (Mureșan, 1995, Savii, 1994).

În continuare se prezintă versiunea fără erori a unei funcții `memchr()` ce caută un caracter în memorie, ce urmează a fi optimistă:

```
void *memchr(void *pv, unsigned char ch , size_t size)
{
    unsigned char *pch = (unsigned char *)pv;
    while(size-- > 0)
    {
        if(*pch == ch)
            return (pch) ;
        pch++;
    }
    return (NULL) ;
}
```

Majoritatea programatorilor, în momentul în care au realizat un cod ce funcționează, își pun problema vitezei de lucru. Acesta nu este un lucru rău, dar dacă se exagerează rezultatele pot fi catastrofale (Steve, 1993).

Având funcția de mai sus, întrebarea care se pune este cum să o facem mai rapidă. Există trei posibilități :

- înlăturarea testului de dimensiune
- înlăturarea testului de caracter
- înlăturarea incrementării pointerului

Pare imposibil să se înlătore oricare din aceste teste, dar dacă cineva e insistent, poate găsi unele metode originale, ca în exemplul următor.

Să vedem de ce avem nevoie de testul de dimensiune. Avem nevoie de acest test doar pentru a întoarce NULL dacă nu găsim caracterul "ch" în primii "size" octeți din memorie. Înlăturând testul de dimensiune noi garantăm că "ch" va fi întotdeauna găsit, prin scrierea lui la sfârșitul blocului de memorie unde va fi găsit dacă nu există deja în blocul de memorie. Rezultă următorul cod :

```
void *memchr(void *pv, unsigned char ch, size_t size)
{
    unsigned char *pch = (unsigned char *)pv;
    unsigned char *pchPlant;
    unsigned char chSave;
// pchPlant indică primul caracter ce urmează după blocul de
// memorie în care se face căutarea. Se pune ch la locația
// respectivă pentru a garanta găsirea lui
    pchPlant = pch + size;
    chSave = *pchPlant;    // Salvează caracterul original
    *pchPlant = ch;
    while(*pch != ch)
        pch++;
    *pchPlant = chSave;    // Pune caracterul original înapoi
    return((pch == pchPlant) ? NULL : pch);
}
```

Pare o idee genială. Viteza a crescut de două ori, funcționează la prima vedere totul OK, dar la un studiu mai atent vom observa problemele mari pe care le poate genera:

- dacă "pchPlant" pointează spre o zonă de memorie read-only, scrierea lui "ch" nu va avea efect și funcția va returna un pointer invalid dacă nu-l va găsi pe "ch" în primele size+1 caractere.
- dacă "pchPlant" indică o zonă de dispozitive I/O mapată în memorie, scrierea lui "ch" va genera tot felul de reacții bizare (pornirea, oprirea floppy discului, rotirea roboților comandați într-o fabrică etc.)
- dacă "pchPlant" indică ultimii "size" octeți din memoria RAM, atât "pch" cât și "size" vor fi valizi dar "pchPlant" va indica o zonă inexistentă sau protejată la scriere.

- dacă "pchPlant" indică o zonă de date folosită de procese concurente, unul îl scrie pe "ch" și poate să distrugă informația de care altul are nevoie.

Ultima variantă este și cea mai periculoasă pentru că se poate manifesta în foarte multe moduri. Ce se întâmplă dacă se apelează `memchr()` pentru a verifica un bloc de memorie ce a fost alocat și funcția alterează o structură de date a managerului de memorie? Oricine va încerca să folosească acel manager (o întrerupere) va bloca sistemul. Sau dacă se scanează o matrice globală se poate scrie într-o variabilă folosită în altă parte etc (Steve, 1993).

În mod normal nu se va sesiza că `memchr()` generează erori, deoarece atâta timp cât nu modifică memoria critică totul pare că merge OK. Dar erorile generate de această funcție sunt foarte greu de depistat deoarece procesul ce execută `memchr()` va funcționa foarte bine, celălalt proces va avea memoria distrusă și se va bloca, deci nu vom avea nici un motiv să suspectăm funcția `memchr()`.

Ca o concluzie generală se poate spune că nu este permisă referirea la memoria asupra căreia nu ai drepturi. A referi înseamnă atât a citi cât și a scrie. Citirea nu dăunează altor procese, dar va bloca programul dacă se referă la memorie protejată, inexistentă sau dispozitive I/O mapate în memorie (Steve, 1993).

5.3 Rezolvarea problemelor legate de memorie

Pentru a stoca informație în cadrul unui algoritm se pot folosi mai multe metode. În primul rând se poate alege între alocarea statică de memorie și cea dinamică. În general alocarea dinamică este mai eficientă, deoarece se poate alocă exact spațiul necesar, dar în aceste situații există riscul de a se uita acea zonă de memorie alocată, ceea ce duce la micșorarea zonei de memorie disponibile. În cazul alocării statice, acest lucru nu se poate întâmpla, deoarece tabloul alocat static se dezalocă automat la părăsirea funcției în care a fost declarat (Steve, 1993, Roger, 1994).

În cazul alocării dinamice se pot alege mai multe metode de stocare a informației:

- listă înlănțuită
- arbori binari
- tablouri
- metoda "hashing"

Cele mai des folosite sunt listele înlănțuite și tablourile.

În alegerea metodei potrivite se iau în considerare doi factori: viteza și eficiența utilizării memoriei (Steve, 1992, Steve, 1993, Tudor, 1995).

Dacă tabloul (matricea) este foarte rară, se alege una din metodele listă înlănțuită sau arbore binar, deoarece doar pentru elementele care se folosesc se alocă memorie.

Tabloul de pointeri nu se poate folosi în acest caz, deoarece el trebuie alocat pentru toate elementele chiar dacă nu toate sunt folosite.

Metoda hashing se plasează undeva între liste/arbori și tablou de pointeri. Ea se caracterizează prin faptul că tabloul logic se alocă pentru toate elementele dar este mai mic decât un tablou de pointeri.

Dacă tabloul este plin se va folosi tabloul de pointeri, deoarece listele/arborii necesită, pe lângă informația din fiecare element, doi pointeri în plus (anterior/următor sau stâng/drept) față de tablourile de pointeri. De exemplu, dacă avem un tablou de 1000 de elemente și considerăm că un pointer ocupă doi octeți, o listă sau arbore va ocupa 4000 octeți pentru pointeri, iar tabloul de pointeri doar 2000 de octeți.

Privind din punct de vedere a vitezei cea mai rapidă metodă este cea cu tabloul de pointeri (Steve, 1993).

Cea mai indicată metodă, când memoria nu reprezintă o problemă, este tabloul de pointeri. Dacă nu este suficientă memorie, se recomandă folosirea listelor înlănțuite sau a arborilor binari.

5.3.1 Memorii tampon reutilizabile

În cazul în care se folosesc variabile locale de tip tablou static și memoria nu este suficientă, se apelează la tablourile dinamice obținute prin alocare dinamică.

De exemplu, avem două procese A() și B() într-un program. Se consideră că A() folosește 60% din memoria disponibilă și B() 55% din memoria disponibilă. Deci A() nu poate apela pe B() și nici B() pe A() deoarece nu se poate folosi memoria 115% (> 100%). Dacă se rulează doar A() sau doar B(), nu apar probleme de memorie. Problema se rezolvă cu alocarea dinamică (Steve, 1992).

Dacă memoria disponibilă este de 10000 de octeți și se rulează un program de forma:

```
A ()
{
    char a[6000];
    .
    .
    .
    B ();
    .
    .
    .
}

B ()
{
    char b[5500];
    .
    .
    .
}
```

Este clar că funcția B() nu mai poate folosi un spațiu de 5500 pentru tabloul b.

Problema se rezolvă astfel: dacă funcția A() nu are nevoie de valorile din tabloul a la revenirea din funcția B(), ambele funcții își pot împărți memoria astfel:

```
A ()
{
    char *a;
    a = (char *) malloc(6000);
    .
    .
    .
    free(a);
}
```

```
B();  
a = (char *) malloc(6000);  
.  
.  
free(a);  
}  
  
B()  
{  
char *b;  
b = malloc(5500);  
.  
.  
free(b);  
}
```

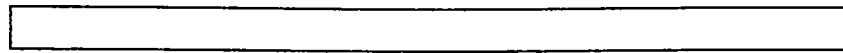
În acest caz a există doar ca pointer pe timpul execuției funcției B().

5.3.2 Fragmentarea memoriei

Deoarece malloc() și free() nu fac parte din limbajul C standard ci din biblioteca C, implementarea lor diferă de la un compilator la altul. Prin folosirea frecventă a funcțiilor în programe poate să apară fragmentarea memoriei, ceea ce duce la eșuarea unor alocări chiar dacă suma zonelor libere este mai mare decât ceea ce trebuie alocat. Fragmentarea apare când sunt intercalate zone alocate cu zone libere scurte (Steve, 1992).

În continuare se prezintă un exemplu de fragmentare a memoriei. Se consideră că sunt alocate patru zone consecutive de memorie A, B, C, D. Dacă se dezalocă B și C, zonele pot fi refolosite împreună. Dacă se dezalocă B și D zonele nu pot fi refolosite împreună deoarece sunt despărțite de zona alocată C. Limbajul nu poate muta zona alocată C în locul zonei D pentru a putea folosi B și C, deoarece programul nu ar putea fi înștiințat de aceasta mutare, ceea ce ar duce la erori în cadrul programului.

Fragmentarea memoriei se elimină greu. Uneori se grupează alocările mărunte pentru a preveni formarea fragmentelor mici. Alteori, în cadrul programului, la un moment dat se salvează toată informația alocată într-un fișier temporar pe disc, se eliberează memoria și se recitește informația din fișier (Steve, 1992).



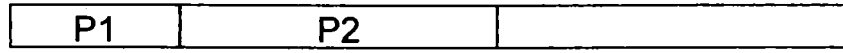
0



0

29

P1 = malloc(30)

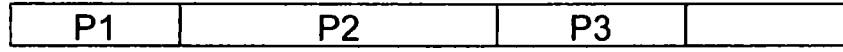


0

29

69

P2 = malloc(40)



0

29

69

P3 = malloc(20)



0

29

69

free(P2)

P4 = malloc(50)

ultima alocare eşuează din lipsă de spațiu.

5.4 Concluzii. Contribuții personale

În continuare se prezintă niște concluzii care nu trebuie neglijate la scrierea unui algoritm (Nielsen, 1995). Acestea au fost toate aplicate în dezvoltarea software-ului din cadrul tezei, multe din ele rezultând în timpul activității de proiectare.

Tipurile de date trebuie să fie alese cu grijă. Chiar dacă standardul ANSI permite tuturor implementărilor să suporte tipurile de date *char*, *int*, *long* și altele, el nu definește concret aceste tipuri. De aceea trebuie să te bazezi numai pe specificațiile pe care standardul ANSI le garantează.

Chiar dacă implementarea este corectă, pot apărea erori la rularea pe un alt sistem. Tipurile de date pentru variabile trebuie alese astfel încât dimensiunea tipului de dată să acopere toate valorile pe care le poate lua variabila, astfel încât să se evite cazurile de *overflow* și *underflow*, generatoare de erori.

Trebuie respectat întocmai algoritmul de implementat; cea mai ușoară cale de a introduce erori este aceea de a "trișa" la implementare.

Fiecare funcție trebuie să aibă un task bine definit, și mai mult decât atât, trebuie să existe o singură cale care rezolvă task-ul. Dacă, indiferent de intrare, codul se execută, scad șansele de a avea erori nedetectate.

Trebuie eliminat fiecare caz special, instrucțiune condițională *if*, care nu este necesară în cod prin înlocuirea sau modificarea algoritmului. Pentru aceasta, uneori este necesară modificarea structurilor de date, alteori este cazul schimbării abordării.

Uneori, instrucțiunea condițională *if* poate apărea deghizată în expresiile de control ale instrucțiunilor de ciclare *while* și *for*. De asemenea, operatorul condițional *?:* este un alt tip de *if*.

Trebuie evitate idiomurile de limbaj. Așa-zisele "trucuri" folosite în implementare, care sunt utilizate cu scopul de a îmbunătăți unele performanțe, deseori nu realizează în mod semnificativ acest lucru, și în plus sunt adevărate capcane;

La scrierea unei expresii nu e bine să se amestece tipurile de operatori, iar în cazul în care acest lucru nu poate fi evitat, se folosesc parantezele pentru separarea operațiilor.

Nu se apelează funcții care pot eșua. Dacă trebuie apelată o astfel de funcție, care poate returna o eroare, se încearcă identificarea cazurilor de tratare a erorilor, și astfel se măresc șansele de a găsi erorile care pot apărea chiar în codul pentru tratarea erorii;

Erorile nu apar în cod în mod spontan; ele sunt rezultatul scrierii unui cod nou de către programator sau a modificării codului existent. Dacă se dorește găsirea erorilor în cod, cea mai bună metodă este de a merge pas cu pas prin fiecare linie a codului. Când trebuie să se facă niște modificări la codul existent și se ajunge la concluzia că aceste

modificări sunt 'fundamentale', trebuie să se analizeze încă o dată ceea ce trebuie schimbat, întrucât noile modificări pot avea și alte influențe în cod. Codul nu trebuie văzut ca un *black-box* la care sunt de interes doar intrările și ieșirile, ci, trebuie urmărite în permanentă și rezultatele intermediare.

În depanare, trebuie mers pe fiecare cale a codului, cel puțin o dată, în special în secvențele de tratare a erorilor și, de asemenea, în cazul instrucțiunilor (*if* și *switch*) și a operatorilor cu mai multe căi (&&, ||, ?:).

Este indicat ca în timpul depanării atenția să fie centrată pe date, nu pe instrucțiuni, astfel existând posibilitatea de a avea o a doua imagine a codului, diferită de cea formată în momentul scrierii codului (Steve, 1992).

Contribuțiile personale la acest capitol sunt:

- Sintetizarea rezultatelor cercetărilor pe plan mondial în domeniul problemelor speciale ce pot să apară în implementarea unui algoritm și metodele de rezolvare a lor
- Evidențierea unor probleme speciale noi și crearea de rezolvări
- Analiza avantajelor și limitelor diferiților algoritmi prezentați în literatura de specialitate, referitoare la evitarea apariției erorilor de precizie
- Abordarea sistematică și unitară a căilor de evitare a erorilor datorate preciziei de calcul în cazul reprezentării datelor în virgulă flotantă.
- Efectuarea unor funcții suplimentare necesare creșterii vitezei algoritmilor și eliminarea cât mai eficientă a erorilor de precizie, fără a crește prea mult cantitatea de memorie utilizată

Concluzii

Prezenta teză de doctorat propune o abordare riguroasă și unitară a procesului de dezvoltare a unei aplicații software. Fiecare algoritm prezentat a fost succesiv îmbunătățit pentru a obține rezultate cât mai bune.

Analizând mulțimea de algoritmi prezentați în literatura de specialitate, am putut evidenția limitările algoritmilor respectivi și, încercând rezolvarea principalelor probleme ridicate, am reușit să obținem rezultate remarcabile.

Rezultatele obținute prin utilizarea programelor originale create în cadrul tezei au confirmat justetea soluțiilor adoptate și corectitudinea modului de aplicare în practică.

Pe parcursul procesului de implementare și testare a algoritmilor, au fost efectuate o serie de îmbunătățiri care să conducă la reducerea timpului de execuție, îmbunătățiri atât la nivelul concepției, cât și la nivelul tehnicilor de programare folosite. În situația în care numărul de prelucrări de pe un corp poate ajunge să fie de ordinul zecilor, este de înțeles dorința utilizatorilor de a avea un produs performant din punct de vedere al vitezei de execuție. Din acest motiv au fost preferate acele metode de implementare care păstrau în structurile de date o cantitate mai mare de informații.

Concluzia la care s-a ajuns în urma efectuării de măsurări la nivelul diferitelor variante ale algoritmilor asupra diferitelor proiecte, este faptul că performanțele algoritmilor sunt dependente de genul de prelucrări care trebuie efectuate asupra corpurilor ce alcătuiesc proiectul.

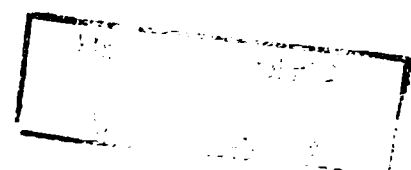
Contribuții personale

Principalele contribuții originale aduse în cadrul tezei sunt:

- Sintetizarea rezultatelor cercetărilor pe plan mondial în domeniul abordat de teză, astfel încât să se realizeze o unealtă utilă celor ce doresc să abordeze în continuare acest domeniu.
- Analiza avantajelor și limitelor diferiților algoritmi prezentați în literatura de specialitate, în scopul conceperii unor algoritmi superiori, care să funcționeze corect fără restricții aplicate solidelor operanzi.
- Analiza avantajelor și limitelor diferitelor metode de modelare a solidelor prezentate în literatura de specialitate, din punct de vedere al problemelor de rezolvat.
- Determinarea și alegerea reprezentării modelelor solidelor celei mai convenabile, din punctul de vedere al prelucrărilor de efectuat, cunoscut fiind faptul că nu orice fel de reprezentare permite orice fel de manipulare a obiectelor
- Analiza datelor necesare algorimilor și a modalităților de organizare a informației, în scopul asigurării eficienței procesului de prelucrare a lor. Această etapă a procesului de proiectare a fost necesar să fie efectuată de mai multe ori, o dată pentru fiecare mod de abordare a rezolvării problemelor, lucru care a evidențiat dezavantajul utilizării metodei de proiectare structurate comparativ cu cea orientată pe obiect.
- Abordarea sistematică și unitară a căilor de evitare a erorilor datorate preciziei de calcul în cazul reprezentării datelor în virgulă flotantă. Deoarece solidelor rezultate prin operații booleene trebuie să li se poată efectua aceleași prelucrări ca și solidelor inițiale a fost necesar ca în procesul de identificare a diferitelor elemene geometrice să se lucreze cu aceeași precizie în ambele probleme tratate în cadrul tezei.
- Abordarea sistematică și unitară a codării prelucrărilor mașină, ceea ce a condus inclusiv la o organizare superioară a programelor din punct de vedere structural și al eficienței.
- Implementarea algoritmilor de îmbinare a două solide prin intermediul unor prelucrări mașină. Îmbinarea presupune generarea codurilor și efectuarea lor prin intermediul operațiilor booleene.
- Proiectarea și implementarea funcțiilor de generare a corpurilor negative ce modelează prelucrările
- Implementarea unui mecanism de reconstruire a corpului pe baza prelucrărilor stocate de el

- Implementarea funcțiilor de calcul al punctelor prelucrărilor pe baza poziției celor două corpuri
- Proiectarea și implementarea funcțiilor de transformare a unor prelucrări în altele. Sunt situațiile în care o prelucrare oarecare are o poziție particulară deci ea poate fi modelată printr-o prelucrare mai simplă. Este preferată prelucrarea mai simplă deoarece ea se va executa mai ușor de către roboți. Prelucrările mai complicate cu poziții oarecare necesită scule speciale.
- Implementarea funcției de recalculare a punctelor prelucrării în vederea transmiterii acestora la roboți.
- Abordarea sistematică și unitară a metodelor de rezolvare a problemelor geometrice de intersecție dintre diferitele elemente (plane, poligoane, drepte, semidrepte, segmente de dreaptă) în spațiul 3D.
- Analiza unitară a cazurilor particulară de contacte dintre solide în spațiul 3D. A fost necesară tratarea acestor situații cu o atenție deosebită, deoarece abordarea generală nu furniza rezultate corecte.
- Luarea în considerare în toate etapele de proiectare a influenței pe care poate să o aibă existența concavităților sau a găurilor asupra modului de evoluție a algoritmilor utilizați. Admiterea solidelor concave sau cu găuri în lista operanzilor a dus la o complicare importantă a subalgoritmilor, dar a fost necesară pentru a putea prelucra toate tipurile de corpuri reale.
- Conceperea algoritmului de clasificare pentru determinarea fețelor ce vor face parte din corpul sau corpurile rezultate în urma operațiilor booleene. Din mulțimea acestor fețe trebuie să se poată stabili submulțimile corespunzătoare fiecărui corp rezultat.
- Conceperea algoritmului de propagare a proprietății unor fețe de a aparține unui corp rezultat, spre fețele învecinate cu apartenență necunoscută, în scopul clasificării lor relativ la corpurile rezultate.
- Stabilirea strategiei optime pentru modul de alegere a operanzilor în cazul operațiilor booleene care conțin mai mult de doi termeni. Criteriul luat în considerare este cel al timpului de execuție minim.
- Implementarea algoritmilor pentru intersecția diferitelor elemente geometrice în spațiul 3D pentru efectuarea operațiilor booleene dintre solide.
- Implementarea algoritmilor pentru stabilirea poziției relative a unui punct față de un poligon, respectiv relativ la un volum.
- Implementarea algoritmului pentru stabilirea poziției relative a două segmente coliniare date prin capetele lor exprimate prin coordonate în spațiul modelului solidului.

- Sintetizarea rezultatelor cercetărilor pe plan mondial în domeniul problemelor speciale ce pot să apară în implementarea unui algoritm și metodele de rezolvare a lor
- Analiza avantajelor și limitelor diferiților algoritmi prezentați în literatura de specialitate, referitoare la evitarea apariției erorilor de precizie
- Abordarea sistematică și unitară a căilor de evitare a erorilor datorate preciziei de calcul în cazul reprezentării datelor în virgulă flotantă.
- Efectuarea unor funcții suplimentare necesare creșterii vitezei algoritmilor și eliminarea cât mai eficientă a erorilor de precizie, fără a crește prea mult cantitatea de memorie utilizată



Anexa 1

Conține codul sursă a definiției clasei utilizată în calcule matematice împreună cu metodele ei, codul sursă a claselor ce descriu prelucrările și codul sursă al funcției de generare a cepului și a găurii de cep în cazul îmbinării a două corpuri.

Codul sursă a clasei matematice Cvector și a metodelor ei:

```
class CVector
{
public:
    double vect[3];    // coordonatele x, y, z ale vectorului
// constructor
    CVector(double x = 0.0, double y = 0.0, double z = 0.0)
    {
        vect[X] = x; vect[Y] = y; vect[Z] = z;
    }
// constructor
    CVector(CVector &v)
    {
        vect[X] = v.vect[X];
        vect[Y] = v.vect[Y];
        vect[Z] = v.vect[Z];
    }
// inițializare a celor trei coordonate cu aceeași valoare
    void operator=(double d)
    {
        vect[X] = d;
        vect[Y] = d;
        vect[Z] = d;
    }
// operator de adunare a doi vectori
    CVector operator+(CVector &v)
    {
        CVector w;
        w.vect[X] = vect[X] + v.vect[X];
        w.vect[Y] = vect[Y] + v.vect[Y];
```

```
        w.vect[Z] = vect[Z] + v.vect[Z];
        return w;
    }
// operator de adunare a unei valori la cele trei coordonate
CVector operator+(double d)
{
    CVector w;
    w.vect[X] = vect[X] + d;
    w.vect[Y] = vect[Y] + d;
    w.vect[Z] = vect[Z] + d;
    return w;
}
// operator de adunare a doi vectori
void operator+=(CVector &v)
{
    vect[X] += v.vect[X];
    vect[Y] += v.vect[Y];
    vect[Z] += v.vect[Z];
}
// operator de scădere a doi vectori
CVector operator-(CVector &v)
{
    CVector w;
    w.vect[X] = vect[X] - v.vect[X];
    w.vect[Y] = vect[Y] - v.vect[Y];
    w.vect[Z] = vect[Z] - v.vect[Z];
    return w;
}
// operator de scădere a doi vectori
void operator-=(CVector &v)
{
    vect[X] -= v.vect[X];
    vect[Y] -= v.vect[Y];
    vect[Z] -= v.vect[Z];
}
// operator de înmulțire a unui vector cu un scalar
CVector operator*(double d)
{
    CVector w;
```

```

    w.vect[X] = vect[X] * d;
    w.vect[Y] = vect[Y] * d;
    w.vect[Z] = vect[Z] * d;
    return w;
}
// operator de înmulțire a celor trei coordonate a unui vector cu
// același scalar
void operator*=(double d)
{
    vect[X] *= d;
    vect[Y] *= d;
    vect[Z] *= d;
}
// operator de tablou
double operator[](int i)
{
    return vect[i];
}
// produsul scalar a doi vectori
double SCALPRD(CVector &v)
{
    return ((vect[X])*(v.vect[X]) + (vect[Y])*(v.vect[Y]) +
            (vect[Z])*(v.vect[Z]));
}
// produsul vectorial a doi vectori
void VECTPRD(CVector &u, CVector &v)
{
    vect[X] = (u.vect[Y])*(v.vect[Z]) - (u.vect[Z])*(v.vect[Y]);
    vect[Y] = (u.vect[Z])*(v.vect[X]) - (u.vect[X])*(v.vect[Z]);
    vect[Z] = (u.vect[X])*(v.vect[Y]) - (u.vect[Y])*(v.vect[X]);
}
// determinarea lungimii unui vector
double VECT_LENGTH(void)
{
    return (sqrt(SCALPRD(*this)));
}

```

```
// determinarea versorului unui vector
void VERSOR(void)
{
    double l = VECT_LENGTH();
    if(fabs(l) < PREC)
    {
        vect[X] = vect[Y] = vect[Z] = 0.0;
        return;
    }
    vect[X] /= l;
    vect[Y] /= l;
    vect[Z] /= l;
}
};
```


Codul sursă a clasei matematice CPrelCode și a metodelor ei:

```
class CPrelCode
{
protected:
    // codurile sunt protejate și se pot obține cu
    // ajutorul metodelor clasei
    char *pointCode; // forma în puncte
    PrelType type; // un tip enumerare ce conține identificatori
    // pentru toate prelucrările posibile
    class CLine *li; // lista de linii (valabil la prelucrările ce nu
    // afectează geometria corpului, de ex. Gaura de
    // burghiu, marcajele, rindeluirea, etc.)
    class CPrelCode *next; // adresa nodului următoarei prelucrări
public:
    // obținerea formei în puncte
    void GetPointCode(char *pCode)
    {
        strcpy(pCode, pointCode);
    }
    void GetParamericCode(char *pCode)
    {
        strcpy(pCode, paramCode);
    }
    // funcție de construcție a corpurilor negative ce modelează
    // prelucrarea
    Cvolume *ConstructNegativeVolume(void);
    // recalcularea formei în puncte pentru a fi transmisă roboților
    void RecalculatePrelPoint(void);
    // testarea corectitudinii punctelor prelucrării
    bool TestCorectPrel(void);
    // conversia din forma în puncte în forma parametrică
    void ConvertPointsInParam(char *paramCode);
    // conversia din forma paramerică în forma în puncte
    void ConvertParamInPoints(char *paramCode);
    // funcție de test a poziției prelucrării față de corp (în interior
    // sau exterior)
    bool TestPosPrel(CVolume *V);
}CPrelCode;
```

Codul sursă al tipului enumerare ce conține identificatorii prelucrărilor.

```
enum { TAIERE_TRANSV, TAIERE_LONGIT, CHERT_SIMPLA, CHERT_SIMPLA_STRAMBA,  
CHERT_CAPAT, CHERT_LUNG,  
TAIERE_CONCAVA, TAIERE_CONVEXA, TAIERE_V,  
ECLIZ_DREAPTA, ECLIZ_STRAMBA, ECLIZ_LUNG,  
CHERT_COLT, CHERT_COLT_DREPT, CHERT_LIM,  
GAURA_CEP_DESCH, GAURA_CEP, CANAL_LIM,  
GAURA_CEP_OARECARE, GAURA_CEP_RANDUN_CAPAT,  
GAURA_CEP_RANDUN_FATA, CEP_DREPT, CEP_DEPLAS,  
CEP_ROTIT, CEP_RANDUN, CEP_CLASIC, CHERT_DUBLA_CAPAT,  
CHERT_DUBLA_FATA, CHERT_TABACITA, CHERT_SAPA,  
ORNAM_CONCAV, ORNAM_CONVEX, ARC_CERC, PROFIL_ROTUNJ,  
CANAL_FIRIZ, GAURA_BURG_OARECARE, GAURA_BURG_PERP,  
MARCAJ, RINDELUIRE } PrelType;
```

Codul sursă al clasei utilizate în cadrul funcției de generare a prelucrărilor pe baza îmbinării de tip cep.

```
class CCepParam
{
protected:
    short seite;
    double H, B, A1, A2, AZ;
    double BD1, BD2, BY, BV, VQ;
public:
    short Load(short i, short flg_trenn);
    void Save(short i, short flg_trenn);
    short AlegeDirectieCepTare(Volume *Vt, CVector *pConturLocalTare,
                               CVector *pOrdonate, CVector *pAles,
                               double maxYTare, double maxZTare,
                               char BStare);
    DetBSTare(CVector *pAles, short val, char BStare);
    DetSistRefBSdinNume(char BStare, CVector *srfBS,
                       short &indexBSTare);
    ConstrPuncteCep(Volume *Vt, Volume *Vm, CVector *pAles,
                   CVector *pOrdonate, CVector *pPrel,
                   short capatTare, short prelCapat);
    short AlegerePozitieCep(BPoly *fataContact, CVector *pPrel1Global,
                           CVector *pPrel2Global);
    VerificarePozitieCorpuri(Plane *plTare, Plane *plMoale,
                             Plane *plComun);
    short DetFunctieVolTare(Volume *Vt, CVector *pPrel1Global,
                           CVector *pContur);
    short CalcGauraCepTaieri(Volume *Vt, Volume *Vm, CVector *pPrel1Global,
                              CVector *pPrel1Local);
    CurataPrelucrariCapatSSZ_Cep(Volume *Vt, CVector *pPrel1Local,
                                  CVector N, double D, short capat);
    short DeterminareTipPrelucrare(short cod, short fctMoale);
}CCepParam;
```

Codul sursă a funcției de generare a prelucrărilor îmbinării.

```
// dimensiunea maximă a unui șir ce conține forma în puncte
#define DIMMAXSIR      500

// Vm - corpul moale
// Vt - corpul tare
// opțiune - modul de generare în prelungire sau nu
// srfBS - fața de referință selectată de utilizator în cutia de dialog
short CCepParam::GenerareCep(CVolume *Vm, CVolume *Vt, short optiune,
                             CVector srfBS)
{
//-----
  CVector pLocalMoale[8], pGlobalMoale[8], pLocalTare[8], pGlobalTare[8];
  ListPrelCode *lPrelCode;
  ListPrel *lPrel;
  short i, valRet, functiaTare, functiaMoale, capatTare, codMoale;
  Short indiceTare, deplPrel;
  Plane plTare[6], plMoale[6];
  short indicePunctDeplasat;
  char sirPuncteMoale[DIMMAXSIR], sirPuncteTare[DIMMAXSIR];
  CVector pContur;
  BPoly fataContact;
//-----
  DeterminaContur(Vm, Vt, pContur, fataContact);
  double unghi = DeterminaUnghiCorpuri(Vm, Vt);

  // axaXTare este vectorul axei corpului tare
  CVector axaXTare;
  axaXTare = Vt->f_line->point_2->coord - Vt->f_line->point_1->coord;

  // se determină capătul corpului tare care va fi afectat
  double sc;
  sc = fataContact->poly->normal.SCALPRD(axaXTare);
  if(sc > 0.0)
    capatTare = BEGIN;
  else
    capatTare = END;
}
```

```
// se trec punctele feței de contact în sistemul local al corpului tare
// pentru a determina valorile maxime pe coordonatele y si z
for(i = 0; i < 4; i ++)  
    pConturLocalTare[i].VECTCOPY(pContur[i]);  
TrecereSistemLocalPuncte(Vt, pConturLocalTare, 4);  
maxYTare = pConturLocalTare[0][1];  
maxZTare = pConturLocalTare[0][2];  
for(i = 1; i < 4; i ++)  
{  
    if(pConturLocalTare[i][1] > maxYTare)  
        maxYTare = pConturLocalTare[i][1];  
    if(pConturLocalTare[i][2] > maxZTare)  
        maxZTare = pConturLocalTare[i][2];  
}  
  
// se determină orientarea cepului  
CVector pOrdonate[5], pAles[2];  
valRet = AlegeDirectieCepTare(Vt, pConturLocalTare, pOrdonate, pAles,  
                               maxYTare, maxZTare, &BStare);  
  
if(!valRet)  
    return 1; // eroare nu s-a putut determina o direcție  
  
// se determină sistemul de coordonate al feței de referință  
short indexBSTare;  
char BStare;  
DetSistRefBSdinNume(BStare, srfBS, &indexBSTare);  
// se calculează punctele cepului și se determină capătul corpului moale  
// de care este apropiată prelucrarea pe el în prelCapat  
short prelCapat;  
CVector pPrel[8];  
  
ConstrPuncteCep(Vt, Vm, pAles, pOrdonate, pPrel, capatTare, prelCapat);  
  
// în pPrelLocalTare se pun punctele cepului în sistemul local al  
// corpului tare  
for(i = 0; i < 8; i ++)  
    pPrelLocalTare[i] = pPrel[i];  
TrecereSistemLocalPuncte(Vt, pPrelLocalTare, 8);
```

```
// în pPrelGlobalTare se pun punctele cepului în sistemul global
for(i = 0; i < 8; i ++)
    pPrelGlobalTare[i] = pPrelLocalTare[i];
TrecereSistemGlobalPuncte(Vt, pPrelGlobalTare, 8);

// dacă utilizatorul a selectat opțiunea ALEGERE și cele două
// deplasamente diferă se propun cele două posibilități de generare a
// cepului
if(optiune == ALEGERE && fabs(A1 - A2) > V_PPPEC)
{
    // se propun spre alegere cele două posibilitati
    ConstrPuncteCep(Vt, Vm, pAles, pOrdonate, pPrel, capatTare,
                    prelCapat);
    for(i = 0; i < 8; i ++)
        pPrelLocalMoale[i] = pPrel[i];
    TrecereSistemLocalPuncte(Vt, pPrelLocalMoale, 8);
    for(i = 0; i < 8; i ++)
        pPrelGlobalMoale[i] = pPrelLocalMoale[i];
    TrecereSistemGlobalPuncte(Vt, pPrelGlobalMoale, 8);
    valRet = AlegerePozitieCep(fataContact, pPrelGlobalTare,
                               pPrelGlobalMoale);
    if(!valRet)
        return 1; // utilizatorul a renunțat la operație
}

// se determină lista de plane comune ale celor două forme brute
Plane plComun;
plComun = new Plane;
CompletarePlanVolum(plTare, Vt);
CompletarePlanVolum(plMoale, Vm);

VerificarePozitieCorpuri(plTare, plMoale, plComun);

// se verifică dacă gaura este perpendiculară pe volumul moale
CVector v_01;
v_01 = pPrelGlobalTare [1] - pPrelGlobalTare [0];
CVector axaMoale;
axaMoale = Vm->f_line->point_2->coord - Vm->f_line->point_1->coord;
sc = v_01.SCALPRD(axaMoale);
```

```

if(sc != 0.0 && fabs(unghi - 90) > V_PPREC && plComun->n &&
  crt_cepParam->A1 < V_PPREC && crt_cepParam->A2 < V_PPREC &&
  (prelCapat && crt_cepParam->AZ < V_PPREC || !prelCapat))
{
  for(i = 0; i < 4; i++)
    pLocalMoale[i] = pPrelGlobalTare [i];
  TrecereSistemLocalPuncte(Vm, pLocalMoale, 4);
  indiceTare = -1;
  for(i = 0; i < 4; i++)
  {
    sc = fataContact->poly->normal.SCALPRD(plTare[i].normal);
    if(fabs(sc) > V_PPREC && sc < 0.0)
    {
      indiceTare = i;
      break;
    }
  }
  if(indiceTare == -1)
    return 1;

// se determină valoarea minimă și maximă pe x a cepului pe corpul moale
double minX, double maxX;
minX = pLocalMoale[1][0]; maxX = pLocalMoale[1][0];
for(i = 1; i < 4; i++)
{
  if(pLocalMoale [i][0] > maxX)
    maxX = pLocalMoale [i][0];
  if(pLocalMoale [i][0] < minX)

    minX = pLocalMoale [i][0];
}

// se determină capătul prelucrării la care se va aplica deplasamentul
specificat de
// utilizator dacă este cazul
for(i = 4; i < 8; i++)
{
  sc = plTare[indiceTare].normal.SCALPRD(pPrelGlobalTare[i]) +
    plTare[indiceTare].D;

```

```

    if(fabs(sc) < V_PPREC)
    {
        if(fabs(pLocalMoale[i - 4][0] - minX) < V_PPREC)
        {
            deplPrel = BEGIN;
            indicePunctDeplasat = i;
            break;
        }
        if(fabs(pLocalMoale[i - 4][0] - maxim_x) < V_PPREC)
        {
            deplPrel = END;
            indicePunctDeplasat = i;
            break;
        }
    }
}
// dacă deplPrel este BEGIN punctele 1 și 2 se vor calcula cu unghiul
// dintre bare
CVector vDepl;
vDepl = pPrelLocalTare[1] - pPrelLocalTare [0];
vDepl.VERSOR();
double xDepl;
xDepl = H * COSINUS(unghi) / SINUS(unghi);
vDepl = vDepl * xDepl;
if(indicePunctDeplasat == 4 || indicePunctDeplasat == 7)
{
    pPrelLocalTare[0] = pPrelLocalTare[0] + vDepl;
    pPrelLocalTare[3] = pPrelLocalTare[3] + vDepl;
}
else
{
    pPrelLocalTare[1] = pPrelLocalTare[1] - vDepl;
    pPrelLocalTare[2] = pPrelLocalTare[2] - vDepl;
}
functiaTare = 4;
}
else
    functiaTare = DetFunctieVolTare(Vt, pPrelGlobalTare, pContur);

```



```
// se determină punctele pe corpul moale pe baza punctelor de pe corpul
// tare inversate
Inverseaza(pPrelGlobalTare, 0, 1);
Inverseaza(pPrelGlobalTare, 2, 3);
Inverseaza(pPrelGlobalTare, 4, 5);
Inverseaza(pPrelGlobalTare, 6, 7);
for(i = 0; i < 8; i ++)
    pPrelLocalMoale[i] = pPrelGlobalTare[i];
TrecereSistemLocalPuncte(Vm, pPrelLocalMoale, 8);

// se verifică dacă gaura este de-a lungul volumului moale
CVector vDir, vPrd;
vDir = pPrelLocalMoale[1] - pPrelLocalMoale[0];
axaXTare.CVector(1.0, 0.0, 0.0);
vPrd.VECTPRD(vDir, axaXTare);
// în funcție de poziția găurii de cep pe corpul moale se determină codul
// și funcția
if(vPrd.VECT_LENGTH() < V_PPREC)
{
    codMoale = 4006;
    functiaMoale = 1;
    sprintf(sirPuncteMoale, "4006,1, ");
}
else
{
    if(optiune == PRELUNGIRE)
    {
        codMoale = 4003;
        functiaMoale = 2;
        sprintf(sirPuncteMoale, "4003,2, ");
    }
    else
    {
        functiaMoale = 3;
        codMoale = 4006;
        sprintf(sirPuncteMoale, "4006,3,1~0");
    }
}
valRet = CalcGauraCepTaieri(Vt, Vm, pPrelLocalMoale, pPrelLocalTare);
```

```
if(valRet && functiaTare != 4)
    functiaTare = 2;

// se generează forma în puncte de pe corpul moale
char sirTmp[301];
for(i = 0; i < 8; i ++)
{
    sprintf(sirTmp, "%lf,%lf,%lf", pPrelLocalMoale[i][0],
            pPrelLocalMoale[i][1], pPrelLocalMoale[i][2]);
    strcat(sirPuncteMoale, sirTmp);
}

// se generează forma în puncte de pe corpul tare
sprintf(sirPuncteTare, "4007,%hi, ", functiaTare);
for(i = 0; i < 8; i ++)
{
    sprintf(sirTmp, ",%lf,%lf,%lf", pPrelLocalTare[i][0],
            pPrelLocalTare[i][1], pPrelLocalTare [i][2]);
    strcat(sirPuncteTare, sirTmp);
}

// se elimină prelucrările care ajung în exteriorul corpului tare din
// cauza cepului generat
CurataPrelucrariCapat_Cep(Vt, pPrelLocalTare,
                            fataContact->poly->normal,
                            fataContact->poly->D, capatTare);

lPrelCode = new ListPrelCode;
strcpy(lPrelCode->code, sirPuncteMoale);
lPrelCode->type = DeterminareTipPrelucrare(codMoale, functiaMoale);

// dacă prelucrarea pe corpul moale nu este în exterior se inserează pe
// corp
if(!AnalizeazaPozitiaUneiPrelucrari(lPrelCode->code, Vm))
{
    lPrel = InsertListPrel(Vm, lPrelCode);
    if(lPrel)
        lPrel->display_mode = VISIBLE;
}
```

```

lPrelCode = new ListNewCode;
strcpy(lPrelCode ->code, sirPuncteTare);
lPrelCode ->type = DeterminareTipPrelucrare(4007, functiaTare);

// se inserează prelucrarea pe corpul tare
lPrel = InsertListPrel(Vt, lPrelCode);
if(lPrel)
    lPrel->display_mode = VISIBLE;
return 0;    // succes prelucrările au fost generate corect
}

```

Clasa CCepParam este folosită la determinarea formei parametrice a prelucrărilor pe corpuri. Clasa CListNewCode se folosește pentru a stoca lista de șiruri de prelucrări pe corpuri. CVector se folosește în cadrul algoritmului la operații vectoriale.

Funcția GenerareCep() are rolul de a genera codul în puncte a unui cep pe corpul Vt și codul în puncte a găurii de cep corespunzătoare pe corpul Vm. Codurile generate depind de poziția relativă a celor două corpuri unul față de celălalt.

Parametrii locali ai funcției au următoarea semnificație:

pLocalMoale – punctele prelucrării de pe corpul moale exprimate în sistemul local al corpului moale

pGlobalMoale – punctele prelucrării de pe corpul moale exprimate în sistemul global

pLocalTare – punctele prelucrării de pe corpul tare exprimate în sistemul local al corpului tare

pGlobalMoale – punctele prelucrării de pe corpul tare exprimate în sistemul global

minX – valoarea minimă a coordonatei x a punctelor prelucrării pe axa x a corpului moale

maxX – valoarea maximă a coordonatei x a punctelor prelucrării pe axa x a corpului moale

fataContact – conturul poligonal închis obținut prin intersectarea muchiilor corpului tare cu fața din corpul moale pe care se v-a pune gaura de cep

pContur – cele 4 puncte ale conturului feței de contact în sistem global

fataContact – poligonul construit pe baza lui pContur

pConturLocalMoale – cele 4 puncte ale conturului feței de contact în sistemul local al corpului moale

prelCapat – specifică capătul corpului moale de care este mai apropiată gaura de cep (BEGIN, END)

capatTare – capătul corpului tare care este afectat de prelucrare

unghi– este unghiul dintre cele două corpuri

sirPuncteMoale – forma în puncte a prelucrării pe corpul moale

sirPuncteTare – forma în puncte a prelucrării pe corpul tare

maxYTare – valoarea maximă a coordonatei y a punctelor conturului în sistemul local al corpului tare

maxZTare – valoarea maximă a coordonatei z a punctelor conturului în sistemul local al corpului tare

BSTare – litera corespunzătoare feței de referință a prelucrării pe corpul tare

IndexBSTare – indexul feței de referință (C = 2, D = 3, E = 4, F = 5)

IPrelCode – clasă ce stochează forma în puncte a prelucrării

IPrel – un nod în lista de prelucrări memorată pe corp

codMoale – numărul corespunzător codului prelucrării pe corpul moale (4xxx)

functiaMoale – numărul corespunzător funcției prelucrării de pe corpul moale

functiaTare – numărul corespunzător funcției prelucrării de pe corpul tare

pOrdonate – punctele conturului ordonate astfel încât direcția 1-2 să fie de-a lungul axei corpului moale

pAles – direcția cepului aleasă de către utilizator

piMoale – planele formei brute a corpului moale

piTare – planele formei brute a corpului tare

indiceTare – indicele din tabloul de plane piTare corespunzătorul planului formei brute care este orientat către fața de contact

piComun – lista de plane din planele celor două forme brute care au aceeași direcție și sens

deplIPrel – poate fi BEGIN sau END și specifică capătul prelucrării (punctele 1 și 4 sau 2 și 3) unde se aplică deplasamentul prelucrării dacă aceasta este aproape de un capăt al corpului moale

indicePunctDeplasat – indicele punctelor prelucrării cărora li se va aplica deplIPrel (poate fi 4 sau 7 pentru un capăt sau 5 sau 6 pentru celălalt capăt)

La începutul funcției se determină conturul de contact al celor două corpuri. Apoi unghiul dintre ele.

Următorul pas este de determinare a capătului la care se va genera cepul pe corpul tare. Pe baza punctelor conturului, trecute în sistemul local al corpului tare, și al opțiunii selectate de utilizator se determină orientarea cepului pe corpul tare. În BSTare se obține litera corespunzătoare feței de referință a cepului.

Funcția `ConstrPuncteCep()` determină cele 8 puncte prin care se codează cepul. Dacă utilizatorul a ales opțiunea de selecție a feței de referință se propun spre alegere cele două posibilități de generare a prelucrării.

În continuare se determină pe baza unghiului dintre cele două corpuri și a poziției corpului tare față de capetele corpului moale dacă se aplică un deplasament suplimentar specificat de utilizator sau nu. Acesta se aplică numai dacă corpul tare este la unul din capetele corpului moale și cele două corpuri sunt perpendiculare unul pe celălalt. Dacă se aplică un deplasament suplimentar funcția din codul prelucrării este 4. Dacă nu funcția se determină pe baza celor 8 puncte de codare.

Următorul pas este de trecere a punctelor cepului în sistemul local al corpului moale pentru a coda gaura de cep. Ordinea lor se inversează corespunzător codului în puncte. Se determină funcția codului prelucrării pe baza poziției găurii de cep față de corpul moale. (adică dacă este de-a lungul axei corpului moale sau perpendicular pe ea)

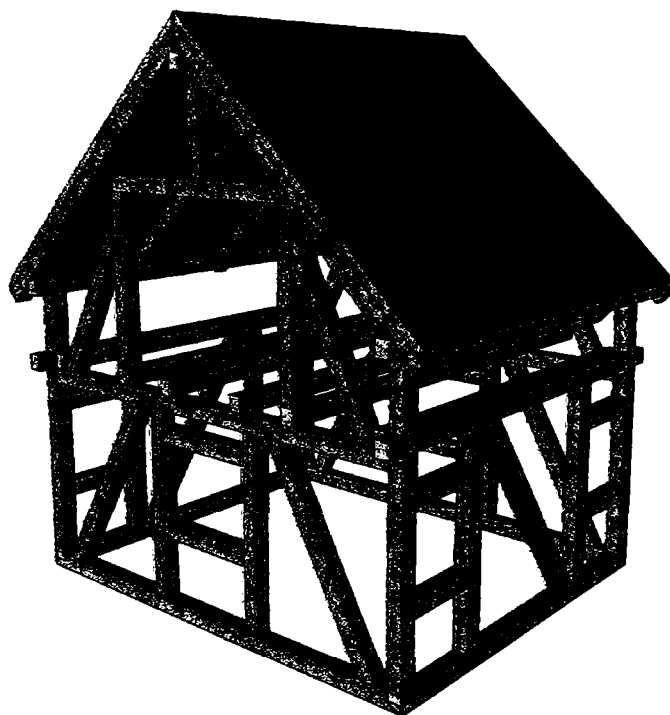
Dacă corpul tare are și tăieri la capătul la care s-a generat cepul se micșorează gaura de cep conform acestor tăieri prin intermediul funcției `CalcGauraCepTaieri()`.

Prelucrările de pe corpul tare care ar elimina cepul creat se elimină cu funcția `CurataPrelucrariCapat_Cep()`. Dacă gaura de cep este în exteriorul corpului moale ea nu se v-a genera. Poziția găurii se determină cu funcția `AnalizeazaPozitiaUneiPrelucrari()`

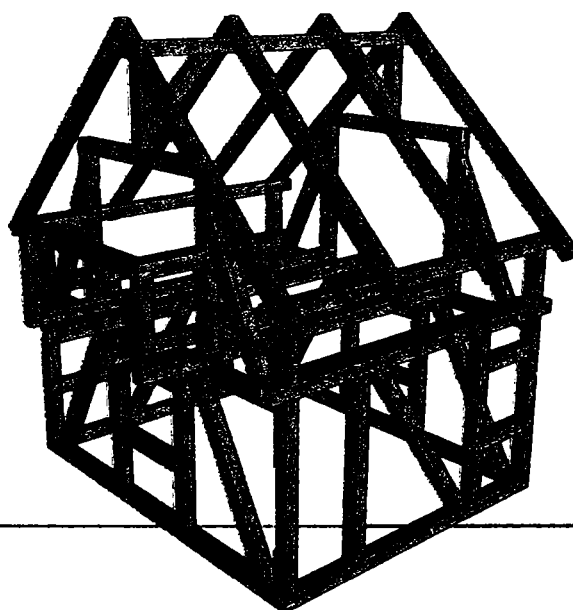
În final se generează șirurile de prelucrări în puncte pe cele două corpuri și se inserează pe acestea.

Anexa 2

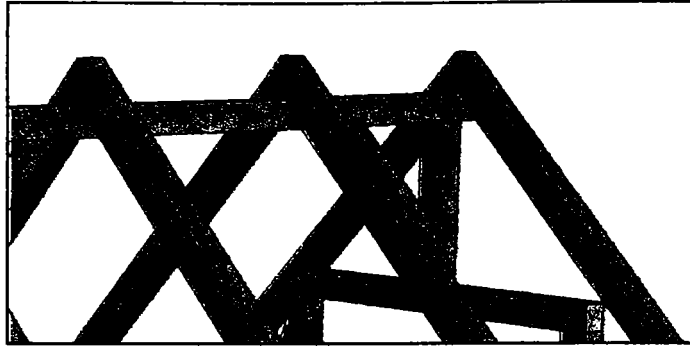
Conține reprezentări ale unei construcții în întregime și cu unele detalii.



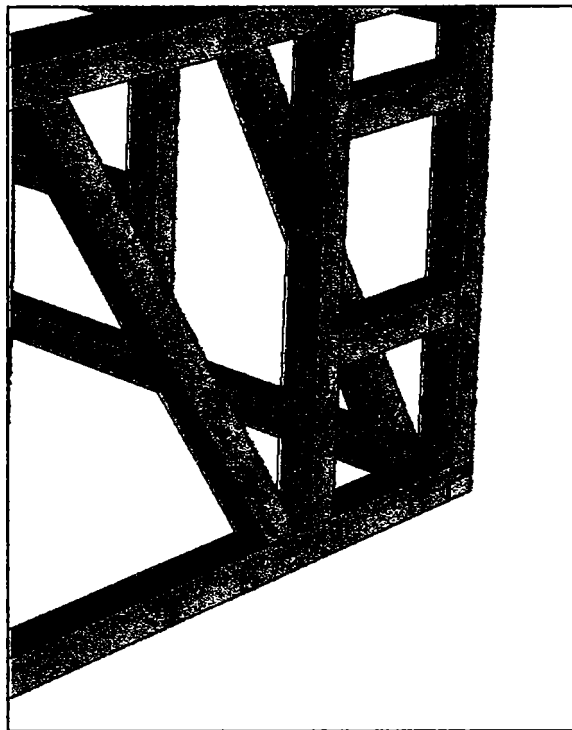
A1: Structura de lemn a întregii construcții cu tot cu acoperiș



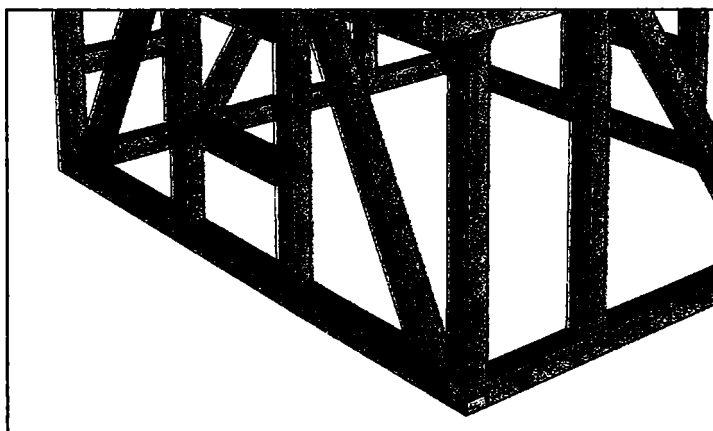
A2: Structura de lemn a întregii construcții fără acoperiș



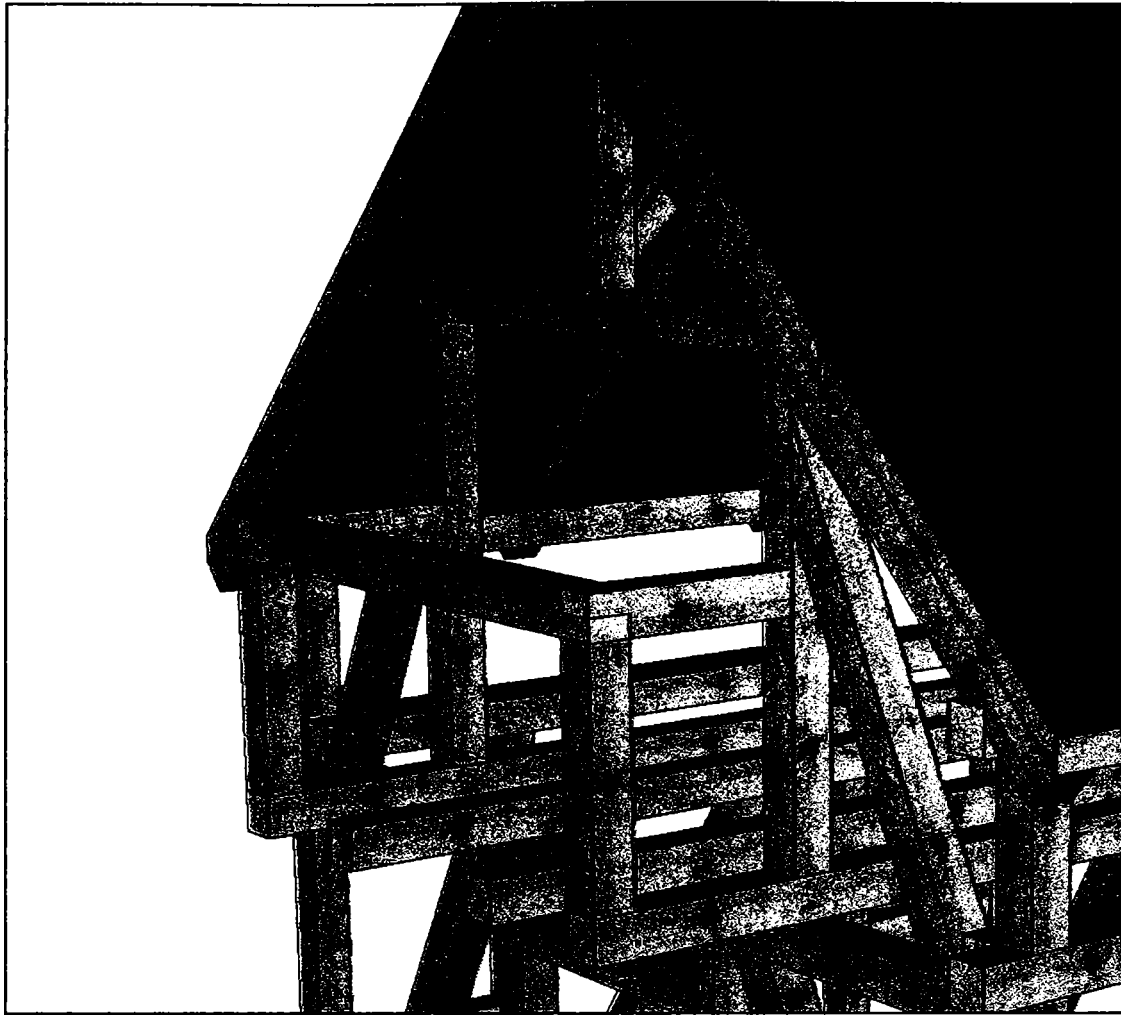
A3: Detaliu coamă acoperiș



A4: Detaliu colț dreapta jos



A5: Detaliu colț față jos



A6: Detaliu balcon mansardă

Definiții

Conține definiții ale termenilor utilizați pe parcursul tezei:

- *bară* (în sensul specific utilizat în teză) – forma inițială a elementului de construcție
- *corp* (în sensul specific utilizat în teză) – forma finală a elementului de construcție
- *forma brută* – paralelipipedul care circumscrie corpul utilizat în construcție pe cele trei dimensiuni: lungime, lățime și înălțime
- *fețele laterale* – suprafețele forme brute care sunt paralele cu axa corpului
- *fețele de capăt* – suprafețele forme brute care sunt perpendiculare pe axa corpului
- *sistem local de referință al unui corp* – este un sistem ortogonal care se atașează unui corp când este generat. Sistemul nu se poate modifica și nu-și poate schimba nici poziția față de corp. Corpul este întodeauna în cadranul I al sistemului.
- *sistem local de referință al unei fețe* – este un sistem ortogonal atașat fiecărei fețe a forme brute. Axele x și y se definesc în planul feței astfel încât fața să fie complet în cadranul I al sistemului. Axa z are direcția normalei feței și sensul către corp.
- *fețele prelucrărilor* – sunt fețele care rezultă în urma prelucrării forme brute
- *forma în puncte a unei prelucrări* – modul de definire internă a unei prelucrări
- *forma parametrică a unei prelucrări* – descrierea pentru utilizator
- *prelucrarea* (în sensul specific utilizat în teză) – este o înșiruire de parametri despărțiți prin virgulă care descriu o operație ce trebuie efectuată pe o bară pentru a putea fi integrată în cadrul unei construcții
- *operație booleană* – operația de modelare a entităților geometrice (de ex. a barelor). Poate fi de tip reuniune, intersecție și diferență. În cazul diferenței una din bare este tare (nu suferă modificări geometrice) și cealaltă este cea moale (cea care suferă modificări geometrice). În urma operației pot să rezulte unul sau mai multe corpuri.
- *roboți* (în sensul specific utilizat în teză) – sisteme automate robotizate
- *șir mașină* – forma în puncte ajustată (astfel încât punctele prelucrărilor să fie situate pe corpul prelucrat) a unei prelucrări care va fi transmisă roboților

Referiri în literatura de specialitate

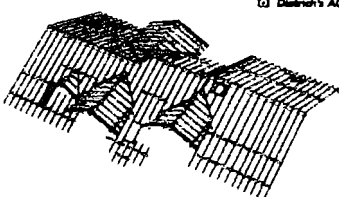
Aplicația *DHP-Bauwerk* a firmei Dietrich's AG este destinată proiectării construcțiilor moderne în lemn și a fabricației asistată de calculator. literatura de specialitate și din pliantele de prezentare a produsului *DHP-Bauwerk*.

În figurile de mai jos, se prezintă extrase din pliantele de prezentare a produsului *DHP-Bauwerk*, evidențiind simplitatea utilizării și eficiența activității de generare și gestiune a prelucrărilor mașină în cadrul aplicației.

BESCHLEUNIGT IHRE ARBEITSVORBEREITUNG

ABBUND KOMPLETT Dialoggeführtes CAD/CAM-Programm für den Holzbau.

Umfangreiche Bearbeitungsassistenten für Dach- und Wandkonstruktionen. Grafische Dachausmittlung mit und ohne Profil, Dachneigungsänderungen in der Fläche, intelligente Gaubenbibliothek mit Bearbeitungsassistenten, schnelle Massenermittlung, Tragwerkseingabe aller Holzler, Spezielle Funktionen für Fachwerk und Holzrahmenbau, Bauteil- und Konstruktionsbibliotheken mit Makrofunktion, Dachdeckermodul für die Eindeckung, Aufmaßlisten für Wände und Dachflächen, Ziegellisten, Holzlisten, Planausgabe mit AUTO-Plan CAD, Integrierte Datenübergabe aller Bearbeitungen an den Postprozessor DIMAS. Mögliche Schnittstellen zu Architektur CAD (z.B. DTH, Dxf, DNS).

 Dietrich's AG

ABBUND KONSTRUKTION Dialoggeführtes 3D-CAD / CAM-Programm mit höchstem Bedienungskomfort.

Schnelle Bearbeitungsassistenten für DACH-WAND-DECKEN Konstruktionen. Wie Abbund Komplett, jedoch mit erweiterten Funktionen für die komfortable grafische Detailbearbeitung. Zusätzlich: windschiefe Dachflächen, freie Trauf-, Ortsgangveränderungen, grafisch, Dachverschnidungen, Bibliothek für Dachaufbauten etc. Aufmaßlisten, Ziegellisten, optimierte Holzliste, Sagellisten, Bestellisten, Plan-

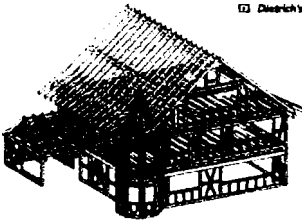
SPART WERTVOLLE

und Abrufen von eigenen Konstruktionen in freien Bibliotheken inklusive Maschinenbearbeitungen, die von Bauten zu Bauteilen vererbt werden. Frei definierbare Modell-Organisations-Strukturen (MOS-Technik), umfangreiche Bauwerks-, Verwaltungs- und Organisationsfunktionen. Integrierte Datenübergabe aller Bearbeitungen an den Postprozessor DIMAS. Fotorealistische Visualisierung, Planausgaben in DIPLAN CAD, Stücklisten, Bestelllisten, Holzlisten etc., umfangreiche Schnittstellen optional.



DIWAND 3D Dialogunterstütztes CAD-Programm zur Erstellung beliebiger Wandkonstruktionen, die ideale Ergänzung zum Abbund in Richtung Fachwerk.

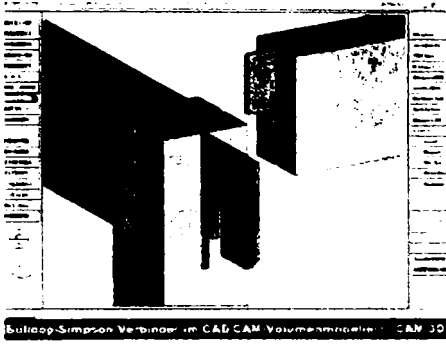
Fachwerk und Ständerbauweise auf frei gestaltbaren Grundrissen und Stockwerken, freie Positionierung und komfortable Endbearbeitung aller Holzler mit

 Dietrich's AG

Revista *Quadrige*, prin numărul 3 din 1999, remarcă prezența la târgul internațional Ligna de la Hanovra a produsului *DICAM*. Figura următoare prezintă un extras al acestui articol.

Firmen und Neuheiten

DICAM - ein neues 3D-Fertigungssystem für den Holzbau



Auf der Liveshow des Woodtech, im Holz- und Holzwirtschafts-Messe-Milieu in Hannover, ist anhand der „Ligna“ Tagungen bei München ins Auge gefasst worden **Dietrich's DHP** seine Rolle als international führender Entwickler und Hersteller von 3D-CAD/CAM-Systemen in der Holzbranche weiter ausbauen können. Unter anderem beliebten Anfragen von größeren Bauwerklichen Holzbetrieblern das Gesamt. Mit der neuen Produktreihe DICAM und DICAM für Zimmerer präsentiert Dietrich's DHP aber auch ein hochmodernes 3D-Fertigungssystem für den gesamten Holzbau mit einer schier unerschöpflichen Vielfalt an Konstruktionsmöglichkeiten. Für den handwerklichen Bereich eröffnen dabei Detailkonstruktionen für Anschlüsse mit Stahlteilen und Holzverbindungen oder vielfältige Möglichkeiten im Wintergarten-, Balken- und Carport-Bau neue Dimensionen. Den professionellen Lohnbetrieblern besetzen dagegen wohl eher die umfangreichen und sehr flexiblen Holzverbindungen die in jeder Ausführung und Position bis auf die Abwandlung gebracht werden können. Die Anfragen aus der internationalen Holzbaufertigung sind heute konzentriert stark schieflich auf umfassende Systemlösungen – von der Architektentelung mit 2D-CAD-Lösungen der Dietrich's Systempartner bis hin zu den vollständigen AW- und schweißtechnischen 3D-CAD/CAM-Holzbauprodukten DICAM.

Pavilionul din figura următoare a fost prezentat la Expo 2000. El a fost proiectat cu ajutorul aplicației *DHP-Bauwerk*.



Der Umgang mit schwierigsten Konstruktionen im Holzbau ist unser Geschäft!

Machen Sie es auch zu Ihrem! Wir helfen Ihnen dabei.

Dietrich's AG
KLEIN, LAMPE & ZIEGLER

Postweg 4
D-82024 Taufkirchen
Fon: +089 6144 21 0
Fax: +089 6144 21 44

info@dietrichs.com
www.dietrichs.com

În numărul 1/2001 al revistei Dietrich's AG 3D-CAD/CAM für den Holzbau se prezintă tipuri mai complexe de îmbinări între barele unei construcții din lemn. Decupajele (ferestre, uși) din zidurile prezentate sunt realizate cu ajutorul operațiilor booleene prezentate în cadrul tezei.

AUS
BRINGT

3D IN DIETRICH'S

Gute Nachrichten für über 10.000 Architekten und alle Holzhausbauer

Dietrich's-System kann Daten mit Allplan von Nemetschek austauschen. Perfekter Holzbau und vollendete Fertigungstechnik können nun auch aus Allplan heraus realisiert werden. In Allplan gezeichnete Bauwerke werden als 3D-Modell in Dietrich's-System eingelesen und können dort, mit allen Raffinessen des Holzbaus technisch ausgearbeitet werden. Noch nie war die Zusammenarbeit zwischen Holzbauer und Architekt so einfach. Geschosse, Wände, Fenster und Türen etc. werden als 3D-Volumenkörper in Dietrich's-System integriert. Mit den Modulen DIWAND 3D bzw. DIWAND HRB (Holzrahmenbau) werden anschließend mehriagige Wandkonstruktionen automatisch erzeugt. Fachwerk-, Brettstapel oder Tafelbaukonstruktionen sind ebenfalls möglich. Das integrierte Bibliothekssystem bietet hier vielfältige Möglichkeiten. Nemetschek und Dietrich's bieten mit diesem Paket eine professionelle Lösung für den modernen Holzbau an. Interesse? - Infos zum Ortstarif: Tel. 01805-214 182

DIMAS STEUERT

Die Gerüchteküche brodelt mal wieder um die Abundanlage von Ballier & Zembrod. Ein neues Einzelstabprogramm soll die Möglichkeiten der Maschine besser ausnutzen. (Ähnliches kennt man auch von Schmiedler.)

Für Dietrich's ein Grund mehr den Postprozessor DIMAS um einen Maschinentyp zu erweitern. Die Vorarbeiten laufen auf vollen Touren, so dass in Kurze die Testläufe gefahren werden können. Kein Grund zur Sorge besteht für alle Betriebe, die noch eine B&Z mit dem "alten" Einzelstab einsetzen. Dietrich's wird weiterhin diese Maschinen an-



steuern und auch in der Softwarepflege berücksichtigen. Wer sich also heute für eine B&Z entscheidet, kann getrost auf Dietrich's setzen


DIETRICH'S STEUERT

Eine interessante Lösung für alle Betriebe, die größere Mengen Stangenware einkurzen, bietet die KDS Kappsäge von Schneider Zimmereimaschinen. Noch interessanter ist jedoch, dass die KDS durch die Datenübergabe vom Dietrich's-System zu einer leistungsfähigen Kappanlage wird. Die Ansteuerung erfolgt direkt aus der Holzliste heraus. Der Vorteil dabei ist, dass die Zugschnittlängen vorher noch optimiert werden können. Das garantiert geringen Verschnitt. Besonders interessant ist die KDS + Dietrich's Ansteuerung für alle Zimmereien, die noch nicht mit Dietrich's Software arbeiten. Die Ansteuerung wie auch die Holzliste sind eigenständige Programme, die notfalls auch zusammen mit anderen Holzbauprogrammen eingesetzt werden können. Infos unter: www.schneider-zimmereimaschinen.de und www.dietrichs.com

DIMAS STEUERT

Noch vor der Auslieferung der ersten Anlagen hat sich unser Entwicklungsteam schon mit der Ansteuerung der neuen Plattenbearbeitungsanlage von Hundegger beschäftigt. Mit einer Pilotinstallation konnten in kürzester Zeit die Programmierarbeiten abgeschlossen werden. Schon in der Testphase sind annähernd 50 Häuser mit Hilfe der neuen Ansteuerung gefertigt worden. Die Möglichkeit, auch zusätzlich im Bedienprogramm der PBA individuelle Bearbeitungsmakros zu speichern, ergänzt die sehr leistungsfähige und flexible Übergabe. Somit werden fast alle Bearbeitungswünsche in Maschinenbewegungen umgesetzt. DIMAS PBA ist ein Sahnestückchen für alle Holzleimbauer oder Betriebe mit ähnlichem Profil, die viel mit Leimbändern und Massiv-



În numărul 4/2000 al revistei Dietrich's AG 3D-CAD/CAM für den Holzbau se specifică avantajele prelucrării automate în cadrul unui acoperiș.

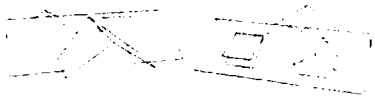
Dietrich's AG

3D-CAD/CAM für den Holzbau

JAHRES-UPDATE

Das zweitausendste Jahr neigt sich langsam dem Ende entgegen und bei Dietrich's laufen die Vorbereitungen wie jedes Jahr auf Hochtouren. Denn wie gewohnt wird das große Update für alle Kunden, die einen gültigen Service- und Betreuungsvertrag haben, in den nächsten Wochen ausgeliefert. Die neueste Version des Programmsystems hält wieder viele Highlights für den Holzbauer bereit.

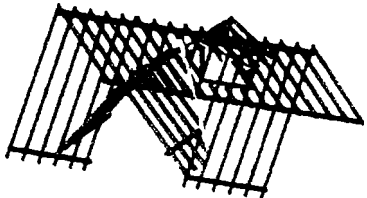
Dachunterschneidung:
Bei der neuen Dachausmittlung sparen Sie viel Zeit! Die Dachunterschneidung in der neuesten Version geschieht vollautomatisch. Dies gilt für Querbauelemente und für die eingerechneten Gåuben im Trauf- und Gåabelbereich. Die zusätzliche kleine



Dachfläche vor der Gåube wird automatisch abgesenkt und mit der Hauptdachfläche verschmolzen. Diese neue Dachunterschneidung bringt in folgenden Bereichen Vorteile:

- die Flächenberechnung im Aufmaßprogramm ist jetzt noch umfangreicher
- die Sparren und Pfetten werden sofort auf die richtige Länge geschnitten und
- eine Dachlattenkorrektur in der freien Dachflächenverformung entfällt.


Verschneidung:
Auch hier wurden viele Funktionen erweitert. So können sowohl das Hauptdach als auch der Aufbau beim Aufsetzen Hålzer enthalten. Diese Hålzer lassen sich beim Wåhlen des zu positionierenden Punktes anwåhlen. Danach wird die Håhe des gewåhlten Punktes angezeigt. Hier kann man jetzt, wie beim Aufsetzen einer Varia-Gåube, in die Perspektive wechseln. Dort lsst sich der Aufbau zustzlich in der Håhe verschieben.



Beim Platzieren kann ein beliebiger Bezugspunkt gewhlt werden, der die Positionierung des Aufbaus erleichtert. Mit dem neuen Knopf "UP/DOWN" kann man die letzte Verschneidung rckgangig machen.

Verschneidung mit Erhalt der Hålzer
Der Aufbau kann bei der Verschneidung Hålzer enthalten. Diese knnen bernommen und danach ohne Einschrnkungen weiterbearbeitet werden, wie auch alle anderen Hålzer des Hauptdaches.

Automatische Verschneidung:
In der Bearbeitung der Verschneidung besteht jetzt die Mglichkeit, die Verschneidung der Dachflchen automatisch ausfhren zu lassen. Es knnen nur das Hauptdach, nur der Aufbau oder beide verschritten werden. Die Nachbearbeitung der Flchen von Hand ist weiterhin mglich. Bei dieser Art der Verschneidung werden Linientypen automatisch erkannt und zugewiesen. Der neue Knopf "KONTROLLE" im Bearbeitungsreich der Dachverschneidung hilft bei der Kontrolle der Linientypen. Mit diesem Knopf erscheint eine Liste aller mglichen Linientypen. Beim Durchblttern dieser Liste werden jeweils die passenden Linien violett markiert.



Tot în numărul 4/2000 se evidențiază capacitatea sistemului de a prelucra și barele ce aparțin lucarnelor din acoperiș.

DIETRICH'S - NEUES SYSTEM

Verschneidung von Varia-Gauben:
 Beim Aufsetzen von Varia-Gauben kann man wählen, ob nach der alten Methode aufgesetzt oder automatisch mit der betreffenden Dachfläche verschnitten werden soll.

Es können bei der Verschneidung alle Hölzer der Gaube, auch Grat- und Kehlsparren, übernommen und vollständig weiterbearbeitet werden. Die Pfetten, Brüstungsriegel und Pfosten in der Gaubenfront lassen sich nachträglich bearbeiten und ergänzen. Zangen, Schalungsflächen, Hobel­flächen und Ziegel lassen sich auf den Gauben­flächen wie auf dem Hauptdach eingeben. Das Loch in der Hauptdachfläche wird wie bisher über die Locheingabe in der Varia-Gaube erzeugt. Die Gaubenöffnung im Hauptdach ist nachträglich änderbar.



Planprogramm, Einzelblattprogramm:
 Legenden können nun frei definiert werden. Selbst erstellte Linien, Texte und Logos in beliebigen Kombinationen werden als Legende abgelegt. In Plänen können diese Legenden frei positioniert oder beim Anlegen des Planes automatisch eingefügt werden. Dabei werden besondere Texte, sogenannte Kennworte, in der Legende automatisch ausgefüllt; Beispiel: Bauvorhaben. Nicht nur die Legenden, sondern auch die übrige Plangestaltung gewinnt durch die Verwendung verschiedener Schriften. Durch Einstellung der Schriftbreite läßt sich nun auch erheblich mehr Information unterbringen.

Neuerungen für alle Maschinen
 In DICAM können Schalungsausklüngen für Grat- und Kehlsparren mit entsprechenden Bearbeitungen erzeugt werden. Der Postprozessor DIMAS sorgt dafür, dass diese sinnvoll auf der Maschine umgesetzt werden. Meist werden die Schalungsausklüngen dabei mit Sägeschlitzen ausgearbeitet (P10, K1, K2). Kurze Schalungsausklüngen können auch vollständig gefräst

BIBLIOGRAFIA

3D-Calc Software for IBM-PC And Compatibles, Markt & Technik

Adams L. (1989) - *High Performance CAD Graphics*, Windcrest

Adams L. (1991) - *Lee Adams' Visualisation Graphics in C*. Windcrest

Adams T. (1992) - *Writing DOS Device Drivers In C*, Academic Press

Angell I. O. (1990) - *High Resolution Computer Graphics Using C*. Macmillan London

Artwick, Bruce A. (1985) - *Microcomputer Displays, Graphics and Animation*, Prentice - Hall, Inc; Englewood Cliffs, New Jersey

Arvo J. (1991) - *Graphics Gems II*, Academic Press, Inc

Bajaj C., Hoffman J., Hopcroft R., Lynch (1988) - *Tracing Surface Intersections Computer Aided Design*

Barsky B. (1988) - *Computer Graphics and Geometric Modeling Using Betasplines*, New York

Barr A. H. (1984) - *Global and Local Deformations of Solid Primitives*, Computer Graphics

Barkakati N. (1990) - *Microsoft C Bible*, Sams Publishing

Blackwell (no. 2, June 1998) - *Computer Graphics forum*

Bouma W.J., Vanecek Jr G. (vol. 26, no. 6, June 1994) - *Modelling contacts in a physically based simulation*, Computer-Aided Design, Butterworths

- Cunningham St. (1992) - *Computer Graphics Using Object-Oriented Programming*.
Wiley&Sons
- Davies B.L., Robotham A.J., Yarwood A. (1991) - *Computer Aided Drawing and Design*.
Chapman&Hall
- Dietrich's Neues System (no. 4, 2000) - *Bauen mit Holz*
- Dietrich's Neues System (Heft 30, 2000) - *Grafische Bedienungsführung, Holzkurier*
- Elsevier oct, nov (1998) - *Computer Aided Design*
- Emmer U., Gerner W., Muresan Voichita (1994) - *DICAM - Wood-manufacturing CAD*,
International Conference on Technical Informatics-Conti '94 , Tehnical
University of Timișoara
- Encarnacao J., Schlechtendahl E. G. (1983) - *Computer Aided Design*. Springer Verlag,
Berlin
- Flaquer J., Carbajal A., Mendez M.A. (vol. 19, no. 5 June 1987) - *Edge - Edge
Relationships in Geometric Modelling*, Computer-Aided Design,
Butterworths,
- Franz K. (1995) - *Grundwissen des Zimmers*, Bruderverlag Karlsruhe
- Fritz K. (1991) - *Der Zimmerpolier*, Bruderverlag Karlsruhe
- Gardan Y. (1991) - *La CFAO. Introduction, techniques, et mise en oeuvre*. Hermes, Paris
- Gui J., Mäntylä M. (vol. 26, no. 6 June, 1994) - *Functional Understanding Of Assembly
Modelling*, Computer-Aided Design, Butterworths
- Hoffman, Cristiph M., Hopcroft, John E. (vol. 19, no. 3 April, 1987) - *Geometric Ambiguities
in Boundary Representations*, Computer-Aided Design, Butterworths

Holzrahmenbau (1992) - *Bund Deutcher Zimmermeister*

John M. Snyder (1992) - *Generative modeling for computer graphics and CAD*,
ACADEMIC PRESS INC

Kajiva J. T., and Timothy L. Kay (July, 1989) - *Rendering Fur with Three Dimensional
Textures*, Computer Graphics

Leigh (1993) - *Solid Modeling With Autocad*, Academic Press

Lignarius (no. 4, 2000) - *Dietrich's AG 3D-CAD/CAM Für den Holzbau*

Lignarius (no. 1, 2001) - *Dietrich's AG 3D-CAD/CAM Für den Holzbau*

Dumitru M. (1985) - *Construcții moderne din lemn*, Editura Tehnică, București

Moldoveanu Florica, Racoviță Zoea, Petrescu Ș., Hera G., Zaharia, M. (1996) - *Grafica pe
calculator*. Teora, București

Mureșan I. (1994) - *Solid modeler for CAD*, International Conference on Technical
Informatics-Conti '94, Technical University of Timișoara

Mureșan I., Savii G. (1996) - *Concepte grafice in CAD/CAM. Tehnici interactive*, Presa
Universitară Română

Mureșan Voichița (1996) – *Modelarea de solide pentru proiectarea asistată de calculator*

Mureșan Voichița, Ogârcin Geraldine (1999) – *Codarea prelucrărilor standard pentru
sisteme CIM în industria lemnului*, Analele Universității Banatului din
Timișoara, vol. 115, pg. 130-134 (ISSN 1224-8576), Editura Presa
Universitară Română

Nielsen K. W. (1995) - *Software development with C++, Maximizing Reuse with Object
Technology*, Academic Press

Ogârcin Geraldine (1995) - "Semantic pitfalls" Sesiunea de comunicări științifice ACADRES '95, Fundația universitară "Banatul", Universitatea Banatului, 5 mai 1995

Ogârcin Geraldine (14/1999) – *Algoritm de stabilire a poziției relative a unui punct față de un poligon, respectiv relativ la un volum*, Raport tehnic intern

Ogârcin Geraldine (23/1999) – *Algoritm pentru intersectarea elementelor 3D*, Raport tehnic intern

Ogârcin Geraldine (1999) – *Tehnici de depanare*, Analele Universității Banatului din Timișoara, vol. 115, pg. 181-187 (ISSN 1224-8576), Editura Presa Universitară Română

Ogârcin Geraldine (2000) – *Codification of standard machinery processing for wood*, Fourth International Conference on Technical Informatics CONTI'2000

Ogârcin Geraldine (18/2000) – *Algoritm de stabilire a poziției relative a două segmente*, Raport tehnic intern

Ogârcin Geraldine (2000) – *Codification of standard machinery processing for wood*, Buletinul științific al Universității "Politehnica" din Timișoara, Seria Automatică și Calculatoare vol. 45/2000, No. 2, pg. 141-146 (ISSN 1224 – 600x)

Ogârcin Geraldine (32/2001) – *Codarea automată a prelucrărilor în cadrul unui acoperiș*, Raport tehnic intern

Paul S. H. (1994) - *Graphics Gems*, AP Professional

Pla-Garcia N. (vol.12, no.3, 1993, pag.C153) - *Boolean Operations and Spatial Complexity of Face Octrees*. In: Computer Graphics Forum

Press W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1986) - *Numerical recipes*, Cambridge University Press, Cambridge, England

- Rogers D. F. (1985) - *Procedural Elements for Computer Graphics*, McGraw-Hill
- Rogers T. Stevens (1994) - *Object – Oriented Graphics Programming in C++*, AP Professional
- Quadriga (no. 5, 1999) - *DICAM – ein neues 3D-Ferigungssystem für den Holzbau*
- Savii G., Ogârcin Geraldine (1994) - *Solid Boolean Operations Implementation on Microstation Shell*, International Conference on Technical Informatics-Conti '94, Tehnical University of Timișoara
- Savii G., Ogârcin Geraldine (2000) – *Special problems of a solid modeler for standard machinery processing*, Buletinul științific al Universității “Politehnica” din Timișoara, Seria Automatică și Calculatoare vol. 45/2000, No. 2, pg. 147-152 (ISSN 1224 – 600x)
- Savii G. (1997) - *Bazele proiectării asistate de calculator, CAD*
- Savii G., Ogârcin Geraldine (2000) – *Special problems of a solid modeler for standard machinery processing*, Fourth International Conference on Technical Informatics CONTI'2000
- Steve M. (1992) - *Writing Solid Code*, Microsoft Press
- Steve M. (1993) - *Writing Solid Code*, Microsoft Press
- Snyder J.M. (1992) - *Generative Modeling for Computer Graphics and CAD*, Academic Press, Inc.
- Taylor D. L. (1992) - *Computer Aided Design*. Addison-Wesley
- Thompson J. F., Z. U. A. Warsi, and C. W. Mastin (1985) - *Numerical Grid Generation*, North-Holland, New York
- Tudor S. (1995) - *Tehnici de programare*, Teora

Zeid I. (1992) - *CAD/CAM Theory and Practice*, McGraw-Hill

Wolfram S. (1991) - *Mathematica: A System for Doing Mathematics by Computer, Second Edition*, Addison-Wesley, Redwood City, CA