

tanulmányok

173 / 1985

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

MOZGÁSPÁLYA GENERÁLÁS BONYOLULT GEOMETRIÁJU
FELÜLETEK 2 1/2D-S NC MEGMUNKÁLÁSÁHOZ

Irta:

KACSUKNÉ BRUCKNER LIVIA

A kiadásért felelős:

DR. VAMOS TIBOR

Fősztályvezető:

NEMES LÁSZLÓ

ISBN 963 311 194 3

ISSN 0324 - 2951

85.473 Alfaprint

TARTALOMJEGYZÉK

BEVEZETÉS	7
1. SZABAD FORMÁJU FELÜLETEK MEGMUNKÁLÁSÁNAK AUTOMATIZÁLÁSA	9
1.1. Az NC technika és az NC programozást segítő első számítógépes rendszerek	10
1.2. Geometriai modellező rendszerek	11
1.2.1 A geometriai modellező rendszerek célja	12
1.2.2 A 3D-s modellező rendszerek típusai	13
1.3. Szabad formájú felületek kezelésének matema- tikai alapjai	15
1.3.1. Általános alapelvek	15
1.3.2. Matematikai módszerek szabad formájú qörbék leírására	17
1.3.3. Matematikai módszerek szabad formájú felületek leírására	22
1.4. Szabad formájú felületek megmunkálásának problémái	26
2. AZ FFS RENDSZER NAGYOLÓ PROCESSZORÁNAK ÁTTEKINTÉSE	28
2.1. Az FFS rendszer megmunkáló processzorai	28
2.2. Az FFS rendszer nagyoló processzorának műkö- dése	30
2.3. A nagyoló processzor algoritmusai	31
3. SZABAD FORMÁJU FELÜLETEK METSZÉSE Z=KONSTANS SIKKAL	33
3.1. A felület megadása, megszorítások	34
3.2. Az eljárás áttekintése	36
3.3. Felületelem metszése sikkal	37

3.4.	Metszésponatok rendezése egy felületelemen belül	38
3.5.	Különböző felületelemeken levő görbék össze- füzése	41
3.6.	Az algoritmus elemzése	46
4.	OFSZET KONTUROK ELŐÁLLITÁSA ZÁRT SIKGÖRBÉKHEZ . .	50
4.1.	A feladat matematikai megfogalmazása	50
4.2.	Az eljárás áttekintése	50
4.3.	Az elemi ofszet görbék meghatározása	51
4.4.	Az elemi ofszet görbék összefüzése	52
4.5.	Ofszet konturok előállítása	56
4.6.	Az algoritmus elemzése	57
5.	OFSZET SIKGÖRBÉK ÁTHATÁSA, EGYESITÉSE	66
5.1.	A probléma felvetése	66
5.2.	Két görbe áthatása	67
5.3.	Sikgörbék körüljárásának meghatározása	70
5.4.	Kiindulási pont típusának meghatározása	72
5.5.	Metszésponatok és érintési ponatok megkülönböz- tetése	74
5.6.	Az algoritmus elemzése	76
6.	SIKBELI SZERSZÁMPÁLYA GENERÁLÁS	79
6.1.	A feladat megfogalmazása	79
6.2.	Első algoritmus	80
6.3.	Javitott algoritmus	83
6.4.	Az algoritmusok elemzése	86
6.5.	Az algoritmusok összehasonlitása	87
7.	IMPLEMENTÁCIÓ	90
7.1.	Implementáció az FFS rendszerben	90
7.2.	Numerikus problémák	91
7.3.	Implementáció a BUILD-4 geometriai modellező rendszerben	92

ÖSSZEFOGLALÁS	94
FÜGGELÉK	95
A síkgörbék irányításával kapcsolatos geometriai fogalmak tisztázása	
IRODALOMJEGYZÉK	100

MELLÉKLET:

- 1/ Az FFS CAD/CAM rendszer nagyoló processzorának felhasználói kézikönyve
- 2/ Müanyag flakon formázó szerszámának terve és a nagyolás ellenőrzési fázisainak plotter-dokumentációja
- 3/ FFS munkadarabok fényképei
- 4/ BUILD rendszerrel tervezett alkatrész és megmunkálásának szerszám pályái

KÖSZÖNETNYILVÁNÍTÁS

Köszönettel tartozom Nemes Lászlónak, az MTA SZTAKI GAFO főosztályvezetőjének és Hermann Gyulának, a Kisérleti Üzem osztályvezetőjének, akik támogatták ezt a témát és biztosították a munkához szükséges feltételeket. Szeretnék köszönetet mondani munkatársaimnak, Várady Tamásnak, Gaál Balázsnak és Renner Gábornak, akik a témába bevezettek és munkámat tanácsaikkal folyamatosan segítették, Czirfusz Györgynek, aki segítségemre volt az algoritmusok kidolgozásában, továbbá Lukács Gábornak, aki dolgozatomat értékes megjegyzéseivel javította.

Köszönöm továbbá Várady Tamásnak, Graham Jarednek és Anthony Parkinsonnak, hogy lehetővé tették a BUILD csoport munkájába való bekapcsolódásomat.

BEVEZETÉS

A számítástechnika bevezetése új korszakot nyitott a gépipar fejlődésében, gyökeres változást hozott a hagyományos gépgyártási technológiákban. Ennek a folyamatnak szükséges velejárója volt az új elméleti alapok kidolgozása és ennek eredményeképpen új tudományág születése.

Ez az új tudományág a számítógépes tervezés és gyártás (Computer Aided Design = CAD, Computer Aided Manufacturing = CAM), melynek eredményei közvetlen ipari felhasználásra kerülnek. A gépipari CAD/CAM rendszerek fejlődése napjainkban olyan mértékben felgyorsult, hogy eredményeinek alkalmazása nélkül a gazdaságos, versenyképes termelés szinte lehetetlenné vált. Ezért hazánkban is egyre nagyobb szükség van az ilyen irányú kutató-fejlesztő munkára és az eredmények gyors ipari bevezetésére.

Az alapvető eszközök - NC-gépek, számítógépek - már Magyarországon is rendelkezésre állnak, de az ipari célokra kifejlesztett számítógépes rendszerek száma igen csekély. A hazai fejlesztés napjainkban azért is nagy jelentőségű, mert a szigorú embargópolitika következtében az ilyen típusú rendszerek vásárlása a legfejlettebb tőkés országokból lehetetlenné vált.

Az MTA SZTAKI-ban több év óta folyik olyan számítógépes rendszerek fejlesztése, amelyek a gépipari tervezés és gyártás különböző problémáinak megoldására szolgálnak. Ezek közül a legújabb az FFS CAD/CAM rendszer, amely termékként forgalmazva tényleges ipari alkalmazásra kerül.

Az FFS (Free-Form Shapes) rendszer szabad formájú felületek által határolt testek tervezését és megmunkálását oldja meg. A tervezés interaktív grafikus úton történik. A rendszer műszaki rajzok és mintadarabok alapján történő reprodukcióra és számítógéppel segített formatervezésre egyaránt alkalmas. A megtervezett felületgeometria alapján a technológiai paraméterek beállítása után a rendszer automatikusan állít elő nagyoló illetve simító megmunkáláshoz szükséges NC vezérlőszalagokat.

Jelen értekezés célja bemutatni azokat a számítógépes algoritmusokat és matematikai hátterüket, amelyek alapján az FFS nagyoló processzora a szerszámpályát kiszámítja. A bemutatásra kerülő algoritmusok lényeges tulajdonsága, hogy a gyakorlati felhasználás eredményességén kívül helyességük elméletileg is bizonyítva van. Alkalmazási területük nem szűkül az FFS rendszerre, könnyen adaptálhatók más gépipari CAD/CAM rendszerekbe is. Így például a 4., 5. és 6. fejezetekben leírt algoritmusokat az értekezés szerzője a Cambridge-i Egyetem CAD kutatócsoportjának felkérésére az ott kifejlesztett BUILD testmodellező rendszer legújabb változatába is beépítette.

1. SZABAD FORMÁJU FELÜLETEK MEGMUNKÁLÁSÁNAK AUTOMATIZÁLÁSA

A hagyományos gépipari tervezés folyamatában a konstruktorok arra törekszenek, hogy az előállítandó alkatrészeket egyszerű matematikai testek, és felületek - sík, henger, gömb, kup - segítségével írják le. Erre azért van szükség, mert a hagyományos megmunkálási technológiákkal bonyolultabb felületek csak igen nehezen, sok idő és költség ráfordítással készíthetők el. Sok esetben azonban az alkatrész funkciója megköveteli a bonyolult, szabad formájú felületek beépítését, például az egyenszilárdságúra méretezett alkatrészek, turbinák, hajócsavarok, járókerekek, vagy a különböző prés- és huzószerszámok, ahol a velük megmunkálható anyagok tulajdonságai követelik meg a bonyolultabb lekerekítések, profilok kialakítását. Másik szempont, amely a szabad formájú felületek felé vezeti a tervezőt, az esztétikai igény, amely ma már egyre több területen jelentkezik, a gépkocsik karosszériájától kezdve a különböző háztartási eszközökig.

Ebben a fejezetben áttekintjük azt a folyamatot, amely elvezetett az ilyen szabad formájú (vagy szoborszerű) felületek megmunkálásának automatizálásához, továbbá az ilyen felületek kezeléséhez szükséges matematikai módszereket és technológiai szempontokat.

1.1. AZ NC TECHNIKA ÉS AZ NC PROGRAMOZÁST SEGÍTŐ ELSŐ SZÁMITÓGÉPES RENDSZEREK

A számítástechnika fejlődése és az NC gépek elterjedése a fejlett ipari országokban már a 60-as évek elejére kialakította az igényt és a lehetőséget arra, hogy számítógépes programrendszereket alkalmazzanak a technológiai tervezésben.

A kézi NC programozás még viszonylag egyszerű esetekben is nagy figyelmet és hozzáértést követelő munka, amelynek számítógépes kiváltása lényeges gyorsítást eredményez a megmunkálás előkészítésében, szabad formájú felületek kézi programozása pedig általában megoldhatatlan.

Az első számítógépes programrendszerek egyedi feladatokat oldottak meg, kezdve a különböző mintázatu furatok megmunkálásával, - KIPPS, AID rendszerek [34] - egyre bonyolultabb problémákig, mint a görbevonalu profilok marásához készült PROFILEDATA [34]. Görbült felületek megmunkálására alkalmas első rendszerek a PMT2 és a MILMAP [34] voltak.

Ezek az első programrendszerek nem terjedtek el széles körben, egyrészt mert az alapul szolgáló számítástechnikai eszközök sem voltak elég fejlettek és egységesek, másrészt mert a technológiai környezet is erősen korlátozott volt.

Az egyedi programok fejlesztése mellett felmerül az igény egységes NC előkészítő rendszerre. Ilyen céllal hozták létre az NSZK-ban az EXAPT nyelvet és programrendszert [41], amelynek alapja az USA-ban az 50-es években kidolgozott APT nyelv volt [4]. (APT = Automatically Programmed Tool, EXAPT = Extended subset of APT). Az EXAPT rendszer első

változata csak furógépek vezérlő lyukszalagjának előállítására volt alkalmas, ma már az összes, maximum 2.5 D-s NC gépen történő gyártási feladat (furás, esztergálás, marás, szikraforgácsolás, lángvágás) programozható vele. Ez a világszerte széles körben elterjedt, ma is fejlődő programrendszer a kézi NC programozáshoz képest óriási előrelépést jelentett. Az EXAPT rendszer előnyei:

- az alkatrész geometriájának könnyen érthető megadása
- a megmunkálási folyamat egyszerű leírása
- a leggyakrabban használt gyártási eljárások technológiai adatainak automatikus meghatározása.
- A rendszer FORTRAN nyelven íródott és így szinte bármely számítógépen futtatható.

Az EXAPT rendszer korlátai:

- A szerszámhálya teljes ütközésmentessége nem biztosított.
- Szabad formájú felületek kezelésére nem alkalmas.

Meg kell jegyezni viszont, hogy maga az APT rendszer már a 60-as években tartalmazott bizonyos lehetőségeket szabad formájú felületek leírására és megmunkálására.

A grafikus hardware eszközök fejlődése az NC programozási rendszerekben is éreztette hatását. A 70-es évek elején megjelentek az első rajzoló rendszerek, amelyek segítségével vizuálisan is ellenőrizhetővé vált a megtervezett alkatrészek síkbeli vetülete és a szerszámhálya (GNC [14]).

Ma is igen sok helyen használnak rajzoló rendszert, legtöbb esetben méretezett műszaki rajzok készítésére, illetve egyszerűbb megmunkálási feladatokra. A rajzoló rendszerek lényegesebb korlátja, hogy térbeli információt közvetlenül nem tudnak nyújtani.

1.2. GEOMETRIAI MODELLEZŐ RENDSZEREK

1.2.1. A geometriai modellező rendszerek célja

A gyártási folyamat automatizálási szintjének növekedésével az alkatrészek geometriai leírása egyre részletesebb és összetettebb lett. Csökkent viszont a technológiai utasítások száma, és a megmunkáláshoz szükséges információk meghatározása egyre inkább a számítógép feladatává vált.

Megerősödött az a tendencia, hogy az alkatrészekről olyan számítógépes modell készüljön, amely egy három dimenziós objektumnak (test, felület) kölcsönösen egyértelműen megfeleltet egy adathalmazt a számítógép tároló egységeiben, az ún. belső reprezentációt. A belső reprezentációnak a lehető legkevesebb információ mennyiséget kell tartalmaznia, amely azonban elegendő a leírt objektum bármely, az adott alkalmazási területen szükséges adatának kiszámításához. Bizonyos esetekben a számítási idő csökkentésére érdemes némi redundanciát is megengedni.

A geometriai modellező rendszerek a számítógépes tervezés és gyártás alapvető eszközei. Használatuk alapja a tervezőmérnök vagy technológus és a számítógépes szoros együttműködése. A felhasználó az alkatrész meghatározó adatait valamilyen formában betáplálja a számítógépbe és annak alapján a program létrehozza a belső reprezentációt. A számítógéphez kapcsolt grafikus berendezés (display, plotter), segítségével a tervezett alkatrész vizuálisan ellenőrizhető majd újabb adatok beállításával módosítható. A számítógép és a felhasználó közötti kommunikáció sokféle módon megoldható, lényeges azonban hogy a tervezőrendszer kezelése egy-

szerű és kényelmes legyen.

A geometriai tervezéshez kapcsolódhatnak különböző konstrukciós számítások, például súlypont, térfogat, nyomaték meghatározása, szilárdságtani analízis végrehajtása, amelyet a program az objektum belső reprezentációja alapján végez el.

A tervezés befejezése után a számítógépes rendszer lehetőséget adhat különböző méretezett rajzok generálására (pl. metszeti, perspektivikus, "robbantott" ábrázolás, összeállítási rajzok stb.), csatlakozhat hozzá technológiai tervezőrendszer és az alkatrész megmunkálásához szükséges NC programok automatikus előállítására alkalmas processzor.

1.2.2. A 3D-s modellező rendszerek típusai

A geometriai modellező rendszerek az objektum szintéziséhez felhasznált építőelemek alapján három típusba sorolhatók:

- drótháló modellezés
- felületmodellezés
- testmodellezés.

A pontokkal és éllel történő "drótháló" modellezés a hagyományos alkatrészleírás (ortogonális nézetek, metszetek megadása), természetes kiterjesztése volt és viszonylag egyszerű továbbfejlesztési lehetőséget adott a meglévő számítógépes rajzoló rendszerekhez. Az ilyen modell azonban nem minden esetben határozza meg egyértelműen a testet, igen sok kiegészítő információra van szükség. A belső kezelés nehézsége és a nagyfokú redundancia ellenére még számos helyen használnak, illetve fejlesztenek drótvázás modellező rendszereket, különösen nyomás és feszültségi viszonyok elemzéséhez, de nézeti ábrák és metszetek készítéséhez is.

Természetesen az ilyen rendszerek a görbült felületeket csak síklapu közelítéssel tudják feldolgozni, ami szabadformájú felületek esetén meglehetősen körülményes és problémás módszer lenne.

A különböző felületmodellező rendszerekben a testek általában sík, illetve másodrendű felületek szintéziseként állnak elő, logikai műveletek segítségével. ("ÉS", "VAGY", "NEGÁLÁS") Sok esetben a felületek helyett félterekkel dolgoznak. Ez azért célszerű, mert így az $F(x,y,z)$ felület-egyenlet pozitív, negatív vagy nulla volta jelenti azt, hogy egy adott pont a test belsejében, külsejében vagy a felületén helyezkedik el. A hagyományos analitikus felületek mellett egyre inkább szükségessé vált a szabad formájú felületek kezelése is, amelyeknél a felületegyenletek megadása (ld. 1.3.3.) bonyolultabbá válik és ezáltal a felületekkel történő műveletvégzés is megnehezedik.

Mindezek ellenére ma már a legtöbb felületmodellező rendszer tartalmazza ezt a lehetőséget valamilyen formában, valamint az újabb fejlesztések is ebbe az irányba mutatnak.

A felületmodellező rendszerek közül a legismertebbek a francia UNISURF [6], a nyugatnémet GEOLAN, WMISURF, a svájci SYSTRID [15], [46], az angol NMG [44], DUCT [27], POLYSURF [21].

A háromdimenziós modellezés legújabb és legkomplexebb irányzata az egyszerű építőkövekkel történő testleírás. Ennek lényege, hogy a tervező rendelkezésére áll bizonyos alapelem készlet, amely lehet például téglá, henger, kup, gömb, stb. továbbá bizonyos halmazelméleti műveletek, például metszet, unió, differencia, amelyek segítségével az alapelemekből a kívánt testek előállíthatók. Egyes rendsze-

rek lehetővé teszik az alapelem-konstrukciót is, például pont-, él vagy felületszintézissel, továbbá 2D-s konturok eltolásával illetve elforgatásával. Vannak olyan rendszerek, amelyek technológiai elemeket is tartalmaznak, mint például cső, furat, tömitő gyűrű, profilidomok, borda, menet, beszúrás, stb.

A szabad formájú felületek kezelése egyre inkább beépül a testmodellező rendszerekbe is. Minden nagyobb testmodellező rendszer tartalmaz már valamilyen lehetőséget erre nézve, ezek általában azonban csak közelítő megoldások. A valódi szintézis létrehozása még csak most kezd elméleti kutatási témából gyakorlattá válni [32].

A testmodellező rendszerek közül a legismertebbek az angol BUILD-4 [2], [10], ROMULUS [7], az amerikai GMSOLID PADL [8], a nyugatnémet PROREN'2 [5], a francia EUCLID [13], a japán TIPS [35].

1.3. SZABAD FORMÁJU FELÜLETEK KEZELÉSÉNEK MATEMATIKAI ALAPJAI

1.3.1. Általános alapelvek

Az ipari automatizálás előfeltétele az alkatrészeket leíró geometriai alakzatok egzakt matematikai kezelése számítógépes algoritmusok segítségével. A klasszikus matematikai analízis $Z=f(x,y)$ vagy $f(x,y,z)=\emptyset$ alakú felületleírás függvényei a műszaki gyakorlatban előforduló felületek nagy részét le tudják ugyan írni, de korántsem mindet.

Sokszor van szükség olyan, matematikai szempontból szinguláris, töréseket, végtelen meredekségű, esetleg visszahajló részeket tartalmazó felületek alkalmazására, amelyeknél más matematikai módszerekre van szükség.

Általánosabb felületleiró eljárást kaphatunk ([46]) kétparaméteres vektor-skalár függvények alkalmazásával, melynek formája:

$$x=X(u,v), \quad y=Y(u,v), \quad z=Z(u,v)$$

koordinátáinként egymástól független kétváltozós függvényrendszer, vagy vektoriálisan:

$$\underline{r}=\underline{R}(u,v).$$

Bonyolultabb felületek leírásához legtöbbször ez önmagában nem elegendő. Coons [12] volt az, aki 1964-ben elsőként vezette be a műszaki felhasználás elméletében a felület elemi részenként történő megadásának koncepcióját, amelynek lényege az, hogy a felületet felosztjuk topológiaiilag négyzethálót alkotó, egymáshoz csatlakozó darabokra és minden felületelemre külön-külön adjuk meg, az $\underline{R}(u,v)$ definiáló függvényt. A felhasznált függvényeknek elegendő szabad paramétert kell tartalmazni ahhoz, hogy a szomszédos felületelemek találkozásánál a fizikai vagy esztétikai okokból megkövetelt első vagy magasabb rendű folytonossági kritériumokat teljesíteni tudják.

Gyakorlati szempontból igen lényeges az a követelmény is, hogy a felhasznált módszerek számítógépes feldolgozásra alkalmasak legyenek. Szükség van olyan gépi reprezentációra, amely viszonylag kevés adatot tárol az egyes felület-

elemekről, ám alkalmas az összes szükséges geometriai számítás (pontok, érintők, normálisok) ésszerű időn belüli elvégzésére.

A szabad formájú felületek elmélete szorosan kapcsolódik a szabad formájú görbék elméletéhez, egyrészt azért, mert számos felülettervező módszer görbeinterpolációs illetve approximációs módszerek általánosításaként adódik, másrészt pedig sok esetben a felület előállításához szükség van bizonyos felületi vagy peremgörbék pontos leírására. (ld. [17])

Jelen dolgozat csupán rövid áttekintést ad a világszer-
te legismertebb és a geometriai modellezésben leggyakrab-
ban használt görbe- és felületelőállító módszerekről, az
irodalomban [3], [12], [17], [20], [22], [28] és [42]
foglalkozik részletesen ezzel a témával.

1.3.2. Matematikai módszerek szabad formájú görbék leírására

Már a klasszikus matematikai analízis is foglalkozott
azzal a feladattal, hogy egy görbe bizonyos adatainak ismer-
etében hogyan lehet előállítani a teljes görbét. Ekkor a-
lakultak ki az első interpolációs és approximációs módsze-
rek (Lagrange, Hermite, Taylor [22], [47]), amelyek tovább-
fejlesztése a számítógépes geometria egyik alapfeladata
lett. Az ipari számítógépes formatervezés kialakulása so-
rán fokozatosan derült ki, hogy bizonyos típusu feladatok-
nál milyen jellegű feltételeket kell a létrehozott görbék-
nek teljesíteni. Így az utóbbi tizenöt év során számos új
módszer fejlődött ki (de Boor, Riesenfeld, Bezier, Gordon
[9], [42], [28]) amely valamilyen szempontból legjobbnak

mondható szabad formájú görbék interpolációjára illetve approximációjára.

Az alapfeladat tehát, hogy bizonyos pontokban ismerjük egy görbe koordinátáit, első vagy magasabb foku deriváltjait. Interpoláció esetén az előállított görbe felveszi az előre megadott értékeket, approximációnál csak közelíti.

A számszerű értékeken kívül lehetnek egyéb követelmények is, amelyet teljesítenie kell a kívánt görbének. Ilyenek például a különböző rendű folytonossági kritériumok, de lehetnek kevésbé egzakt módon megfogalmazható esztétikai követelmények, például hogy a görbe a megadott pontok által kijelölt ivet kövesse, ne tartalmazzon felesleges oszcillációkat. Az egyik legfontosabb igény, amit a klasszikus interpolációs módszerek nem elégítenek ki, a görbe részenként való "alakíthatósága". Ez azt jelenti, hogy a görbe egy adatának változtatása csak a változtatás helyének egy kisebb környezetét módosítsa, ne az egész görbét.

Gyakorlatban felmerülő feltétel sokszor az is, hogy egyenes szakaszokat vagy köríveket lehessen beilleszteni egy legalább másodrendben folytonos görbe részeként.

A görbeközelítő eljárások egyik alapvető jellemzője, hogy melyik függvényosztály elemeit használják fel. Leggyakoribb módszer a polinomos közelítés valamelyik fajtája, mivel a számítási feladatok ezen a függvényosztályon belül a legegyszerűbbek. A felhasznált polinomok fokszáma különböző lehet. Tul magas fokszám esetén nehézséget okoz a nemkívánatos oszcillációk elkerülése és a számítások elvégzése is bonyodalmasabb, mégis van olyan módszer - Bezier [6] - amely nem korlátozza a polinomok fokszámát. Általánosabb azonban az alacsonyabb fokszámú görbékkel való közelítés, amikor viszont több, egymáshoz csatlakozó polinomot kell alkalmazni, amelyek a csatlakozási pontokban az előirt foly-

tonossági feltételeket teljesítik. Tapasztalatok szerint a harmadfoku paraméteres görbék használata számos előnnyel jár, mivel elegendő szabadsági fokkal rendelkeznek ahhoz, hogy a különböző simasági feltételeket teljesíteni tudják, viszont numerikus szempontból viszonylag egyszerűen kezelhetők.

Természetesen a fokszám rögzítése után is sokféle módszer közül választhatunk egyik sem mondható általános érvénnyel a legjobbnak.

Sokszor hosszú kísérletezés, próbálkozás szükséges ahhoz, hogy egy adott feladat jellegének legjobban megfelelő módszert kiválasszunk. Jó megoldásnak látszik az FFS rendszer koncepciója, amely több módszert kínál a felhasználónak választási lehetőségként. (ld. [19])

Tekintsük át röviden a legismertebb paraméteres harmadfoku görbék közelítő eljárásokat. Jelöljön $g(u)$ egy részgörbét, ahol

$$\underline{g}(u) = x(u)\underline{i} + y(u)\underline{j} + z(u)\underline{k}$$

paraméteres vektor-skalár függvény, $x(u)$, $y(u)$ és $z(u)$ harmadfoku polinom, a paramétertartomány pedig legyen rögzítve:

$$0 \leq u \leq 1.$$

Ily módon mindegyik polinom előállításához négy adatra van szükség, amelyet a görbedarab előre megadott adatainak függvényében kell beállítani. Általánosan felírhatjuk a következő képletet:

$$g(u) = \underline{u}^T \underline{A} \underline{p}$$

ahol \underline{u}^T vektor a paraméter-változó hatványaiból áll:

$$\underline{u}^T = [u^3, u^2, u, 1]$$

\underline{A} mátrix a konkrét módszerre jellemző konstansokat tartalmazza:

$$\underline{A} = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

\underline{p} vektor elemei pedig térbeli vektorok, amelyek az előállítandó görbedarab geometriai adatait tartalmazza:

$$\underline{p} = [p_1, p_2, p_3, p_4]$$

Az alkalmazandó módszert a \underline{p} vektorban megadott geometriai adatoknak megfelelően kell megválasztani, így az \underline{A} mátrix meghatározásánál is azt kell figyelembe venni, hogy melyek a rendelkezésre álló adatok (koordináták, deriváltak). Néhány példa a különböző kiinduló geometriai adatokra épülő interpolációs és approximációs módszerekre:

a/ Egy pont körüli görbeközelítés:

Taylor módszer:

p_1 = a pont koordinátái

p_2 = első derivált a pontban

p_3 = második derivált

p_4 = harmadik derivált

b/ Görbeszegmens interpoláció:

Hermite módszer:

p_1 = kezdőpont koordinátái

p_2 = végpont koordinátái

p_3 = első derivált a kezdőpontban

p_4 = első derivált a végpontban

c/ Görbeszegmens approximációja:

B-spline módszer:

p_1 = első tartópont koordinátái

p_2 = második tartópont koordinátái

p_3 = harmadik tartópont koordinátái

p_4 = negyedik tartópont koordinátái

A görbeinterpoláció problémáját tovább mélyíthetjük annak vizsgálatával, hogy kevesebb adat esetén hogyan lehet a hiányzó adatokat kiszámítani úgy, hogy a simasági vagy egyéb feltételek teljesüljenek. Feloldhatjuk a paraméter-tartomány hosszára vonatkozó megkötést is. Számos interpolációs módszer van, amelyek alapvetően Hermite típusu, de a derivált értékek és a paramétertartomány hosszának kiszámítása speciális mellékfeltételek figyelembe vételével történik. Ilyen például a Renner módszer, amelyik az érintőket

ugy határozza meg, hogy három, egy egyenesbe eső pont esetén a keletkező görbe szintén egyenes szakaszt tartalmazzon a megfelelő helyen. Ebben az esetben viszont a folytonosság csak elsőrendű lehet, hiszen a rendszer szabadsági foka kötött, ám a gyakorlatban ez nem látszik zavaró tényezőnek. Az FFS rendszer tapasztalatai szerint a Renner módszer az, amelyik a gépészetben szokásos vonalak leírására a legalkalmasabb.

1.3.3. Matematikai módszerek szabad formájú felületek leírására

A felületek leírására használt kétparaméteres vektor-skalár függvény általános alakja a következő:

$$\underline{R}(u,v) = X(u,v)\underline{i} + Y(u,v)\underline{j} + Z(u,v)\underline{k}$$

Ezt a függvényt felületelemenként kell meghatározni úgy, hogy a felület a megadott geometriai adatokat tartalmazza és a csatlakozási görbéken az előirt folytonossági feltételek teljesüljenek. Korlátozzuk a paramétertartományt az egységnégyzetre

$$0 \leq u \leq 1 \quad \text{és} \quad 0 \leq v \leq 1 .$$

A kiindulási adatok, amelyekhez a felületinterpolációs illetve approximációs módszert ki kell választani különböző típusúak lehetnek:

- a/ felületi pontok koordinátái, érintők, keresztderiváltak
- b/ felületi görbék valamely paraméterirányban
- c/ felületi görbék mindkét paraméter-irányban.

Tekintsük a Coons módszert ([12]), amely az egyik legismertebb eljárás egy felületelem interpolációjára. Legyenek $\underline{P}(0,v)$, $\underline{P}(1,v)$, $\underline{P}(u,0)$ és $\underline{P}(u,1)$ a felületelem peremén haladó görbék. Jelölje $\underline{P}_u(u,v)$ a perem egy pontjának deriváltját u paraméter szerint, $\underline{P}_v(u,v)$ a deriváltat v szerint, $\underline{P}_{uv}(u,v)$ pedig a keresztderiváltat, azaz:

$$\frac{\partial P}{\partial u} = \underline{P}_u$$

$$\frac{\partial P}{\partial v} = \underline{P}_v$$

$$\frac{\partial^2 P}{\partial u \partial v} = \underline{P}_{uv}$$

Ekkor ha adott a négy peremgörbe a saját paraméterezésének megfelelő deriváltakkal, továbbá a négy sarokpontban a keresztderiváltak, a ráillesztett felületet a következő egyenlet írja le:

$$\begin{aligned}
 \underline{R}(u,v) = & [s_{0,0}(u), s_{0,1}(u), s_{1,0}(u), s_{1,1}(u)] \begin{bmatrix} \underline{P}(0,v) \\ \underline{P}(1,v) \\ \underline{P}_u(0,v) \\ \underline{P}_u(1,v) \end{bmatrix} + \\
 (1) \quad & + [s_{0,0}(v), s_{0,1}(v), s_{1,0}(v), s_{1,1}(v)] \begin{bmatrix} \underline{P}(u,0) \\ \underline{P}(u,1) \\ \underline{P}_v(u,0) \\ \underline{P}_v(u,1) \end{bmatrix} - \\
 & - [s_{0,0}(u), s_{0,1}(u), s_{1,0}(u), s_{1,1}(u)] \underline{P} \begin{bmatrix} s_{0,0}(v) \\ s_{0,1}(v) \\ s_{1,0}(v) \\ s_{1,1}(v) \end{bmatrix}
 \end{aligned}$$

ahol

$$\underline{\underline{P}} = \begin{bmatrix} \underline{P}(0,0) & \underline{P}(0,1) & \underline{P}_v(0,0) & \underline{P}_v(0,1) \\ \underline{P}(1,0) & \underline{P}(1,1) & \underline{P}_v(1,0) & \underline{P}_v(1,1) \\ \underline{P}_u(0,0) & \underline{P}_u(0,1) & \underline{P}_{uv}(0,0) & \underline{P}_{uv}(0,1) \\ \underline{P}_u(1,0) & \underline{P}_u(1,1) & \underline{P}_{uv}(1,0) & \underline{P}_{uv}(1,1) \end{bmatrix}$$

továbbá $s_{0,0}$, $s_{0,1}$, $s_{1,0}$, $s_{1,1}$ olyan súlyfüggvények, amelyek a szomszédos felületelemek illesztéséhez szükséges feltételeket kielégítik:

$$\left. \begin{array}{l} s_{r,i}^{(p)} u_k = \\ s_{r,i}^{(p)} v_k = \end{array} \right\} \delta_{p,r} \delta_{i,k}$$

ahol δ a Kronecker szimbólum.

A Coons-féle felületleíró módszer kevesebb geometriai adat esetén is használható. Az (1) képletben szereplő összeg első tagja megoldást ad arra az esetre, ha csak az u paraméter irányában adottak a peremgörbék, a második tag pedig a v irányu paramétergörbékre vonatkozik. Az összeg harmadik tagja, amely itt negatív előjellel szerepel, a négy sarokpontra vonatkozó feltételeket kielégítő felületet ad.

Meg kell jegyezni, hogy ha a súlyfüggvényeket azonos függvényosztályból választjuk, és az (1) képletben szereplő deriváltak és keresztderiváltak egymással kompatibilisek, akkor az összeg mindhárom tagja ugyanazt a felületet írja le.

Célszerű a súlyfüggvényeket a görbékhez hasonlóan a harmadfoku polinomok osztályából választani. Ez esetben az (1)-ben szereplő összeg bármelyik tagja használható a felület leírására, leggyakoribb azonban a tenzorszorzat alakban történő megoldás, amely az (1) összeg harmadik tagjának felel meg.

Ez a megadás az 1.3.2.-ben leírt görbeinterpolációs módszerek általánosításának is tekinthető, írjuk fel ugyanis a következő alakban:

$$(2) \quad F(u,v) = \underline{u}^T \underline{A} \underline{P} \underline{A}^T \underline{v}$$

ahol

$$\underline{u}^T = (u^3, u^2, u, 1)$$
$$\underline{v} = \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}$$

\underline{A} ismét a módszerre jellemző konstans mátrix, \underline{P} pedig az előre megadott adatokat tartalmazza, amely nem csak az (1)-ben megadott típusu lehet, hanem a görbékhez hasonló módon tartalmazhatja több felületi pont koordinátáját a deriváltak helyett, vagy éppenséggel kevesebb pontot és magasabb fokú deriváltakat. Alkalmazhatjuk ezt a formulát felület approximációjára is (B-spline, Bezier [28], [6]).

Látható tehát, hogy a szabad formájú felületek matematikai leírására sokféle módszer használható. Azt, hogy egy modellező rendszerbe melyiket célszerű alkalmazni, a görbékhez hasonlóan a felhasználási terület jellege határozza meg.

Ha a számítástechnikai erőforrások lehetővé teszik, a felhasználó számára meg kell adni a lehetőséget, hogy több módszer közül választhassa ki a számára legalkalmasabbat, mint ahogy erre az FFS rendszer is módot nyújt.

1.4. SZABAD FORMÁJU FELÜLETEK MEGMUNKÁLÁSÁNAK PROBLÉMÁI

A szabad formájú felületek leírásának módszerei, a különböző geometriai tervezőrendszerek létrehozása végső soron azt a célt szolgálja, hogy valamilyen térbeli alakzatot ténylegesen elkészítsünk, és a gyártást minél jobban automatizáljuk. Ehhez pályavezérlésű NC marógépekre van szükség és egy olyan számítógépes programra, amely a tervezőrendszerrel előállított objektum geometriai adatai és a megadott technológiai adatok alapján előállítja a szerszám-pálya vezérlőszalagját.

Az automatizáció követelményeként egyre több helyen fejlesztenek ki a geometriai tervező rendszerhez csatlakozó számítógépes technológiai tervező rendszert, azzal a céllal, hogy valamilyen szempontból (idő, költség, stb.) optimalizálják a gyártási technológiát. Ilyen rendszerhez szükség van egy ún. technológiai adatbázisra, amely tartalmazza a szerszámgépekre, szerszámokra és az anyagokra vonatkozó adatokat. Így például a szerszámgépekről

- a vezérlés lehetőségeit
- a megmunkálható méreteket
- a pontosságot
- a fordulatszám érték-tartományait

- az előtolástartományt,
- a főorsó teljesítményét;

a szerszámokról

- az alakjellemzőket
- a méreteket
- a megengedett technológiai adatokat
- a szerszámbe fogás méreteit, alakját
- a kopottsági állapotot
- a merevséget,

az anyagokról pedig a különböző anyagállandókat.

Mindezek mellett az egyes munkadarabokhoz is meg kell adni a szükséges technológiai adatokat, anyag, pontosság, felületi simaság, az előgyártmány mérete, stb.

A geometriai leírás és a technológiai adatbázis alapján történő NC vezérlőszalag generálásának automatizáltsága a rendszer fejlettségi szintjétől függően nagyon különböző lehet. Minimális követelmény a kijelölt felületrész, a maró alakja és méretei valamint a ráhagyás ismeretében a referenciapont koordinátáinak kiszámítása egy valamilyen stratégia szerint haladó pálya mentén. A jelenlegi legismertebb geometriai tervezőrendszerek közül csak igen kevés (pl. a japán TIPS [35]) képes arra, hogy teljes technológiai tervet készítsen, tehát meghatározza a szükséges műveleteket, azok sorrendjét, a szükséges szerszámokat és mozgásstratégiát.

2. AZ FFS RENDSZER NAGYOLÓ PROCESSZORÁNAK ÁTTEKINTÉSE

2.1. AZ FFS RENDSZER MEGMUNKÁLÓ PROCESSZORAI

Az FFS rendszerben kialakított megmunkálási lehetőségek a hazai fejlettségi szinthez igazodnak, azaz ahhoz a tényhez, hogy a Magyarországon jelenleg használatban levő NC marógépek döntő többsége 2 1/2 D-s illetve 3D-s.

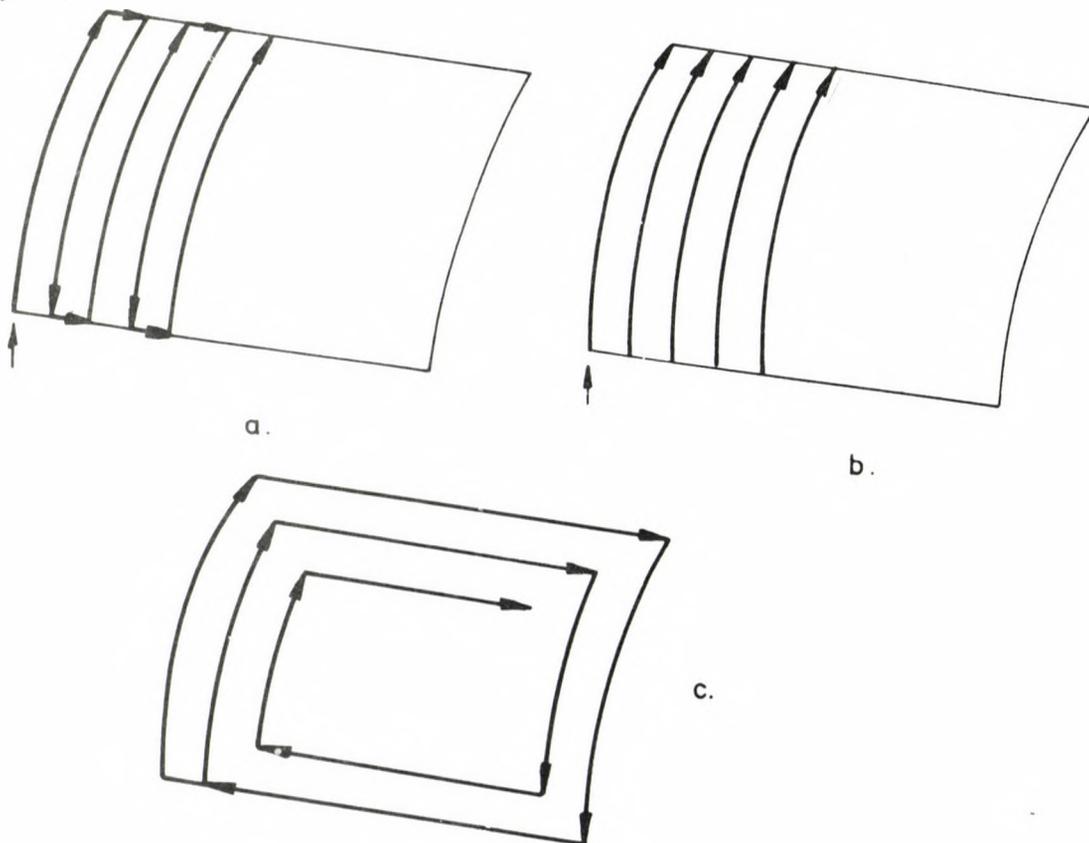
A 2 1/2 D-s NC marógépek egy irányban szakaszvezérlésre, az arra merőleges síkban pedig pályavezérlésre képesek. Ebből következik, hogy szabad formájú felületek megmunkálása csak a pályavezérléssel párhuzamos síkmetszetek mentén lehetséges és a felület a szerszámtengely irányából nézve alámetszést nem tartalmazhat. Ez a feltétel leszűkíti ugyan a megmunkálható felületek osztályát, de a rendszer még sem veszti el ipari jelentőségét és használhatóságát hiszen a különböző műanyag fröccsöntő szerszámok, kovácslószerszámok felületei éppen ilyen tulajdonságaik.

3D-s gépek esetén is érvényes ez a megkötés, de a térbeli pályavezérlési lehetőség következtében jobb felületi minőséget eredményező megmunkálási stratégiák alakíthatók ki.

A fenti szempontok alapján az FFS rendszerben a felülettervező modulhoz kétféle megmunkáló processzor csatlakozik, egy 2 1/2D-s és egy 3D-s. 2 1/2D-ben az (x,y) síkkal párhuzamos síkmetszetek mentén hengeres maróval történik a megmunkálás és használata nagyolási célokra ajánlott. Teljes megmunkáláshoz a teraszokat egymástól igen kis távolságra kell képezni, ami hosszú számítási időt és kevésbé terme-

lékeny technológiát eredményez. Ennek ellenére ilyen irányu felhasználásra is sor került már, olyan üzemben, amely csupán 2 1/2D-s marógéppel rendelkezik, és az eredmény a helyi körülmények között megfelelőnek bizonyult.

Az FFS rendszer simító processzora 3D-s marógéphez generál szerszám pályát, a felület paraméterezésének megfelelő irányokat követve. Ezen belül különböző stratégiák alakíthatók ki (ld. 2.1. ábra). A felületi normális ismeretében a gömbszerű szerszám referenciapontjának koordinátái egyszerű képlettel számíthatók. A teljes ütközésmentesség automatikus vizsgálata nincs kialakítva, mivel erre egyelőre csak rendkívül számításigényes eljárások állnak rendelkezésünkre. A megoldás jelenleg a technológus interaktív közreműködése, amelyhez a rendszer elegendő segédletet nyújt (felület szegmentálása, lehatárolása, görbületi térkép stb.) ([19]).



2.1 ábra

2.2. AZ FFS RENDSZER NAGYOLÓ PROCESSZORÁNAK MŰKÖDÉSE

Az FFS rendszer nagyoló processzora szoborszerű felületek 2 1/2D-s megmunkálásának NC programjait állítja elő. Bemenő adatai a következők:

- 1/ az FFS felülettervező moduljával készített valamely felület belső, számítógépes reprezentációja (ld. [19])
- 2/ technológiai adatok. (ld. 1.sz. melléklet)

A rendszer a bemenő adatok alapján kiszámítja a szerszám pályát egy belső adatstruktúrában, és a csatlakozó posztprocesszorok segítségével teljes NC programot készít DIALOG 8860 vagy UNIMERIC 723 vezérlésre.

A megmunkálás z tengelyen negatív irányba haladva az (x,y) sikkal párhuzamos síkmetszetek mentén történik. A rendszer technológiai tervezést nem végez, hanem lehetőséget ad a technológusnak a technológiai paraméterek előzetes megadására, továbbá a processzor működése közben történő módosítására, grafikus ellenőrzési lehetőség támogatásával.

Az 1. sz. melléklet tartalmazza a nagyoló processzor felhasználói kézikönyvét, amely ismerteti a rendszer használatát. Jelen dolgozat a processzorba beépített számítógépes geometriai algoritmusok leírásával és elemzésével foglalkozik.

2.3. A NAGYOLÓ PROCESSZOR ALGORITMUSAI

A megmunkálási stratégiának megfelelően a nagyoló processzornak a következő problémákat kell megoldania.

- 1/ Szoborszerű felület metszése z =konstans sikkal
- 2/ Sikkbeli szerszám-pálya előállítás az előgyártmány, a metszetkonturok, a marósugár és a simitási ráhagyás ismeretében.

Mivel egy szabad formájú felület síkmetszete szabad formájú görbék halmaza lesz, a síkmetszés eredménye csak valamilyen formájú közelítés lehet. A 3. fejezetben leírt metszési eljárás egyenes szakaszokkal közelíti a metszetkonturokat. Léteznek azonban olyan eljárások, amelyek egy kis egyenes szakaszokból álló görbe helyett egy adott tőrésen belül körivekből és egyenes szakaszokból álló, sokkal kevesebb darabból összetevődő görbét számítanak ki (ld. [38]). Gondolva erre a lehetőségre, valamint arra, hogy egy 2D-s megmunkáló processzor a szoborszerű felületektől függetlenül is jól használható, a sikkbeli szerszám-pálya-generálást köriveket és egyenes szakaszokat tartalmazó görbékre dolgoztuk ki. Elengethetetlen viszont, hogy a konturok zártak legyenek, hiszen az anyagi és a megmunkálási tartományt el kell egymástól választani. A megmunkálási tartomány kijelölésére azt követeljük meg a síkgörbéktől, hogy irányítottak legyenek, és az irányításuk szerinti jobb oldalon legyen az anyag tartománya, balról pedig a megmunkálásé. A Függelékben matematikailag bebizonyítjuk, hogy ezzel egyértelműen feloszthatjuk a síkot diszjunkt tartományokra, ha a görbéink nem önmetszőek. A 3. fejezetben leírt síkmetsző eljárás ezekhez a szempontokhoz igazodik.

A síkbeli szerszám-pálya előállítás három, egymást követő algoritmusból áll:

1/ Ofszet görbék meghatározása mindegyik síkmetszeti görbéhez külön-külön.

A metszetkonturokat a marósugár nagyságával "hizlalva" vagy zsugorítva megkapjuk az egyes metszetgörbékhez azokat a konturokat, amelyeken a maró referenciapontját mozgatva mindig a megmunkálási tartomány felül érintjük az eredeti görbét (ld. 4. fejezet).

2/ Az egymást metsző ofszet görbék egyesítése.

Erre abban az esetben van szükség, ha több síkmetszeti görbénk van és nem mindegyiket befelé ofszeteljük. Ekkor ugyanis előfordulhat, hogy az ofszet görbék egymásba metszenek, azaz az egyik eredeti konturt követve belemetszünk a másikba. Ebben a fázisban az ilyen ofszet görbéket egyesítjük és így meghatározzuk az előgyártmány síkmetszésre vonatkozó megmunkálási tartományokat. (ld. 5. fejezet).

3/ A szerszám-pálya meghatározása

Ez az algoritmus cikk-cakk pályát számít a hengeres maró referencia pontja számára, amelyen végighaladva a kijelölt tartományok teljes egészében megmunkálhatók, különböző optimalizálási szempontok figyelembe vételével. (ld. 6. fejezet).

3. SZABAD FORMÁJU FELÜLETEK METSZÉSE Z =KONSTANS SIKKAL

Szabad formájú felületek tervezése során fontos ellenőrzési szerepe lehet a különböző síkmetszeteknek. Teraszos megmunkálás esetén a síkmetszeti görbék képezik a szerszám pálya számításának alapját. Szükség van tehát olyan eljárásra, amely meghatározza azokat a görbéket, amelyeket egy adott sík metsz ki egy adott szoborszerű felületből. Elegendő azonban az algoritmust z =konstans síkra kidolgozni, mivel a felületek egyszerű térbeli transzformációval a megfelelő helyzetbe állíthatók és ezeket a transzformációkat a tervezőrendszerek általában tartalmazzák is.

Paraméteres harmadfokú felület esetén a síkmetszet kiszámítása alapvetően kétféle módon történhet [45]. Vagy úgy, hogy a felület és a sík közös pontjaiból elegendő sokat meghatározunk és azokat valamilyen eljárással görbéké fűzzük össze, vagy pedig a metszetgörbe mentén haladva közelítő eljárással számítjuk ki a közös pontokat.

Az első módszer alkalmazása esetén komoly elvi problémát jelent a metszéspontok összefűzése. Nem ismeretes olyan eljárás, amely mindig teljesen pontos eredményt ad, (ld. 3.6), de kialakítható olyan algoritmus, amelynek pontossága a gyakorlatban elegendő, illetve interaktív alkalmazás során tetszőlegesen fokozható.

A második módszer használata szintén számos problémát vet fel, melyek között legsúlyosabb az, hogy az alkalmazható közelítő számítások konvergenciájának sebessége általánosan nem meghatározható.

A kétféle megközelítésből adódó problémákat összehasonlítva látható, hogy egy interaktív rendszerben, amilyen az FFS rendszer, előnyösebb az első módszer használata. Ilyen síkmetsző algoritmust ismertet a 3. fejezet, amelyet a szerző az FFS rendszer nagyoló processzorához dolgozott ki és abban implementált. Az eljárás célja, hogy egy adott paramétereiben harmadfoku felület és egy adott z =konstans sík metszetkonturjait előállítsa, oly módon hogy a metszetgörbék zártak legyenek és irányításukkal kijelöljék a síkbeli megmunkálási tartományt.

3.1. A FELÜLET MEGADÁSA, MEGSZORÍTÁSOK

A továbbiakban leírt síkmetsző algoritmus olyan felületekre alkalmazható, amely a következő feltételeket kielégíti.

a/ A felület olyan, topológiailag $n \times m$ -es négyzethálót alkotó elemi felületekből áll, amelyeket koordinátáinként harmadfoku skalár-vektor függvények írnak le:

$$x = X_{i,j}(u,v), \quad y = Y_{i,j}(u,v), \quad z = Z_{i,j}(u,v)$$

ahol: $n, m \geq 1$,
 $0 \leq u, v \leq 1$,
 $1 \leq i \leq n, \quad 1 \leq j \leq m$

b/ A felületelemek folyamatosan csatlakoznak egymáshoz, azaz:

$$\left. \begin{aligned} X_{i,j}(u,1) &= X_{i,j+1}(u,0), \\ Y_{i,j}(u,1) &= Y_{i,j+1}(u,0), \\ Z_{i,j}(u,1) &= Z_{i,j+1}(u,0) \end{aligned} \right\} \begin{aligned} 0 \leq u \leq 1 \\ 0 \leq i \leq n \\ 0 \leq j \leq m-1 \end{aligned}$$

és

$$\begin{aligned} X_{i,j}(1,v) &= X_{i+1,j}(0,v), & 0 \leq v \leq 1 \\ Y_{i,j}(1,v) &= Y_{i+1,j}(0,v), & 0 \leq i \leq n-1 \\ Z_{i,j}(1,v) &= Z_{i+1,j}(0,v) & 0 \leq j \leq m \end{aligned}$$

c/ A felület nem önmetsző, azaz

$$\text{ha } X_{i,j}(u_1, v_1) = X_{k,l}(u_2, v_2)$$

$$\text{és } Y_{i,j}(u_1, v_1) = Y_{k,l}(u_2, v_2)$$

$$\text{és } Z_{i,j}(u_1, v_1) = Z_{k,l}(u_2, v_2)$$

akkor $i=k, j=l,$

továbbá $u_1=u_2$ és $v_1=v_2.$

d/ A felület a z-tengely irányából nézve nem alámetsző, azaz

$$\text{ha } X_{i,j}(u_1, v_1) = X_{k,l}(u_2, v_2)$$

$$\text{és } Y_{i,j}(u_1, v_1) = Y_{k,l}(u_2, v_2)$$

$$\text{és } Z_{i,j}(u_1, v_1) = Z_1 < Z_{k,l}(u_2, v_2) = Z_2$$

akkor létezik olyan $Z_0,$ és olyan g, h és $u, v,$ amelyre

$$Z_1 < Z_0 < Z_2$$

$$1 \leq g \leq n, 1 \leq h \leq m$$

$$0 \leq u, v \leq 1$$

$$X_{g,h}(u,v) = X_{i,j}(u_1, v_1)$$

$$Y_{g,h}(u,v) = Y_{i,j}(u_1, v_1)$$

$$Z_{g,h}(u,v) = Z_0$$

e/ A felületi normális a metszetkonturok mentén valamilyen tartományban nem párhuzamos az (x,y) sikkal.

Megjegyzés:

A síkmetszet kiszámításához az a , b , és c feltételek elegendőek. A d feltételre a $2\ 1/2D$ -s megmunkálhatósághoz, e -re pedig a megmunkálási terület automatikus kijelöléséhez van szükség.

3.1.2. A síkmetszéshez szükséges további adatok

a/ Mivel az eljárás kifejezetten szerszám-pálya generáláshoz készült, már ebben a fázisban szükség lehet az előgyártmány méreteire. Feltétezzük, hogy az előgyártmány bármely (x,y) sikkal párhuzamos metszetét ismerjük.

b/ Mivel a metszetkonturoknak csak bizonyos pontjait számítjuk ki, a közelítés finomságát előre meg kell határozni.

Ez a paramétersík felosztásának megadását jelenti u és v irányokban.

3.2. AZ ELJÁRÁS ÁTTEKINTÉSE

A síkmetsző algoritmus három fő részből áll:

a/ Egy felületelem és a sík metszéspontjainak kiszámítása egy megadott sűrűségű paraméterháló mentén.

b/ Összefüggő metszetvonalak kialakítása egy felületelemen belül

c/ A különböző felületelemeken levő metszéspontok összekapcsolása irányított, zárt görbéké.

3.3. FELÜLETELEM METSZÉSE SIKKAL

A felületelemet leíró paraméteres $\underline{r}(u,v)$ függvénybe konstans u illetve v értékeket helyettesítve az előírt $z=z_0$ síkot a felületen haladó görbékkel metszük el. A metszéspontokat a következő harmadfokú egyenletek megoldásaként kapjuk:

$$(1) \quad z(i\Delta u, v) = z_0$$

$$(2) \quad z(u, j\Delta v) = z_0$$

ahol IU és IV a felhasználó által megadott paraméter-felosztási érték u illetve v irányban, továbbá

$$0 \leq i \leq IU, \quad \Delta u = \frac{1}{IU}$$

és

$$0 \leq j \leq IV, \quad \Delta v = \frac{1}{IV}.$$

A harmadfokú egyenlet három gyöke közül minden esetben a $[0,1]$ intervallumba esők szolgáltatnak metszéspontot, mivel az egyenletek paraméterértékekre vonatkoznak (3.1. ábra).

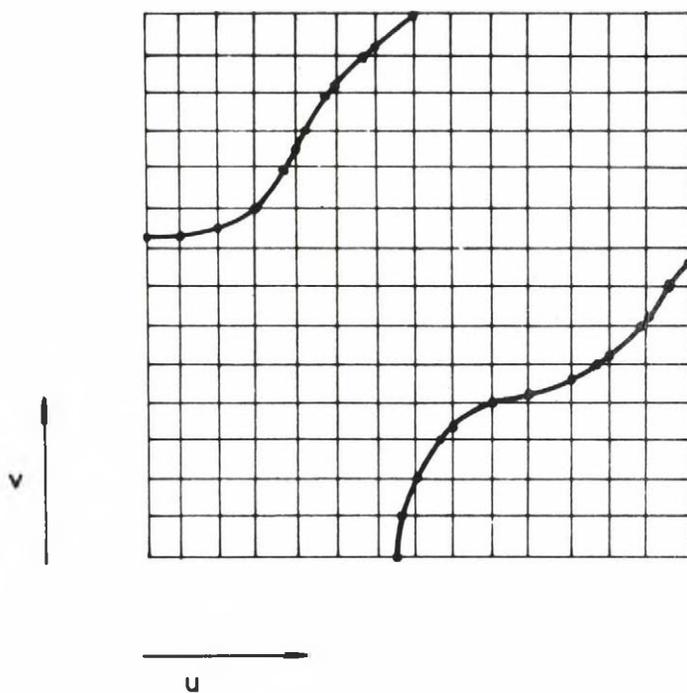
3.4. METSZÉSPONTOK RENDEZÉSE EGY FELÜLETELEMEN BELÜL

A sík és a felületelem metszéspontjainak halmazából olyan összefüggő görbedarabokat kell létrehozni, amelyek a felületelem határán kezdődnek és végződnek, vagy pedig zártak. Az algoritmus során a metszéspontok felfűzésén az eredménylistához való csatlakoztatást és a rendezetlen pontok listájából való törlést értjük.

- a/ Megvizsgáljuk, hogy van-e metszéspont a felületelem négy határoló oldalán. Az azonos oldalon levő, Δu illetve Δv távolságra levő pontokat összefűzzük, de az így keletkezett görbedarabok végpontjait nem töröljük az eredeti listából.
- b/ Kiválasztunk egy metszéspontot, amely a felületelem határán fekszik. Ezt egy új görbedarab kezdőpontjaként felfűzzük. Ha nincs ilyen pont, h/-nál folytatjuk.
- c/ Az utoljára felfűzött pont $\Delta u, \Delta v$ környezetében levő mesztéspontok közül (ha vannak) felfűzzük az euklideszi távolságban legközelebb levőt (3.2. ábra).
- d/ Visszatérünk c/-hez.
- e/ Nincs közeli metszéspont. Ha a felületelem határán vagyunk, vége a megkezdett görbedarabnak, b/-nél folytatjuk.
- f/ Ha az utolsóként felfűzött pont a kezdőponthoz $\Delta u, \Delta v$ közelségben van, zárt konturnak tekintjük a darabot, és b/-nél folytatjuk az eljárást.
- g/ Az adott felosztási finomság mellett a metszetkontur helyessége bizonytalan, a program figyelmeztetést küld a felhasználónak vagy pedig automatikusan változtatja a

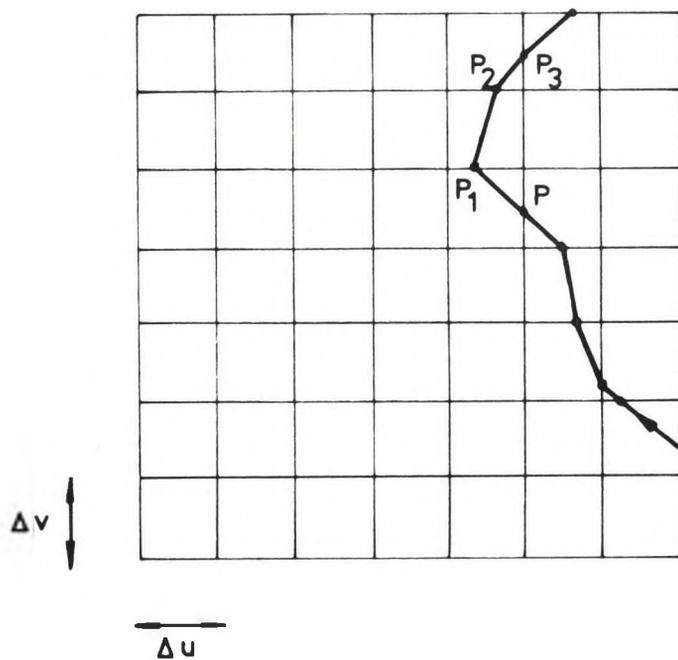
paramétereket és újra kezdi a metszési eljárást, előzetes beállítás szerint.

- h/ Ha van még felfüzetlen metszéspont, megkeressük a paramétertartomány széléhez legközelebb esőt, és azzal új görbedarabot kezdünk. Az eljárást c/-nél folytatjuk.
- i/ Vége a felületelemen belüli rendezésnek. Az egyetlen pontból álló görbéket elhagyjuk az eredménylistából.



3.1 ábra

Metszetgörbék vetülete a paraméterhálón



3.2 ábra

Metszetgörbe pontjainak összekötése egy felületelemen

3.5. KÜLÖNBÖZŐ FELÜLETELEMEN KEN LEVŐ GÖRBÉK ÖSSZEFÜZÉSE

3.5.1. Problémák

Az egyes felületelemeken keletkezett metszetgörbékét olyan módon kell összefűzni, hogy a létrejövő konturok zártak legyenek és irányításuk egyértelműen kijelölje a megmunkálási tartományt.

A konturok zártsága nem feltétlenül adódik magától, mivel nem testet, hanem felületet metszünk el, tehát a metszetgörbe nyílt is lehet. A megmunkáláshoz azonban tartományokat kell kijelölni, ezért a nyílt görbékét valamilyen módon le kell zárni. Erre több lehetőség is kínálkozik. Ha a nyílt konturdarabok elérik az előgyártmány határát, akkor az előgyártmány síkmetszete mentén értelmesen összeköthetők, és sok esetben technológiailag is ez adja a legcélszerűbb megmunkálási tartományt. (3.3. ábra).

A másik eljárás, amely mindig elvégezhető, hogy a felületet testté egészítjük a $2\ 1/2D$ -s megmunkálásnak megfelelően. A felület határoló görbéit a kijelölt paraméterosztással pontonként levetítjük a metsző síkra. Az így keletkezett síkbeli konturt (nevezzük befoglaló konturnak) a síkmetszet görbék mindig elérik, így a lezárás a befoglaló kontur mentén mindig elvégezhető. (3.4. ábra) A vetítést elegendő az (x,y) síkra elvégezni, mivel a metsző sík mindig (x,y) síkkal párhuzamos, a vetítés pedig erre merőleges.

A konturdarabok összefűzése úgy történik, hogy a kezdő illetve a végpontokat próbáljuk csatlakoztatni. Ha sikerült, a konturdarabot töröljük a bemeneti listából. A felfűzést úgy irányítjuk, hogy a megmunkálási tartomány a görbétől balra legyen, ez egyértelműen megtehető (ld. Függelék).

3.5.2. Algoritmus

a/ Uj konturt kezdünk.

Keresünk egy olyan görbedarabot, amelyet még nem próbáltunk felfűzni. Ha nincs ilyen, akkor vége a csatlakoztató eljárásnak. Ha van, legyen a kezdőpontja P_0 , a következő P_1 , az utolsó pontja pedig P_v .

b/ Tekintsük a görbedarab kezdőpontjában a kifelé mutató felületi normálist. Ha ez elfajuló vagy párhuzamos az (x,y) sikkal, folytassuk az eljárást, a/-nál. Ha viszont merőleges az (x,y) sikkra, akkor töröljük a konturdarabot a bemeneti listából és folytassuk az eljárást a/-nál.

c/ Vegyünk a normális egyenesén egy pontot, amelynek z koordinátája nagyobb mint a metsző síké, és vegyük ennek a pontnak a vetületét a metsző síkra. Legyen ez a pont Q. (Q biztosan a megmunkálási tartományban van, mivel a normális a felületből kifelé mutat.)

d/ Vizsgáljuk a $\overline{P_0P_1} \times \overline{P_0Q}$ vektorális szorzatot. Ha a szorzatvektor z koordinátája nulla, akkor a/-nál folytatjuk az eljárást. Ha pozitív, akkor Q a görbedarab eredeti irányítása szerint balra van, a fűzést P_0 -nál kezdjük. Ha ez az érték negatív, a fűzést P_v -től kezdjük, visszafelé. (Ez megfelel a Függelék 1. definíciója szerint annak, hogy a megmunkálási tartomány a konturtól balra van).

e/ A kijelölt irányítás szerint felfűzzük a görbedarabot.

f/ Ha az utolsónak felfűzött pont egybeesik a kontur kezdőpontjával, akkor egy konturt befejeztünk, a/-nál folytatjuk az eljárást.

g/ Az utolsónak felfűzött ponttal egybeesőt keresünk a felfűzetlen görbedarabok kezdő illetve végpontjai között. (Ez már a fűzési irányt is kijelöli). Ha több ilyen darab van, keresünk egy olyant, amelyik nem az előzőleg

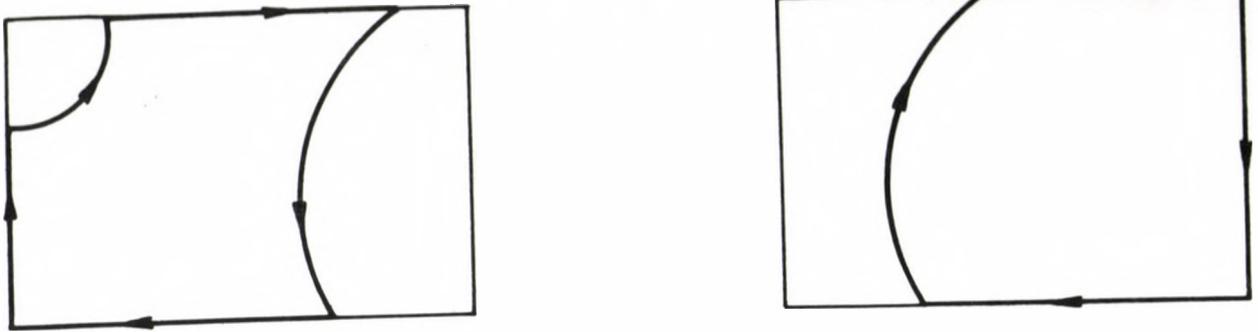
felfüzdött darabon indul visszafelé, a többit pedig töröljük a bemeneti listából. Az eljárást e/-nél folytatjuk.

- h/ Ha nincs csatlakozó pont, megnézzük hogy az utolsónak felfüzdött pont rajta van-e az előgyártmány síkmetszetén. Ha nics, l/-nél folytatjuk az eljárást.
- i/ Az előgyártmány síkmetszete mentén negatív körüljárással haladva addig folytatjuk a görbét, amig egy másik, az anyaghatárra kifutó felfüzetlen görbedarabot nem találunk (3.3. ábra).
- j/ Ha találunk ilyen görbedarabot, e/-nél folytatjuk az eljárást.
- k/ Ha nem, a felfüzdött anyaghatárt töröljük az eredménylistából és az utolsó felfüzdött konturponttól a felület határa mentén keresünk folytatást.
- l/ Ha az utolsó pont nincs a befoglaló konturon, o/-nál folytatjuk az eljárást.
- m/ Ha van olyan felfüzetlen görbedarab, amelyik a befoglaló konturt eléri, akkor a befoglaló kontur mentén való összefüzés lehetséges. Meghatározzuk, hogy melyik irányban kell elindulni a befoglaló kontur felfüzdésével. A vetítés során felosztottuk a felület szélét a Δu illetve Δv paraméterosztás szerint, tekintsük most az aktuális P pontot közrefogó két vetületi pontot. Amelyiknek Z koordinátája kisebb a metsző sík szintjénél, annak a vetülete jelöli ki a felfüzdési irányt. Ha mindkettő teljesíti ezt a feltételt, vagy pedig egyik sem, o/-nál folytatjuk az eljárást.

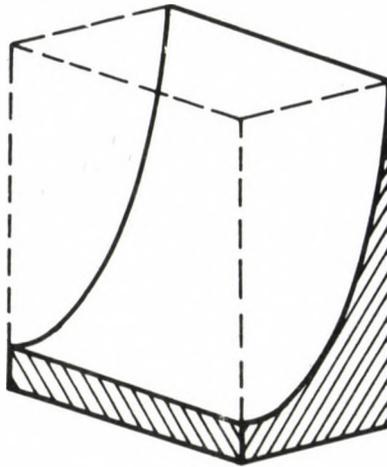
n/ A befoglaló kontur pontjait felfűzzük addig, amig az első felfűzetlen görbedarab kezdő vagy végpontját el nem érjük rajta.

Az eljárást e/-nél folytatjuk.

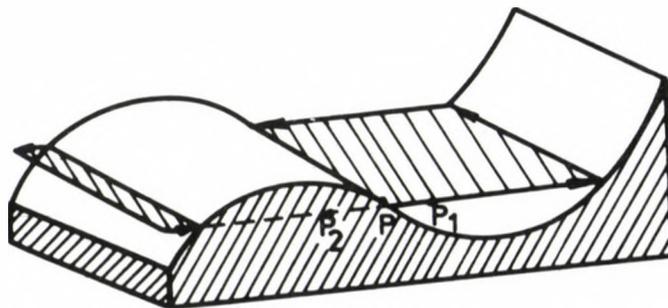
o/ A megkezdett kontur nem fejezhető be, töröljük az eredménylistából és a/-nál folytatjuk az eljárást.



3.3 ábra
Metszetkontúr lezárása az előgyártmány síkmetszete mentén



3.4 ábra
Felület kiegészítése testté, vetítéssel



3.5 ábra
Metszetkontúr lezárása a felülethatár vetületén

3.6. AZ ALGORITMUS ELEMZÉSE

3.6.1. Felületelem metszése sikkal

A 3.3.-ban leirt eljárás elvileg korrekt, a harmadfoku egyenlet megoldásaként kapott pontok mind a felületen, mind a metsző sikon rajta vannak. Felmerülnek viszont a számábrázolási és számítási pontatlanságból adódó hibák, amelyek egyes esetekben olyan súlyosak lehetnek, hogy a kapcsolódó algoritmusok helyes működését veszélyeztetik. Ezért ennek az eljárásnak a beprogramozása során különösen ügyelni kell a numerikus problémákra, amellyel részletesen a 7.2. fejezet foglalkozik.

3.6.2. Metszéspontok rendezése egy felületelemen belül

A 3.4.-ben leirt összefüzési eljárás minden olyan esetben biztosan jó (azaz nem köt össze olyan pontokat, amelyek nem szomszédosak), amikor egy felületelemen belül csak egy metszéspont van, és a felületi normális nem változik nagyon gyorsan. Egyéb esetekben a paraméterosztás sűritésével kapható meg a helyes eredmény, ami történhet automatikusan, vagy pedig a felhasználó interaktív beavatkozásával. (3.6. ábra) Igen speciális esetben előfordulhat, hogy az osztás sűritése sem hoz megoldást, pl. ha egy metsző sík érintővé válik egy felületelemen belül. (3.7/b ábra) Az ilyen esetekre való felkészülés aránytalanul megnövelné a számítási időket, viszont a metsző sík igen kis eltolásával már helyes metszetgörbéhez juthatunk. Ez a módosítás szintén végezhető interaktívan vagy automatikusan.

Szólni kell még arról az esetről, amikor a felület és a sík metszete olyan kicsi kontur, hogy az adott paraméterosztás-sűrűség mellett a metszetkontur nem áll elő (nincs kellő számú metszéspont). Ezért az osztást úgy kell megválasztani, hogy a kieső metszetkonturok a kis méretük miatt a gyakorlatban elhanyagolhatók legyenek.

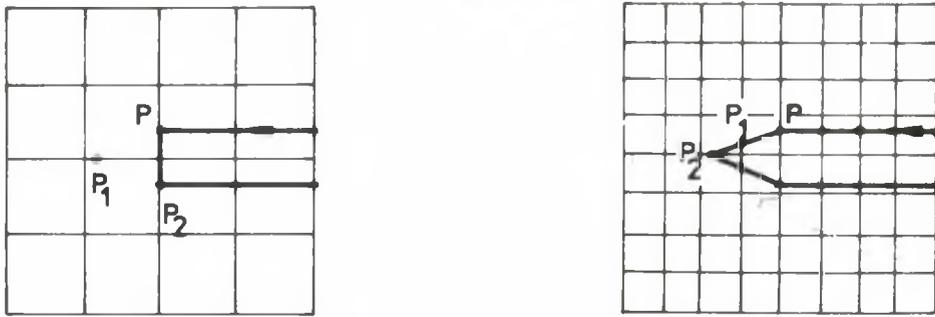
3.6.3. Különböző felületelemeken levő görbedarabok összefüzése

A felületelemek közötti összefüzés háromféle problémát vet fel.

- a/ Meg kell határozni a metszetkontur irányítását. Ezt az első felfűzött darabnál kell eldönteni, mivel a fűzés a többiek irányítását már meghatározza. A 3.1/d feltétel biztosítja, hogy létezik olyan görbedarab, ahol a 3.5/b szerinti módon a kontur irányítása meghatározható. Az eljárás kiharaja azokat a görbedarabokat, amelyek a sík és a felület érintkezéséből adódnak (3.7/a ábra), így a metszetkonturok nem lesznek degeneráltak.
- b/ Ha a metsző sík tartalmazza két szomszédos felületelem határvonalát, vagy annak egy részét (3.8. ábra), akkor a határon fekvő metszéspontok mindkét felületelemen kiszámíthatódnak. 3.4/a-ban azonban gondoskodtunk róla, hogy az ilyen határmenti szakaszok különálló konturdarabok legyenek még akkor is, ha a felületelemen belül folytatni lehetett volna őket, 3.5/g pedig a többszörösen jelentkező darabok közül csak egyet fűz fel, a többit törli.

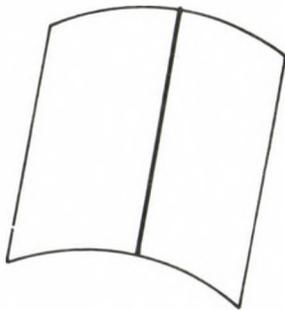
c/ Mivel nem testet, hanem felületet metszünk el, előfordulhat, hogy nyílt metszetkonturt kapunk. Ezeknek a lezárását meg kell oldani, hogy a megmunkálási tartomány egyértelmű legyen. Az előgyártmány határa mentén történő lezárásnál az állandó körüljárás biztosítja a helyes eredményt. A felülethatár vetülete mentén történő lezárásnál pedig 3.5/m-ben a csatlakoztatás irányát úgy állítjuk be, hogy a keletkező metszetkontur kipótolt darabjai mindig a felület felett haladnak, tehát a megmunkálhatóság biztosítva van.

A felsorolt problémákat ilyen módokon megoldva a konturok csatlakoztatása már egyértelmű.

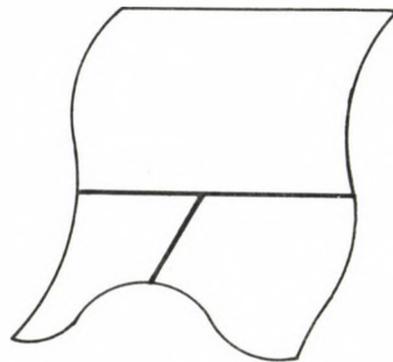


3.6 ábra

Metszetgörbe összefűzésének hibája és javítása



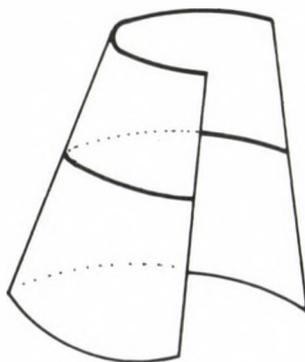
a.)



b.)

3.7 ábra

Degenerált metszetgörbék



3.8 ábra

Síkmetszés két felületelem határán

4. OFSZET KONTUROK ELŐÁLLÍTÁSA ZÁRT SIKGÖRBÉKHEZ

4.1. A FELADAT MATEMATIKAI MEGFOGALMAZÁSA

Adott a síkon egy G görbe, amely a következő tulajdonságokkal rendelkezik.

- a/ folytonos
- b/ zárt
- c/ irányított
- d/ egyenes szakaszokból és 180° -nál nem nagyobb középponti szögű körivekből áll
- e/ nem önmetsző és nem elfajuló, azaz alkotóelemei csak végpontjaikban találkozhatnak és mindig csak az irányítás szerinti szomszédos elemmel.

Adott továbbá egy $R > \emptyset$ szám.

Keressük azon pontok mértani helyét, amelyek a G görbe baloldalán, (ld. Függelék 1.) a görbétől pontosan R távolságra vannak. Ezt a mértani helyet (ha létezik) a G görbe ofszetjének nevezzük.

4.2. AZ ELJÁRÁS ÁTTEKINTÉSE

A továbbiakban be fogjuk látni, hogy egy 4.1.-ben meghatározott tulajdonságú síkgörbe ofszetje (ha létezik) egy vagy több, zárt, esetleg elfajuló görbét alkot. Nevezzük ezt a G görbe ofszet görbehalmazának és jelöljük a következőképpen:

$$O = \{O_1, O_2, \dots, O_n\}$$

Az ofszet görbék meghatározó algoritmus logikailag három fázisra bontható, számítógépes implementáció során azonban az első kettő összevonható, ami által a számítási idő és a tárigény csökkenthető. A három fázis a következő:

- a/ Elemi ofszet görbék generálása G minden alkotóeleméhez.
- b/ Az elemi ofszet görbék összefűzése egyetlen összefüggő, zárt (többnyire önmetsző) görbévé.
- c/ Az önmetszések által keletkező hurkok szelektálása.

4.3. AZ ELEMI OFSZET GÖRBÉK MEGHATÁROZÁSA

Tekintsük a G görbe alkotó elemeit, az egyenes szakaszokat és köríveket nyiltaknak, azaz végpontjaik figyelmen kívül hagyásával. Határozzuk meg az egyes elemekhez azon pontok mértani helyét, amelyek tőle pontosan R távolságra, az irányítás szerinti bal oldalon vannak. Elemi geometriai megfontolások alapján ezek a következők lesznek:

- a/ Egyenes szakaszhoz egy vele párhuzamos, egyenlő hosszú, irányítás szerinti bal oldalon R távolságra haladó nyílt egyenes szakasz (ld. 4.1/a ábra).
- b/ Olyan körívhez, amelynek sugara r ;
 - I. negatív körüljárás esetén $R+r$ sugaru, nyílt, azonos középpontu és nyílásszögű körív (4.1/b ábra),
 - II. pozitív körüljárás és $r > R$ esetén $r-R$ sugaru azonos középpontu és nyílásszögű körív (4.1/c ábra),

III. pozitív körüljárás és $r < R$ esetén nincs ilyen pont
(4.1/d ábra)

Készítsük el ezeket az elemi ofszet görbéket G minden komponenséhez és az így keletkezett görbehalmazhoz vegyük hozzá a G görbe csucspontjai köré írt R sugarú köröket. Nevezzük ezt a görbehalmazt M -nek (pl. 4.3. ábra).

4.4. AZ ELEMI OFSZET GÖRBÉK ÖSSZEFÜZÉSE

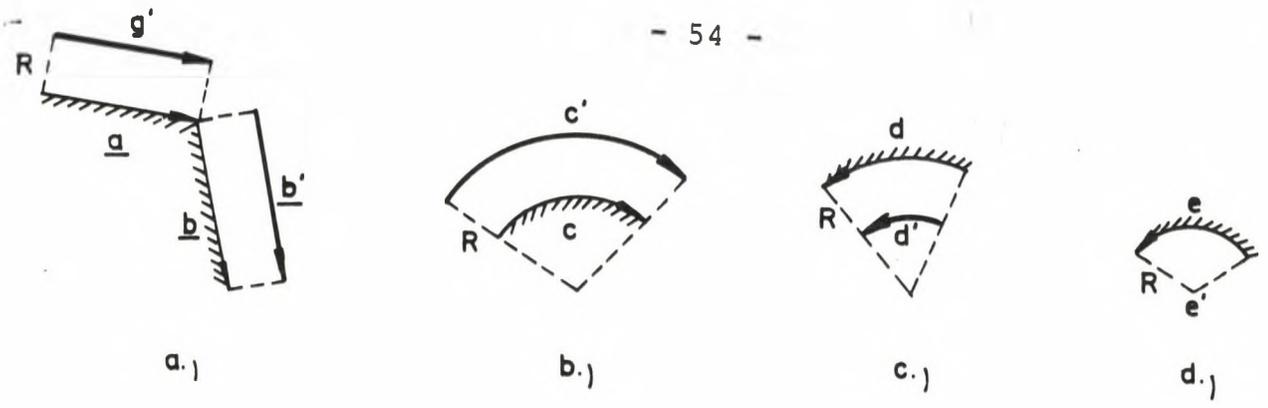
Az eredeti görbe irányítása átvihető a 4.3/a és 4.3/b alapján képezett ofszet elemekre, mivel pontonkénti egy-egy értelmű megfeleltetés létesíthető közöttük. A csucspontok köré írt köröket irányítsuk negatív körüljárással, mivel ekkor lesz a középpont a körív jobb oldalán.

Az összefüztést a következőképpen végezzük. Kövessük a G görbe irányításának megfelelő sorrendben az ofszet elemeket és csatlakoztassuk őket a következő módon:

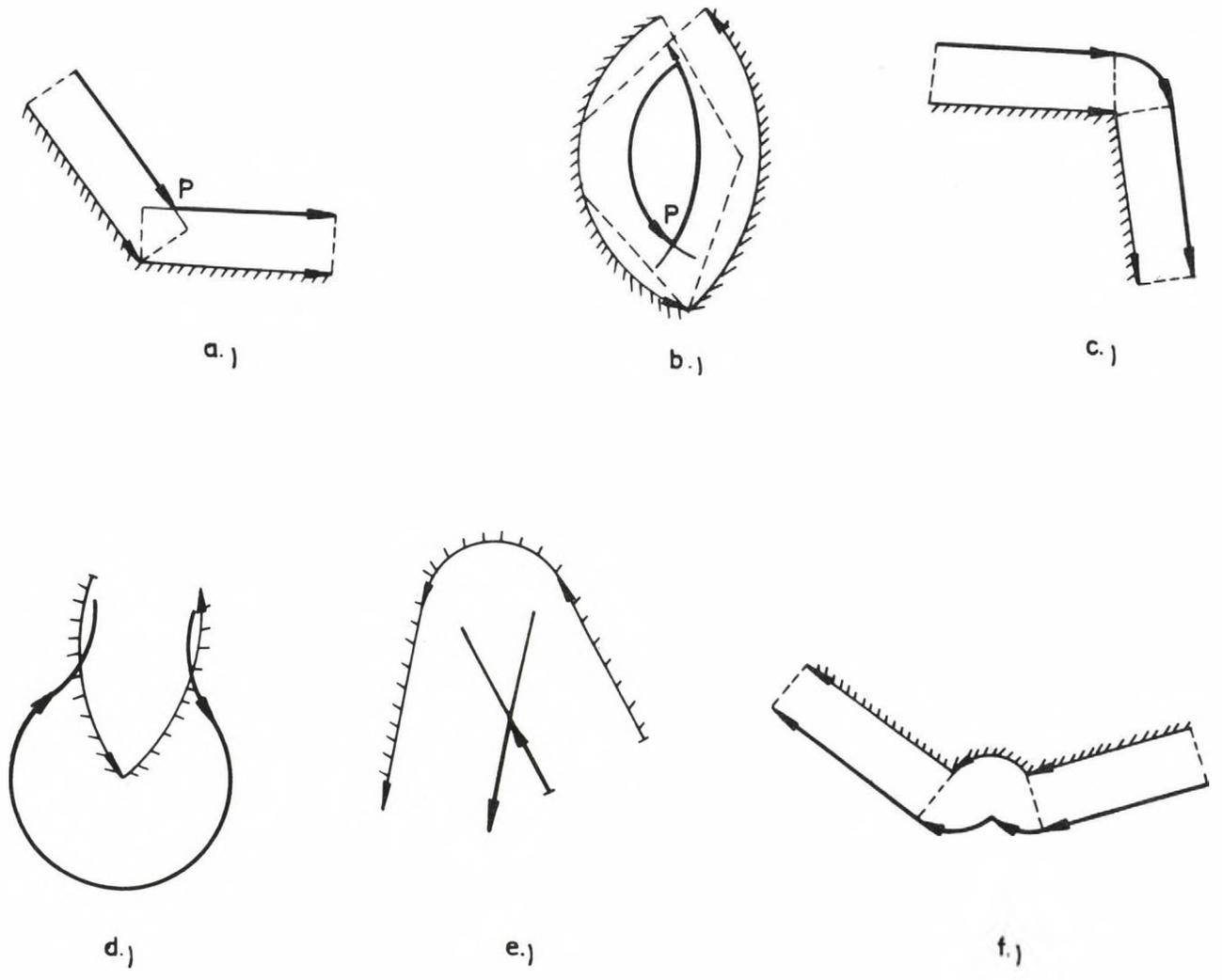
a/ Ha két, G -ben szomszédos komponensnek megfelelő ofszet elem metszi egymást (egy vagy több pontban) és az első görbedarabot tekintve az irányítás szerinti utolsó metszéspont P , akkor az első ofszet elemet csak P -ig tekintjük érvényesnek, a másodiknak pedig a P -ből induló darabjával dolgozunk tovább. Ez a két részgörbe P -ben csatlakozik egymáshoz (ld. 4.2/a,b ábra)

- b/ Ha két, egymást sorrendben követő ofszet elem nem metszi egymást, de mindkettő létezik, akkor kössük össze őket a hozzájuk tartozó G-beli csucs körüli körívvel, amelyet az irányítás egyértelműen kijelöl az első ofszet görbe végpontjától a második kezdőpontjáig. (Nyilvánvaló, hogy mindkettő rajta van ezen a körön.) (ld 4.2/c,d ábra)
- c/ Ha egy, vagy több egymást követő ofszet elem nem létezik (4.3/b. III.) akkor csatlakoztassuk a két közrefogó ofszet elemet a/ szerint, ha metszik egymást (4.2/e ábra). Ha nem, akkor G-beli csucspontok körüli köríveken történik a csatlakoztatás (ld. 4.2/f ábra)

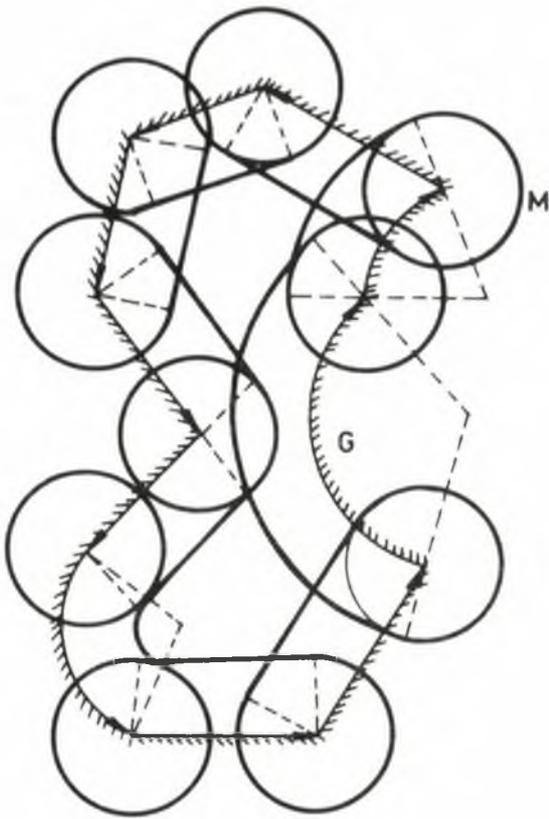
Nevezzük H-nak az így keletkezett, irányított, zárt, esetleg elfajuló görbét. (ld. 4.4. ábra)



4.1 ábra
Kontúrelemek ofszetje

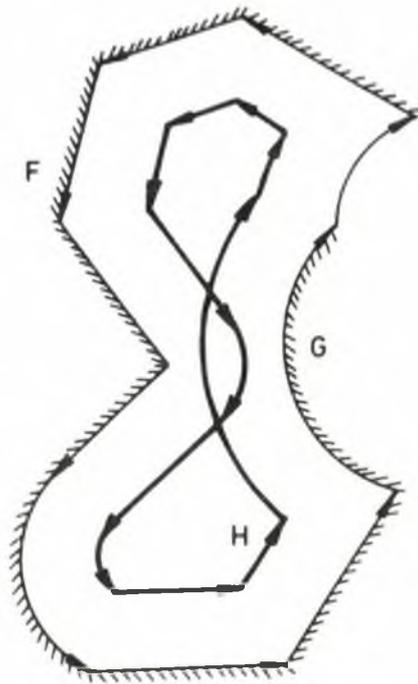


4.2 ábra
Ofszet elemek csatlakoztatása



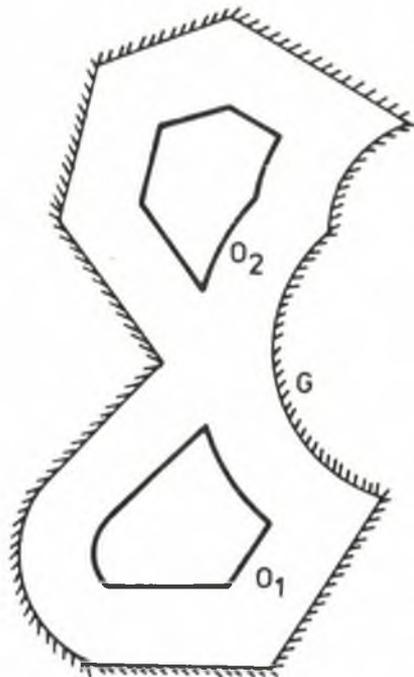
4.3 ábra

Az elemi ofszetelés eredménye



4.4 ábra

Összezsztatolt ofszet elemek



4.5 ábra

Ofszet kontúrok

4.5. OFSZET KONTUROK ELŐÁLLITÁSA

- a/ Határozzuk meg H összes önmetszési pontját. (Egybeeső szakaszok kezdő és végpontját tekintjük annak.)
- b/ Keressünk H-n egy olyan P_0 pontot, amely körüli R sugaru kör nem metsz bele az eredeti G görbébe.
- c/ Járjuk végig H-t P_0 -ból indulva és az egymást követő ofszet elemekre vonatkozó hivatkozást helyezzük el egy verem típusu V adatstrukturába, az önmetszési pontokra való hivatkozással együtt.
- d/ Mikor egy metszésponthoz másodszor érkezünk, egy hurkot zárunk be amely része a H görbének. Erre a hurokra vonatkozó információk a hurok bezárásának pillanatában a V verem tetején helyezkednek el, tehát jól vizsgálhatók. Vegyük a hurok egy tetszőleges P pontját, amely nem önmetszési pont. Ha P egy G-beli csucsponthoz tartozó ofszet elemen van akkor megnézzük, hogy a hozzátartozó G-beli csucsszög 180° -nál kisebb-e. Ha igen, ez a hurok biztosan nem jó ofszet görbének, f/-re megyünk.
- e/ Megnézzük, hogy a P köré irt R sugaru kör belemetsz-e G-be. Ha nem, akkor az így keletkezett hurok G egyik ofszet görbéje lesz, eredményként kiadjuk.
- f/ A hurokra vonatkozó információkat töröljük V-ből.
- g/ Ha a verem üres, vége az eljárásnak. Ha nem, folytatjuk H körbejárását c/-nél.

4.6. AZ ALGORITMUS ELEMZÉSE

4.6.1. Állítás

A 4.3.-ban előállított M halmaz görbéi tartalmazzák a keresett ofszet görbék minden pontját.

Bizonyítás

M görbehalmaz minden pontja maximum R távolságra van a G görbétől, hiszen egy pontjától pontosan R távolságra van, továbbá minden más pont, amely a görbe valamely pontjától pontosan R távolságra van, csak a G görbe jobb oldalán lehet.

4.6.2. Állítás

A 4.4.-ben előállított H görbe tartalmazza a keresett ofszet görbék minden pontját.

Bizonyítás

Mivel a H görbe, mint pontthalmaz, része M -nek, ezért csak azt kell belátni, hogy a 4.4-ben leírt eljárás során M -ből elhagyott görbedarabok nem lehetnek az ofszet kontur részei. A 4.4./a,b,c esetekre külön-külön belátjuk, hogy az elhagyott görbedarabok R -nél kisebb távolságra vannak a G görbétől, vagy pedig G jobb oldalán helyezkednek el.

a/ A 4.6. ábra a metszésponttal való csatlakoztatás különböző eseteit mutatja. Q pont a G görbe olyan csucspontja, ahol a szomszédos görbeelemek érintője különböző. Ekkor az ofszet elemeken Q -nak két különböző pont felel meg, Q_1 és Q_2 . Legyen az ofszet elemek metszéspontja P . Mivel

a Q pont közös, két oldal pedig metsző, az eredeti és az ofszet elemek által meghatározott téglalap vagy körgyűrű-darab átmetszi egymást. Nevezzük ezeket hizlalási sávoknak. Ha Q_1 illetve Q_2 benne van a szomszédos elemhez tartozó hizlalási sávban, akkor az attól való távolsága kisebb mint R .

Ha nincs benne, akkor viszont a G szerinti jobb oldali siktartományban kell lennie (másként nem lehetne a két hizlalási sáv metsző), tehát nem tartozik a keresett ofszet pontok közé. Ez igaz $\overline{Q_1P}$ és $\overline{Q_2P}$ minden pontjára.

A Q körüli R sugaru kört azért hagyhattuk el, mert konvex szögtartomány belső pontjainak valamely oldaltól való távolsága kisebb, mint a csucstól való távolság (ami pedig R), a konkáv tartomány pontjai pedig G jobb oldalán vannak.

- b/ Ebben az esetben csak a Q csúcspont körüli R sugaru kör egyik ivét hagytuk el. Nevezzük ezt a kört K -nak. Könnyű belátni, hogy a negatív körüljárású iv érintőlegesen köti össze a két ofszet elemet. Legyen ugyanis a Q csúcspontban a bemenő oldali érintő \underline{e} , a bemenő elemhez tartozó ofszet elem a Q -nak megfelelő pont Q_1 (4.7. ábra). A K kör Q_1 -beli érintője párhuzamos \underline{e} -vel. Az elemi ofszet képzés feltétele szerint $z(\underline{ex}\overline{QQ_1}) > \emptyset$ (ld. Függelék 1.) tehát az $z(\underline{ex}\overline{Q_1Q}) < \emptyset$. Ez viszont konvex sikidomoknál éppen a negatív körüljárás definíciója. Az érintővel ellenétes irányú körív viszont belemetsz a hizlalási sávokba majd a jobb oldali siktartományba, tehát nem tartozhat az ofszet konturhoz.

c/ Hasonlóan belátható, hogy az elhagyott iv κ -nál közelebb van G -hez, illetve a jobb oldali siktartományba esik.

4.6.3. Állítás

A H görbe minden pontjához hozzárendelhető egy tőle R távolságra levő pont G -n, az irányítás által létrehozott rendezés megtartásával, azaz ha P_1 és $P_2 \in H$, és $P_1 \prec P_2$ (azaz P_1 megelőzi P_2 -t), akkor létezik olyan $X_1, X_2 \in G$, hogy $X_1 \rightarrow P_1 = X_2 \rightarrow P_2 = R$ és $X_1 \prec X_2$ (azaz X_1 távolsága P_1 -től és X_2 távolsága P_2 -től egyaránt R , és X_1 megelőzi X_2 -t vagy azonos vele).

Bizonyítás

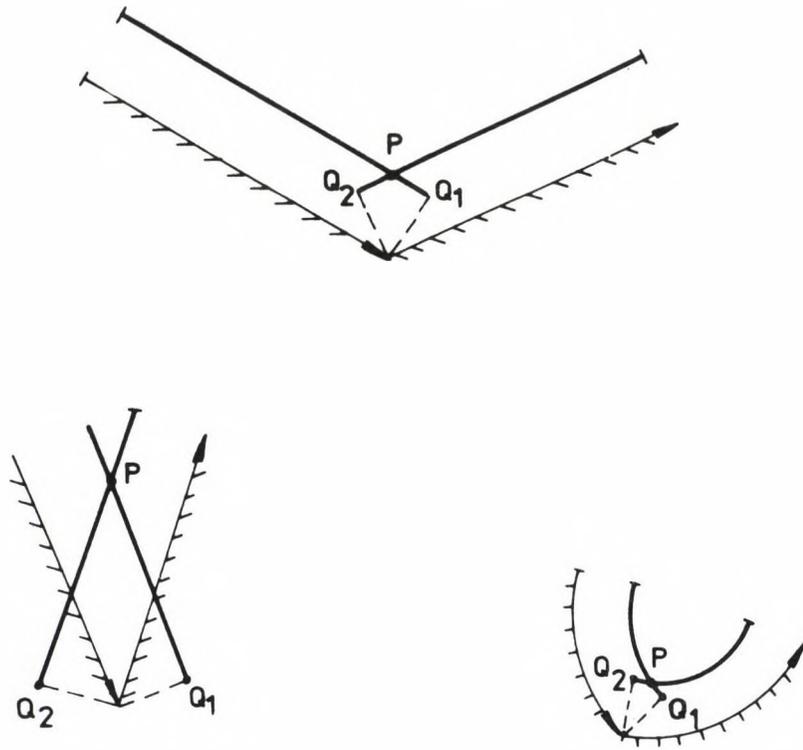
H előállításából következik a megfeleltetés.

4.6.4. Állítás

A G -hez tartozó ofszet konturok olyan részgörbéi H -nak, amelyek zártak és nem önmetszők, vagy pedig teljesen elfajulóak, azaz minden pontjuk több elemhez tartozik egyszerre.

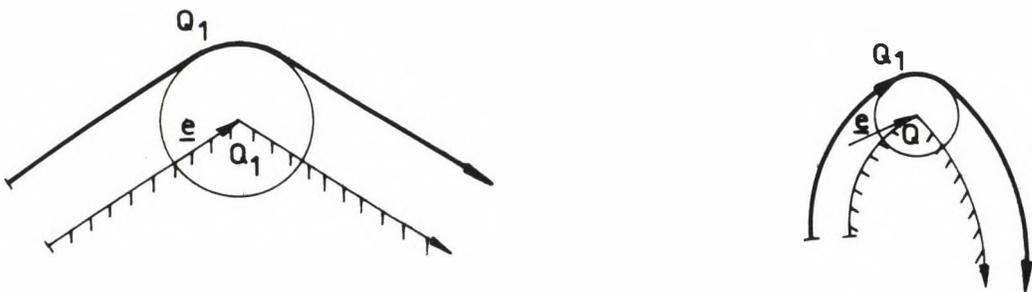
Bizonyítás

Az önmetsző görbék szétbonthatók nem önmetszőekre. A zárt-ság a következőképpen bizonyítható. Az ofszet görbék határolják azokat a területeket, amelyek a G görbe baloldali tartományában R -nél nagyobb távolságra vannak G -től. Ugyanis bármely két pont között, amelyek egyike R -nél távolabb, másika R -nél közelebb van G -hez, kell lenni pontosan R távolságra levő pontnak.



4.6 ábra

Metsző hizlalási sávok



4.7 ábra

Érintőleges csatlakoztatás

Az elfajuló görbék nem határolnak területet, ám ezek is zárt görbeként állnak elő, egybeeső éllel. Tekintsünk ugyanis egy olyan görbedarabot, amely pontosan R távolságra van G -től és mindkét oldalán R -nél kisebb távolságra levő pontok vannak. Akkor mindkét oldalán kell haladni G -beli elemnek, tehát ez a H -beli görbedarab kétszeresen áll elő az elemi ofszetelés során, mégpedig ellentétes irányítással.

Az egy ponttá elfajuló görbe tekinthető zártnak.

4.6.5. Állítás

H -nak bármely összefüggő darabja, amely nem tartalmaz önmetszési pontot, vagy teljes egészében része egy ofszet konturnak, vagy egyetlen pontja sem.

Bizonvítás

Egy P pont, amely nem csatlakozási pontja az ofszet elemeknek, akkor önmetszési pontja H -nak, ha van olyan $X_1, X_2 \in G$ hogy $X_1 \neq X_2$ és $P \rightarrow X_1 = P \rightarrow X_2$.

Tegyük fel, hogy az állítás nem igaz egy $K \subset H$ görbedarabon. Ekkor van olyan két pont, P_1 és P_2 , amely azonos szakaszon vagy köríven van, és $P_1 \rightarrow G < R$, $P_2 \rightarrow G = R$ (ld. 4.8. ábra). Akkor viszont 4.6.3. szerint van olyan $X_1, X_2 \in G$, hogy $X_1 \rightarrow P_1 = X_2 \rightarrow P_2 = R$, és X_1, X_2 azonos G -beli elemek van.

Az indirekt feltevés miatt létezik $Y \in G$, $Y \rightarrow P_1 < R$ egy másik G -beli elem.

Vegyünk Y körül egy R sugaru kört. Ez tartalmazza P_1 -et, de P_2 -t nem, tehát a P_1 -et P_2 -vel összekötő K görbe átmet-

szi ezt a kört egy P_3 pontban. Ez a P_3 pont R távolságra van Y -től, viszont 4.6.3. miatt X_1 és X_2 között is kell lenni G -n egy P_3 -től R távolságra levő X_3 pontnak. P_1 és P_2 azonos H -beli elemek, tehát közöttük csatlakoztatási pont nem lehet. Tehát P_3 önmetszési pontja H -nak, ami elmentmond az indirekt feltevésnek, hogy K nem tartalmaz önmetszési pontot, tehát az állítás bizonyítva van.

4.6.6. Állítás

Ha P pont két H -beli elem metszéspontja, akkor amelyik elem P belső pont, arra igaz, hogy P olyan görbedarabokat határol, amelyek közül legfeljebb az egyik része valamelyik ofszet konturnak.

Bizonyítás

A metsző elem keresztülhalad a másik elem hízalási sávján.

4.6.7. Állítás

Ha G elemeit az irányításnak megfelelően úgy indexeljük, hogy a csúcspontokat is beleszámítjuk és ezt átvisszük az ofszet konturokra H előállításának megfelelően, akkor minden ofszet konturon választható olyan kezdőpont, hogy a konturt körbejárva az átvitt indexek szigorúan monotonon nőnek.

Bizonyítás

A monotonitás abból következik, hogy G nem önmetsző. Egyenlőség csak a szomszédos elemeknél léphetne fel, de azt 4.6.6. kizárja.

4.6.8. Állítás

A 4.5.-ben leirt algoritmusban keletkező H-beli hurkok nem tartalmazhatnak olyan görbedarabokat, amelyek egy része hozzátartozik az ofszethez, más része nem. (Ebből következik, hogy c/-ben elég egy pontot vizsgálni.)

Bizonyítás

Tekintsünk egy $O \subset H$ ofszet görbét, amelynek 4.6.7. szerinti G-ből származó indexei $i_1 < i_2 < \dots < i_n$.

Tegyük fel, hogy O valamely darabja az eljárás során egy olyan hurokba kerül, amelynek nem felel meg az ofszet kontur feltételeinek, és ez az első ilyen darab. 4.6.5-ből tudjuk, hogy az ilyen darabokat H önmetszési pontjai határolják. Legyen P a rossz hurokba kerülő darab kezdőpontja, az i_j és i_{j+1} H-beli elemek metszéspontja. Ha P-ből az i_j elem végpontja felé haladunk H-n, akkor az i_{j+1} elem előtt egyik H-beli elem sem metszi az O görbét, i_{j+1} elem viszont éppen P-ben metszi. Tehát P-ben bezárult egy hurok, amely nem metszett O-ba és így nem tartalmazhat O-beli részgörbét. Ezt a hurkot megvizsgálva és a V adatstrukturából elhagyva O képzését az i_{j+1} elemmel folytatjuk. Ez minden metszéspontban igaz, tehát O-n körbejártunk anélkül, hogy közben egy darabja másik hurokba került volna.

Megjegyzés

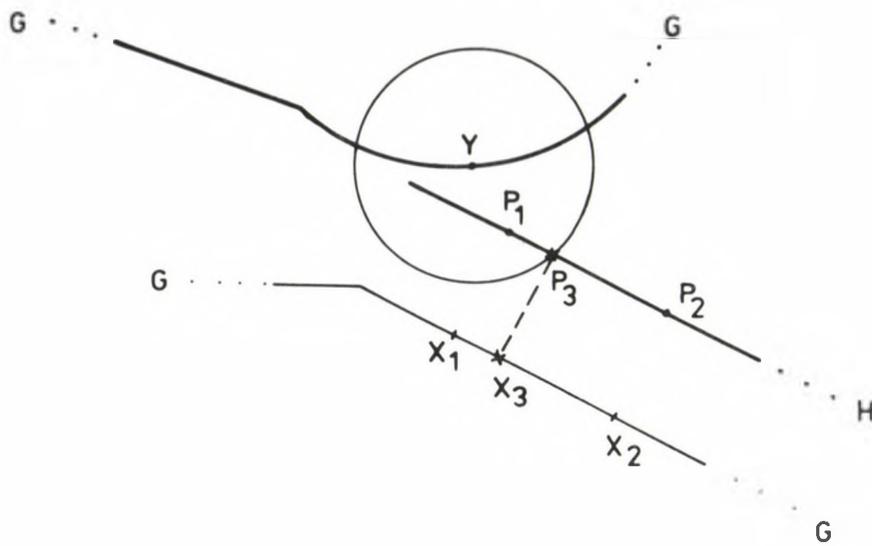
A 4.5/b kiindulási feltétel szükséges, mert különben az állítás nem igaz (ld. 4.9. ábra), hiszen a bizonyításban feltételeztük, hogy egy ofszet konturon haladunk.

4.6.9. Állítás

A 4.5. eljárás pontosan kiadja a G görbe ofszet konturjait.

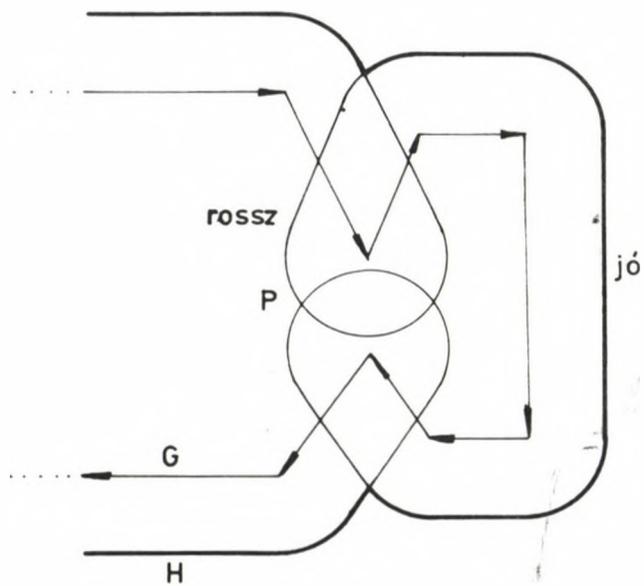
Bizonyítás

4.6.8. szerint az összes ofszet kontur előáll H -bel hurokként és ezeket az eljárás végigjárja.



4.8 ábra

Ofszet kontúrhoz tartozás vizsgálata



4.9 ábra

Hurokképzés H-ban rossz kiinduló pontból (P)

5. OFSZET SIKGÖRBÉK ÁTHATÁSA, EGYESITÉSE

5.1. A PROBLÉMA FELVETÉSE

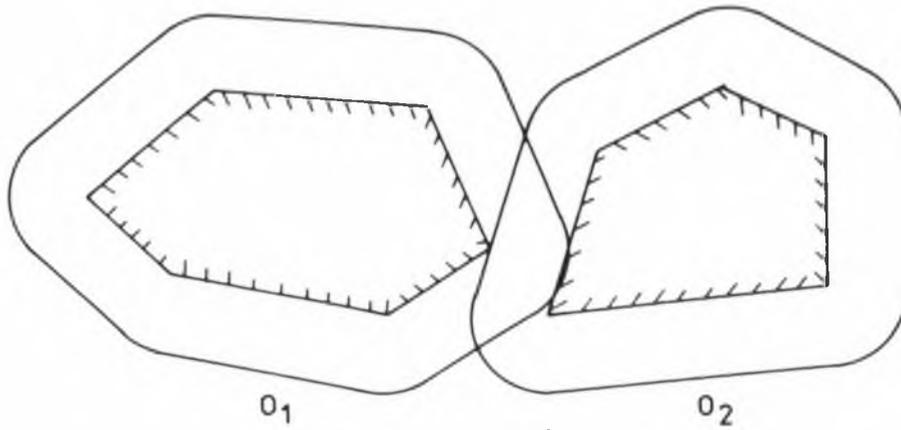
A 2.3.-ban leírt 2D-s megmunkálási folyamat második fázisára akkor van szükség, ha a kiindulási görbék száma (az ofszetelés előtt) egynél nagyobb. A 4. fejezetben leírt ofszetelő eljárást az egyes görbékre külön-külön végezzük, az így kapott eredmény nem tükrözi az együttes korlátozásokat (ld. 5.1. ábra). Ha a kapott ofszet görbék nem metszenek egymásba és nem tartalmazzák egymást, akkor az általuk kijelölt megmunkálási tartomány egyértelmű és a 6. fejezetben leírt szerszámpálya-generáló eljárás elvégezhető. Ha viszont metszik vagy tartalmazzák egymást, a megmunkálási tartományok halmazelméleti metszetét kell képezni, ami azonos az anyagi tartományok diszjunkt egyesítésével. Jelen fejezet eljárást ad az így keletkező tartományok határgörbéinek meghatározására. A halmazelméleti metszet illetve unióképzés disztributivitása miatt az algoritmust elegendő két metsző görbére kidolgozni, a teljes egyesítés páronként szekvenciálisan elvégezhető.

Az 5.2. alfejezet ismerteti az algoritmust, amely két görbe áthatásából egy vagy több egyesített ofszet konturt képez, a hozzá szükséges részfeladatok megoldását pedig az 5.3, 5.4, 5.5 alfejezetek tartalmazzák.

5.2. KÉT GÖRBE ÁTHATÁSA

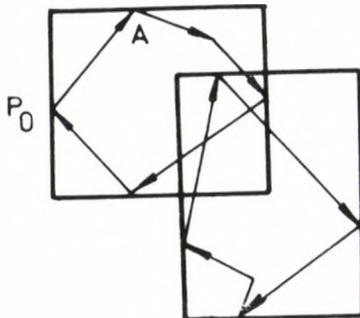
- a/ Mindkét görbét befoglaljuk egy-egy, a koordináta tengelyekkel párhuzamos oldalu téglalapba az x és y szélsőértékeik alapján. (5.2. ábra) Ha ez a két téglalap diszjunkt, a két ofszet görbe biztosan nem metsződik, az eljárás véget ér.
- b/ Meghatározzuk a görbék körüljárását (ld. 5.3).
- c/ Keresünk egy olyan P_0 pontot a két görbe valamelyikén, amelyik a két görbére nézve extrémális x vagy y értékkel rendelkezik és csak az egyik görbén van rajta. Nevezzük a P_0 -t tartalmazó görbét A-nak, a másikat pedig B-nek (5.2. ábra).
- d/ A körüljárás segítségével eldöntjük, hogy P_0 pont hozzátartozik-e az egyesített ofszet konturok valamelyikéhez (ld. 5.4). Azt mondjuk, hogy P_0 "jó" ha hozzátartozik, "rossz", ha nem.
- e/ P_0 -ból indulva A görbe mentén az irányítás szerint haladva meghatározzuk a két görbe közös pontjait. Jelölje ezeket rendre Q_0, Q_1, \dots, Q_n (ld. 5.7. ábra) Ha nincs közös pont, akkor a két kontur nem metszi egymást, az eljárás véget ér. Ha P_0 "jó" volt, mindkét kontur megmunkálható, ha "rossz", akkor egyik sem (5.2. ábra).
- f/ Q_0 -ból kiindulva elkezdjük az első egyesített ofszet kontur felfűzését, ha P_0 "jó" volt, akkor a B konturon, egyébként az A konturon haladva. A felfűzés azt jelenti, hogy az eredeti kontur elemeit csatlakoztatjuk az egyesített konturhoz.

- g/ A fűzés során elért Q_i közös pontokról eldöntjük, hogy metszéspontok vagy érintési pontok. Az elsőként talált metszéspontnál (legyen ez Q_j) áttértünk a másik konturra és ott folytatjuk az egyesített konturt.
- h/ Az aktuális kontur elemeit felfűzzük, amíg visszajutunk Q_0 -hoz. Ezzel létrehoztunk egy egyesített ofszet konturt, amely eredményként kiadható. Q_0 -t és Q_j -t töröljük a metszéspontok listájából.
- i/ Ha a metszéspontok listája üres, az eljárás véget ér.
- j/ Legyen Q_i az első nem törölt metszéspont a listán. Q_i -ből kiindulva elkezdjük felfűzni az A kontur elemeit.
- k/ Az elsőként talált metszéspontnál áttérünk a B konturra és addig folytatjuk a felfűzést amíg vissza nem érkezünk Q_i -hez.
- l/ Töröljük Q_i -t és Q_k -t a metszéspontok listájáról, az egyesített konturt pedig eredményként kiadjuk.
- m/ Folytatjuk az algoritmust i/-nél.

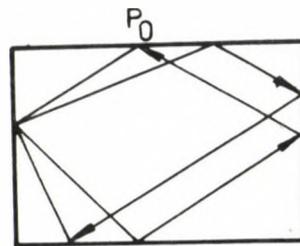


5.1 ábra

Egymást metsző ofszet kontúrok



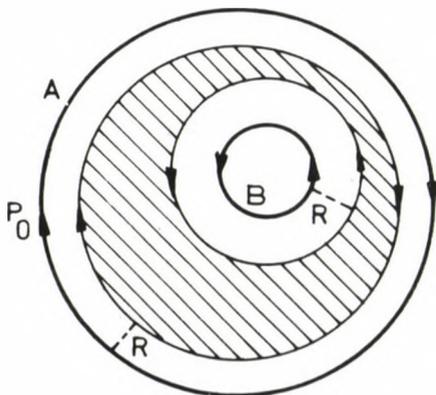
a.1



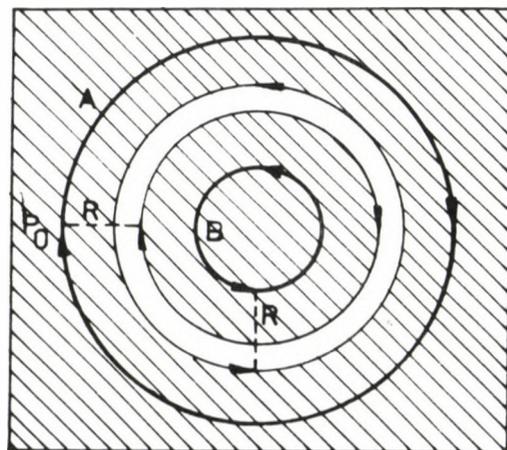
b.1

5.2 ábra

Extremális téglalapok



a.1



b.1

5.3 ábra

Megmunkálható és meg nem munkálható ofszet kontúr-párok

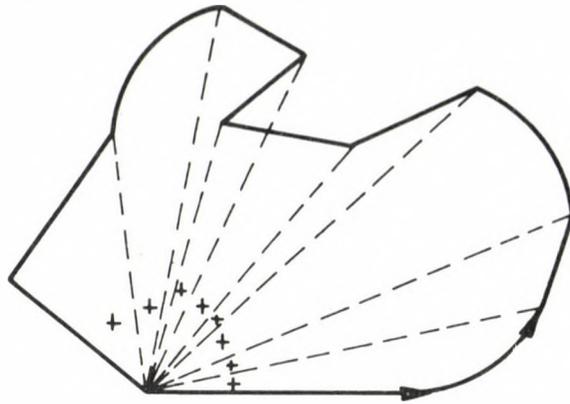
5.3. SIKGÖRBÉK KÖRÜLJÁRÁSÁNAK MEGHATÁROZÁSA

A körüljárás meghatározására egy szokásos módszer a következő. Tekintsük a körivekből és egyenes szakaszokból álló görbe egy tetszőleges csúcspontjából az összes, ezt a csúcst nem tartalmazó konturelem látószögét. Irányítsuk ezeket a szögeket a kontur irányításának megfelelően és előjelesen összegezzük őket. Az eredmény előjele pontosan a körüljárást adja meg. (5.4. ábra) Nulla eredmény esetén a görbe elfajuló.

Ez az eljárás akkor gyors, ha a csúcspontok száma kevés. Nagy elemszámú offset konturok esetén hatékonyabb a következő módszer.

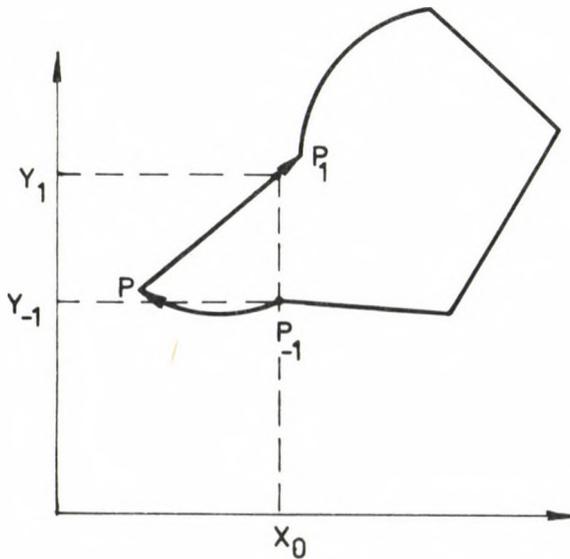
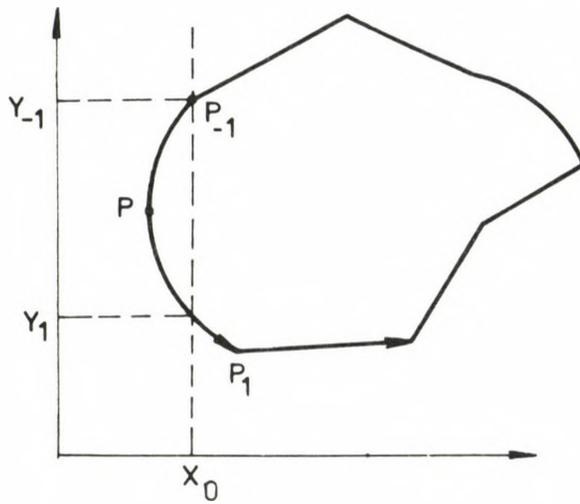
A teljesen elfajuló görbéket tekintsük pozitív körüljárásuaknak, ez megfelel annak az elvnek, hogy a pozitív körüljárásu görbék a belső görbék, azaz a belsejüket kell megmunkálni (ld. Függelék 4.). Részben elfajuló görbék esetén a körüljárást a nem elfajuló rész határozza meg.

Nem elfajuló görbén keressünk egy P pontot, ahol a görbe felveszi x -beli minimumát. (Jelen esetben ez nem növeli a számításokat, mert az x és y -beli szélsőértékekre amugy is szükség van). Ha P egy körív belső pontja, legyen P_{-1} a körív kezdőpontja, P_1 pedig a végpontja. Más esetben P egy csúcspont, legyen P_{-1} az irányítás szerinti megelőző, P_1 pedig a rákövetkező csúcspont. Legyen x_0 a P_{-1} és P_1 -beli x értékek minimuma. Messük el a görbe szóban forgó elemeit $x=x_0$ egyenessel, legyen a P_{-1} P elemmel való metszéspont második koordinátája y_{-1} , P_1 -hez pedig y_1 . (5.5. ábra) Mivel a görbe nem elfajuló, ezért $y_1 \neq y_{-1}$. Ha $y_{-1} > y_1$, akkor



5.4 ábra

Kontúr körüljárásának meghatározása szögek ből



5.5 ábra

Kontúr körüljárásának meghatározása szélsőérték - pont környezetéből

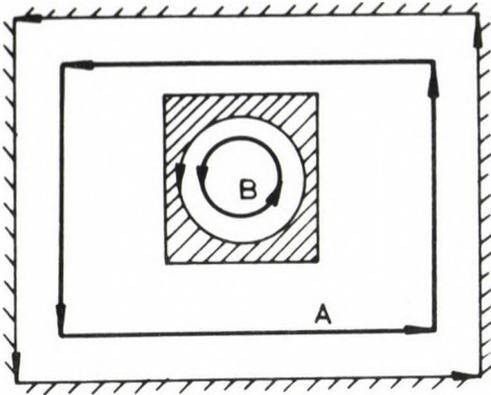
a görbe körüljárása pozitív, egyébként negatív. Ez megfelel a körüljárás definíciójának, mivel a $P_{-1}P_1$ görbedarabot a ki-metszett függőleges szakasszal helyettesítve a síkgörbe bel-seje pozitív x irányban van (ld. [30]).

5.4. KIINDULÁSI PONT TIPUSÁNAK MEGHATÁROZÁSA

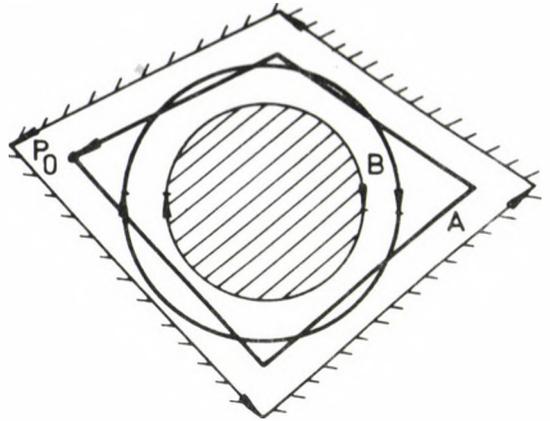
Az esetek többségében a P_0 pont (ld. 5.2/c) típusa el-dönthető a két kontur körüljárásának ismeretében. Az ábrák alapján könnyen ellenőrizhetők az alábbi esetek:

"A" kontur irányítása	"B" kontur irányítása	P_0 pont típusa	ábra
pozitív	pozitív	"jó"	5.6/a
pozitív	negatív	"jó"	5.6/b
negatív	negatív	"jó"	5.6/c

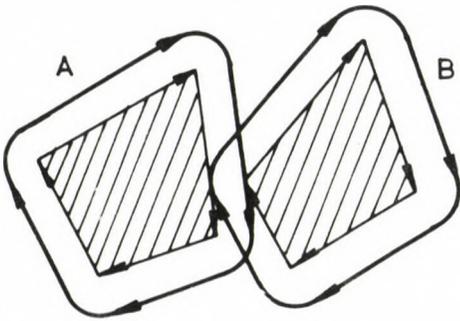
A negyedik eset, ha az A kontur irányítása negatív, B konturé pedig pozitív, az 5.5/d ábrán láthatóan "rossz" esetet ad P_0 -ra. Ez azonban csak akkor igaz, ha a két kontur metszi egymást. Tartalmazás esetén azonban külön vizsgálatokra van szükség (5.3 ábra). Megnézzük, hogy a P_0 körüli $2R$ sugaru kör (R volt az ofszetelés sugara) belemetsz-e a B konturba. Ha igen, P_0 "rossz", egyébként "jó".



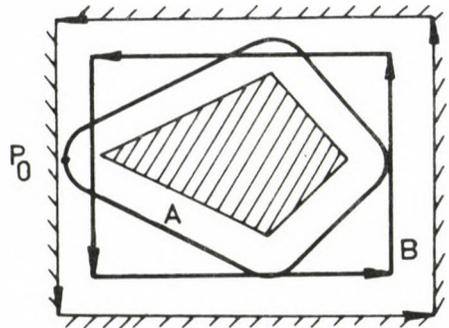
a.)



b.)



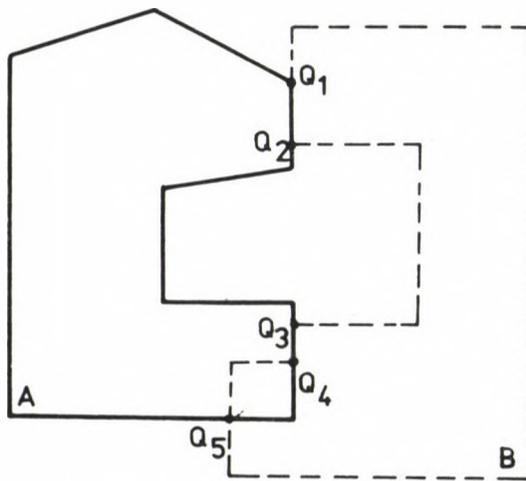
c.)



d.)

5.6 ábra

Kiindulási pont típusának meghatározása



5.7 ábra

Kontúrok érintkezése és metszése

5.5. METSZÉSPONTOK ÉS ÉRINTÉSI PONTOK MEGKÜLÖNBÖZTETÉSE

Az 5.2/e-ben szükséges metszéspont-sorozatot a két görbe elemeinek metszésével képezzük. Az elemenkénti metszés során az érintési esetek nagy részét ki lehet szűrni köriv-köriv, köriv-szakasz, amikor a kritikus pont mindkét elemnek belső pontja. A fennmaradó esetekben (pl. 5.7. ábra) utólag az érintési pontoktól.

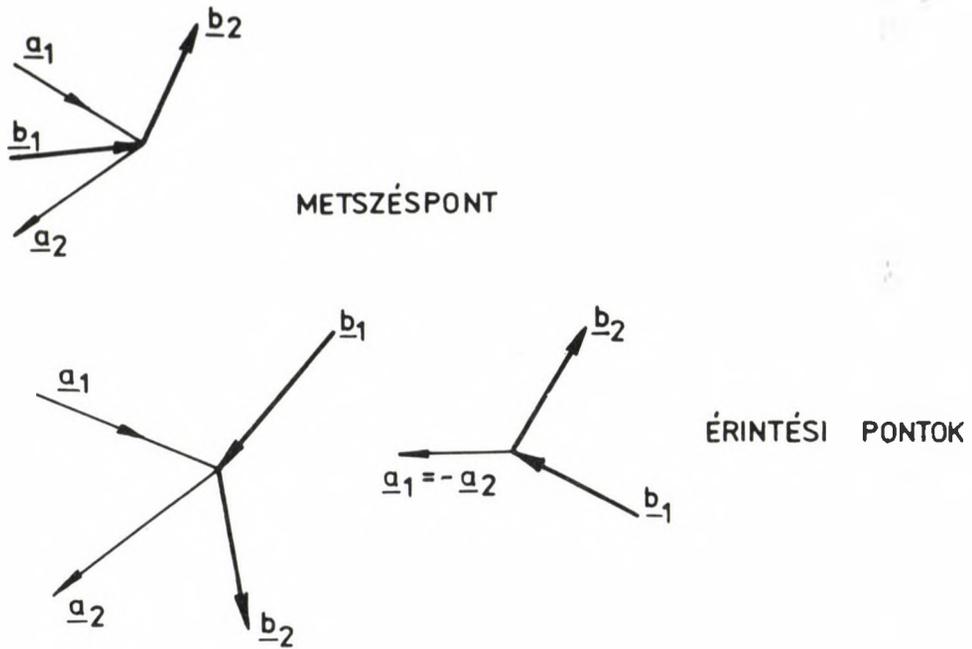
Tekintsük tehát az A és B görbe egy közös Q_i pontját. (ld. 5.2/e). Ha ez egyik görbén sem csucspont, akkor a fentiek miatt biztos, hogy metszéspont.

Ha viszont Q_i legalább az egyik görbén csucspont, akkor további vizsgálatra van szükség. Legyen az A görbén Q_i -be befutó konturelem érintője \underline{a}_1 , a kifutóé \underline{a}_2 , az irányításnak megfelelő egységvektorok, B görbén hasonlóan \underline{b}_1 és \underline{b}_2 .

a/ Ha ezek egyike sem esik egy félegyenesre (5.8. ábra), tekintsük az $(\underline{a}_1 \underline{a}_2)$ szög által létrehozott két siktartományt. Ha \underline{b}_1 és \underline{b}_2 különböző tartományban van, akkor Q_i metszéspont, egyébként érintési pont.

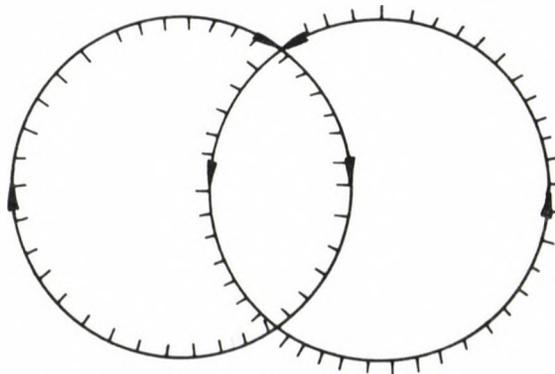
b/ Ha a vektorok között azonosak vannak, akkor vagy egybeesik a két görbe egy darabon (ld. 5.7. ábra) vagy merőlegesen érintkező köriveink vannak. Merőlegesen találkozó köriveknél az érintők helyett a kezdőpontból a kritikus pontba, illetve onnan a végpontba mutató szelőt vizsgáljuk a/ szerint.

c/ Ha a két görbe egy darabon illeszkedik egymáshoz, ha Q_i a sorrendben első közös pont, Q_{i+1} lesz az utolsó, mivel csak ezek jelentkeznek metszéspontként. Vonjuk össze ezt a két pontot, azaz tekintsük Q_i -ben \underline{a}_1 -et és \underline{b}_2 -t,



5.8 ábra

Metszési és érintési pontok megkülönböztetése



5.9 ábra

Ellentmondásos kontúrirányítás

Q_{i+1} -ben pedig \underline{b}_1 -et és \underline{a}_2 -t, és ezekre vizsgáljuk meg az a/ feltételt. Ha ez a vizsgálat érintési pontot mutat, akkor Q_i és Q_{i+1} is az. Ha ez metszéspont, a két pont közül csak egyet tekinthetünk metszéspontnak, Q_i -t akkor, ha 5.2/j-ben vagyunk, vagy 5.2/f-ben úgy, hogy P_0 pont "rossz" volt, egyéb esetekben Q_{i+1} -et. Ezáltal az illeszkedő darabokat elfajuló részként hozzáfűzzük az egyik közös ofszet konturhoz.

5.6. AZ ALGORITMUS ELEMZÉSE

A segéd eljárások leírása során már megindokoltuk a nem nyilvánvaló lépéseket, ezért most csak az 5.2 algoritmus helyességének bizonyításával foglalkozunk.

- 1/ Mivel zárt görbékről van szó, két kontur metszéspontjainak a száma mindig páros. A metszéspontok mindkét görbén olyan darabokat határolnak, amelyek egyike a másik görbe jobb oldali, másika a bal oldali tartományába esik, tehát mindkét görbén "jó" és "rossz" darabok váltják egymást.
- 2/ Mivel az ofszetelés előtti, kiinduló görbék irányítása egyértelműen szétválasztja az anyagi és a megmunkálási tartományt, az ofszet görbék irányítása pedig követi az eredetiekét, a két metsző ofszet görbe által létrehozott siktartományokról is egyértelműen eldönthető, hogy melyik megmunkálható és melyik nem.
- 3/ A megmunkálási tartományok nem feltétlenül egyszeresen összefüggőek, tehát határuk több görbéből is állhat. Minden határoló görbe "jó" darabokból áll, még hozzá pontosan

két metszéspont által kijelölt egy-egy A és B görbéhez tartozó darabból. (1/ miatt)

4/ Állítás

Minden határgörbén, amely megmunkálható és anyagi tartományt választ el, az A és B konturbeli darabok irányítása egymáshoz csatlakozó.

Bizonyítás

Tegyük fel, hogy ellentétes (ld. 5.9. ábra). Mivel mindkét görbedarabnak jobb oldalán van az anyagi tartománya, viszont az egyik a zárt görbe belsejébe, a másik pedig a külsejébe esik, tehát egyik sem lehet megmunkálható tartomány.

5/ Állítás

Az 5.2. eljárás előállítja a megmunkálási tartományok határgörbéit és csak azokat.

Bizonyítás

Az 5.2. eljárásban elsőként előállított kontur megkezdésénél külön vizsgálattal biztosítjuk, hogy "jó" darabon kezdjük el a felfűzést. A továbbiakban mindig A görbén kezdjük a felfűzést, mivel a metszéspontok rendezése A szerint történt és ha egy "jó" görbedarab lezáró metszéspontját elhagyjuk, akkor az utána következő "rossz" darab kezdete is törlődik. Tehát a soron következő metszésponttól A-n irányítás szerint haladva ismét "jó" görbedarabon vagyunk.

A B görbére való áttéréskor 4/ miatt szintén "jó" darabon folytatjuk a felfűzést, 3/ miatt pedig visszajutunk a kezdőpontba.

Mivel a metszéspontok száma véges, az eljárás véget ér. Láttuk, hogy az összes felfűzött görbedarab "jó" volt, másrészt minden metszéspontnál mindkét görbéről felfűztünk egy-egy határolt görbedarabot, tehát csak "rossz"-akat hagyhattunk el.

6. SIKBELI SZERSZÁMPÁLYA GENERÁLÁS

6.1. A FELADAT MEGFOGALMAZÁSA

A síkmetszeti konturok ofszetelése és az egymást metsző ofszet konturok egyesítése után keletkező görbék az előgyártmány síkmetszetén belül egyértelműen kijelölik a megmunkálási tartományokat. Feltételezzük, hogy az előgyártmány síkmetszete olyan zárt görbe, amely szintén egyenes szakaszokból és körivekből áll, nevezzük E-nek. A megmunkálási tartományokat az E görbe belsejében levő, körivekből és egyenes szakaszokból álló zárt ofszet görbék határozzák meg olyan módon, hogy pozitív körüljárás esetén a göbe belsejét, negatívnál a külsejét kell megmunkálni. Mivel az ofszet görbék a síkmetszeti konturoktól mindenütt R távolságra vannak, a szerszámpanyát az ofszet görbéken is végig kell vezetni.

Keresünk tehát olyan görbéket, amelyeken egy adott R sugaru kört végigfuttatva a kijelölt megmunkálási tartományokat, és csak azokat, teljesen lefedjük.

2D-s szerszámpanya előállításához általában kétféle stratégiát használnak:

- 1/ cikk-cakk
- 2/ konturkövető (meander)

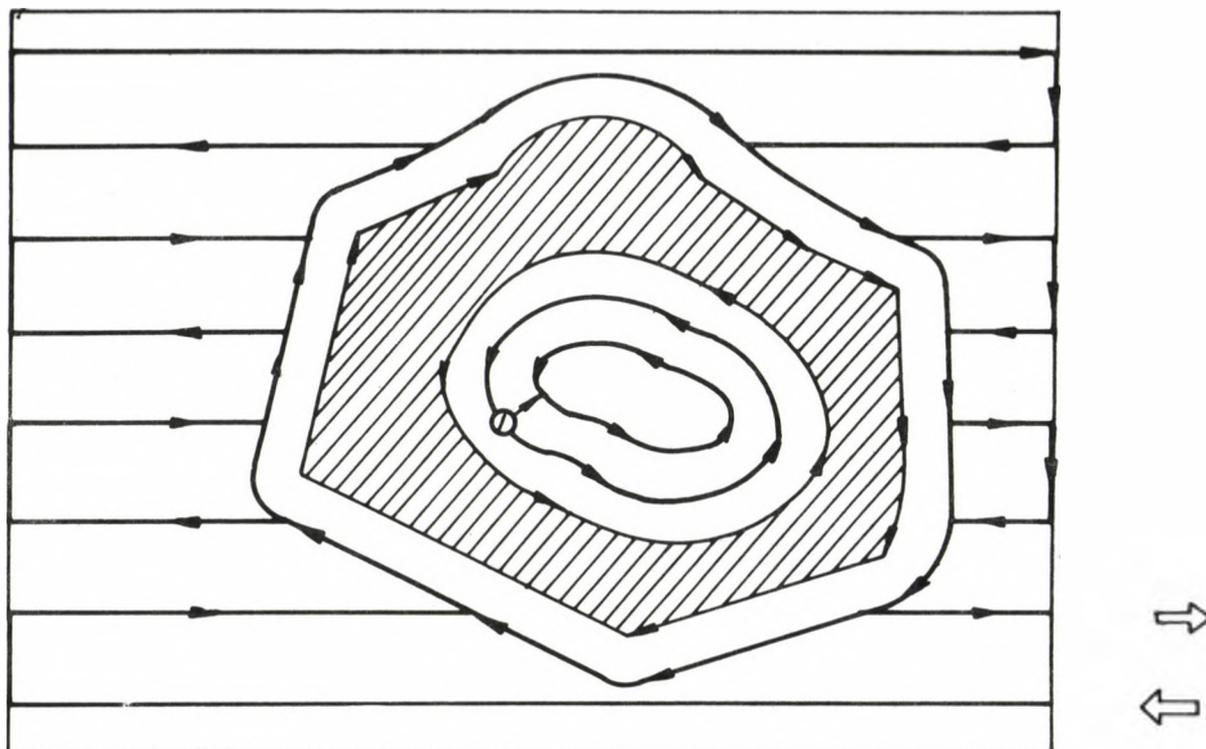
Cikk-cakk stratégia esetén a szerszámpanya egy adott egyenessel párhuzamosan, alternáló irányítással halad. Konturkövető szerszámpanyát zsebek belsejének megmunkálásához szokás alkalmazni, ez a kontur egymás utáni többszöri ofszetelésével áll elő. (ld. 6.1. ábra)

A 4. fejezetben ismertetett ofszetelő eljárás iterált alkalmazásával lehetőségünk nyílna konturkövető pálya előállítására, de ez igen hosszú számítási időt igényelne, ezért olyan algoritmust kerestünk, amely cikk-cakk pályát eredményez. Jelen fejezet két ilyen algoritmust ismertet. Az első algoritmus lényegesen egyszerűbb ám a második algoritmus által eredményezett szerszám-pálya sokkal gyorsabb és gazdaságosabb megmunkálást biztosít. Mindkét algoritmus esetén feltételzzük, hogy a megmunkálás iránya (amellyel párhuzamos a pálya) tetszőleges, de előre meghatározott egyenes.

6.2. ELSŐ ALGORITMUS

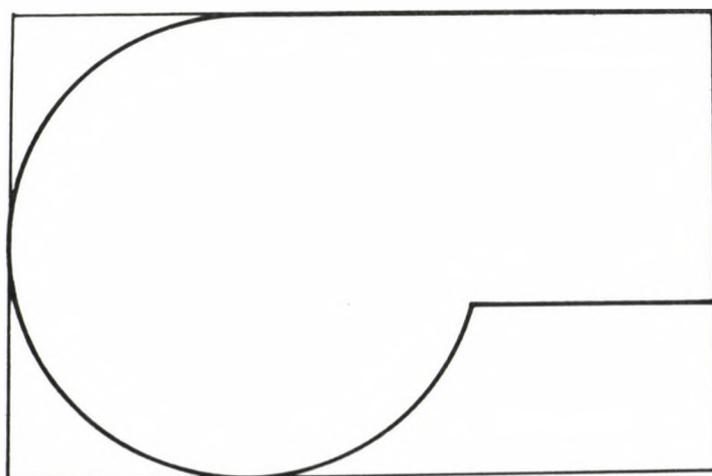
Az első algoritmus lényege, hogy a szerszám az egész területet, amely megmunkálandó részeket tartalmazhat, bejárja, vagy megmunkálva, vagy pedig kiemelve, üresjáratban. Ezen kívül a tartományokat szétválasztó (ofszet) konturok mentén is végighalad. Az algoritmus a 6.3. ábrán követhető.

- a/ Meghatározzuk azt a legkisebb T téglalapot, amely az E előgyártmányi síkmetszetet tartalmazza és egyik oldala párhuzamos a megmunkálási iránnyal (6.2. ábra).
- b/ T téglalap azon oldalát, amelyik a megmunkálási irányra merőleges, felosztjuk úgy, hogy a kapott szakaszok hossza ne legyen nagyobb, mint $1.6 \cdot R$, a két szélsőé pedig pontosan R legyen. Az így kapott felosztást megmunkálási szinteknek nevezzük.



6.1 ábr 1

Cikk-cakk és kontúrkövető megmunkálás kombinálása



6.2 ábra

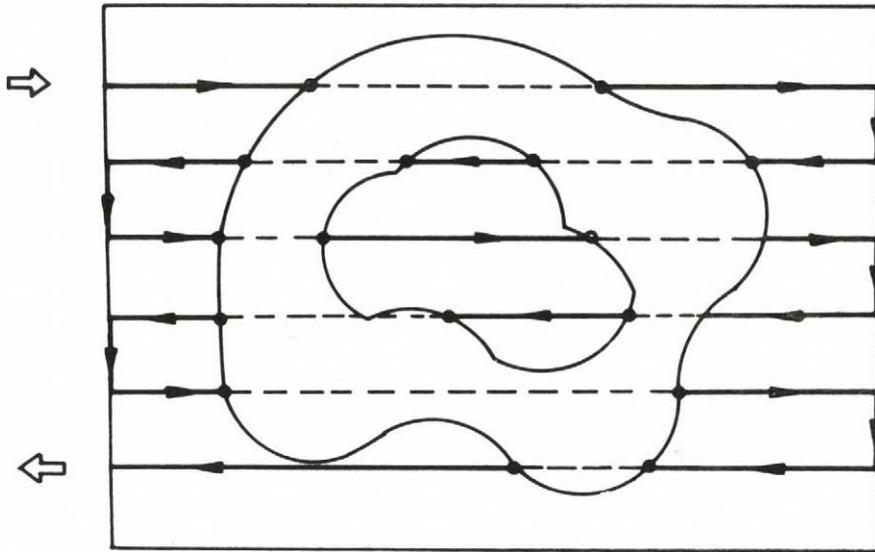
Az előgyártmány és a betoglaló téglalap

- k/ Ha van még az aktuális megmunkálási szinten, a haladási irányban metszéspont, g/-nél folytatjuk az eljárást.
- l/ Ha van még megmunkálási szint, vesszük a sorrendben következőt, megfordítjuk a haladási irányt és e/-nél folytatjuk az eljárást.
- m/ Azokat a konturokat, amelyeket az eljárás során nem érintettünk, munkameneti pályával körbejárjuk, negatív körüljárás esetén furóciklussal kezdve.

6.3. JAVITOTT ALGORITMUS

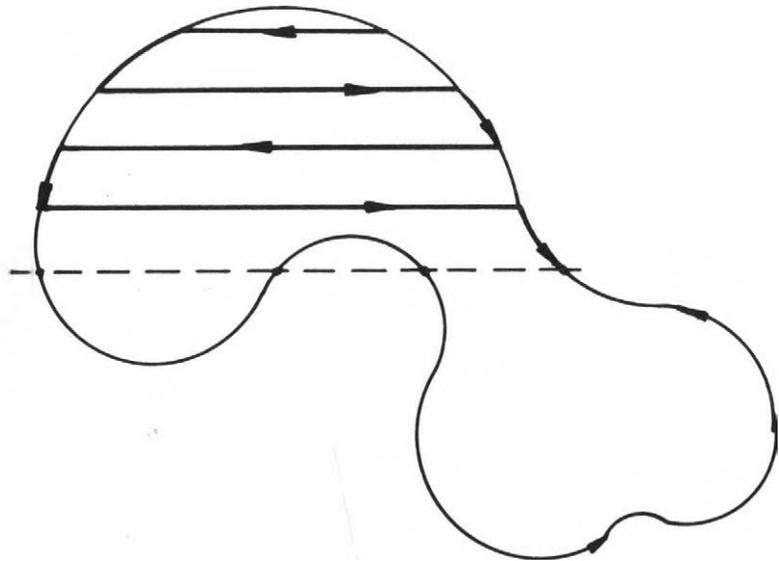
A második algoritmus lényege, hogy a szerszám nem járja be az E által meghatározott egész területet, hanem minimalizálja az üresjáratit mozgást. Az algoritmus a 6.6/b ábrán követhető.

- a,b,c/ lépések megegyeznek az első algoritmussal
- d/ Beállítjuk az első megmunkálási szintet, a szinten való haladási irányt és a szintek közötti haladási irányt, mert ebben az algoritmusban az is változni fog.
- e/ Az összes konturt az összes megmunkálási szinten elmetszük a megmunkálás irányával párhuzamos egyenessel, ha $KJ = \text{negatív}$ akkor az E konturt is. A metszéspontokat a szinten való haladási irány szerint szintenként sorba rendezzük és egy M táblában eltároljuk. Az egybeeső metszéspontokat töröljük.



6.3 ábra

Szerszámpálya az első algoritmus szerint



6.4 ábra

Áttérés új megmunkálási szintre

- f/ Megkeressük M táblában a szintek közötti haladási irány és a szinten való haladási irány szerinti első olyan metszéspontot, amelyet még nem dolgoztunk fel. Ha nincs ilyen, p/-nél folytatjuk az eljárást.
- g/-h/ Mint 6.2.-ben. A feldolgozott metszéspontokat az M táblában megjelöljük.
- i/ Megnézzük, hogy az utoljára érintett konturnak van-e metszéspontja a következő bejárási szinten. Ha nincs, n/-nél folytatjuk az eljárást.
- j/ Ha több metszéspontja van, kiválasztjuk a követési irány szerinti legközelebbit (6.4. ábra).
- k/ Ha ez a metszéspont már föl volt dolgozva, n/-nél folytatjuk az eljárást.
- l/ Konturt követve munkameneti pályát generálunk az új metszéspontig, de csak akkor, ha ez nem halad keresztül az előző szinten, mert akkor o/-nál folytatjuk az eljárást.
- m/ Megfordítjuk a szinten való haladási irányt és h/-nál folytatjuk az eljárást.
- n/ Megnézzük, hogy az adott megmunkálási szinten az aktuális haladási iránynak megfelelő metszéspont (ha van ilyen) ugyanazon a konturon van-e. Ha igen, megfordítjuk a szinten való és a szintek közötti haladási irányt és l/-nél folytatjuk az eljárást.
- o/ Megnézzük, hogy a pálya utolsó pontja a T téglalap melyik csucsához van legközelebb és annak megfelelően beállítjuk a szinten való és a szintek közötti haladási irányt. Az eljárást f/-nél folytatjuk.
- p/ Mint 6.2/m.

6.4. AZ ALGORITMUSOK ELEMZÉSE

6.4.1. Első algoritmus

- 1/ Minden kontur mentén végighalad a szerszámpálya, ezt m , biztosítja.
- 2/ Mivel a konturok zártak, az egybeeső metszéspontokat pedig töröltük, a megmunkálandó és a nem megmunkálandó szakaszok minden szinten váltakoznak.
- 3/ Egy megmunkálási szinten a kezdő szakaszt jól választjuk meg, mert ha az egyik legkülső ofszet kontur körüljárása negatív, akkor annak a külsejét kell megmunkálni, tehát E megmunkálandó tartományt határol.
- 4/ Minden megmunkálási szinten végighaladunk, tehát biztos, hogy nem marad ki megmunkálandó szakasz.
- 5/ A megmunkálási szintek és a konturokkal való metszéspontok száma véges, tehát az eljárás véget ér.

6.4.2. Javitott algoritmus

- 1/ A 6.4.1/1,2,3 és 5 erről algoritmusról is elmondható.
- 2/ Egy megmunkálási szinten a széléről indulva az elsőnek talált metszésponttól kezdődő szakasz megmunkálási szakasz lesz. Ha viszont kontur mentén térünk át a következő szintre (6.3/1), akkor a következő szinten az a metszéspont, ahová a pálya beérkezik, ugyanolyan paritású lesz, mint amiből kiindultunk, tehát a haladási irány megfordításával ismét megmunkálható szakaszt kezdünk. (ld. 6.4. ábra)

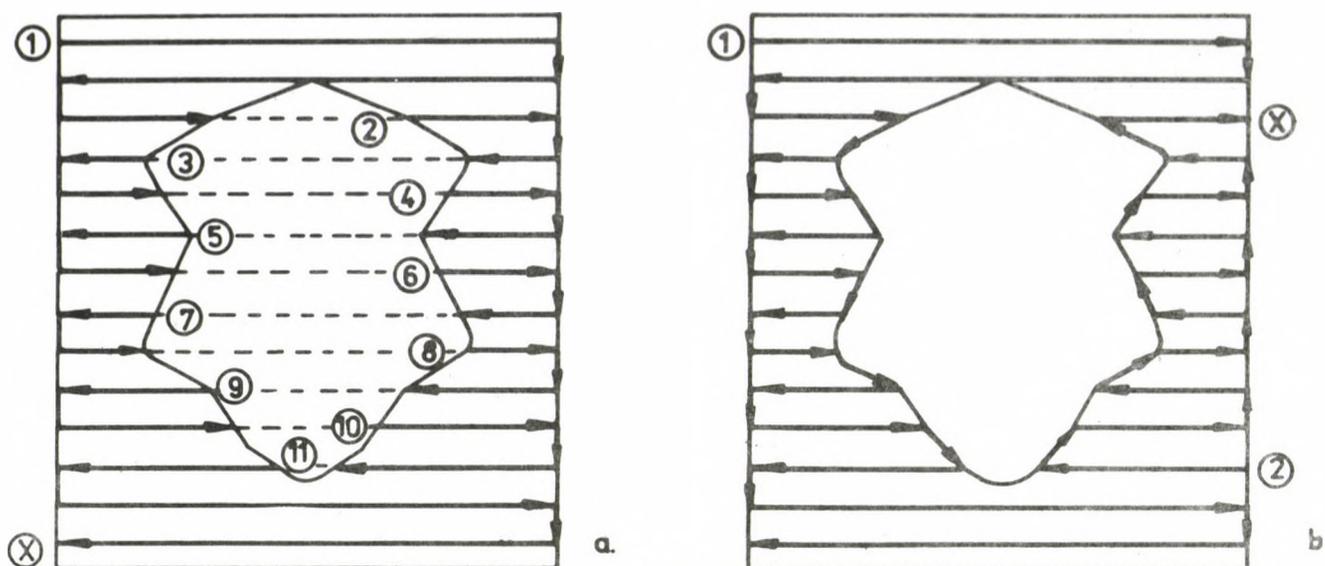
- 3/ Ha egy megmunkálási szinten kigenerálunk egy szakaszt, mindkét végpontját töröljük M táblából, tehát a metszéspontok száma mindig páros marad.
- 4/ A 6.3/f beli ciklikus vizsgálat biztosítja, hogy minden metszéspontot feldolgozunk.

6.5. AZ ALGORITMUSOK ÖSSZEHASONLITÁSA

A 6.5. ábrán látható a két algoritmus által generált mozgáspálya, 6.5/a-n az első, 6.5/b-n a második. Mindkettő teljesíti a 6.1-ben leirt feltételeket, tehát a kijelölt tartományok megmunkálására alkalmas. Az ábrákon az összefüggő munkameneti pályadarabok kezdőpontját sorszám jelzi, a megmunkálás utolsó pontját x . A gyorsmeneti szakaszokat szaggatott vonal jelöli. Látható, hogy a második algoritmussal generált pálya rövidebb megmunkálási időt biztosít, mivel a szerszámkiemelések száma és az üresjáratú mozgás lényegesen kevesebb. Ez általánosan is kimutatható.

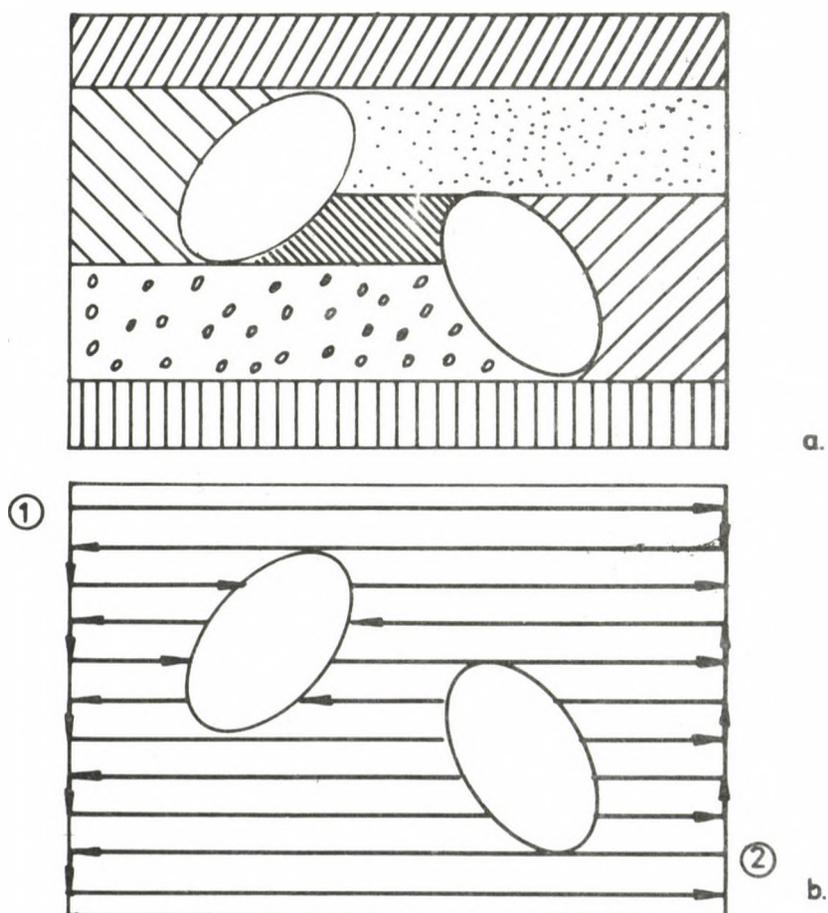
Tegyük fel először, hogy minden kontur konvex. Nem megszorítás, ha feltesszük, hogy a megmunkálási irány az x tengellyel párhuzamos. A 6.6/a ábrán különbözőképpen jelölt területeket vízszintes szakaszok és pontosan két kontur egy-egy monoton darabja határolja (E-t is beleértve). Az algoritmusból következik, hogy egy-egy ilyen területet mindig összefüggően, a szerszám kiemelése nélkül munkálunk meg. Ha az ofszet konturok száma n , akkor az ilyen területek száma $3:n+1$. Tehát a szerszámkiemelések száma a második algoritmusnál legfeljebb $3.n$.

Az első algoritmusnál a kiemelések száma attól is függ, hogy egyetlen konvex zsebet kell megmunkálni, vagy bonyolultabb, esetleg többszörösen összefüggő alakzatot. Egyetlen konvex zseb kiemelés nélkül végigjárható, csakugy mint a javított algoritmusnál. Egyéb esetekben azonban a kiemelések száma $\sum_{i=1}^n k_i$, ahol k_i megmutatja, hogy i konturt hány megmunkálási szint metszi. Ez tehát függ a szerszám átmérőjétől is, és azt jelenti, hogy ha a marósugár ugy aránylik a konturok méretéhez, hogy a megmunkálási szintek metszési száma egy konturral átlagosan legalább 3, akkor már biztos, hogy a második algoritmus kevesebb szerszámkiemelést eredményez. Az is igaz továbbá, hogy ha egy kontur kettőnél több, a második algoritmus szerint összefüggően megmunkálandó területet határol, akkor ha a kontur és a megmunkálási szintek metszési száma nem több mint egy, akkor ezek közül a területek közül lesznek olyanok, amelyiken nem halad megmunkálási szint, tehát nem is kell külön megmunkálni. Ha tekintetbe vesszük azt is, hogy a szomszédos területek a javított algoritmus esetén igen gyakran csatlakoztathatók, látható, hogy akkor sem ad rosszabb eredményt, ha a konturok és a megmunkálási szintek átlagos metszési száma háromnál kevesebb.



a.
6.5 ábra

Cikk-cakk megmunkálás kétféle algoritmussal



6.6 ábra

Területi felosztás és szerszám-pálya a javított algorit-mussal

7. IMPLEMENTÁCIÓ

7.1. IMPLEMENTÁCIÓ AZ FFS RENDSZERBEN

Mint a bevezetésben és a 2. fejezetben említettük, a dolgozatban leírt algoritmusok az FFS felülettervező-megmunkáló rendszer [23], [47] fejlesztése során készültek, és a rendszer nagyoló processzorának alapját képezik. Az implementációhoz szükséges programokat az értekezés szerzője írta, TPA 11/40 kisszámítógépre, GESAL programozási nyelven [26], RSX 11/M operációs rendszer alatt. Az MTA SZTAKI által fejlesztett GD80 grafikus display [1] szolgáltatja az interaktív ellenőrzési lehetőségeket, illetve hardcopy célra NE 2000 típusu Videoton plotter.

A program 140 K byte memóriát foglal el, kihasználva a RSX rendszer által nyújtott overlay lehetőséget. A nagyobb adatstruktúrák virtuális kezelését a SZTAKI-ban készült MOTOR virtuális memória kezelő rendszer oldja meg [11].

Az FFS rendszer, és így vele a nagyoló processzor is 1982 vége óta próbaüzem alatt áll. Azóta számos munkadarab készült el a segítségével, részben tesztelési célokra, részben ipari megrendelésekre. A 2. sz. melléklet egy műanyag flakon formáló szerszámjának FFS tervét és a nagyoló NC szalag készítése közbeni ellenőrzési fázisokat mutatja, plotter-rajz formájában. A 3. sz. melléklet fényképeket tartalmaz az FFS rendszer segítségével tervezett és megmunkált munkadarabokról, nagyolás illetve simítás után. A megmunkálás a BME-GTI-SZTAKI közös kísérleti üzemében készült,

UNIMERIC 723 illetve DIALOG 8060 CNC vezérlésű TC3 illetve POLYAX - VF1 marógépen.

7.2. NUMERIKUS PROBLÉMÁK

A próbaüzem során végzett munkák számos olyan problémát vetettek fel, amelynek megoldása a technológiai szolgáltatások bővítését igényelte. (ld. 1.sz. melléklet) Algoritmusaink szempontjából azt mondhatjuk, hogy helyességüket a felhasználási tapasztalatok teljes mértékben alátámasztják. Szembe kellett azonban nézni a számítógépes megvalósítás korlátaival, nevezetesen azzal, hogy a valós számok 32-bites számábrázolási pontossága a feladathoz képest igen alacsony. Ez főként a síkmetszés során jelent problémát, ahol a harmadfoku egyenlet megoldása nagyszámu algebrai művelet elvégzését igényli, lebegőpontos ábrázolási számokkal. Növeli a pontatlanságot az is, hogy a számítások nagy része a paraméter-tartományra vonatkozik, amely a $[0,1]$ intervallumban van, utána viszont át kell térni a térbeli koordinátákra, amelyek 2-3-4 nagyságrenddel nagyobb értékhatárok közé esnek.

A számítási pontatlanságból eredő hibák elkerülésére a következő lépéseket tettük:

- 1/ Az elvégzendő műveletek sorrendjét úgy határoztuk meg, hogy a fellépő hibák összegződése a lehető legkisebb legyen.

- 2/ Minden vizsgálatot, amely lebegőpontos ábrázolásu számokra vonatkozik, valamely, az illető számok nagyságrendjétől függő hibakorlát figyelembe vételével végeztünk.
- 3/ A harmadfoku egyenlet megoldásait helyettesítéssel ellenőriztük, és nem megfelelő pontosság esetén a gyököt iterációs eljárással korrigáltuk.

Ezekkel a módszerekkel sikerült kiküszöbölni a pontatlanságból eredő hibákat és a program működése megbízhatóvá vált.

7.3. IMPLEMENTÁCIÓ A BUILD-4 GEOMETRIAI MODELLEZŐ RENDSZERBEN

A BUILD [10], [32] az egyik legelső testmodellező rendszer volt, és jelenlegi 4. verziója a világ egyik legismertebb és koncepcionálisan legfejlettebb geometriai modellező rendszere. Első változatát 1970-ben kezdte fejleszteni a Cambridge-i Egyetem kutató csoportja. A BUILD alapvetően az 1.2.2. fejezetben leirt, térfogati alapelemekből halmazelméleti műveletekkel "építkező" (build=építeni) rendszer, amelyet azonban a folyamatos továbbfejlesztés során igen alaposan kibővítettek. A jelentősebb eredmények közé tartozik a tőrésezett aritmetika, az alkatrész-osztályozás, a véges elem háló generálás, a lokális operátorok és a szabad formájú felületek használatának lehetősége.

Az új lehetőségek úgy megváltoztatták a BUILD szerkezetét, hogy az eredeti verzióhoz készült megmunkáló rendszer [29] használhatatlanná vált. Új megmunkálási lehetőség az "AUTOMAC" program ([36]) a hozzá csatlakozó 2.5D-s NC-processzorral, amely nem más mint a jelen dolgozat 4.5.6. fejezetének megvalósítása a BUILD keretei között. Az implementációt az említett angol-magyar együttműködési szerződés keretében az értekezés szerzője végezte, IBM/370 számítógépen, ALGOL68C programozási nyelven, CMS operációs rendszer alatt.

A 4. sz. mellékleten látható egy alkatrész "GEHAUSE", amely a geometriai modellező rendszerek egyik alapvető teszt-objektuma, és a hozzá generált szerszámpanyák.

ÖSSZEFOGLALÁS

Az NC technika és a hozzá kapcsolódó számítógépes rendszerek megváltoztatták a gépipari tervezés és gyártás arcukat. Világszerte nagy erőfeszítéseket tesznek komplett CAD/CAM rendszerek kifejlesztésére és ipari alkalmazására, és ebbe a folyamatba már hazánk is bekapcsolódott.

Jelen dolgozat bemutatta a gépipari CAD/CAM rendszerek fejlődésének főbb lépcsőit különös tekintettel a geometriai modellező rendszerekre. Az MTA SZTAKI-ban kifejlesztett FFS CAD/CAM rendszerhez kapcsolódva eljárást adott szabad formájú felületek 2.5 NC megmunkálásához szükséges szerszám-pálya előállítására. Az algoritmusok használhatósági feltételeinek és korrektségének részletes elemzése megtörtént. A mellékletek dokumentálják a leírt algoritmusok implementálását, egyrészt az FFS rendszerben, másrészt az angol BUILD-4 testmodellező rendszerben.

FÜGGELÉK

A síkgörbék irányításával kapcsolatos geometriai fogalmak tisztázása.

1. Definíció

Legyen G olyan görbe az (x,y) síkon, amely

- irányított,
- folytonos,
- zárt,
- nem önmetsző,
- egyenes szakaszokból és 180° -nál nem nagyobb középponti szögű körivekből áll.

Egy $P \in G$ síkbeli pont a G görbétől balra van, ha a következő feltétel teljesül minden Q pontban, amely $Q \in G$ és $Q \rightarrow P = P \rightarrow G$:

- Legyen \underline{e} \overline{QP} -re merőleges egységvektor Q -ban, melynek iránya a G görbe irányítását követi (F.l. ábra) akkor

$$z(\underline{e} \times \overline{QP}) > 0.$$

azaz a vektorszorzat pozitív z irányba mutat.

Megjegyzés

\underline{e} vektor a G görbe Q -beli érintő egységvektora, ha G másodrendben folytonos Q -ban. Ha Q -ban nincs érintő, a G görbét Q kis környezetében \overline{QP} -re merőlegesen elmeteszve a kapott metszéspontok irányítás szerinti sorrendje jelöli ki \underline{e} irányát.

2. Definíció

A sík azon pontjai, amelyekre a fenti vektorszorzat negatív, a G görbétől jobbra vannak.

3. Állítás

A sík bármely $P \notin G$ pontjáról egyértelműen eldönthető, hogy a G görbétől jobbra vagy balra van.

4. Bizonyítás

Minden $P \notin G$ -hez létezik legalább egy Q pont, amelyre $P \rightarrow Q = P \rightarrow G$ és $Q \in G$. Q -hoz egyértelműen definiált az \underline{e} vektor iránya, amely a döntés alapjául szolgál. Belátjuk, hogy több, egyenlően közel levő pont esetén sem jutunk ellentmondásra.

Tegyük fel, hogy Q_1 és Q_2 olyan pontjai G -nek, melyek P -hez legközelebb vannak, azaz $Q_1, Q_2 \in G$, $Q_1 \rightarrow P = Q_2 \rightarrow P = P \rightarrow G$. Legyen $\underline{e}_1 \overline{Q_1 P}$ és $\underline{e}_2 \overline{Q_2 P}$ a görbe irányát követő merőleges egységvektorok Q_1 -ben illetve Q_2 -ben.

Indirekt módon tegyük fel, hogy

$$\text{sign } z(\underline{e}_1 \times \overline{Q_1 P}) \neq \text{sign } z(\underline{e}_2 \times \overline{Q_2 P}).$$

Tekintsük a $Q_1 P Q_2$ szöveget, \underline{e}_1 és \underline{e}_2 ennek száraira merőleges. Ha \underline{e}_1 és \underline{e}_2 különböző szögtartományba mutat (F.2. ábra) akkor P körüli forgatással egymásba vihetők, tehát a vektorszorzat nem különbözhet.

Tegyük fel, hogy különböző szögtartományba mutatnak. A P körüli $r = \overline{Q_1 P} = \overline{Q_2 P}$ sugaru nyílt körlemez nem tartalmazhat G -beli pontot, hiszen akkor nem lenne igaz, hogy $P \rightarrow G = r$.

Belátjuk, hogy nincs olyan irányított, zárt, nem önmetsző görbe a síkon, amely Q_1 -en és Q_2 -n az \underline{e}_1 illetve \underline{e}_2 irányításának megfelelően áthalad és nem metszi a P körüli r sugarú kört.

Kössük össze Q_1 -et Q_2 -vel egy tetszőleges, nyílt q görbével, amely nem önmetsző és a P körüli r sugarú körbe nem metsz. Zárjuk le ezt a görbét a kör azon ívével, amely az \underline{e}_1 és \underline{e}_2 irányával kijelölt szögtartományba esik. (F.3. ábra), és nevezzük ezt a zárt görbét q' -nak. Ha a q görbét az irányításnak megfelelően akarjuk folytatni, Q_1 -ben a q' görbe külsejében kell haladni, Q_1 -be viszont q' belsejéből kell beérkeznie, \underline{e}_1 és \underline{e}_2 irányva szerint. Ekkor viszont metszenie kell q' -ot, azaz q görbe vagy önmetsző lesz, vagy a P körüli r sugarú körbe metsz bele, tehát a feltételeknek megfelelően nem zárható be.

4. Állítás

- a/ A G görbétől jobbra illetve balra levő pontok osztálya megegyezik a G görbe által meghatározott külső és belső siktartományokkal.
- b/ Pozitív irányítás esetén a görbén kívül levő pontok vannak a jobb oldalon, negatív irányításnál pedig a görbén belül levők.

Bizonyítás

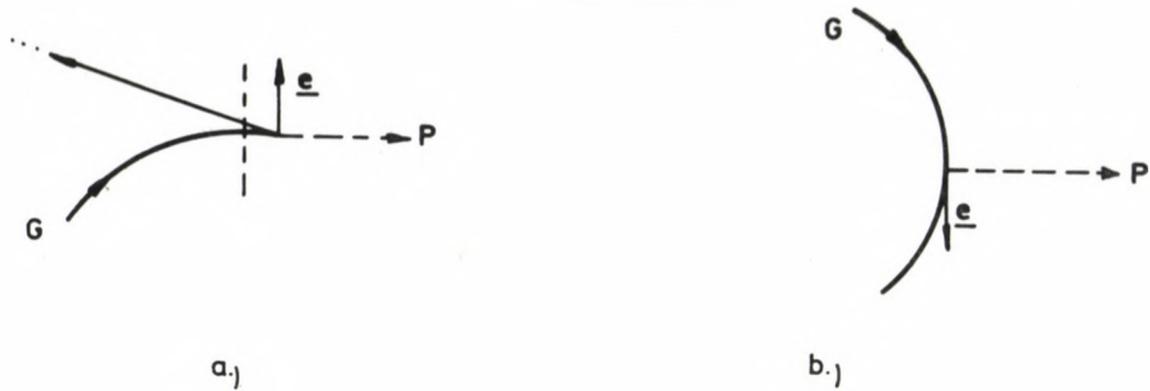
- a/ Tegyük fel, hogy az állítás nem igaz. Ekkor van az egyik siktartományban olyan P_1 és P_2 pont, hogy P_1 a görbéhez képest jobbra, P_2 pedig balra van. Mivel azonos siktartományban vannak, összeköthetők olyan nem önmetsző siktartományban (nevezzük H -nak) amely nem metszi G -t. Csatol-

juk a H görbe két végéhez az $\overline{X_1P}$ és $\overline{X_2P}$ szakaszokat, amelyekre igaz, hogy $X_1, X_2 \in G$ és $P_1 \rightarrow G = P_1 \rightarrow X_1$, $P_2 \rightarrow G = P_2 \rightarrow X_2$.

Vegyük fel X_1 -ben és X_2 -ben a G görbe irányát követő \underline{e}_1 és \underline{e}_2 merőleges egységvektorokat. $\underline{e}_1 \perp \overline{X_1P_1}$, $\underline{e}_2 \perp \overline{X_2P_2}$ (F.4. ábra).

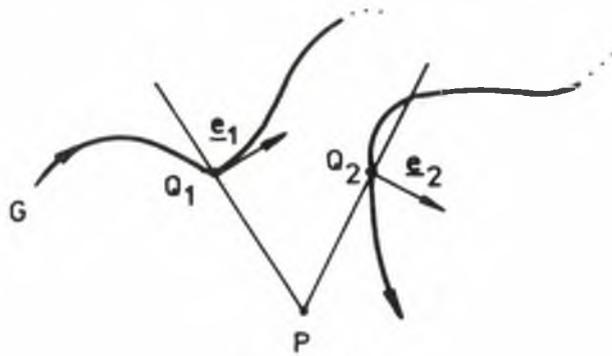
Ha $\text{sign } z(\underline{e}_1 \times \overline{Q_1P_1}) \neq \text{sign } z(\underline{e}_2 \times \overline{Q_2P_2})$ az indirekt feltevésnek megfelelően, akkor léteznie kell olyan zárt, irányított, nem önmetsző görbének, amely X_1 -en és X_2 -n az \underline{e}_1 illetve \underline{e}_2 irányításának megfelelően áthalad és nem metsz H-t. Ez a feladat topológiailag ekvivalens a 3. állítás bizonyításában leírt feladattal (F.3. ábra) tehát itt is ellentmondásra jutottunk.

b/ Az állítás második fele a körüljárás definíciójából következik. (ld [30])



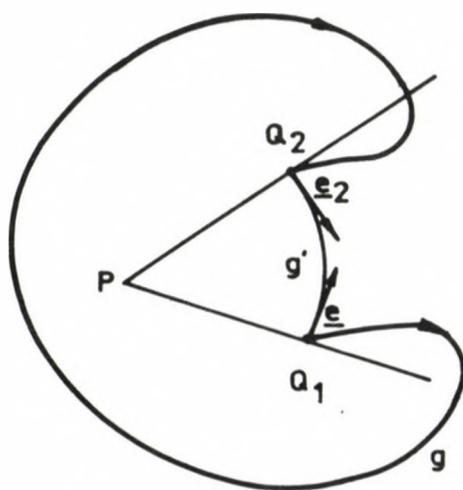
F.1 ábra

Görbe irányítását követő vektor meghatározása



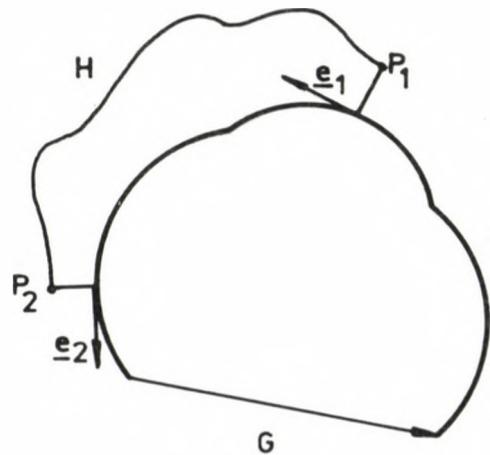
F.2 ábra

Több minimális távolságra lévő pont



F.3 ábra

Görbe jobb és bal oldalának
egyértelműsége



F.4 ábra

A sík pontjainak
osztályozása



IRODALOM

- [1] All About GD-80 MTA SZTAKI 1979.
- [2] Anderson, C.: The new BUILD User's Guide, CAD Group Document 116, Cambridge University Engineering Department, (1983)
- [3] Barnhill, R.E., Boehm, W.: Surfaces in Computer Aided Geometric Design. Proc. of a conference at Oberwolfach, 1982.
- [4] Baugham, J.A.: Többkoordinátás megmunkálás APT segítségével. NC alkalmazási kézikönyv, Műszaki Könyvkiadó, Budapest, 1973.
- [5] Benutzerhandbuch PROREN2
Firmenschrift der ISYKON software GmbH Bochum, 1980
- [6] Bezier, P.: Mathematical and Practical Possibilities of UNISURF, in Computer Aided Geometric Design, (Barnhill and Riesenfeld eds.), Academic Press, (1974)
- [7] Bishop, A.W.: ROMULUS2 and its Role in CAE, in: Proc. MICAD 84, Paris, (1984)
- [8] Boyse, J.W. and Gilchrist, J.E.: GMSOLID: Interactive Modelling for Design and Analysis of Solids, IEEE Computer Graphics and Applications, March (1982)
- [9] De Boor, c.: On Calculating with B-splines Journal of Approximation Theory 1972/6.
- [10] Braid, I.C., Lang, C.A.: Computer-Aided Design of mechanical components with volume building bricks. Proc. PROLAMAT'73 (Budapest 1973)

- [11] Bródy, F.: MOTOR - A Mini-Micro Blow-Up
MTA SZTAKI, Budapest 1979
- [12] Coons, S.A.: Surfaces for Computer Aided Geometric
Design of Space Figures M.I.T. ESL 9442-M-139,
January, 1964.
- [13] Darvid, B.: 'EUCLID' and its application at Computer-
Aided Design in machinery topic. Proc. CAD in
Mechanical Engineering, Politecnico di Milano (1976)
- [14] Davies, K.J.: GNC-A graphical NC processor. Proc.
PROLAMAT'73 (Budapest 1973)
- [15] Encarnacao, J., Shlechtendahl, E.G.: Computer Aided
Design. Fundamentals and system Architectures.
Berlin, Springer Verlag, 1983.
- [16] Engeli, M.: Geometric Modelling of Complex Forms with
EUKLID, in: Proc. 12th CIRP International Seminar on
Manufacturing Systems, Belgrade, (1980)
- [17] Faux, I.D. and Pratt, M.J.: Computational Geometry
for Design and Manufacture, Ellis Horwood, Chichester,
U.K., (1980)
- [18] FFS CAD/CAM - Rendszerprogramozói kézikönyv
MTA SZTAKI GSZTO, 1984
- [19] FFS CAD/CAM - Felhasználói kézikönyv
MTA SZTAKI GSZTO, 1984
- [20] Forrest, A.R.: Curves and Surfaces for Computer Aided
Geometric Design, PhD Thesis, University of Cambridge,
(1968)
- [21] Flutter, A.: The POLYSURF system
Proc. PROLAMAT'73 (Budapest 1973)
- [22] Gaál B., Hermann Gy., Horváth L., Renner G., Várady
T.: Szoborszerű felületek tervezése és megmunkálása
MTA SZTAKI tanulmány 1978.

- [23] Gaál, B., Kacsukné, B.L., Várady, T.: The FFS (Free-form shapes) system in Practice. Papers of a Joint Anglo-Hungarian Seminar on CAGD.
- [24] Gaál, B. and Várady T.: Experiences and further development of the FFS Free-Form Shapes system, Proc. of the 9th IFAC World Congress, Budapest, (1984).
- [25] Geisow, A.D.: Results of Benchmark Tests Using MEDUSA in: Proc. 2nd CAM-I Geometric Modelling Seminar, Cambridge, Report No. P-83-GM-01, CAM-I, (1983)
- [26] GESAL - Language Reference Manual
MTA SZTAKI 1980
- [27] Golssling, T.H.: The 'DUCT' system for practical objects.
Proc. CAD in Mechanical Engineering, Politecnico di Milano (1976)
- [28] Gordon, W.J. and Riesenfeld, R.F.: B-spline curves and surfaces, in: Computer Aided Geometric Design, (Barnhill and Riesenfeld eds.), Academic Press, (1974)
- [29] Grayer, A.L.: A Computer Link Between Design and Manufacture
PhD. Thesis, Cambridge, University, 1976
- [30] Hajós Gy.: Bevezetés a geometriába
Tankönyvkiadó, Budapest 1960.
- [31] Holló, K., Várady T.: Számítógépes alkatrészmodellezés
MTA SZTAKI belső tanulmány, 1978.

- [32] Jared, G.E.M. and Várady T.: Synthesis of volume modelling and sculptured surfaces in BUILD, in: Proc. CAD 84. Conf., Brighton, (1984)
- [33] Kacsukné, B.L.: Geometric Algorithms for 2.5D Roughing Process of Sculptured Surfaces. Papers of a Joint Anglo-Hungarian Seminar on CADG.
- [34] Leslie, W.H.P.: NC alkalmazási kézikönyv. Műszaki Könyvkiadó, Budapest, 1973.
- [35] Okino, N., et al: TIPS-1 Technical Information Processing System for Computer Aided Design, Drawing and Manufacturing, in: Proc. Prolamat 73, Budapest (1973)
- [36] Parkinson, A.: An Automatic NC Data Generation Facility for the BUILD Solid Modelling System Proc. of 16th CIRP International Seminar on Manufacturing Systems
- [37] Person, H.: NC machining of arbitrarily shaped pockets Computer Aided Design 1978/3
- [38] Piegl, L.: A curve fitting method for rough cutting CAD Journal, to be published
- [39] Renner, G.: A Method of Shape Description for Mechanical Engineering Practice Computers in Industry 1982/3 North-Holland Publishing Company (1982)
- [40] Renner, G.: Conventional elements of engineering drawing - free-form curves, Proc. Eurographics'84, Copenhagen, North Holland, (1984)
- [41] Reckziegel, D.: EXAPT1, EXAPT2 NC alkalmazási kézikönyv Műszaki Könyvkiadó, Budapest 1973.

- [42] Riesenfeld, R.F.: Nonuniform B-spline Curves Proc 2nd USA-Japan Computer Conference 1975
- [43] Sabin, M.A.: The Use of Piecewise Forms for the Numerical Representation of Shapes, Dissertation Hungarian Academy of Sciences, 1977.
- [44] Sabin, M.A.: Computer-Aided part program generation Proc PROLAMAT, 1973
- [45] Sabin, M.A.: Contouring - A review of methods for scattered data, Mathematical Methods in Computer Graphics and Design, Academic Press, (1980)
- [46] Spur, G.: Rechnerunterstützte Zeichnungserstellung und Arbeitsplanung, Case Hanser Verlag München Wien 1980.
- [47] Szőkefalvi, N.Gy., Gehér L., Nagy P.: Differenciálgeometria, Műszaki Könyvkiadó, Budapest 1979.
- [48] Várady T.: Some practical aspects of an experimental system for design and manufacture of sculptured surfaces, Computers in Industry, Vol. 3, (Coons' Memorial Issue), (1982)
- [49] Várady T.: Synthesis of volume modelling and sculptured surfaces in BUILD-2, In: Proc. of Joint Anglo-Hungarian Seminar on CAGD, Budapest, (1982)

1.SZ. MELLÉKLET

Az FFS CAD/CAM rendszer nagyoló procesz-
szorának felhasználói kézikönyve.

10. NAGYOLAS

10.1. A RENDSZER CELJA

AZ INTERAKTIV FELULETTERVEZO RENDSZER SEGITSEGEVEL TERVEZETT SZOBORSZERU FELULETEK MEGMUNKALASA NC VEZERLESU MAROGEPEKEN TORTENIK. A FELULET NAGYOLASAT POZITIV Z IRANYBOL LEFELE, RETEGENKENT TERASZOLVA VEGEZZUK, HENGERES MAROVAL. A RENDSZER A RETEGENKENTI NAGYOLO MARAS NC VEZERLOSZALAGJAT ALLITJA ELO A FELULET BELSO TPW REPREZENTACIOJABOL, AMELY A FELULETTERVEZO RENDSZER OUTPUTJA, TOVABBA A TECHNOLOGIAI ADATOKBOL, AMELYEK EGY EGYSZERU TECHNOLOGIAI FILE-BAN, ILLETVE A DIALOGUSRENDSZEREN KERESZTUL ADHATOK MEG. A RENDSZER MUKODHET AUTOMATIKUSAN, DE TARTALMAZHAT INTERAKTIV GRAFIKUS ELLENORZESI ES BEAVATKOZASI LEHETOSEGEKET. EZ A PROGRAM INDITASAKOR BEALLITHATO OPCIO.

10.1.1. A PROGRAM ALGORITMUSA

A SZERSZAMPALYA ELOALLITASA MINDEN Z SZINTEN NEGY FAZISBAN TORTENIK:

A/ SIKMETSZES:

A FELULET ES AZ AKTUALIS Z SIK METSZETENEK KISZAMITASA. ENNEK EREDMENYE EGY, VAGY TOBB ZART SIKGORBE LEHET, KIS EGYENES SZAKASZOKKAL KOZELITVE.

B/ OFSZETELES:

A SIKMETSZETI KONTUROK MODOSITASA A SZERSZAMSUGAR+SIMITASI RAHAGYAS ERTEKKEL A MEGMUNKALASNAK MEGFELELOEN KIFELE VAGY BEFELE.

C/ PASSZITAS:

HA A SIKMETSZET EGYNEL TOBB KONTURBOL ALL, AKKOR ELOFORDULHAT, HOGY AZ OFSZET KONTUROK EGYMASBA METSZENEK.

EBBEN A FAZISBAN AZ ILYEN KONTUROK OSSZEVONASA TORTENIK MEG.

D/ PalyAGENERALAS:

CIKK-CAKK PALYA ELOALLITASA X VAGY Y TENGELLYEL PARHUZAMOSAN, A KEZDOPONT A NEGY SAROKPONT KOZUL KIVALASZTHATO.

A PROGRAM MUKODESENEK LENYEGE, HOGY A MEGADOTT MAXIMALIS Z ERTEKTOL KEZDVE A MINIMALIS Z ERTEKIG FOGASMELYSEGENKENT HALADVA A FENTI NEGY FAZIS CIKLIKUSAN ISMETLODIK. INTERAKTIV HASZNALAT ESETEN LEHETOSEG VAN EGY-EGY FAZIS MEGISMETLESERE, VAGY PEDIG EGY RESZBEN VEGREHAJTOTT CIKLUSBOL VALO KILEPESRE.

10.2.A RENDSZER HASZNALATA

A PROGRAM FUTASA SORAN MUNKAMEZOKENT EGY "MOTOR.WRK" NEVU MUNKAFILE-T NYIT MEG AZ SY: LEMEZEN. EHHEZ EGY 210 (DEC) BLOKKOS (NEM FELTETLENUL OSSZEFUGGO) TERULETNEK KELL A LEMEZEN LENNIE. AZ NC SZALAG GENERALASANAK EREDMENYET, -AZ NC PROGRAMOT - IS A LEMEZEN KELL MEGORIZNUNK, EZERT CELSZERU MEGFELELO NAGYSAGU URES TERULETROL GONDOSKODNI.

TERMESZETESEN A FUTTATAS BEFEJEZESE UTAN A "MOTOR.WRK" -T A RENDSZER LETORLI,

KIVEVE HA A PROGRAM FUTASA NEM TERMESZETES MODON TERMINALODIK (PL: ABORT PARANCSRA). ILYENKOR A FILE-T A

>PIP MOTOR.WRK;*/SD

PARANCCSAL KELL ELTAVOLITANI.

A PROGRAM FUTTATASAHOZ BIZONYOS GEOMETRIAI ES TECHNOLOGIAI PARAMETEREKRE VAN SZUKSEG. EZEK BEALLITASA EGY IGEN EGYSZERU TECHNOLOGIAI FILE SEGITSEGEVEL (LD.10.2.2.) ILLETHEZ AZ SZIAL EGYSZERU DIALOGUS RENDSZEREN KERESZTUL TORTENIK.

A PROGRAM ELEJEN MEGJELEND NEGY DIALOGUS AZ OSSZES PARAMETERT TARTALMAZZA, NEGY CSOPORTBAN (LSD 10.2.3. 1-4.SZAMU DIALOGUSOK).

1. GEOMETRIAI PARAMETEREK
2. TECHNOLOGIAI PARAMETEREK
3. OUTPUT PARAMETEREK
4. INTERAKTIVITASI PARAMETEREK

A NEGY DIALOGUS KULON-KULON JELENIK MEG A KEPERNYON. A PARAMETEREK BEALLITASA UTAN "RETURN" LEUTESEVEL ATTERHETUNK A KOVETKEZO DIALOGUSRA, VAGY "BACK SPACE" LEUTESEVEL VISSZATERHETUNK A MEGELOZORE.

HA AZ INTERAKTIVITASI PARAMETEREKET NULLARA ALLITJUK, A PROGRAM AUTOMATIKUSAN FUT LE ES ELOALLITJA AZ NC PROGRAMOT EGY VAGY TOBB

DISC FILE-BAN, HA A FELHASZNALO KERT INTERAKTIVITAST, A PARAMETEREK MINDEN FAZIS UTAN MODOSITHATOK. TERMESZETESEN VIGYAZNI KELL, HOGY AZ UJ PARAMETER ERTEKEK NE KERULJENEK ELLENTMONDASBA AZ ELOZOEKKEL. (PL. SZERSZAMSUGAR MEGVALTOZTATASA OFSZETELES UTAN).

A KERT FAZIS UTAN A SIKBELI KONTUROK GRAFIKUS DISPLAY-N MEGJELENITODNEK, AZ ALFANUMERIKUS KEPERNYON PEDIG AZ 1. DIALOGUS-SOROZAT JELENIK MEG. A PARAMETEREK VALTOZTATASA UTAN "V" KILEPESEL A VIZSGALT FAZIS ISMETHETO, "E" KILEPESEL PEDIG AZ AKTUALIS SZINT ELHAGYHATO. EZ ESETBEN A KOVETKEZO Z SZINT METSZESEVEL FOLYTATODIK A PROGRAM. "M" KILEPESEL A PROGRAM RATER A KOVETKEZO FAZIS VEGREHAJTASARA.

HIBAJELZES UTAN "RETURN LEUTESSEL A PROGRAM A DIALOGUS RENDSZERBE TER VISSZA AKKOR IS, HA AZ INTERAKTIVITAST ELAG NEM VOLT BEALLITVA. EBBEN AZ ESETBEN A MEGADOTT DISC FILE LEZARASAROL A PROGRAM GONDOSKODIK.

10.3.1. GEOMETRIAI MEGKÖTÉSEK A FELULETRE

AZ ELJÁRÁSSAL OLYAN SZOBORSZERŰ FELÜLETEK MUNKALHATOK MEG, AMELYEK:

A, $x=x(u,v)$ $y=y(u,v)$ $z=z(u,v)$
 PARAMETERES ALAKBAN LEÍRHATÓK, ES ELOALLITOTTUK AZ FFS SEGÍTSÉGÉVEL
 A TPW REPRÉZENTÁCIÓJUKAT,

B, ONMAGUKBA NEM METSZENEK, AZAZ HA

$$x(u_1, v_1) = x(u_2, v_2) \text{ ES}$$

$$y(u_1, v_1) = y(u_2, v_2) \text{ ES}$$

$$z(u_1, v_1) = z(u_2, v_2)$$

KÖVETKEZIK HOGY $u_1 = u_2$ ES $v_1 = v_2$

C, $x=y$ SIKRA NEZVE $+z$ IRÁNYBÓL ALAMETSZEST NEM TARTALMAZNAK, AZAZ HA

$$x(u_1, v_1) = x(u_2, v_2) \text{ ES}$$

$$y(u_1, v_1) = y(u_2, v_2)$$

KÖVETKEZIK HOGY $z(u_1, v_1) = z(u_2, v_2)$

MEGJEGYZÉS:

AZ ALAMETSZESI KRITÉRIUM A FELÜLRŐL LEFELE TÖRTENŐ MEGMUNKALÁS MIATT NYILVANVALÓAN SZÜKSÉGES, HA AZ ALAMETSZES OLYAN KISMÉRTEKŰ, HOGY GYAKORLATILAG NEM OKOZNA HIBÁT, A NAGYÓLO PROCESSZORRAN AKKOR IS ELOFORDULHAT HOGY TELJESEN RÜSSZ PÁLYA GENERALÓDIK, (ELLENTETES ÖLDALRA KERÜL AZ OFFSZET). FÜGGŐLEGES ÖLDALAK LEHETNEK A FELÜLETEN, AM MINDEN TERASZ SZINTJEN KELL LENNI OLYAN SZAKASZNAK, AHOL NEM FÜGGŐLEGES A FELÜLET.

10.2.2 A BEMENŐ ADATOK FORMATUMA,

 KAPCSOLÓDÁS A FELÜLETTERVEZŐ RENDSZERHEZ

A PROGRAM KÖZVETLENUL CSAK A DIALOGUSOKON ILLETVE A
 TECHNOLOGIAI FILE-ON KERSZTUL BEALLITOTT GEOMETRIAI

ADATOKKAL HASZNÁLJA; A FELÜLET MEGADÁSA ITT CSUPÁN EGY HAT KARAKTERES AZO-
 ELOALLITOTT ÉS KÖNYVTARAZOTT FELÜLETRE VONATKOZIK.

A PROGRAM ELINDÍTÁSA

 A PROGRAM ELINDÍTÁSA AZ RSX RENDSZERNEK MEGFELELŐEN

"RUN NAGYULÓ"

PARANCCSAL TÖRTÉNIK.
 EZUTÁN A PROGRAM A FELÜLETET TARTALMAZÓ ".TPW" TÍPUSÚ FILE NEVET KERI.
 A TÍPUS MEGJELELÉSE ITT NEM SZÜKSÉGES.

"FELÜLET:"

A FELÜLET MEGADÁSA UTÁN A TECHNOLOGIAI FILE ADHATÓ MEG.

"TECHNOLOGIAI FILE (I/N)"

"N" VALÁSZ ESETÉN A PROGRAM BELEP A DIALOGUS RENDSZERBE.
 "I" VALÁSZ ESETÉN A TECHNOLOGIAI FILE NEVET KERT A PROGRAM, ENNEK TÍPUSA
 ".TEC" ÉS A TÍPUST NEM KELL MEGADNI.

"INPUT:"

HA A TECHNOLOGIAI FILE SZINTAKTIKAI HIBÁT TARTALMAZ, HIBAJELEZÉS VAN (10.3.)
 UTÁN BELEP A DIALOGUS RENDSZERBE.
 HELYES BEOLVASÁS UTÁN A PROGRAM A DIALOGUS RENDSZERRE TER ÁT.

A DIALOGUSRENDSZERRŐL VALÓ KILÉPÉS "M" KARAKTERREL TÖRTÉNIK, HA A 3. DIALOGUS
 DF VÁLTOZÓJA 1 VOLT, A PROGRAM AZ NC OUTPUT FILE NEVET KERI.

"OUTPUT:"

AZ OUTPUT FILE NEVE AZ RSX KONVENCIOKNAK MEGFELELŐ ALFANUMERIKUS KARAKTER-
 SOROZAT LEHET, AMELYHEZ A NAGYULÓ PROGRAM EGY KÉTJEGYŰ SZÁMOT ÉS ".NNC"
 TÍPUST ILLESZ. A SZÁM ARRÁ SZOLGAL, HOGY HA A MEGADOTT MAXIMALIS MONDATSZÁM-
 NAL HOSSZABB AZ NC PROGRAM, AKKOR A NEVBEN EZT A SZÁMOT MEGNOVELVE ÚJ
 OUTPUT FILE NYÍLIK MEG. AZ ÚJ OUTPUT FILE NEVE A KEPERNYŐRE MINDIG KIÍRODIK.

A PROGRAM BEFEJÉZÉSE

 1/ A TELJES NC SZALAG GENERALÁSA UTÁN A PROGRAM A KEZDŐ ÁLLAPOTBA TER VISSZA
 ÉS ÚJABB FELÜLETET KER. KILÉPÉS CTRL Z LEUTÉSEVEL TÖRTÉNHET.

A TECHNOLOGIAI FILE

A TECHNOLOGIAI FILE ARRÁ SZOLGAL, HOGY MEGKÖNYVITSE A RENDSZER INTERAKTÍV ELLENŐRZÉS NÉLKÜLI, AUTOMATIKUS HASZNÁLATÁT, ILLETVE ARRÁ, HOGY EGY FELLELI ISMÉTELT FELDOLGOZÁSA ESETÉN CSOKKENTSE A DIALOGUSON KERESZTÜL BEJELT ADATOK MENNYISÉGÉT. EZENKÍVÜL TARTALMAZ EGY OLYAN LEHETŐSÉGET, AMELY A DIALOGUS RENDSZEREN KERESZTÜL NEM ELÉRHETŐ; ELOFURÁSI HELYEKET LEHET DEFTNÁLNI AZ ANYAGBAN.

A TECHNOLOGIAI FILE HARMADAT-BLOKKOT TARTALMAZ, AMELYEK KÖZÜL

AZ ELSŐ KÖTELEZŐ, A MASIK KÉTTO OPCIONÁLIS.

1/ AZ ELOGYARIMÁNY MÉRTEI AZ 1. DIALOGUSNAK MEGFELEŐ SORRENDEN:

X MIN, X MAX, Y MIN, Y MAX, Z MIN, Z MAX;

2/ TECHNOLOGIAI ADATOK VÁLTOZÁSA A MEGMUNKÁLÁS SZINTJENEK FÜGGVÉNYFÉRN; Z ALSÓ HATAK, FOGASMÉLYSÉG, SZERSZÁMSUGAR, SZERSZÁMHÉLY, FORDULATSZÁM, ELŐTOLÁS;

JELENTÉSE: A MEGADOTT Z SZINTIG LEFELE A HOZZARENDELTE ADATOK

ALLITÓDNÁK BE AZ AUTOMATIKUS MEGMUNKÁLÁS SORÁN.

A TECHNOLOGIAI RÉTEGEK MAXIMÁLIS SZÁMA=6

3/ ELOFURÁSI HELYEK :

A LEGMELYEBB PONT X,Y,Z, KOORDINATAI, SZERSZÁMHÉLY, FORDULATSZÁM, ELŐTOLÁS;

HATÁSA: AZ NC PROGRAM FLEJÉN AZ ADOTT HELYEKRE A HOZZARENDELTE TECHNOLOGIAI ADATOKKAL FURÓCIKLUS GENERALÓDIK, MAJD AZ ÉGYES SZINTEK MEGMUNKÁLÁSA MINDIG AZ ELOFURTE HELYEKEN FOG KEZDŐDNI, HA VAN ILYEN AZ AKTUALIS SZINTEN. HA A SZERSZÁMHÉLY ÉRTEKET NEGATIVAN DEFINIÁLJUK, A FURÓCIKLUSOK NEM GENERALÓDNÁK AZ NC PROGRAMBA, DE A MEGMUNKÁLÁS KEZDESE OLYAN LESZ, MINTHA ELOFURÁS TORTENT VOLNA. HA A SZERSZÁMHÉLY ÉRTEKE 0, AKKOR NINCS SZERSZÁMCSERE, DE A FURÓCIKLUSOK KIGENERALÓDNÁK.

ELOFURÁSOK MAXIMÁLIS SZÁMA = 6

MINDHARMAD BLOKKBAN REAL SZÁMOK SZEREPELNEK, AMELYEKET VESSŐ TERMINÁL,

KIVEVE AZ UTOLSÓ SZÁMOT AMELYET MINDHARMAD ESETREN PONTOSVESSŐ TERMINÁL.

A 2. BLOKKOT 'M' KARAKTER VÉZETI BE, A HARMADIKAT PEDIG 'F'.

A MÉRTEFILET "\$" ZÁRJA LE, SPACE ES SÖREMELES BARRHOL LEHET (KIVEVE TERMESZETÉSEN SZÁM KÖZEPEN)

10.2.6. A DIALOGUSOK

1.SZ. DIALOGUS

** SZUBORSZERU FELULET NAGYOLASA **

** GEOMETRIA **

AZ ELOGYAKTMANY MERETEI:	X MIN [MM]	XH=	-58.888
	Y MIN [MM]	YL=	-50.000
	Y MAX [MM]	YH=	50.000
	Z MIN [MM]	ZL=	-10.000
	Z MAX [MM]	ZH=	20.000

FELULETELEM OSZTAS U MENTEN	IU=	4
FELULETELEM OSZTAS V MENTEN	IV=	4
ONMETSZES VIZSGALAT [0=NINCSE,1=VAN,2=EROS]	OC=	1
LINEARIS KOZELITTES TURESE [MM]	LT=	0.100
ALGORITMIJUS HTBAKORLAT	HK=	0.003
KONTUROK LEZARASA [0=AUT,-1=+,-2=-,1=Y,2=X]	KO=	0

KILEPES A DIALOGUSBOL: KOVETKEZO FAZIS	(M)
VISSZA A FAZIS ELEJERE	(V)
SZINT ELDOBASA,UJ SZINT	(E)
"RETURN" "BACK SPACE	

ERTELMEZES:

- TU, IV: A PARAMETERSIK FELSZTASA A SIKMETSZESHEZ. ERTEKUKET
NOVELVE A METSZETKONTUROK KOZELITESE FINOMODIK, A SZAMITASI
IDO VISZONT NOVEKSZIK.
- LT: A SIKMETSZES EREDMENYEKENT KAPOTT KONTUROK IGEN SOK
KIS SZAKASZBOL ALLNAK, AZ EGY EGYENESBE ESO RESZEKEN IS.
LT VALTOZO ARRA SZOLGAL HOGY AZ IGY MEGADOTT
TAVOLSAGNAI NEM NAGYOBB ELTERSSEL EGY EGYENESBE
ESO PONTOK HELYERE EGYETLEN EGYENES SZAKASZ KERULJON A
METSZETKONTUROKBAN. EZ JELENTOSEN MEGROVIDITHETI
A TOVABBI FUTASI IDOT ES AZ EREDMENYEZETT
NC SZALAGOT.
- OC: AZ OPSZETELES GYORSITASA ERDEKRENI AZ UTKOZESELLENORZER
CSOKKENTHETO ILLETVE ELHAGYHATO OC=1 ILLETVE OC=0
REALLITASSAL. EBEN AZ ESETBEN TERMESZTESEN AZ EREDMENY
HELYESSEGET ELLENORIZINT KELL, TELJSEN AUTOMATIKUS
MUNKATERTES ESEKEN OC=2 ERTEKET KELL HASZNALNI.
INTERAKTIV HASZNALAT ESEKEN AJANLATOS AZ OC=1 ERTEK
HASZNALATA, MIVEL AZ ESETEK DONTO TOBBSEGEREN EZ MAR JO
ERDMENYT AD, VISZONT A FUTASI IDO ROVIDEBB LESZ.
- HK: A REAL SZAMOK GEPI ABRAZOLASANAK PONTATLANSAGA MIATT
A SIKMETSZO ALGORITMUSBA BEEPITETT HIBAKORLAT, A PROGRAM
AZ ILYEN KULONRSEGGEL EGYMASTOL ELTERO METSZESPONTOKAT
AZONOSNAK TEKINTI, METSZESI HIRA ESETEN HK NOVELESE
ELOSEGITHETI A HELYES EREDMENY KISZAMITASAT.
- KO: HA A FELULET ES EGY ADOTT SZINT METSZETE NEM LENNE ZART
KONTUR, AKKOR A SIKMETSZO PROGRAM EZEBOL ZARTAT KESZIT.
KO=0 ESETEN HA AZ ELOGYARTMANYT ELERI, AZ ELOGYARTMANY
METSZETNEK KERULETE MENTEN, HA NEM, AKKOR OLYAN KONTUROKAT
KESZIT, HOGY A MEGMUNKALANDO RESZEK ZSEBEKET ALKOSSANAK.
KO=-1, -2 ERTEKEK ESETEN A PARAMETERSIKON POZITIV ILLETVE
NEGATIV KORULJARASSAL ZARJA BE A KONTUROKAT. EZEK EGYIKE
MEGEGYEZIK A KO=0 ESETEN KAPOTT EREDMENNYEL, A MASIKNAL
PEDIG A MEGMARADO RESZEK VANNAK SZIGETKENT BEZARVA.
A MEGMUNKALAS KOZOTT A KULONRSEG CSAK ABBAN VAN, HOGY
AZ ELOGYARTMANY SZELE AZ ELSO ESETREN FALKENT MEGMARAD,
A MASODIKNAL PEDIG LEMUNKALODIK.
KO=1, 2 SEGITSEGEVEL X, ILLETVE Y TENGELY MENTEN TORTENIK
A KONTUROK BEZARASA, HA EZ LEHETSEGES. ERRE VISZONYLAG
RITKAN VAN SZUKSEG.

2. SZ. DIALOGUS

** TECHNOLOGIA **

VEZERLES [1=DIALOG, 2=UNIMERIC]	SG=	2
HUTES [0=NINCSE, 1=VAN]	HU=	1
SZERSZAMSUGAR [MM]	RR=	7,000
SZERSZAM HELYE [0=NINCSE VALTAS]	SH=	0
FORDULATSZAM [FORD/PERC]	FF=	4000
ELOTOLAS [MM/PERC]	SE=	400
FOGASMEYLSÉG [MM]	FM=	4.
SIMITASI KAHAGYAS [MM]	TU=	0,3
MEGMUNKALASI IRANY [1=X, 2=Y]	IR=	1
KEZDOPONT [00, 01, 10, 11]	KP=	0
ELOGYARTMANY KORULMARASA [0=NEM, 1=IGEN]	KM=	0
TERULETMAKAS [0=NEM, 1=IGEN]	TM=	1

KILEPES A DIALOGUSBOL: KOVETKEZO FAZIS (M)
 VISSZA A FAZIS ELEJERE (V)
 SZINT ELDOBASA, UJ SZINT (E)

"RETURN "BACK SPACE

ERTELMEZES:

IR: KIJELOLI, HOGY A CIKK-CAKK PALYA MELYIK TENGELLYEL PARHUZAMOSAN GENERALODJON.

KP: KIJELOLI, HOGY AZ ELOGYARTMANY MELYIK SAROKPONTJABOL KEZDODJON A CIKK-CAKK PALYA GENERALASA. HA AZ ELOGYARTMANY KIJELOLT SARKA NEM ESIK A MEGMUNKALASI TERULETRE, FZ CSAK A SZAMITAS MEGKEZDESENEK A HELYET JELENTI. KP ERTEKENEK A ZAROJELBEN FELSOROLT ERTEKEK VALAMELYIKET KELL ADNI. A SZAMOK A KOVETKEZO MODON RENDELODNEK HOZZA A SAROKPONTOKHOZ:

```

Y ↑
  1
  1  01 ----- 11
  1      I           I
  1      I           I
  1      I           I
  1  00 ----- 10
  1
  1 ----->X

```

TM: MEGMUNKALAS, ES ETER-ESAK A PALYA SZELKONTROLKODAS ENTEL TORTENIK.

3.SZ. DIALOGUS

** OUTPUT **

NC AZONOSITÓ	NA=123456
MAX.MONDAISZAM	LM= 600
DIALOGUS ADATOK A PRINTERRE [0=NEM, 1=IGEN]	LP= 0
NC FILE	DF= 1
[0=NEM, 1=IGEN]	

KILEPÉS A DIALOGUSRÓL: KÖVETKEZŐ FÁZIS	(M)
VISSZA A FÁZIS ELEJÉRE	(V)
SZINT ELDOBÁSA, ÚJ SZINT	(E)

"RETURN "BACK SPACE

ERTELMÉZÉS:

NA: DIALOG VÉZÉRELES ESETEN AZ NC PROGRAM AZONOSÍTÓJA.

LP: LP=1 ÉRTEK ESETEN MINDEN AMI A KÉPERNYŐRE KERÜL,
EGYIDEJÜLEG A SORNYOMTATORA IS KIIRODIK.

DF: DF=0 ÉRTEK MELLET NEM KÉSZÜL NC PROGRAM, A PARAMETEREK
BEÁLLÍTÁSÁHOZ ÉS ELLENŐRZÉSÉHEZ CELSZERŰ EZT AZ
ÜZEMMÓDOT HASZNÁLNI, A TENYLEGES MŰNKA MEGKEZDÉSE ELŐTT.

4.SZ. DIALOGUS

** INTERAKTIVITAS **

SIKMETSZES UTAN	SI=	1	
OFSZETELES UTAN	OI=	1	
PASSZITAS UTAN	PI=	0	
PALYAGENERALAS UTAN	GI=	1	
TRACE	TR=	0	
TRACE PRINTERRE	TP=	0	
KILEPES A DIALOGUSBOL:	KOVETKEZO FAZIS		(M)
	VISSZA A FAZIS ELEJERE		(V)
	SZINT ELDOBASA,UJ SZINT		(E)
"RETURN	"BACK SPACE		

ERTELMEZFS:

SI,OI,PI,GI: HA AZ ERTEK 1, AKKOR MINDEN SZINTEN A KIJELOLT FAZIS UTAN GRAFIKUS MEGJELENITES KOVETKEZIK ES A VEZERLES A DIALOGUSON KERESZTUL VISSZATER A FELHASZNALOHOZ.

TR: NYOMKOVETES BEALLITASA HIBA ESETERE, EZ MEGKONNYITI A RENDSZERPROGRAMOZOK SZAMARA A HIBA OKANAK FELDERITESET, TELJES NYOMKOVETES: TR=-1.

TP: TP=1 ESETEN A NYOMKOVETES ADATAI A SORNYOMTATORA IS KIRODNAK.

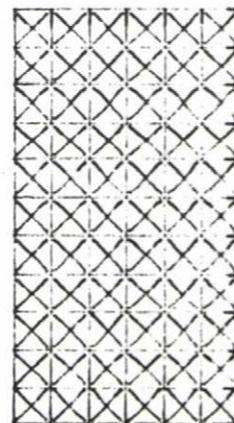
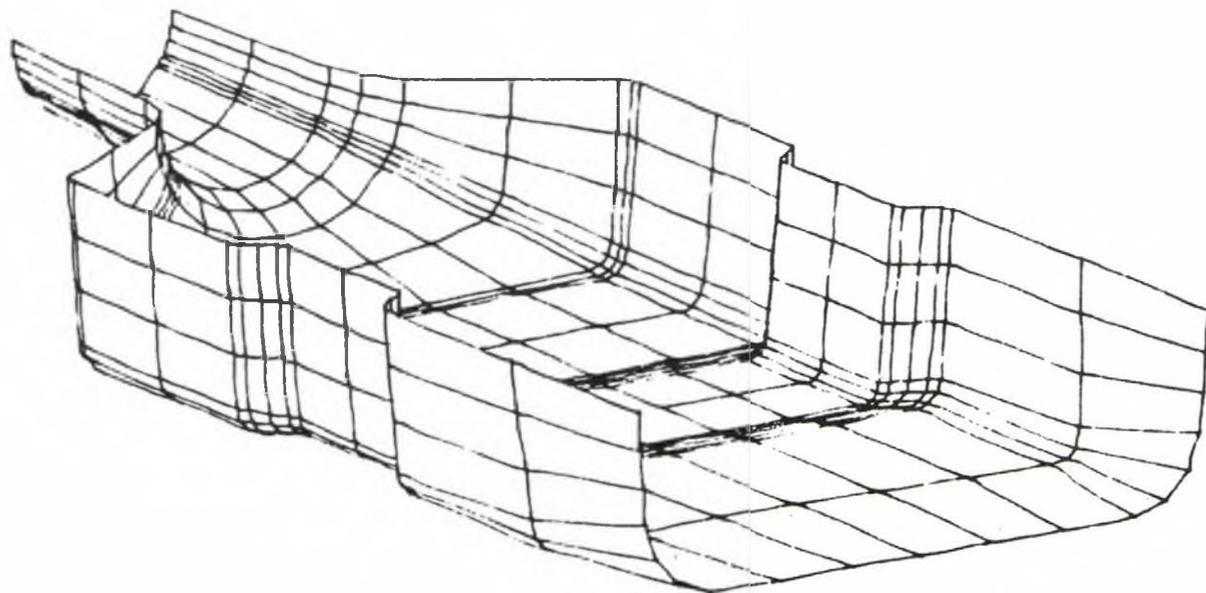
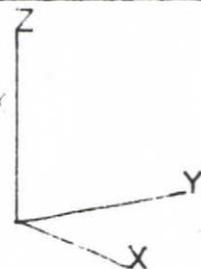
2.SZ. MELLÉKLET

Műanyag flakon formázó szerszámának terve és nagyolás ellenőrzési fázisainak plotter-dokumentációja.

MTA SZTAKI
C.A.D.

FREE-FORM SHAPES
VERSION 1.2. 84.03.01.

JKRAUTER

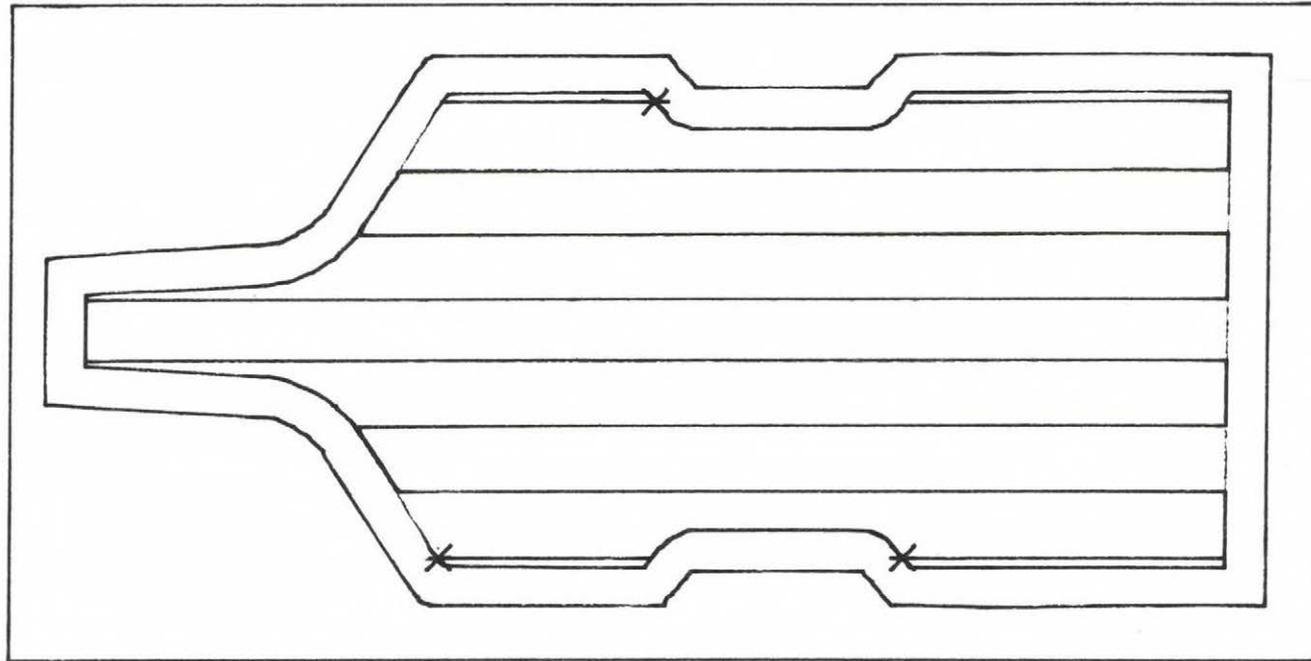


F F S CAD/CAM
NAGYOLO PROCESSZOR

MTA SZTAKI

VERZIO 2.1.
84.11.15.

FAZIS
PALYAGEN.



FELULET

DK1 JKRAUTER

Z= 25.600

Y



X

3.SZ. MELLÉKLET

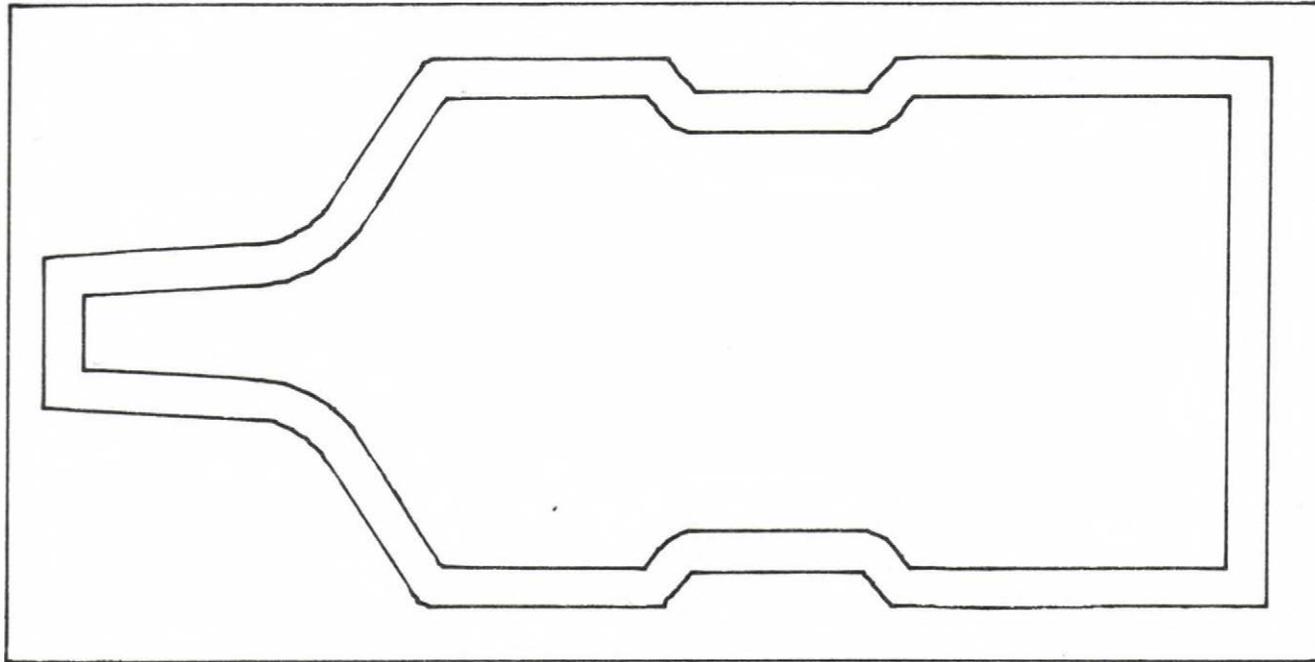
FFS munkadarabok fényképei

F F S CAD/CAM
NAGYOLO PROCESSZOR

MTA SZTAKI

VERZIO 2.1.
84.11.15.

FAZIS
OFSZETELES



FELULET

DK1 JKRAUTER

Z= 25.600

Y



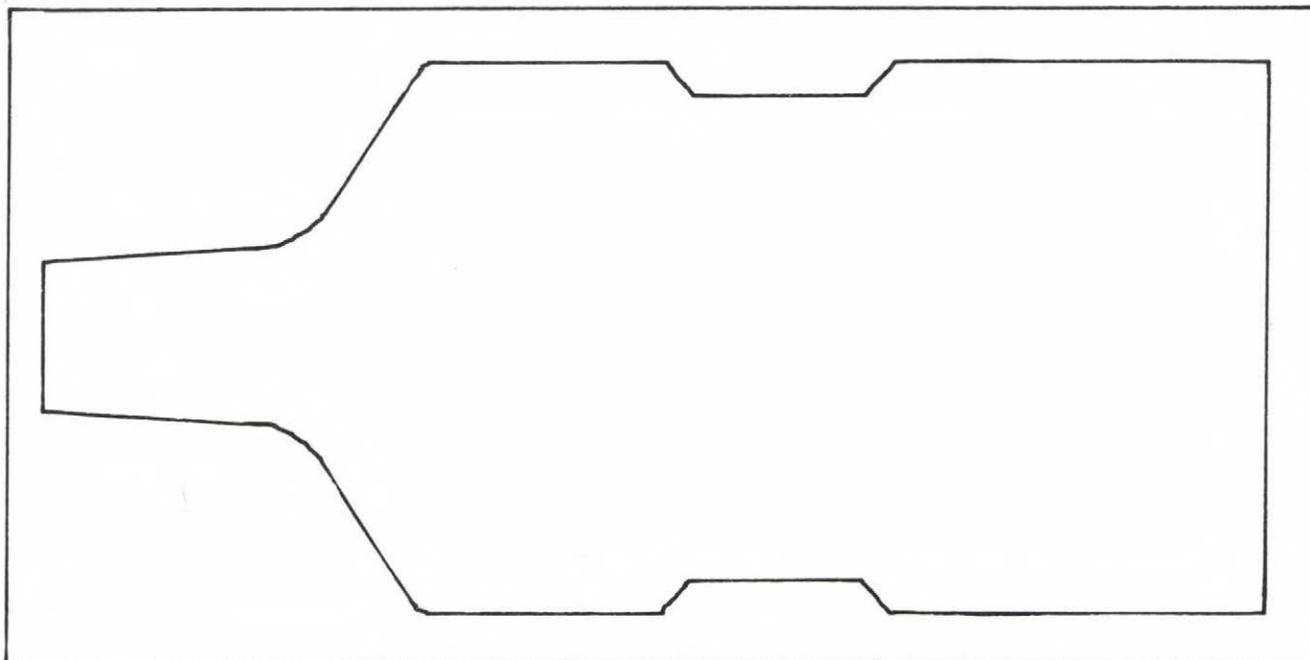
X

FFS CAD/CAM
NAGYOLO PROCESSZOR

MTA SZTAKI

VERZIO 2.1.
84.11.15.

FAZIS
SIKMETSZES



FELULET

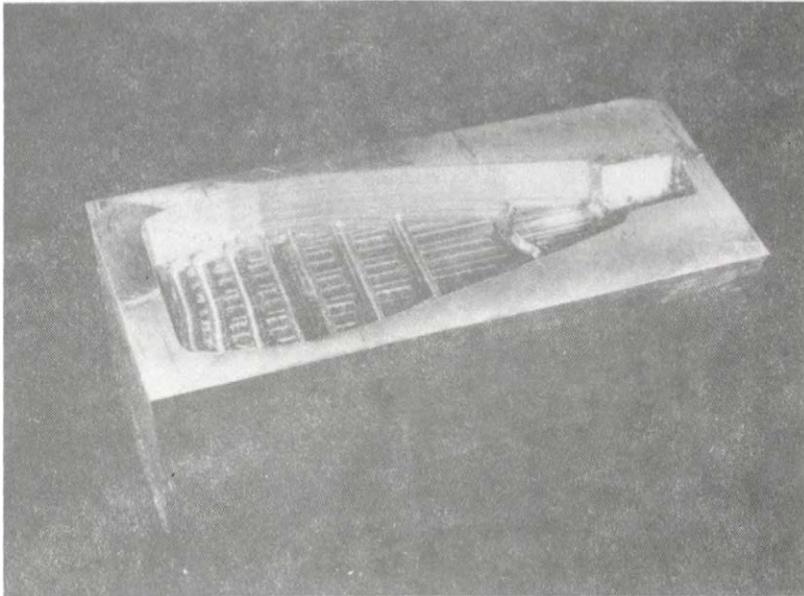
DK1 JKRAUTER

Z= 25.600

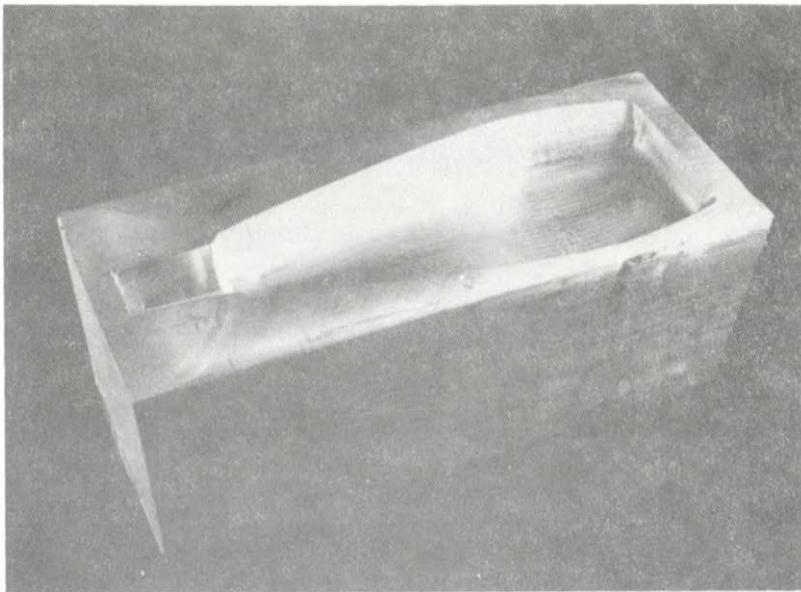
Y



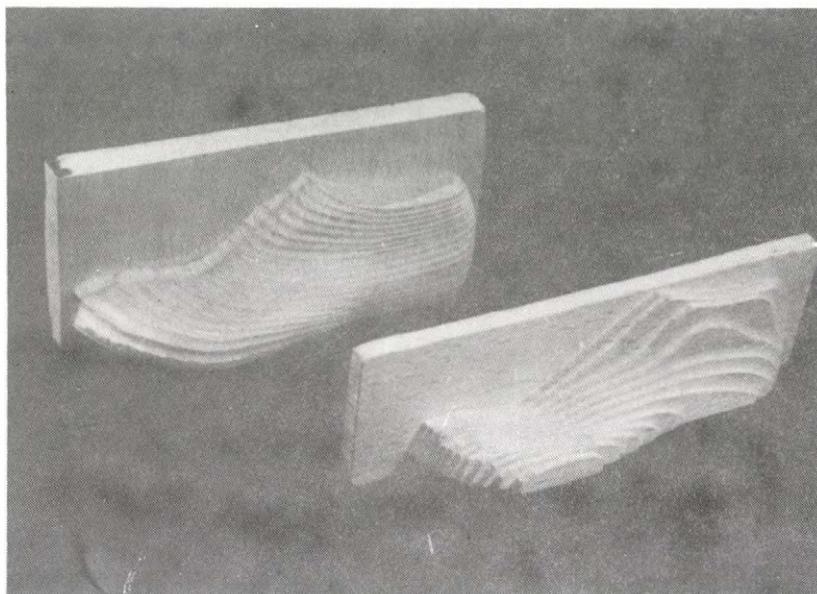
X



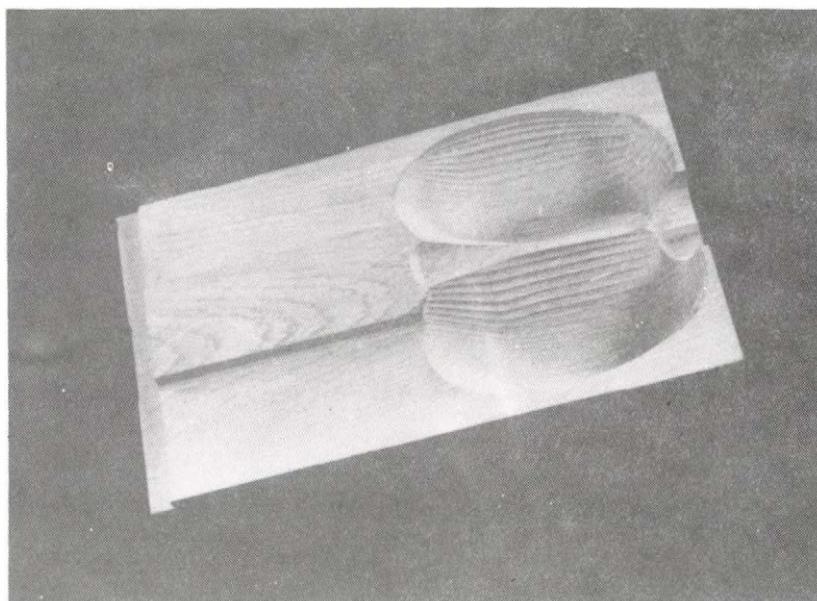
1. Műanyag flakon formázó szerszáma nagyolás után



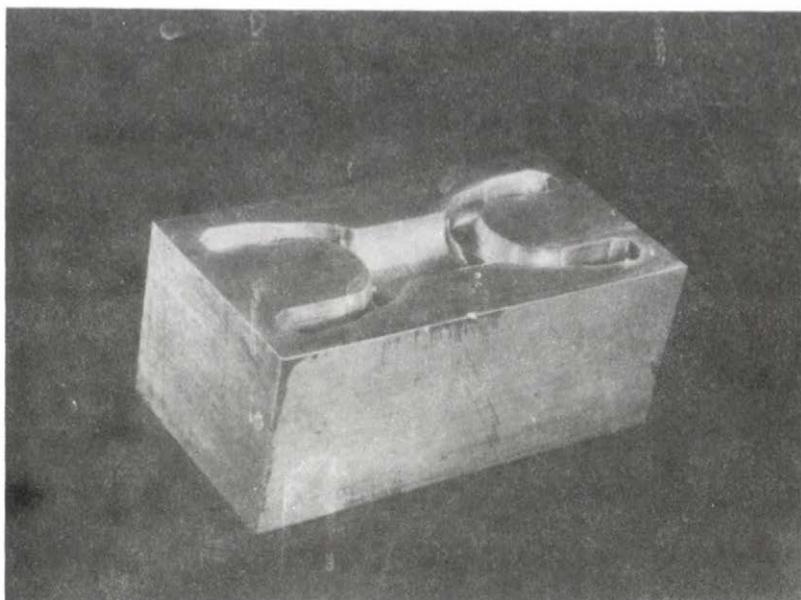
2. Műanyag flakon formázó szerszám simítás után



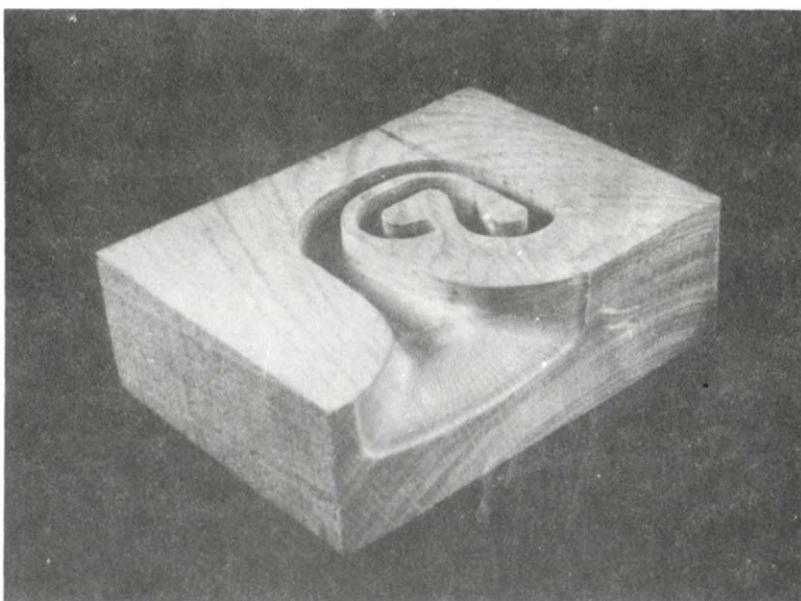
3. Cipőipari szerszám famintája nagyolás és
simitás után



4. Autókürt szerszámának famintája



5. Kardánvilla alakító szerszáma

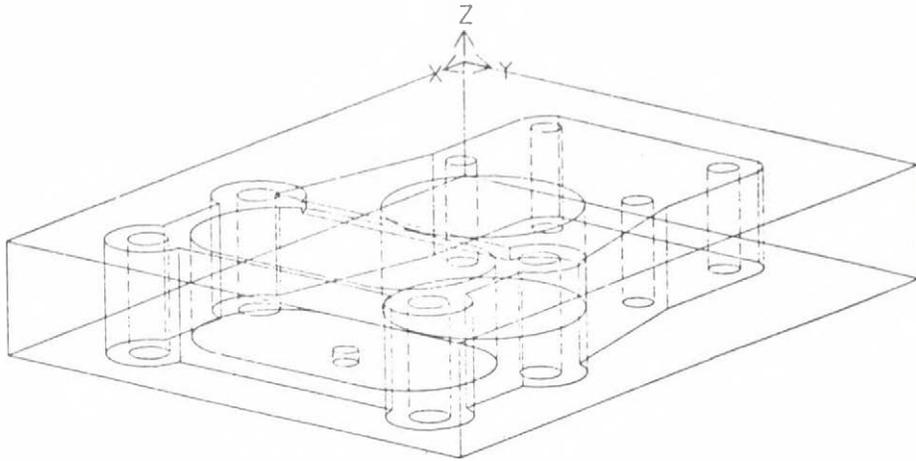


6. Pelton turbinalapát szerszámának famintája

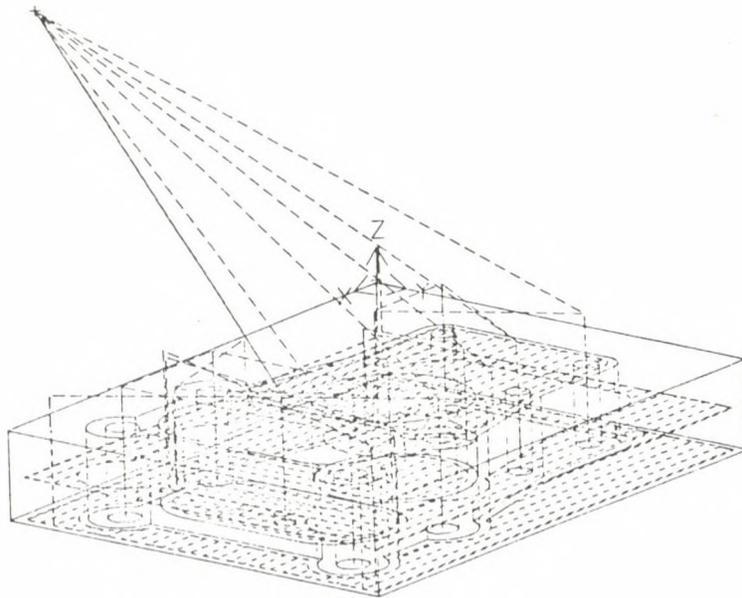
4.SZ. MELLÉKLET

BUILD rendszerrel tervezett alkatrész és
megmunkálásának szerzámpanyái

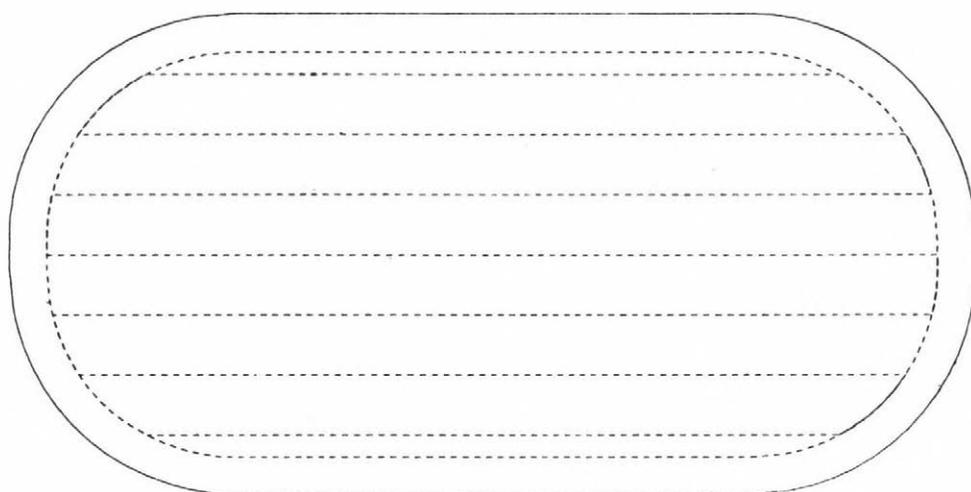
/Tektronix 4014 display-n készült ábrák/



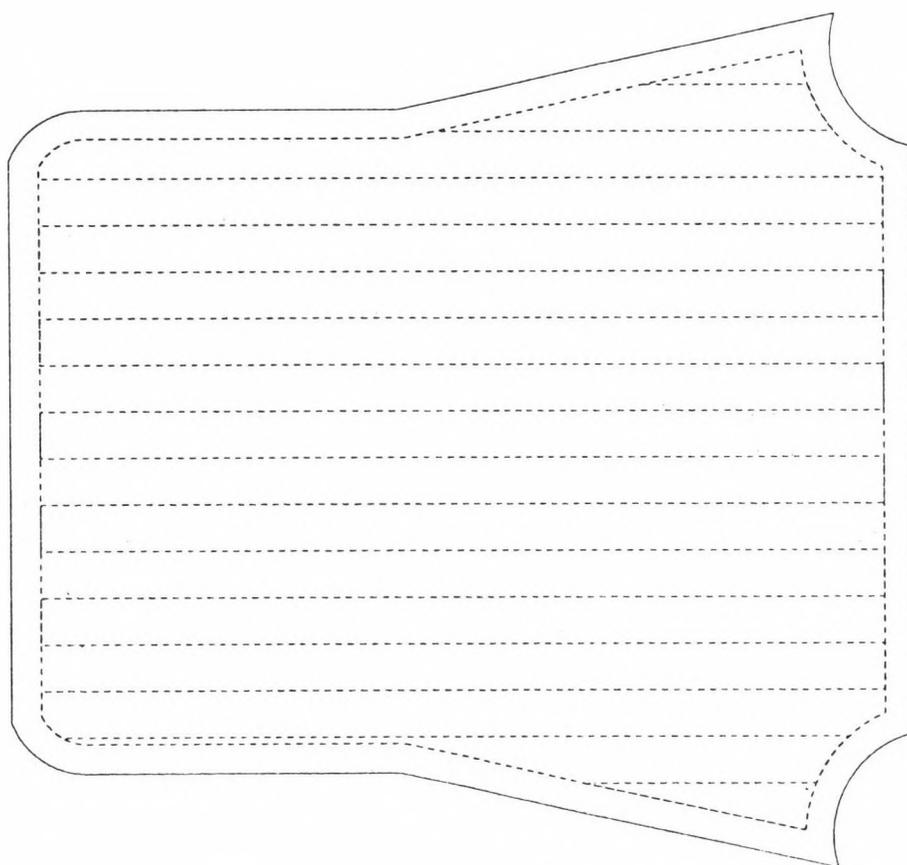
1. "GEHAUSE" alkatrész terve



2. Szerszám pálya a "GEHAUSE" alkatrész megmunkálásához



3. Zsebmarás



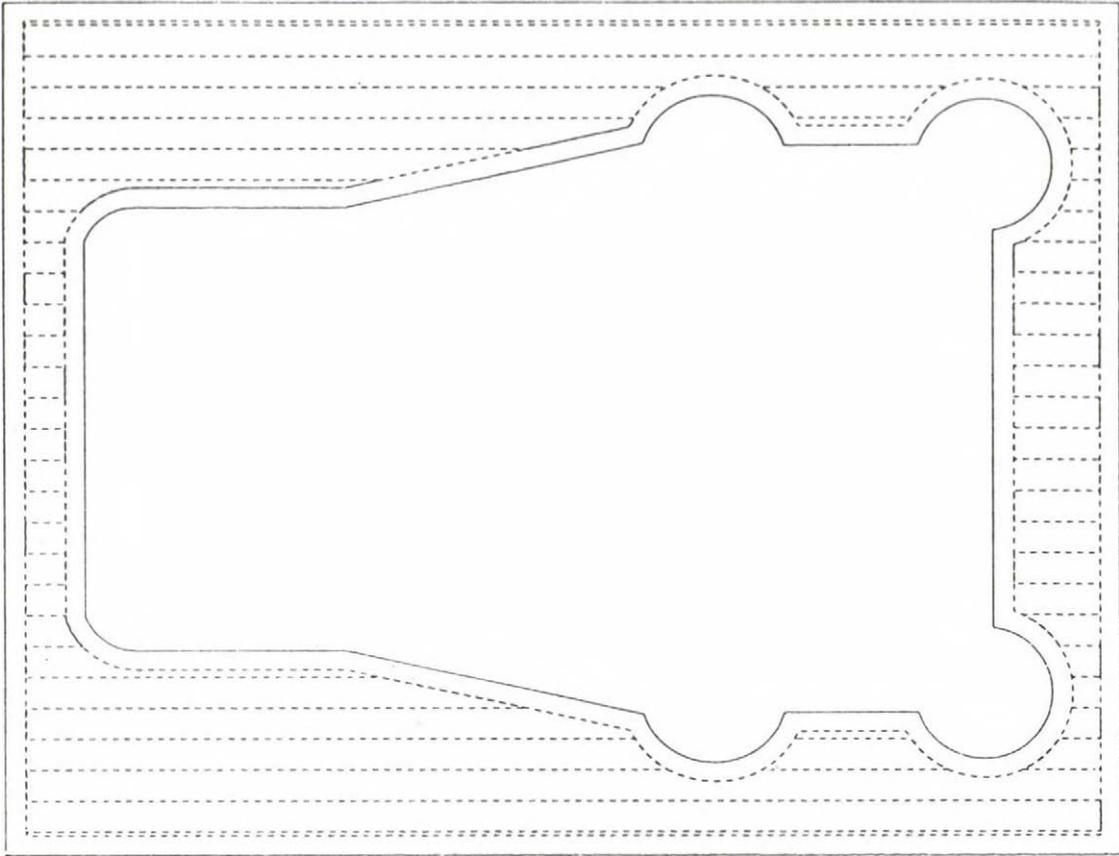
4. Zsebmarás

1984-BEN JELENTEK MEG:

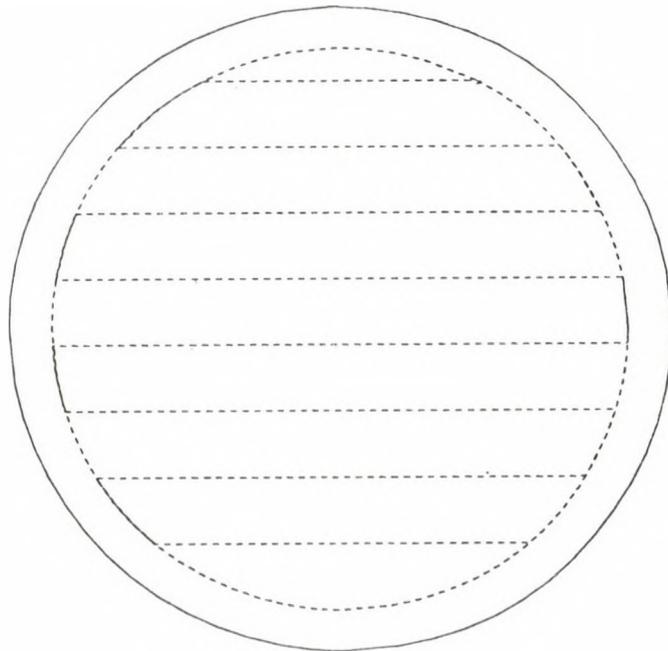
- 155/1984 Deák, Hoffer, Mayer, Németh, Potecz, Prékopa, Straziczky: Termikus erőműveken alapuló villamos-energiarendszerek rövidtávu, optimális, erőművi menetrendjének meghatározása hálózati feltételek figyelembevételével.
- 156/1984 Radó Péter: Relációs adatbáziskezelő rendszerek összehasonlító vizsgálata
- 157/1984 Ho Ngoc Luat: A geometriai programozás fejlődései és megoldási módszerei
- 158/1984 PROCEEDINGS of the 3rd International Meeting of Young Computer Scientists,
Edited by: J. Demetrovics and J. Kelemen
- 159/1984 Bertók Péter: A system for monitoring the machining operation in automatic manufacturing systems
- 160/1984 Ratkó István: Válogatott számítástechnikai és matematikai módszerek orvosi alkalmazása
- 161/1984 Hannák László: Többértékű logikák szerkezetéről.
- 162/1984 Kocsis J. - Fetyiszov V.: Rugalmas automatizált rendszerek: megbízhatóság és irányítási problémák
- 163/1984 Kalavszky Dezső: Meleghengerművi villamos hurokemelő hajtás vizsgálata
- 164/1984 Knuth Előd: Specifikációs adatbázis modellek
- 165/1984 Petróczy Judit: Publikációk 1983

1985-BEN EDDIG MEGJELENTEK:

- 166/1985 Radó Péter: Információs rendszerek számítógépes tervezése
- 167/1985 Studies in Applied Stochastic Programming I,
Szerkesztette: Prékopa András
- 168/1985 Böszörményi László - Kovács László - Martos Balázs
Szabó Miklós: LILIPUTH
- 169/1985 Horváth Mátyás: Alkatrészgyártási folyamatok
automatizált tervezése
- 170/1985 Márkus Gábor: Algoritmus mátrix alapu logaritmus
kiszámítása kriptográfiai alkalmazásokkal
- 171/1985 Tamás Várady: Integration of free-form surfaces
into a volumetric modeller
- 172/1985 Reviczky János: A számítógépes grafika terület-
kitöltő algoritmusai



5. Külső kontur kialakítása



6. Zsebmarás



MS. B. 1. 7.