Nagyné dr. Szilvási Márta

CAD-iskola

TypoTEX Kft Elektronikus Kiadó
1991

Lektorálta: **Reiman István**

© Nagyné dr. Szilvási Márta, 1991

ISBN 963 7546 02 2

Kiadja a TypoTEX Kft
Felelős kiadó: Votisky Zsuzsa
Felelős szerkesztő: Köves Gabriella
A szedés az EMTEX+MATEX3.0, a korrektúra
a NyelvÉsz programcsomaggal készült
a TypoTEX Kft Elektronikus Kiadóban.
NYOMDA:
MEMPHIS GMK.
A borítót és a kötést tervezte: Hörömpő Gabriella
A könyv formátuma: B/5
Ívterjedelme: 8,4 (A/5) ív
A kézirat lezárva: 1991. május

Tartalomjegyzék

Előszó **5**

1. **Derékszögű koordináta-rendszerek** **7**

1.1. Descartes-féle koordináták 7
1.2. Koordinátatranszformációk 8
1.2.1. A koordináta-rendszer eltolása 8
1.2.2. A koordináta-rendszer elforgatása 8
1.2.3. A hosszegységek megváltoztatása a koordinátatengelyeken (skálázás) 9
1.2.4. Összetett transzformációk 10
1.3. Homogén koordináták 12

2. **Térbeli alakzatok transzformációi** **15**

2.1. Lineáris transzformációk, ponttranszformációk 15
2.2. Egyenes körüli forgatás 16
2.3. Síkra vonatkozó tükrözés 19
2.4. Nyújtás, zsugorítás a koordinátatengelyek irányában 20
2.5. Merőleges vetítés 21
2.5.1. Vetítés a koordinátasíkokra 21
2.5.2. Merőleges vetítés origón áthaladó általános helyzetű síkra 22
2.6. Ponttranszformációk felírása homogén koordinátákkal 24

3. **Felületek meghatározó adatai és előállítása** **26**

3.1. Csavarfelületek 26
3.2. Forgásfelületek 27
3.3. Általános hengerfelületek 29
3.4. Általános kúpfelületek 30
3.5. Összetett alakzatok szerkesztése 31

4. **A közelítő poliéderek adatrendszere** **34**

5. **A megjelenítés módszerei** **37**

5.1. Merőleges nézetek egymásra merőleges képsíkokon 37
5.2. Axonometrikus leképezések 38
5.2.1. Ortogonális axonometria 39
5.2.2. Ferdeszögű axonometria 40
5.3. Centrális projekció 41

6. **A CAD-rendszerek felépítése és típusai** **43**

6.1. A számítógépi geometriai modellezés műveletei 45
6.2. Modellezés diszkrét és analitikus adatrendszerekkel 49

6.2.1. Modellezés diszkrét adatrendszerrel 49
6.2.2. Modellezés analitikus adatrendszerekkel 50
6.3. Adatrendszerek szerkezete, drótvázas modellek, felületmodellek és testmodellek 56
6.4. A láthatóság szerinti ábrázolás eljárásai 58
6.5. A grafikus rendszerek közötti adatátvitel problémái 64
7. CAD-környezet **66**
8. A CAD-iskola programjai **68**
Önálló gyakorlásra szánt feladatok
Irodalom

Előszó

A CAD-iskola a CAD-rendszerek leendő felhasználói számára készült, de sok támpontot nyújt azoknak is, akik térbeli alakzatok numerikus leírását igénylő programozási feladatok megoldására vállalkoznak.

A könyv bevezetést ad a számítógépi geometriai modellezés elméletébe és gyakorlatába, és ezáltal érthetővé teszi a CAD-rendszerek működésének alapelveit.

A mellékelt, PC/AT számítógépre készített gyakorló programmal elvégezhetők a geometriai modellezés alapműveletei, vagyis a háromdimenziós felületek és testek definiálása, azokon geometriai transzformációk végrehajtása, összetett alakzatok szerkesztése, azoknak különböző vetületekben való megjelenítése és a megfelelő típusú adatfile-ok kezelése. A programmal való gyakorlás nemcsak a CAD-rendszerek használatára készít fel, de segít a térbeli látás és gondolkodás képességének fejlesztésében is. Új oktatási segédeszközként jó hasznát vehetik a középiskolák és felsőfokú oktatási intézmények tanárai és diákjai.

A számítógépi modellezés nemcsak hasznos, de szórakoztató tevékenység is. Ez azonban ne csábítson senkit arra, hogy próbálgatással, találgatással fogjon hozzá egy elképzelt modell elkészítéséhez.

Azt javaslom, készítsünk vázlatot a tervezett modellről, gondoljuk meg, milyen részekből áll, a részeket hogyan fogjuk definiálni és összeépíteni.

E módszer elsajátításához adnak segítséget a könyv végén található gyakorló oldalak.

A CAD-iskola programjait mindenki tudja használni, aki a számítógép kezelését olyan szinten ismeri, amennyire azt az egyszerűbb szövegszerkesztő programok megkívánják. A könyv első gyakorlata még csak a háromkoordinátás pontok és vektorok ismeretét feltételezi, de a harmadik fejezet már összetett térbeli alakzatok szerkesztésére készít fel. A további fejezetek pedig a CAD-rendszerek működésébe és belső felépítésébe adnak betekintést.

A szerző

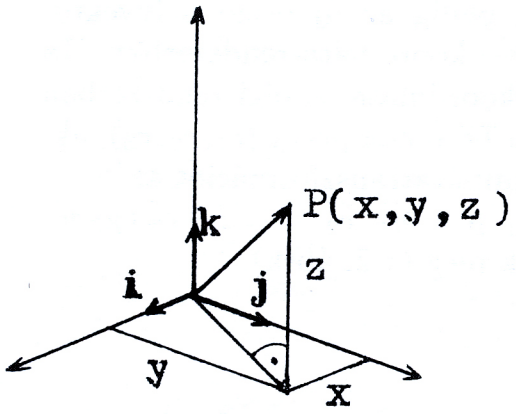
1. Derékszögű koordináta-rendszerek

A háromdimenziós euklideszi tér pontjainak koordinátázásáról lesz szó, miközben feltételezzük a vektorműveletek és a mátrixműveletek ismeretét.

1.1. Descartes-féle koordináták

Legyenek $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ páronként merőleges jobbrendszert alkotó egységvektorok. A $\mathbf{v}$ vektornak a $\mathbf{v} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ egyértelműen létező előállításában az (x, y, z) számhármast a $\mathbf{v}$ vektor $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ bázisra vonatkozó koordinátáinak nevezzük. Ha a térben megadunk egy O kezdőpontot (origót), akkor bármely P pont helyét megadhatjuk az $\overrightarrow{OP}$ helyvektorral. A pont helyvektorának koordinátáit a pont derékszögű vagy *Descartes*-féle koordinátáinak nevezzük. Ha $\overrightarrow{OP} = x\mathbf{i}+y\mathbf{j}+z\mathbf{k}$, a P pont (x, y, z) koordinátái meghatározzák a pont helyét abban a koordináta-rendszerben, amelynek kezdőpontja O, tengelyei pedig az $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ koordináta-egységvektorokkal párhuzamos egyenesek az O ponton át (1.1. ábra).

Mi van a lemezen ?



1.1. ábra. Descartes-féle koordináták

A CAD-iskola programjai minden számítást *Descartes*-féle koordináta-rendszerben végeznek el az analitikus geometria eszközeivel.

A görbét és poligont előállító CURVE program az xy-koordinátasíkban számol, és ezt a síkot úgy ábrázolja, hogy az origót a képernyő középpontjába helyezi, az x-tengely vízszintes és jobbra mutat, az y-tengely függőlegesen felfelé mutat, és az egység a tengelyeken egy képpontnak az x-irányú mérete. Az aktuális poligon pontjainak koordinátáit a g (get info) utasítás kiírja a képernyőre.

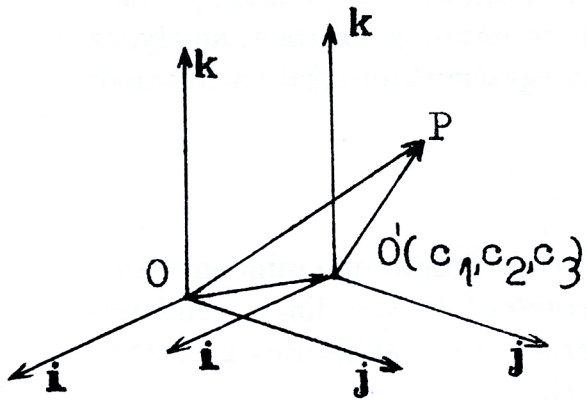
A térbeli alakzatokat előállító SURF program az x-, y-, z- tengelyű derékszögű koordináta-rendszerben számol, amelynek a vetületét az aktuális alakzat vetületével együtt akkor láthatjuk, ha a rajzoltatás előtt (d utasítás) kiadjuk a c utasítást is. Egy vetület megrajzolásakor az origó a képernyő középpontjába kerül, és a pozitív koordinátatengelyeknek 100 egységnyi szakasza jelenik meg az aktuális leképezésnek (*a:* axonometrikus, *f:* elölnézet, *t:* felülnézet) megfelelően vetítve. Négy vetület

elkészítésekor (v parancs) a koordináta-rendszer megfelelő vetülete a képernyő egy negyedének a középpontjába kerül. A koordináta- egységek választása összhangban van a CURVE programmal. A g utasítás itt az alakzatot befoglaló téglatest méretét írja ki, azaz az alakzat pontjaira az x-, y-, z-koordináták minimális és maximális értékét.

1.2. Koordinátatranszformációk

Ha más koordináta-rendszerre térünk át, kiszámíthatjuk az egyik rendszerbeli koordinátákból a másik rendszerbeli koordinátákat, amit koordinátatranszformációnak nevezünk

1.2.1. A koordináta-rendszer eltolása



1.2. ábra. A koordináta-rendszer eltolása

Toljuk el a koordináta-rendszer kezdőpontját egy O' pontba. Ekkor az $\overrightarrow{O'P} = x'\mathbf{i} + y'\mathbf{j} + z'\mathbf{k}$ helyvektorra $\overrightarrow{O'P} = \overrightarrow{OP} - \overrightarrow{OO'}$ teljesül, ahol $\overrightarrow{OP}$ a P pont helyvektora, $\overrightarrow{OO'}$ pedig az új origó helyvektora az eredeti koordináta-rendszerben. Ha a P pont koordinátái a régi rendszerben (x, y, z), az O' ponté pedig (c_1, c_2, c_3), akkor a koordinátatranszformációt az $x' = x - c_1, y' = y - c_2, z' = z - c_3$ egyenletek adják meg (1.2. ábra).

1.2.2. A koordináta-rendszer elforgatása

Válasszunk három új, egymásra páronként merőleges jobbrendszert alkotó egységvektort bázisvektornak, miközben az origó helyben marad. Legyenek $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ az új koordináta-egységvektorok a régi bázisban a következőképpen kifejezve

$$\mathbf{e}_m = a_{1m}\mathbf{i} + a_{2m}\mathbf{j} + a_{3m}\mathbf{k}, \qquad m = 1, 2, 3.$$

Az együtthatókat egy mátrixba írjuk, amelyet a bázistranszformáció mátrixának nevezünk.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

vagy tömörebb formában írva

$$A = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3].$$

Ha a P pont koordinátáit az eredeti $\mathbf{i}, \mathbf{j}, \mathbf{k}$ bázisban (x, y, z), az új $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ bázisban pedig (u, v, w) jelöli, akkor

$$\begin{aligned}\overrightarrow{OP} &= u\mathbf{e}_1 + v\mathbf{e}_2 + w\mathbf{e}_3 \\ &= u(a_{11}\mathbf{i} + a_{21}\mathbf{j} + a_{31}\mathbf{k}) + v(a_{12}\mathbf{i} + a_{22}\mathbf{j} + a_{32}\mathbf{k}) + w(a_{13}\mathbf{i} + a_{23}\mathbf{j} + a_{33}\mathbf{k}),\end{aligned}$$

másrészt

$$\overrightarrow{OP} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}.$$

Az $\mathbf{i}, \mathbf{j}, \mathbf{k}$ vektorok együtthatóinak összehasonlításával

$$\begin{aligned}x &= a_{11}u + a_{12}v + a_{13}w \\ y &= a_{21}u + a_{22}v + a_{23}w \\ z &= a_{31}u + a_{32}v + a_{33}w\end{aligned}$$

adódik. Tömörebb formában

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}.$$

Mivel a bázistranszformáció mátrixának determinánsa az $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ oszlopvektorok lineáris függetlensége miatt nem zérus, létezik az inverze. Az inverz pedig az oszlopvektorok ortonormáltsága miatt a mátrix transzponáltja (a főátlóra tükrözött mátrix), ezért a pont koordinátái az új koordináta-rendszerben az

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = A^* \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

egyenletekkel számíthatók ki, ahol A^* az A transzponáltját jelöli.

1.2.3. A hosszegységek megváltoztatása a koordinátatengelyeken (skálázás)

Legyen az új koordináta-rendszer kezdőpontja ugyanaz mint az eredetié, a koordináta-egységvektorok pedig az $\mathbf{i}, \mathbf{j}, \mathbf{k}$ vektorokkal egyirányúak. Ekkor

$$\mathbf{e}_1 = q_x\mathbf{i}, \quad \mathbf{e}_2 = q_y\mathbf{j}, \quad \mathbf{e}_3 = q_z\mathbf{k}, \qquad q_x > 0, q_y > 0, q_z > 0.$$

Kiszámítjuk az **i, j, k** rendszerben adott $P(x, y, z)$ pont koordinátáit az új koordináta-rendszerben. Mivel

$$\overrightarrow{OP} = u\mathbf{e}_1 + v\mathbf{e}_2 + w\mathbf{e}_3 = u \cdot q_x\mathbf{i} + v \cdot q_y\mathbf{j} + w \cdot q_z\mathbf{k},$$

a régi és az új koordináták között a következő összefüggéseket kapjuk

$$x = q_x u, \quad y = q_y v, \quad z = q_z w,$$

illetve

$$u = x/q_x, \quad v = y/q_y, \quad w = z/q_z.$$

1.2.4. Összetett transzformációk

Általános esetben, ha a koordináta-rendszert eltoljuk, elforgatjuk és a tengelyeken a léptéket is megváltoztatjuk, a koordinátatranszformációt e transzformációk egymás után való elvégzésével számíthatjuk ki. Az új koordináta-rendszerből a régibe való átszámításra az

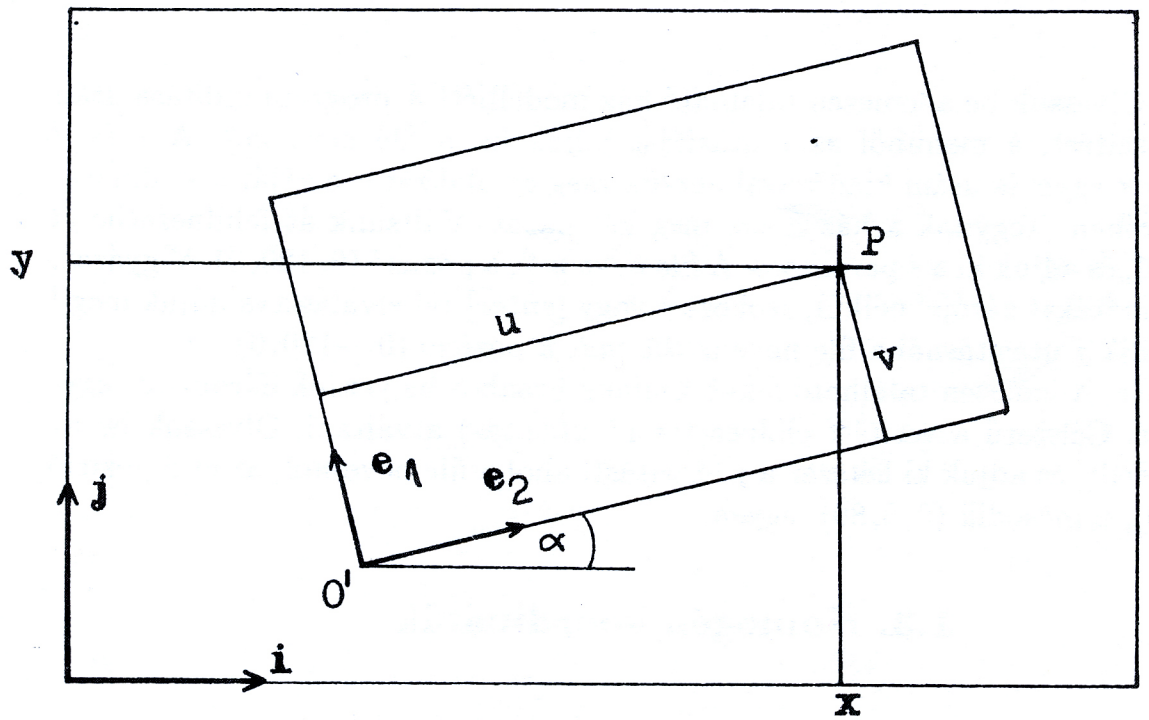
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A \cdot \begin{bmatrix} q_x u \\ q_y v \\ q_z w \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

egyenleteket kapjuk, ahol az A mátrix oszlopaiban az új bázisvektoroknak az eredeti bázisra vonatkozó koordinátái állnak. Az u, v, w innen úgy számítható ki, hogy c_1, c_2, c_3 mindkét oldalból való kivonása után mindkét oldalt az $A^{-1} = A^*$ mátrixszal balról megszorozzuk, majd a megfelelő q_x, q_y, q_z tényezőkkel átosztunk.

Példaképpen egy síkbeli koordinátatranszformációt írunk fel. Tegyük fel, hogy egy bizonyos görbét akarunk a rajzlap adott helyére adott állásban felrajzolni, és a görbe pontjainak (u, v) koordinátáit a görbe saját koordináta-rendszerében ismerjük. E koordináta-rendszernek a kezdőpontja O', bázisvektorai pedig $\mathbf{e}_1$ és $\mathbf{e}_2$ (1.3. ábra).

Rajzoltatás céljából a pontok koordinátáit át kell számolni a rajzlap koordináta-rendszerébe, amelynek tengelyei az **i, j** bázisvektorokkal egyirányúak. Ha a görbe koordináta-rendszerének skálázása nem egyezik meg a rajzlap koordináta-rendszerének skálázásával, akkor először a rajzlapon adott léptékre térünk át. Ez $|\mathbf{e}_1| = q_x|\mathbf{i}|$, $|\mathbf{e}_2| = q_y|\mathbf{j}|$ esetén azt jelenti, hogy a $q_x u$, $q_y v$ koordinátákkal számolunk tovább. Mivel az $\mathbf{e}_1$ és $\mathbf{e}_2$ irányú, (rajzlap) egységnyi hosszú vektorok koordinátái az **i, j** bázisban $(\cos\alpha, \sin\alpha)$, illetve $(-\sin\alpha, \cos\alpha)$, a bázistranszformáció mátrixa

$$A = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$



1.3. ábra. Új koordináta-rendszer választása a síkban

Az $A \cdot \begin{bmatrix} q_x u \\ q_y v \end{bmatrix}$ szorzás elvégzése után a P pont koordinátáit az O' kezdőpontú, **i**, **j** koordináta-egységvektorú rendszerben kapjuk meg. Amennyiben O' koordinátái a rajzlapon (c_1, c_2), ezeket az értékeket kell a megfelelő koordinátákhoz hozzáadni, hogy az (x, y) koordinátákat megkapjuk.

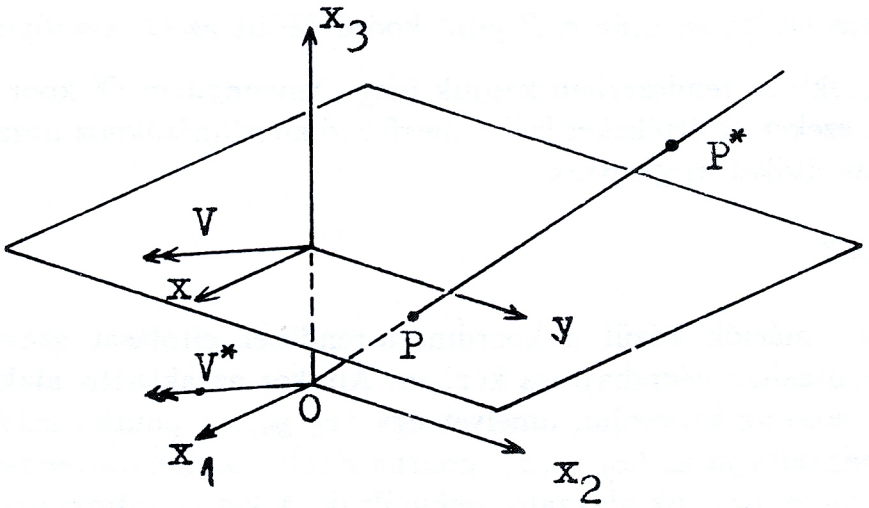
Mi van a lemezen ?

A koordinátatranszformációk közül a koordináta-rendszer eltolását számítja a SURF program a *j* utasítás végrehajtása közben. Amikor az aktuális alakzathoz egy másikat hozzá akarunk kapcsolni, amelyet egy (x_0, y_0, z_0) pontba helyezünk, akkor a program megmutatja az (x_0, y_0, z_0) pontba eltolt koordináta-rendszert is, amelyben a szóban forgó második alakzatot definiáltuk. A két koordináta-rendszer megfelelő tengelyei párhuzamosak. Az utasítás végrehajtásakor a program átszámítja a második alakzat pontjainak koordinátáit az első koordináta-rendszerébe. (A *j* utasításra a 3.5. fejezetben még visszatérünk.)

Gyakorlatok

1. Házsor: Olvassuk be a lemezen található ház modelljét! A program indítása után az *o* utasítást, a menüből az *i* utasítást adjuk ki, a file neve *m3*. A *c* és *d* parancsok egymás utáni kiadásával nézzük meg az alakzatot a saját koordináta-rendszerében. Tegyünk a ház mellé még két házat. Váltsunk át felülnézetbe (*t* utasítás), és adjuk ki a *j* parancsot. A file neve *m3*, a pozíció $(0, 120, 0)$. Vigyázat, a koordinátákat zárójel nélkül, szóközzel vagy [*enter*]-rel elválasztva adjuk meg ! A második *j* utasításnál a file neve ismét *m3*, a pozíció $(0, -120, 0)$.
2. Sarokpolc: A lemezen található fekvő L-alakú hasábot helyezzük háromszor egymás fölé. Célszerű a vetítést elölnézetre (*f* utasítás) átváltani. Olvassuk be az *m2* modellt, és adjuk ki kétszer a *j* utasítást, ahol a file neve *m2*, az első pozíció $(0, 0, 40)$, a második $(0, 0, 80)$ legyen.

1.3. Homogén koordináták



1.4. ábra. A sík pontjainak homogén koordinátái

Először csak az xy-koordinátasík pontjaira értelmezzük a homogén koordinátákat. Csatoljunk az x, y derékszögű koordináta-rendszerhez egy olyan térbeli derékszögű koordináta-rendszert, amelynek x_1, x_2 tengelyei az x, y tengelyekkel párhuzamosak, x_3 tengelyének az egységpontja pedig az eredeti x, y koordináta-rendszer kezdőpontja (1.4. ábra).

Az xy-sík bármely P pontját megadhatjuk az O ponton áthaladó OP egyenessel, vagy ami ugyanaz, az OP^* egyenessel úgy, hogy az egyenesnek az xy-síkkal alkotott döféspontja jelöli ki a szóban forgó pontot. Minden olyan egyenes pedig, amely az xy-síkot metszi, megadható egy olyan $\overrightarrow{OP^*}(x_1, x_2, x_3)$ vektorral, amelyre $x_3 \neq 0$. Ezt az $[x_1, x_2, x_3]$ koordinátahármast a P pont homogén koordinátáinak nevezzük, és szögletes zárójelbe írjuk. A pont homogén koordinátáit egy zérustól különböző λ számmal megszorozhatjuk, hiszen a $(\lambda x_1, \lambda x_2, \lambda x_3)$ koordinátájú vektor ugyanazt az egyenest, és ezáltal ugyanazt a P pontot határozza meg, mint az (x_1, x_2, x_3) koordinátájú vektor. Lehet természetesen az is, hogy a P pontot kijelölő egyenest az $(x, y, 1)$ koordinátájú vektorral adjuk meg, ahol $x = x_1/x_3$, $y = x_2/x_3$. Vagyis az $[x_1, x_2, x_3]$ és $[x, y, 1]$ ugyanannak a $P(x, y)$ pontnak a homogén koordinátái.

Az x_1, x_2, x_3 koordináta-rendszer x_1, x_2 síkjában adott $\overrightarrow{OV^*}(v_1, v_2, 0)$ vektor egyenese nem metszi az xy-síkot. Az ilyen $[v_1, v_2, 0]$ homogén koordinátahármasnak az xy-sík (v_1, v_2) vektorát tudjuk megfeleltetni. Ha ezzel a (v_1, v_2) vektorral párhuzamos egyenesekhez egy „ideális pontot" (az ábrán V jelöli) rendelünk, akkor azt mondjuk, hogy $[v_1, v_2, 0]$ ennek a V ideális pontnak a homogén koordinátái. Két egyeneshez akkor és csak akkor rendeljük ugyanazt az ideális pontot, ha a két egyenes párhuzamos. Ily módon az ideális pontokkal kibővített sík bármely két egyenesének egyetlen közös pontja van. A sík ideális pontjainak összessége a sík ideális egyenesét alkotja. Az ideális pontokkal kibővített euklideszi síkot projektív síknak nevezik. Szokás az ideális pontokat végtelen távoli pontoknak is mondani.

A zérusvektor nem határoz meg egyenest, ezért egy pont homogén koordinátáinak mindegyike nem lehet 0! Egyébként minden számhármas egyetlen pontot határoz meg, az $[x_1, x_2, 0]$ típusúak ideális, az $[x_1, x_2, x_3]$ $(x_3 \neq 0)$ típusúak pedig „közönséges" pontokat. Az ideális pontoknak nincsenek *Descartes*-féle koordinátái, a közönséges pontok homogén és *Descartes*-féle koordinátáira $x_1 : x_2 : x_3 = x : y : 1$ teljesül.

Példaképpen felírjuk az egyenes homogén koordinátás egyenletét. Az $ax + by + c = 0$ egyenletbe $x = x_1/x_3$ és $y = x_2/x_3$ helyettesítésével az $ax_1 + bx_2 + cx_3 = 0$ homogén lineáris egyenlet adódik. Az egyenes ideális pontja $[-b, a, 0]$, amelyet valóban az egyenes irányvektora jelöl ki.

Következő példánk az ellipszis, parabola és hiperbola homogén koordinátás egyenlete, amelyeket a kanonikus egyenletekből írunk fel. Ezek rendre az alábbi homogén másodfokú egyenletek

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = x_3^2, \qquad x_2^2 = 2px_1x_3, \qquad \frac{x_1^2}{a^2} - \frac{x_2^2}{b^2} = x_3^2.$$

Ahhoz, hogy meghatározzuk e a görbék ideális pontjait, meg kell vizsgálnunk, hogy mely $[x_1, x_2, 0]$ számhármasok elégítik ki az egyenletüket. Az ellipszis esetében csak az $x_1 = x_2 = 0$ megoldást kapjuk, de mivel egy pont mindhárom koordinátája nem lehet zérus, az ellipszisnek nincs ideális pontja. A parabolára $x_2 = 0$ adódik, tehát egy ideális pontja van, az $[1, 0, 0]$, amely éppen a parabola tengelyével azonos

x-tengely ideális pontja. A hiperbola esetében az egyenletből az $x_1 : x_2 = \pm a/b$ arányt,vagyis az $[a, b, 0]$ és $[a, -b, 0]$ ideális pontokat kapjuk, amelyek a hiperbola aszimptótáinak ideális pontjai.

A háromdimenziós térben a homogén koordinátákat a következő megállapodással vezetjük be: a $P(x, y, z)$ pont homogén koordinátái olyan x_1, x_2, x_3, x_4 számok, amelyeknek nem mindegyike zérus, és amelyekre

$$x : y : z : 1 = x_1 : x_2 : x_3 : x_4$$

teljesül. A $\mathbf{v}(v_1, v_2, v_3)$ vektor iránya által meghatározott ideális pont homogén koordinátái olyan x_1, x_2, x_3, x_4 számok, amelyeknek nem mindegyike zérus, és amelyekre

$$v_1 : v_2 : v_3 : 0 = x_1 : x_2 : x_3 : x_4.$$

Az ideális pontokra tehát $x_4 = 0$. A közönséges pontok homogén koordinátáiból kiszámíthatjuk a *Descartes*-féle koordinátáit az

$$x = \frac{x_1}{x_4}, \quad y = \frac{x_2}{x_4}, \quad z = \frac{x_3}{x_4}$$

összefüggésekkel. Egy alakzat egyenletét homogén koordinátás egyenletre úgy írhatjuk át, hogy x, y, z helyére a fenti törteket helyettesítjük, majd az egyenletet x_4 megfelelő hatványával beszorozzuk. Ezáltal az alakzathoz ideális pontokat is csatolunk, amelyek nem függenek a koordináta-rendszer megválasztásától, de ennek igazolására itt nem térünk ki.

2. Térbeli alakzatok transzformációi

2.1. Lineáris transzformációk, ponttranszformációk

A háromdimenziós vektortérnek olyan A leképezéseiről lesz szó, amelyek minden $\mathbf{v}$ vektorhoz egy egyértelműen meghatározott $A\mathbf{v}$ „képvektort" rendelnek és lineárisak, vagyis két tetszőleges $\mathbf{u}$, $\mathbf{v}$ vektorra és tetszőleges c valós számra

$$A(\mathbf{u}+\mathbf{v}) = A(\mathbf{u}) + A(\mathbf{v}),$$
$$A(c\mathbf{u}) = cA(\mathbf{u})$$

teljesül.
Írjuk fel a lineáris leképezést egy $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ bázisban. Ha $\mathbf{v} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$, akkor A fenti tulajdonságai miatt

$$A\mathbf{v} = xA\mathbf{e}_1 + yA\mathbf{e}_2 + zA\mathbf{e}_3.$$

Legyen

$$A\mathbf{e}_i = a_{1i}\mathbf{e}_1 + a_{2i}\mathbf{e}_2 + a_{3i}\mathbf{e}_3, \qquad (i = 1,2,3)$$

hiszen a bázisvektorok képe is vektor, ezért felírhatók $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ lineáris kombinációjaként. Innen

$$A\mathbf{v} = x(a_{11}\mathbf{e}_1 + a_{21}\mathbf{e}_2 + a_{31}\mathbf{e}_3) + y(a_{12}\mathbf{e}_1 + a_{22}\mathbf{e}_2 + a_{32}\mathbf{e}_3) + z(a_{13}\mathbf{e}_1 + a_{23}\mathbf{e}_2 + a_{33}\mathbf{e}_3)$$

adódik. Rendezzük a jobb oldalt $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ szerint, az együtthatók éppen az $A\mathbf{v}$ vektor koordinátái lesznek

$$x' = a_{11}x + a_{12}y + a_{13}z$$
$$y' = a_{21}x + a_{22}y + a_{23}z$$
$$z' = a_{31}x + a_{32}y + a_{33}z$$

Ebből azt látjuk, hogy az A lineáris leképezésnek az adott $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ bázisban egy mátrix felel meg,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

A mátrix oszlopaiban a bázisvektorok képei állnak. Megfordítva az teljesül, hogy minden 3 × 3-as mátrix egy adott bázisban a háromdimenziós vektorok terén egy lineáris leképezés mátrixának tekinthető, de ennek az igazolására itt nem térünk ki. A lineáris leképezést és a neki megfelelő mátrixot ugyanazzal a betűvel jelöltük.

Az eddig elmondottakból következik, hogy egy lineáris leképezés mátrixát egy adott bázisban fel tudjuk írni, ha ismerjük a bázisvektorok képét. A mátrix ismeretében pedig bármely további vektor képe meghatározható.

Ha egy vektor képére újabb lineáris transzformációt alkalmazunk, vagyis $\mathbf{v}' = A\mathbf{v}$ és $\mathbf{v}'' = B\mathbf{v}'$, akkor a $\mathbf{v}'' = B(A\mathbf{v})$ összetett transzformáció mátrixa a BA szorzatmátrix. Ezt a tényt úgy is fogalmazhatjuk, hogy lineáris transzformációk szorzata ismét lineáris transzformáció, aminek a bizonyítására nem térünk ki.

Indítsuk ki a vektortér minden elemét az origóból. Így minden vektor az euklideszi tér egy pontjának a helyvektora. A $\mathbf{v} = \overrightarrow{OP}$ helyvektor $A\mathbf{v}$ képe egy P' pontnak a helyvektora, amelyet P képének tekintünk. Íly módon az A mátrixhoz az euklideszi tér egy ponttranszformációját rendeltük. A $P(x, y, z)$ pont képének koordinátái tehát a

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

képlettel adódnak. Minden ponttranszformációt viszont nem lehet ilyen alakban felírni. Az eltoláshoz pl. nem lehet 3 × 3-as mátrixot rendelni. Az összefüggések részletes tárgyalásához az affin tér és affin transzformációk fogalmát kellene bevezetni.

Invertálhatónak nevezzük a lineáris transzformációt, ha kölcsönösen egyértelmű megfeleltetést létesít a vektortér és a képtér elemei között. Ezzel ekvivalens, hogy az $A\mathbf{v} = \mathbf{0}$ egyenletből $\mathbf{v} = \mathbf{0}$ következik. Az invertálható transzformációt „nemelfajuló"-nak is mondják. Az ilyen transzformációnak van inverze, amelynek mátrixa A^{-1}. Az invertálható lineáris transzformáció mátrixának oszlopvektorai lineárisan független vektorok.

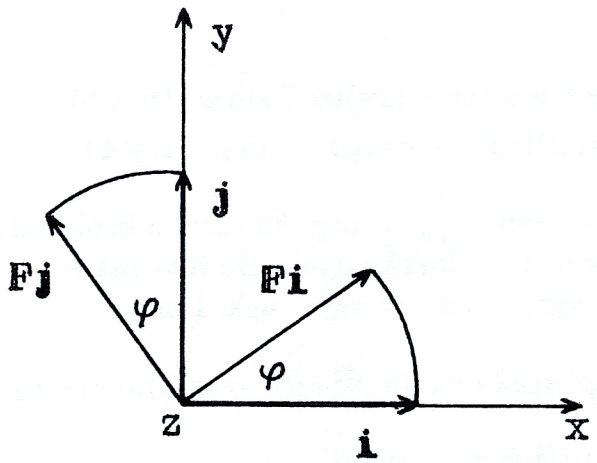
Ha egy lineáris transzformáció nem invertálható, akkor elfajulónak nevezzük. Az elfajuló transzformáció képterének dimenziója kisebb mint 3.

A továbbiakban áttekintjük a háromdimenziós euklideszi tér néhány, a gyakorlatban legtöbbször előforduló transzformációját. Figyeljük meg, hogy mindegyik transzformáció helyben hagyja az origót.

2.2. Egyenes körüli forgatás

Forgassuk el a tér pontjait a koordináta-rendszer z-tengelye körül pozitív irányban φ szöggel. Korábbi megállapításaink szerint a koordináta-egységvektorok elforgatottjának koordinátáit kell kiszámítanunk. A 2.1. ábrán látható, hogy

$$\begin{aligned} F\mathbf{i} &= \cos\varphi\mathbf{i} + \sin\varphi\mathbf{j} \\ F\mathbf{j} &= -\sin\varphi\mathbf{i} + \cos\varphi\mathbf{j} \\ F\mathbf{k} &= \mathbf{k} \end{aligned}$$

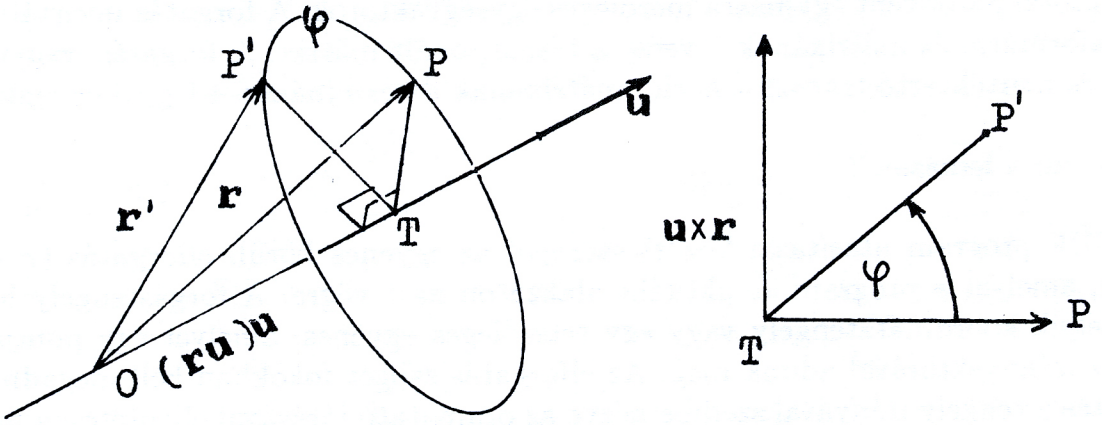


2.1. ábra. Forgatás a z-tengely körül

Ezért

$$F = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Következő példaként egy tetszőleges irányú, origón áthaladó egyenes körüli forgatás mátrixát írjuk fel. Az adott forgástengely irányvektora legyen az **u** egységvektor, egy tetszőleges adott pont pedig a P pont (2.2. ábra).



2.2. ábra. Pont egyenes körüli forgatása

Az $\mathbf{r}' = \overrightarrow{OP'}$ helyvektor az $\mathbf{r} = \overrightarrow{OP}$ helyvektornak a tengely körül φ szöggel való elforgatásával keletkezik. A P pont pályája a tengelyre merőleges $[TPP']$ síkban fekvő körív. Az OTP derékszögű háromszögben $\overrightarrow{OT} = (\mathbf{ru})\mathbf{u}$ és $\overrightarrow{TP} = \mathbf{r} - \overrightarrow{OT} = \mathbf{r} - (\mathbf{ru})\mathbf{u}$. Most először a $\overrightarrow{TP'}$ vektort számítjuk ki. Ehhez a $[TPP']$ síknak elkészítjük egy olyan vektorát, amely a $\overrightarrow{TP}$ vektorra merőleges és vele egyenlő hosszú. Ez éppen az $\mathbf{u} \times \overrightarrow{TP}$ lesz, mivel az **u** egységvektor és merőleges $\overrightarrow{TP}$-re. $\mathbf{u} \times \overrightarrow{TP} = \mathbf{u} \times (\mathbf{r} - (\mathbf{ru})\mathbf{u}) = \mathbf{u} \times \mathbf{r}$. A $\overrightarrow{TP}$ és $\mathbf{u} \times \mathbf{r}$ vektorokkal, valamint az adott φ szöggel kifejezve

$$\overrightarrow{TP'} = \cos\varphi \cdot (\mathbf{r} - (\mathbf{ru})\mathbf{u}) + \sin\varphi \cdot (\mathbf{u} \times \mathbf{r})$$

Most tekintsük az OTP' derékszögű háromszöget, amelynek két befogóvektorát már ismerjük, az átfogóvektor tehát

$$\begin{aligned}\mathbf{r}' = (\mathbf{ru})\mathbf{u} + \overrightarrow{TP'} &= (\mathbf{ru})\mathbf{u} + \cos\varphi \cdot (\mathbf{r} - (\mathbf{ru})\mathbf{u}) + \sin\varphi \cdot (\mathbf{u} \times \mathbf{r}) \\ &= (1 - \cos\varphi)(\mathbf{ru})\mathbf{u} + \cos\varphi\mathbf{r} + \sin\varphi \cdot (\mathbf{u} \times \mathbf{r})\end{aligned}$$

A forgatás U mátrixának az oszlopvektorait kapjuk meg, ha ezzel a képlettel az $\mathbf{i}, \mathbf{j}, \mathbf{k}$ bázisvektorok elforgatottjának koordinátáit kiszámítjuk. Ha $\mathbf{u} = u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$, akkor $(\mathbf{iu})\mathbf{u} = u_1\mathbf{u} = u_1^2\mathbf{i} + u_1u_2\mathbf{j} + u_1u_3\mathbf{k}$, $\mathbf{u} \times \mathbf{i} = u_3\mathbf{j} - u_2\mathbf{k}$. Innen

$$U\mathbf{i} = [(1-\cos\varphi)u_1^2+\cos\varphi]\mathbf{i}+[(1-\cos\varphi)u_1u_2+u_3\sin\varphi]\mathbf{j}+[(1-\cos\varphi)u_1u_3-u_2\sin\varphi]\mathbf{k}$$

adódik. Hasonlóképpen számítható ki $U\mathbf{j}$ és $U\mathbf{k}$. Tehát

$$U = \begin{bmatrix} (1-\cos\varphi)u_1^2 + \cos\varphi & (1-\cos\varphi)u_1u_2 - u_3\sin\varphi & (1-\cos\varphi)u_1u_3 + u_2\sin\varphi \\ (1-\cos\varphi)u_1u_2 + u_3\sin\varphi & (1-\cos\varphi)u_2^2 + \cos\varphi & (1-\cos\varphi)u_2u_3 - u_1\sin\varphi \\ (1-\cos\varphi)u_1u_3 - u_2\sin\varphi & (1-\cos\varphi)u_2u_3 + u_1\sin\varphi & (1-\cos\varphi)u_3^2 + \cos\varphi \end{bmatrix}$$

Megjegyezzük, hogy a forgatások mátrixa ún. ortogonális mátrix, amelynek oszlopvektorai páronként egymásra merőleges egységvektorok. A forgatás invertálható transzformáció és mátrixának inverze a transzponált mátrix. A forgatás irányítástartó és mértéktartó transzformáció, mátrixának determinánsa +1-gyel egyenlő.

Mi van a lemezen ?

A SURF program utasításai között szerepel az egyenes körüli elforgatás (r utasítás), amelyet a program az aktuális alakzaton hajt végre. A forgástengely lehet bármelyik koordinátatengely vagy egy tetszőleges egyenes, amelyet egy pontjával és egy irányvektorával adunk meg. Az elforgatás szögét fokokban kell megadni. A forgatás a tengely irányával szembe nézve az óramutató járásával ellentétesen megy végbe, ha az adott szög pozitív, és az óramutató járásával megegyező irányban, ha az adott szög negatív.

Gyakorlatok

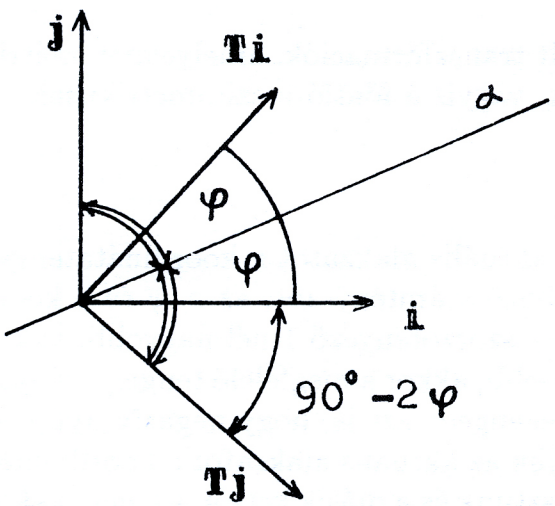
1. Álló henger vízszintes helyzetbe forgatása: Olvassuk be a lemezről az *m1* fileban található álló henger adatait, és adjuk ki az r utasítást. Válasszuk forgástengelynek az x-tengelyt, és a szög legyen 90°. Most a henger tengelye a negatív y-tengelyre került. Ha még egyszer elforgatjuk a z-tengely körül 90°-kal, akkor a henger a pozitív x-tengelyre kerül.
2. Fekvő alakzat felállítása: Olvassuk be a fekvő L-alakú hasábot az *m2* file-ból és forgassuk el az y-tengely körül 90°-kal. Vegyük észre, hogy y-tengelyre illeszkedő

éle helyben maradt. Ha most az újabb forgatásnál az elülső élét (a pont $(20,0,0,)$, irányvektor $(0,1,0)$) választjuk forgástengelynek, és $-90°$-kal visszaforgatjuk, akkor a hasáb az xy-sík alá kerül.

3. Alakzat hátsó oldalának előre forgatása: Nézzük meg az *m3* file-ban található ház hátsó falát a z- tengely körüli vagy egy azzal párhuzamos, tetszőleges egyenes körüli $180°$-os elforgatással.

2.3. Síkra vonatkozó tükrözés

Tükrözzük a tér pontjait egy olyan α síkra, amely a z-tengelyt tartalmazza, és az xz-koordinátasíkkal φ szöget alkot. Ha a transzformáció mátrixát T jelöli, $T\mathbf{k} = \mathbf{k}$ teljesül, mivel a z-tengely tükörképe önmaga (2.3. ábra).



2.3. ábra. Tükrözés az α síkra

Az ábrán tehát csak az xy-síkot és az adott α síknak az xy-síkkal alkotott metszésvonalát rajzoltuk fel, amely az x-tengellyel φ szöget alkot. Az ábrából könnyen leolvasható, hogy $T\mathbf{i} = \cos 2\varphi\mathbf{i} + \sin 2\varphi\mathbf{j}$ és $T\mathbf{j} = \sin 2\varphi\mathbf{i} - \cos 2\varphi\mathbf{j}$, ezért

$$T = \begin{bmatrix} \cos 2\varphi & \sin 2\varphi & 0 \\ \sin 2\varphi & -\cos 2\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Egyszerűbb példa az xy-koordinátasíkra vonatkozó tükrözés mátrixának felírása, mert az $\mathbf{i}$ és $\mathbf{j}$ bázisvektorok képe önmaga, a $\mathbf{k}$ tükörképe pedig $-\mathbf{k}$. A mátrix tehát

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

A síkra vonatkozó tükrözés ortogonális irányításváltó transzformáció, amely egy ortonormált jobbsodrású bázist ortonormált balsodrásúba visz át. A mátrix inverze a transzponáltja és determinánsának értéke -1.

Mi van a lemezen ?

A síkra vonatkozó tükrözés a SURF programban külön nincs definiálva, de a koordinátasíkokra lehet tükrözni az *s* (nyújtás) utasítással. Erre a 2.4. fejezetben kitérünk.

2.4. Nyújtás, zsugorítás a koordinátatengelyek irányában

Ezt a transzformációt úgy definiáljuk, hogy egy tetszőleges $P(x,y,z)$ pont koordinátáit rendre megszorozzuk a $q_1 > 0$, $q_2 > 0$ és $q_3 > 0$ nyújtási tényezőkkel. A transzformáció mátrixát megkapjuk, ha oszlopaiba a megnyújtott bázisvektorokat írjuk, azaz

$$N = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix}.$$

Ha $q_1 = q_2 = q_3 = q$, akkor speciális esetként az origó középpontú q arányú hasonlósági leképezésről van szó, amelynek mátrixa

$$H = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{bmatrix}.$$

A most felírt transzformációk ún. önadjungált transzformációk, amelyeknek mátrixai önmaguk transzponáltjával egyeznek meg, vagyis a főátlóra szimmetrikusak.

Mi van a lemezen ?

A SURF program *s* utasítása megnyújtja az aktuális alakzatot a koordinátatengelyek irányában. Ehhez három zérustól különböző számtényezőt, az *x*, *y* és *z* koordináták nyújtási szorzóit kell megadni. Ha egy szorzótényező 1-nél nagyobb, akkor valóban nyújtásról van szó, ha viszont 1-nél kisebb, akkor a megfelelő tengely irányában zsugorítás következik be. Az utasítás megengedi azt is, hogy negatív nyújtási tényezőt adjunk meg, ezáltal tehát tükrözhetjük az aktuális alakzatot a koordinátasíkokra. Ha ugyanis egy tényezőt -1-nek választunk és a másik kettőt $+1$-nek, akkor az alakzat minden pontjának ugyanazt a koordinátáját megszorozzuk -1-gyel, ami a megfelelő koordinátatengelyre merőleges koordinátasíkra vonatkozó tükrözést jelenti.

Gyakorlatok

1. Különböző mértékű nyújtás a koordinátatengelyek irányában:
   Olvassuk be az L-alakú hasábot az *m2* file-ból, és adjuk ki az *s* parancsot. Legyenek az *x*, *y*, *z* irányú tényezők rendre 1, 0.5 és 5. Most egy ötszörös magasságú L-hasábot kaptunk, amelynek a talphossza nem változott, és a szára felére zsugorodott.
2. Nagyítás-kicsinyítés:
   Készítsük el az 1.2. fejezet 1. gyakorlatában leírt házsort. Adjuk ki az *s* parancsot, és válasszuk a három nyújtási tényezőt egyaránt 0.4-nek. Ezzel az aktuális

alakzatot lekicsinyítettük, most négy vetülete belefér a képernyőbe (v utasítás). Az eredeti méretet az $1/0.4 = 2.5$ tényezőkkel való nagyítással állíthatjuk vissza.

3. Tükrözés az yz-koordinátasíkra:
   Adjuk ki az s parancsot vagy az L-alakú hasábra, vagy a házsorra, és válasszuk az x-irányú tényezőt -1-nek, a másik kettőt pedig 1-nek.

2.5. Merőleges vetítés

Az eddig felírt transzformációk mind invertálhatók voltak, a most következő transzformáció, amely a háromdimenziós teret egy síkra képezi le, azonban elfajuló. Ennek megfelelően mátrixának determinánsa zérus, és nincs inverze.

2.5.1. Vetítés a koordinátasíkokra

Vetítsük a tér pontjait az xy-koordinátasíkra. Mivel az $\mathbf{i}$ és $\mathbf{j}$ bázisvektorok vetülete önmaga, a $\mathbf{k}$ vetülete pedig a zérus vektor, a vetítés mátrixa a következő:

$$V_{xy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Hasonlóképpen írhatjuk fel az yz-koordinátasíkra való x-irányú vetítés

$$V_{yz} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

mátrixát és az xz-koordinátasíkra való y-irányú vetítés

$$V_{xz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

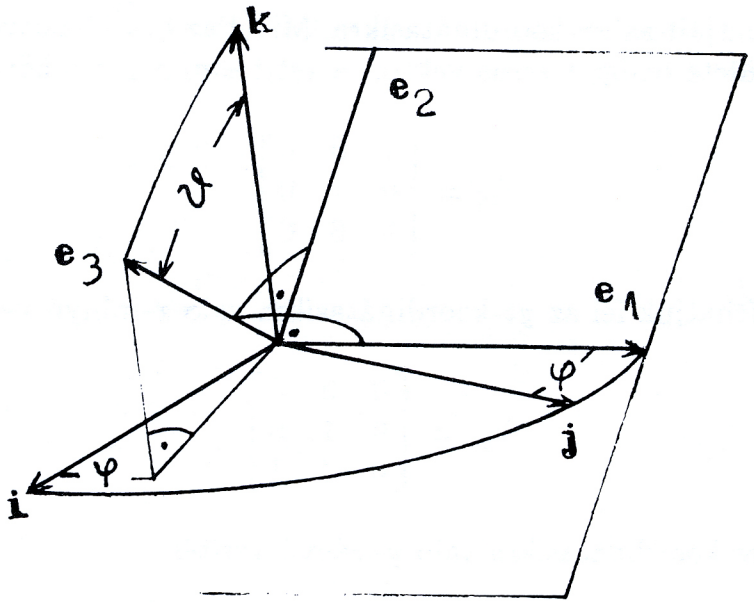
mátrixát.

Mi van a lemezen ?

A SURF program t utasítása az aktuális alakzat pontjait az xy- koordinátasíkra, f utasítása pedig az yz-koordinátasíkra vetíti.

2.5.2. Merőleges vetítés origón áthaladó általános helyzetű síkra

Ezt a feladatot visszavezetjük az xy-koordinátasíkra való merőleges vetítésre úgy, hogy először elforgatjuk a koordináta-rendszert olyan helyzetbe, hogy az xy-sík az adott általános helyzetű síkba kerüljön. Ebben a helyzetben végrehajtjuk a z-tengely irányából való vetítést, majd visszatérünk az eredeti koordináta-rendszerbe. Az általános helyzetű képsík állását a rá merőleges vetítési irány megadásával határozzuk meg. A vetítési irányt két szöggel adjuk meg: a z-tengellyel bezárt szöge legyen ϑ, az xy-síkra való merőleges vetületének az x-tengellyel bezárt szöge legyen φ. Ezek a szögek tulajdonképpen a vetítési irány egységvektorának polárkoordinátái. A vetítési iránnyal ellentétes irányú egységvektor lesz az $\mathbf{e}_3$ új bázisvektor. A 2.4. ábrából megállapítható, hogy $\mathbf{e}_3 = \sin\vartheta\cos\varphi\mathbf{i} + \sin\vartheta\sin\varphi\mathbf{j} + \cos\vartheta\mathbf{k}$.



2.4. ábra. Merőleges vetítés az $[\mathbf{e}_1, \mathbf{e}_2]$ síkra

Az $\mathbf{e}_3$-ra merőleges képsíkban az új koordináta-egységvektorokat szabadon választhatjuk meg. Legyen pl. $\mathbf{e}_1$ vízszintes, vagyis legyen benne $\mathbf{i}$ és $\mathbf{j}$ síkjában. Könnyen ellenőrizhető, hogy az $\mathbf{e}_1 = -\sin\varphi\mathbf{i} + \cos\varphi\mathbf{j}$ egységnyi hosszú és merőleges az $\mathbf{e}_3$-ra. Ezután a jobbsodrású ortonormált bázis $\mathbf{e}_2$ vektora $\mathbf{e}_3 \times \mathbf{e}_1$ vektoriális szorzással adódik, tehát $\mathbf{e}_2 = -\cos\vartheta\cos\varphi\mathbf{i} - \cos\vartheta\sin\varphi\mathbf{j} + \sin\vartheta\mathbf{k}$. A koordinátatranszformáció mátrixa tehát a következő:

$$A = \begin{bmatrix} -\sin\varphi & -\cos\vartheta\cos\varphi & \sin\vartheta\cos\varphi \\ \cos\varphi & -\cos\vartheta\sin\varphi & \sin\vartheta\sin\varphi \\ 0 & \sin\vartheta & \cos\vartheta \end{bmatrix}.$$

Mivel A ortogonális mátrix, $A^{-1} = A^*$.

Az $\mathbf{e}_1$ és $\mathbf{e}_2$ bázisvektorok síkjára való merőleges vetítés mátrixa az $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ bázisban

$$V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

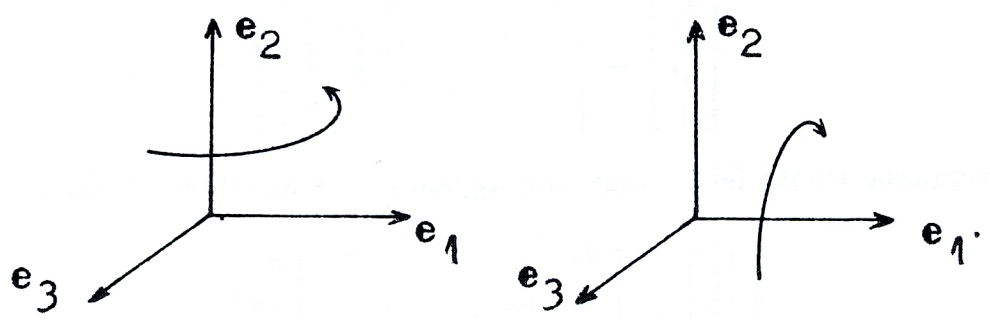
Most pedig határozzuk meg egy tetszőleges $P(x, y, z)$ pontnak az adott síkra való merőleges vetületét. P koordinátáit az $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ bázisban az

$$A^* \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

mátrixszorzással kapjuk (lásd az 1.2. fejezetben). Ezekre az új koordinátákra kell alkalmazni a V mátrixú vetítést, majd az A bázistranszformációval visszaszámolni a vetület koordinátáit az eredeti koordináta-rendszerbe. Ha P vetületét $P'(x', y', z')$ jelöli, akkor

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = A \cdot V \cdot A^* \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Példaként megjegyezzük, hogy az itt tárgyalt általános síkra való vetítést valósítjuk meg, amikor egy tárgyat „körüljárunk", és közben magasabbról vagy alacsonyabbról nézzük. Mivel szemléletünknek az felel meg, hogy eközben a függőleges függőleges marad, a képsík koordináta-rendszerének x-tengelyét, azaz az $\mathbf{e}_1$ bázisvektort vízszintesnek választottuk (2.5. ábra).



2.5. ábra. A vetítési irány megváltoztatása

A képsík koordináta-rendszerében, amelynek koordináta-egységvektorai $\mathbf{e}_1$ és $\mathbf{e}_2$, az $\mathbf{e}_3$ pedig az a vetítési irányt adja meg (a vetítési iránnyal szembe mutat), a tárgy körüljárása az $\mathbf{e}_2$ körüli φ szöggel való forgatást, a vetítési iránynak az $[\mathbf{e}_2, \mathbf{e}_3]$ síkban történő változtatása pedig $\mathbf{e}_1$ körüli ϑ szöggel való forgatást jelenti. A megfelelő mátrixokat felírva így is eljuthattunk volna az általános helyzetű síkra való vetítés mátrixához.

Mi van a lemezen ?

A SURF program *u* és *l* utasítása az aktuális vetítési irányt forgatja el a megadott szöggel fel és le, illetve jobbra és balra a képsík koordináta-rendszerében, pontosabban az *u* az e_1 egyenese körül, az *l* pedig az e_2 egyenese körül forgat (2.5. ábra). Ezután az új vetítési irányra merőleges képsík jelenik meg a képernyőn. Mivel a programban oldalnézet nem definiálható közvetlenül, azt az *l* parancs kiadásával 90°-os szögérték-választással állíthatjuk elő. Ha a parancs kiadása előtt elölnézetet (*f* utasítás) definiáltunk, akkor az így elkészített oldalnézet a szokásos helyzetben (vízszintesen) áll. A szemléletes képek előállításakor, ha élek vetülete egymásra esik, az *u* utasítással állítsuk a vetítési irányt néhány fokkal meredekebbre vagy laposabbra.

2.6. Ponttranszformációk felírása homogén koordinátákkal

A homogén koordináták segítségével olyan transzformációk mátrixát fogjuk felírni, amelyek egy, az origót helyben hagyó lineáris transzformáció és egy eltolás egymás utáni végrehajtásának felelnek meg. Ezeket affin transzformációknak nevezik. Végezzük el a háromdimenziós euklideszi térben a (d_x, d_y, d_z) vektorral való eltolást. A $P(x, y, z)$ pont képe ekkor az

$$x' = x + d_x, \quad y' = y + d_y, \quad z' = z + d_z$$

koordinátájú pont lesz. Homogén koordinátákkal ugyanez az eltolás a

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & & & d_x \\ & 1 & & d_y \\ & & 1 & d_z \\ & & & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

mátrixszorzással írható fel — vagy ami ugyanaz — a következő alakban

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix} = \begin{bmatrix} d_4 & & & d_1 \\ & d_4 & & d_2 \\ & & d_4 & d_3 \\ & & & d_4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix},$$

ahol

$$x : y : z : 1 = x_1 : x_2 : x_3 : x_4$$

és

$$d_x : d_y : d_z : 1 = d_1 : d_2 : d_3 : d_4.$$

A mátrix ki nem írt elemei nullák.

Az A transzformáció mátrixa homogén koordináta-rendszerben

$$\begin{bmatrix} & & & 0 \\ & kA & & 0 \\ & & & 0 \\ 0 & 0 & 0 & k \end{bmatrix}$$

alakú lesz, ezért az A transzformáció és az eltolás kompozíciója a

$$\begin{bmatrix} & & & d_1 \\ & d_4A & & d_2 \\ & & & d_3 \\ 0 & 0 & 0 & d_4 \end{bmatrix}$$

mátrixszal számítható ki.

A homogén koordináták használatának egyik előnye tehát, hogy az eltolás is mátrixszorzással számítható ki. Másik előnye az, hogy a numerikus számolás kiegyensúlyozottabbá tehető az adatok kedvezőtlen nagyságrendje esetén. Például ha 10^{-3} nagyságrendű pontkoordinátákkal forgatást számolunk, célszerű a forgatás mátrixát egy alkalmas konstanssal megszorozni, mert annak elemei is egynél kisebbek. Ezáltal elkerülhetjük azt, hogy az origóhoz közeli pontok elforgatottjai az origóba essenek.

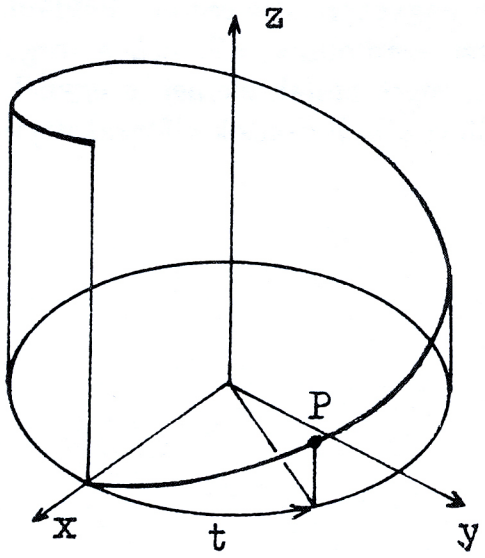
3. Felületek meghatározó adatai és előállítása

3.1. Csavarfelületek

Egy pont csavarmozgást végez, ha egy tengely körül állandó szögsebességgel forog, és ugyanakkor a tengely irányában állandó sebességgel halad. A pont pályagörbéje hengeres csavarvonal, amelynek tetszőleges pontja az

$$x(t) = a\cos t, \quad y(t) = a\sin t, \quad z(t) = ct, \quad (-\infty < t < +\infty)$$

koordinátafüggvényekkel adható meg (3.1. ábra). A t változó az elfordulás szöge, a a henger sugara, c pedig az 1 radián szögelforduláshoz tartozó menetemelkedés. A csavarvonal menetmagassága $c2\pi$. Csavarmozgást végző szakasz csavarfelületen mozog. A csavarfelületek alakja nagyon változatos aszerint, hogy a szakasz a csavarmozgás tengelyéhez viszonyítva milyen helyzetű. Ha a szakasz egyenese a tengelyre merőleges, akkor laposmenetű, egyébként élesmenetű csavarfelületen mozog. Ha a szakasz egyenese metszi a tengelyt, akkor zárt, egyébként nyitott csavarfelület keletkezik.



3.1. ábra. Hengeres csavarvonal

Mi van a lemezen ?

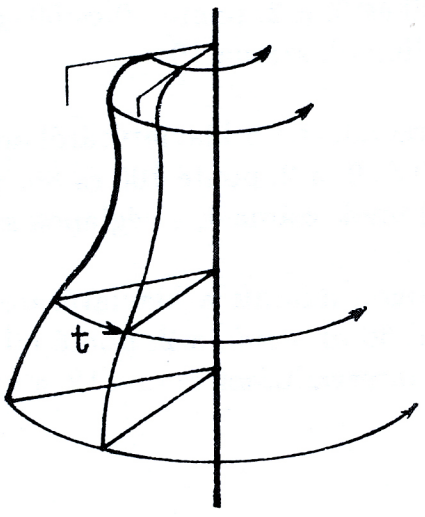
A CAD-iskola SURF programja a csavarfelületet két csavarvonal megfelelő (ugyanahhoz a t értékhez tartozó) pontjait összekötő szakaszok halmazaként hozza létre. A két csavarvonal az $x(t) = a\cos t$, $y(t) = b\sin t$, $z(t) = ct + d$ koordinátafüggvényekben szereplő a, b, c, d értékek megválasztásával egymástól függetlenül definiálható. Ahhoz azonban, hogy valóban csavarfelület keletkezzen, a c értékének mindkét görbére meg kell egyeznie (egyenlő menetemelkedés). Ha t kezdőértékét 0°-nak, végértékét pedig 360°-nak választjuk, pontosan egy menetet kapunk. Meg

kell adni, hogy a görbe a választott paramétertartományon hány pontjával legyen előállítva, ennyi alkotója lesz megrajzolva a definiált felületnek.

Gyakorlatok

1. Zárt lyukas, laposmenetű csavarfelület:
   Az első csavarvonal adatai: $a_1 = 50, b_1 = 50, c_1 = 10, d_1 = 0$. A második csavarvonal adatai: $a_2 = 90, b_2 = 90, c_2 = 10, d_2 = 0$. $t_1 = 0, t_2 = 540$, a pontok száma 25.
2. Zárt lyukas, élesmenetű csavarfelület:
   $a_1 = 50, b_1 = 50, c_1 = 10, d_1 = 0$. $a_2 = 90, b_2 = 90, c_2 = 10, d_2 = -30$. $t_1 = 0, t_2 = 540$, a pontok száma 25.
3. Zárt, élesmenetű csavarfelület:
   $a_1 = 0, b_1 = 0, c_1 = 10, d_1 = 0$. $a_2 = 70, b_2 = 70, c_2 = 10, d_2 = 30$. $t_1 = -30, t_2 = 450$, a pontok száma 21.
4. Egyéb vonalfelületek zérus menetemelkedésű csavarvonalakkal:
   Csonkakúp keletkezik, ha két párhuzamos síkú kört adunk meg, azaz $a_1 = b_1 = 70, c_1 = 0, d_1 = 0, a_2 = b_2 = 30, c_2 = 0, d_2 = 60$.
   Tubus alakú felület áll elő, ha két párhuzamos síkú ellipszist adunk meg úgy, hogy az egyiknek a kistengelye nullává zsugorodik. Az adatok lehetnek pl. a következők: $a_1 = 40, b_1 = 0, c_1 = 0, d_1 = 0, a_2 = 60, b_2 = 50, c_2 = 0, d_2 = 80$.
   A paraméter (szög) mindkét esetben 0-tól 360-ig változik és a pontok száma 25.

3.2. Forgásfelületek



3.2. ábra. Forgásfelület származtatása

Egy forgásfelület meghatározásához egy forgástengelyt és egy meridiángörbét kell megadni úgy, hogy a meridiángörbe és a tengely ugyanabban a síkban legyen, és a görbe ne metssze a tengelyt. Ha a meridiángörbe minden pontját a tengely körül egy meghatározott szöggel elforgatjuk, forgásfelületet kapunk. A teljes forgásfelület a t forgásszög $[0, 2\pi]$ intervallumához tartozik, ekkor a meridiángörbe minden pontja egy tengelyre merőleges síkú kört ír le (3.2. ábra).

Mi van a lemezen ?

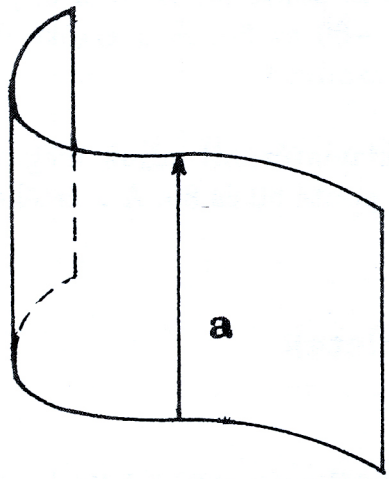
A SURF program a három koordinátatengely körül tud forgatni koordinátasíkban fekvő, véges sok pontjával adott meridiángörbét, vagyis meridiánpoligont. Ez lehet a CURVE programmal készített, file-ból név szerint beolvasott poligon vagy a klaviatúráról koordinátapárokkal megadott pontok sorozata. Ha megválasztottuk a forgástengelyt, akkor a program az egyik olyan koordinátasíkot választja meridiánsíknak, amely az adott tengelyt tartalmazza. Ez a z-tengely esetén az xz-sík, az x- és y-tengely esetén pedig az xy-sík. A meridiánpoligon nem metszheti a tengelyt, de a végpontjai rajta lehetnek a tengelyen. Meg kell adni a forgásszög kezdő- és végértékét fokokban és azt, hogy hány részintervallumból álljon a megadott szögtartomány. A program ennek a felosztásnak megfelelően számítja ki a meridiánpoligon pontjainak elforgatott helyzeteit, vagyis a körpályát szabályos sokszögnek tekinti. Ha a forgatás szöge 0°-tól 360°-ig változik, akkor a forgásfelülethez véglapokat is lehet csatolni.

Gyakorlatok

1. Tórusz:
   A forgástengely a z-tengely, a meridiánpoligon adatait file-ból olvassuk be, a file neve $c2$. A meridiánpoligon pontjaira az y értéke 0. A forgásszög 0°-tól 360°-ig változik, az intervallumok száma 12.
2. Váza:
   A forgástengely a z-tengely, a meridiánpoligon adatait file-ból olvassuk be, a file neve $c3$. Az y értéke 0. A forgásszög 0°-tól 360°-ig változik, az intervallumok száma 12, a véglapok száma 0.
3. Fél csonkakúp:
   A forgástengely a z-tengely, a meridiánpoligon adatait a klaviatúráról adjuk be, a pontok száma 2, az 1. pont koordinátái 100 és 0, a 2. ponté 50 és 50, $y = 0$, a forgásszög 0°-tól 180°-ig változik, az intervallumok száma 10.
4. Álló négyzetes hasáb:
   A forgástengely a z-tengely, a meridiánpoligon adatait a klaviatúráról adjuk be, a pontok száma 2, az 1. pont koordinátái 100 és 0, a 2. ponté 100 és 80, $y = 0$, a forgásszög 0°-tól 360°-ig változik, az intervallumok száma 4, a véglapok száma 2.
5. Fekvő henger:
   A forgástengely az y-tengely, a meridiánpoligon adatait a klaviatúráról adjuk be, a pontok száma 2, az 1. pont koordinátái 60 és -100, a 2. ponté 60 és 100, $z = 0$, a forgásszög 0° -tól 360°-ig változik, az intervallumok száma 12, a véglapok száma 0.

3.3. Általános hengerfelületek

Az általános hengerfelületet vezérgörbéje és alkotóvektora határozza meg. A vezérgörbe egy tetszőleges síkgörbe, és az alkotó nem párhuzamos a görbe síkjával (3.3. ábra).



3.3. ábra. **Általános hengerfelület**

Ezekből az adatokból a felületet úgy állítjuk elő, hogy a vezérgörbe minden pontján át az alkotóval párhuzamos egyenest húzunk. Így egy végtelen hengerfelületet kapunk. Ha a vezérgörbe sokszög, akkor a hengerfelület származtatásával egy végtelen hasáb palástjához jutunk. Ha a vezérgörbe kör és az alkotóvektor merőleges a kör síkjára, akkor egyenes körhengerfelületet kapunk. A szokásos értelemben ha hengerről beszélünk, nem a végtelen hengerfelületre gondolunk, hanem annak a két vezérgörbe közé eső darabjára. A lezáró véglapok azonban nem tartoznak a hengerfelülethez.

Mi van a lemezen?

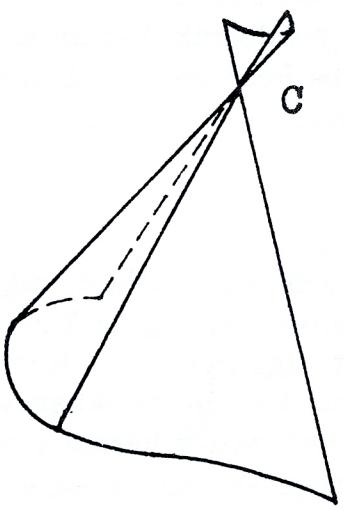
A SURF program által előállított általános hengerfelület vezérgörbéje vagy a CURVE programmal készített és file-ból név szerint beolvasott, vagy a klaviatúráról pontonként megadott poligon. A poligon lehet zárt vagy nyitott, és legalább két csúcsa van. Ezt a poligont a program a kiválasztott koordinátasíkban vagy azzal párhuzamosan helyezi el, és alapnak nevezi. Az alkotóvektort három koordinátájával kell megadni. A program ezzel a vektorral eltolja az adott poligont, és a két poligon közötti hasábpalástot (hengerszerű felületdarabot) generálja, amelynek annyi oldallapja van, ahány éle az adott poligonnak. Zárt alappoligon esetén véglapokat is lehet definiálni.

Gyakorlatok

1. Négyoldalú álló egyenes hasáb:
   Az alaplap az xy-síkban van, adatait a klaviatúráról adjuk meg, a poligonnak 5 csúcsa van, mert a négyszög zárásához az 1. pontot végpontként is meg kell adni. Az 1. pont koordinátái 0 és 0, a második ponté 50 és 0, a 3. ponté 50 és 100, a 4. ponté 0 és 100, az 5. ponté 0 és 0. A z értéke az összes ponthoz 0. Az alkotóvektor $(0, 0, 80)$. Vigyázat, ne írjunk zárójelet, és a koordinátákat szóközzel vagy [*enter*]-rel válasszuk el! A véglapok száma 2.

2. Vízszintes tengelyű körhenger:
   Az alapgörbe az yz-síkban van, az adatait file-ból olvassuk be, a file neve *c1*. Az x értéke -100, az alkotóvektor koordinátái $(200, 0, 0)$, a véglapok száma 0.
3. Ferde négyoldalú, felül nyitott hasáb:
   Az alaplap az xy-síkban van, az adatait a klaviatúráról adjuk meg, a sokszögnek 5 csúcsa van. Az 1. pont koordinátái -60 és 60, a 2. ponté 60 és 60, a 3. ponté 60 és -60, a 4. ponté -60 és -60, az 5. ponté -60 és 60. A z értéke 0, az alkotóvektor koordinátái $(10, 20, 100)$, a véglapok száma 1.
4. Vízszintes síkú téglalap:
   Az alaplap az xy-síkkal párhuzamos, adatait a klaviatúráról adjuk meg, és 2 csúcsa van. Az 1. pont koordinátái 60 és -80, a 2. ponté 60 és 80. A z értéke 30, az alkotóvektor $(-120, 0, 0)$.

3.4. Általános kúpfelületek



3.4. ábra. Általános kúpfelület

Az általános kúpfelületet vezérgörbéje és csúcspontja határozza meg. A vezérgörbe egy tetszőleges síkgörbe, és a csúcspont a görbe síkján kívül van. Ezekből az adatokból a felületet úgy állítjuk elő, hogy a vezérgörbe minden pontján át a csúcsponton áthaladó egyenest illesztünk. Így egy végtelen kettős kúpfelület keletkezik. Ha a vezérgörbe sokszög, akkor ez a felületszármaztatás gúlapalástot eredményez. Ha a vezérgörbe kör és a csúcspont a kör középpontján át a kör síkjára állított merőlegesen van, akkor forgáskúpfelületet (3.4. ábra).

Mi van a lemezen ?

A SURF program által előállított általános kúpfelület vezérgörbéje vagy a CURVE programmal készített és file-ból név szerint beolvasott, vagy a klaviatúráról pontonként megadott poligon. A poligon lehet zárt vagy nyitott, és legalább két csúcsa van. Ezt a poligont a program a kiválasztott koordinátasíkkal párhuzamosan helyezi el, és alapnak nevezi. A csúcspontot három koordinátájával kell megadni. A program az alappoligon élei és a csúcspont által meghatározott háromszöglapokból állítja elő a definiált gúlapalástnak (általános kúpfelületnek) a vezérgörbe és a csúcspont közé eső darabját. Zárt alappoligon alaplapként a gúlához rendelhető.

Gyakorlatok

1. Négyzetes gúla:
   Az alaplap az xy-síkban van, az adatait a klaviatúráról adjuk meg, a poligonnak 5 csúcsa van. Az 1. pont koordinátái -80 és 80, a 2. ponté 80 és 80, a 3. ponté 80 és -80, a 4. ponté -80 és -80, az 5. ponté ismét -80 és 80. A z értéke 0. A csúcspont koordinátái $(0, 0, 100)$. A véglapok száma 1.
2. Vízszintes tengelyű körkúp:
   Az alap párhuzamos az xz-síkkal, az adatait file-ból olvassuk be, a file neve *c1*. Az y értéke 120, a csúcspont koordinátái $(0, 0, 0)$. A véglapok száma 0.
3. Háromszöglemez ferde síkban:
   Az alaplap az xz-síkkal párhuzamos, az adatait a klaviatúráról adjuk meg, a csúcspontok száma 2. Az 1. pont koordinátái -120 és 30, a 2. ponté 120 és 30, az y értéke -40. A csúcs koordinátái $(20, 90, -30)$.

3.5. Összetett alakzatok szerkesztése

Az összetett alakzat szerkesztése abból áll, hogy a már definiált komponenseket egy előre megválasztott koordináta-rendszerben megfelelően pozicionáljuk, és az alakzathoz rendeljük. Az előre megválasztott koordináta-rendszer lesz az eredményül kapott összetett alakzat koordináta-rendszere. Ezt nevezzük globális koordináta-rendszernek. Minden komponensnek van saját koordináta-rendszere, amelyben definiáltuk. Ezt lokális koordináta-rendszernek nevezzük. Ha egy komponenst a globális koordináta-rendszerben forgatással és eltolással a megfelelő helyre teszünk, akkor lokális koordináta-rendszerének kezdőpontját és bázisvektorait kell megadnunk a globális koordináta-rendszerbeli koordinátáival. A szóban forgó komponensnek az alakzathoz való rendelése azt jelenti, hogy ki kell számítani a komponens pontjainak új, a globális koordináta-rendszerre vonatkozó koordinátáit, vagyis koordinátatranszformációt kell elvégezni. Ha a két koordináta-rendszer bázisvektorai megegyeznek, akkor a koordinátatranszformáció eltolás.

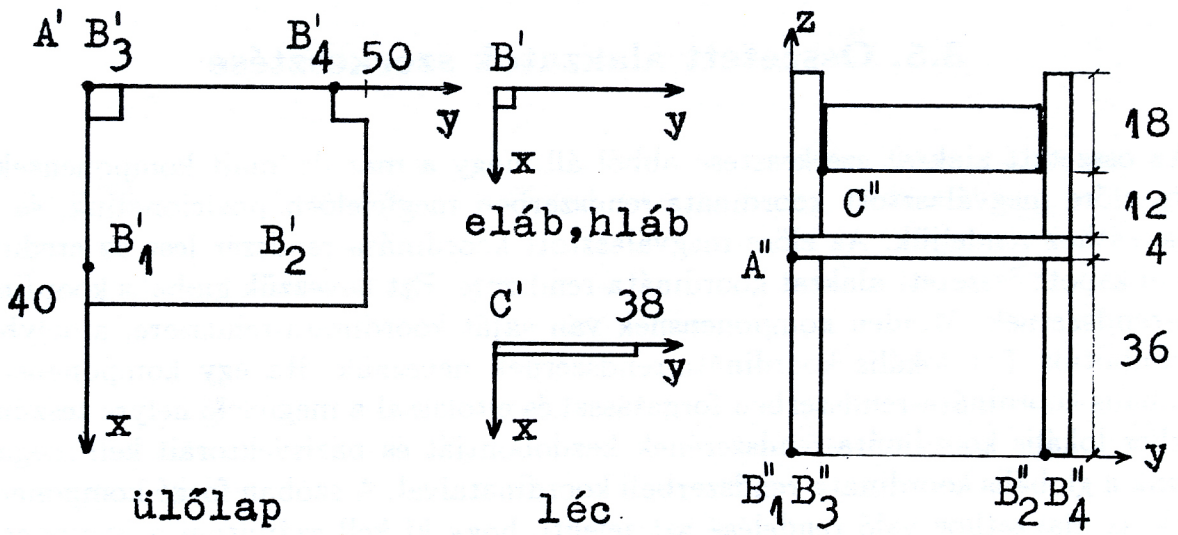
Mi van a lemezen ?

A már definiált felületekből való építkezés a SURF programban két utasítással (j és p) végezhető el. Mindkét esetben a komponensek pozicionálása csak eltolást jelent, tehát a lokális koordináta-rendszer kezdőpontjának koordinátáit kell megadni abban a globális rendszerben, amelyben építkezünk. Mindkét műveletnél azt a vetületet válasszuk ki előre, amelyben a két alakzat kölcsönös helyzete jól látszik. A j utasítással az aktuális alakzathoz kapcsolunk egy másikat, amelynek adatait fileból kell behívni. A művelet elvégzése után az összekapcsolt alakzat lesz az aktuális, amelyhez újabb komponenst kapcsolhatunk. A p utasításnál az építkezést elölről

kezdjük, vagyis az aktuális alakzat kezdetben az üres halmaz. Gondoljuk meg, hogy hogyan helyezkedjen el az összetett alakzat a globális koordináta-rendszerben, és ennek megfelelően adjuk meg a komponensek pozícióit, amelyeket egyesével fileból hívunk. Ennél a műveletnél a szerkesztett modellkonfiguráció elemeinek nevét és pozícióit a program egy *.mcf* típusú szövegfile-ba írja, amelyből a program elején található *r* utasítással építhetjük fel ismét az alakzatot. Eközben a pozíciókat korrigálhatjuk is.

Gyakorlatok

1. Szék szerkesztése:
   A szék komponensei az ülőlap, elülső láb (eláb), hátulsó láb (hláb) és keresztléc (léc) (3.5. ábra).

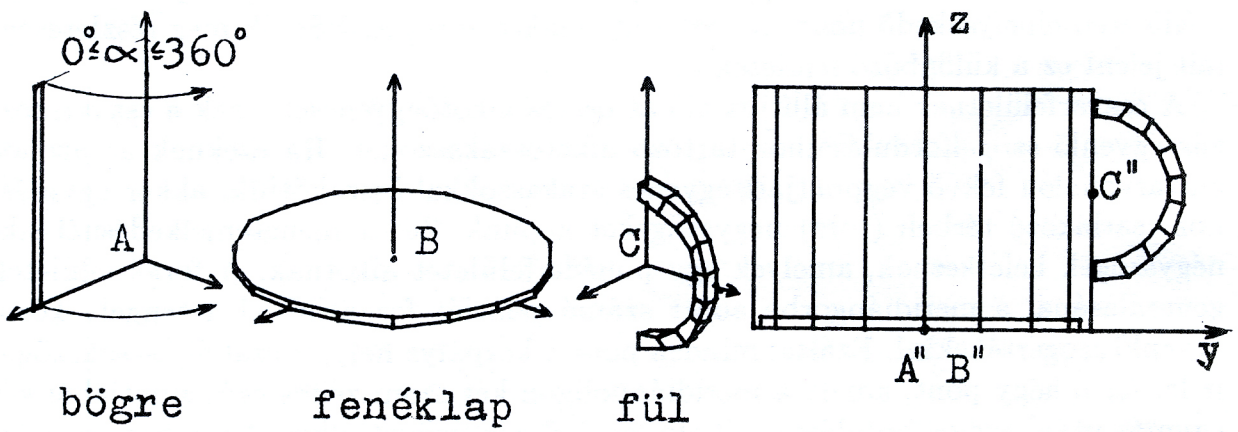


3.5. ábra. A szék komponensei

Az ábrán látható méretezés szerint definiáljuk a komponenseket. Mindegyik komponens hasáb (hengerszerű test), amelynek alapja az xy-síkban van. Az ülőlap alaplapját 9 csúcspontal kell megadni és alkotóvektora $(0, 0, 4)$, a véglapok száma 2. A lábak alapja ugyanaz az 5 × 5-ös négyzet, az eláb alkotóvektora $(0, 0, 36)$, a hlábé $(0, 0, 70)$, a véglapok száma 2. A keresztléc is hasáb, $(0, 0, 12)$ alkotóvektorral. A szék koordináta-rendszerének kezdőpontja legyen a bal hátsó láb bal hátsó alsó csúcspontjában. Az építkezéshez válasszuk ki a felülnézetet a *t* paranccsal. A *p* utasítással helyezzük el ezeket a testeket a globális koordináta-rendszerben! A konfigurációba a következő komponenseket és pozíciókat írjuk be: ülőlap $(0, 0, 36)$, eláb $(35, 0, 0)$, eláb $(35, 45, 0)$, hláb $(0, 0, 0)$, hláb $(0, 45, 0)$, léc $(1, 6, 52)$. Megjegyezzük, hogy ha a keresztléc alaplapját rögtön a $z = 52$

magasságban definiáljuk, akkor az összeépítésnél pozíciója $(1,6,0)$. Nézzük meg az eredményt négy vetületben a v utasítással, vagy váltsunk át axonometrikus vetítésre az a paranccsal, majd adjuk ki a h utasítást.

2. Füles bögre szerkesztése:
A modell komponensei forgásfelületek (3.6. ábra).



3.6. ábra. A füles bögre komponensei

A bögre és a fenéklap forgástengelye a z-tengely, a forgásszög 0°-tól 360°-ig változik és az intervallumok száma 18. A bögre meridiánpoligonjának csúcspontjai $(40,0)$, $(42,0)$, $(42,60)$, $(40,60)$, $(40,0)$ és a véglapok száma 0. A fenéklap meridiánpoligonja egy szakasz, amelynek csúcspontjai $(40,0)$ és $(40,2)$, a véglapok száma 2. A fül forgástengelye az x-tengely, a meridiánpoligon csúcspontjai $(2,18)$,$(2,22)$,$(-2,22)$,$(-2,18)$,$(2,18)$. A forgásszög $-90°$-tól 90°-ig változik, az intervallumok száma 9. A modellkonfigurációba (p utasítás) a következőt írjuk: bögre $(0,0,0)$, fenéklap $(0,0,0)$, fül $(0,42,35)$.

3. Metsző hengerfelületek:
Az első modell a 3.3. fejezet 2. számú gyakorlatában definiált fekvő henger. Olvassuk be ezt a modellt (i utasítás), majd csatoljuk hozzá a j utasítással a 3.2. fejezet 5. gyakorlatában leírt hengert, a pozíció $(1,1,1)$. Adjuk ki a h (hidden-line) utasítást (a: minden élre), amely előállítja a metszésvonalat is. Váltsuk át a vetítést felülnézetre (t utasítás), és ismét nézzük meg a metszésvonalat (h utasítás).

4. Gúla és háromszöglemez metszésvonala:
Olvassuk be a 3.4. fejezet 3. gyakorlatában elkészített háromszöglemezt, és a j utasítással csatoljuk hozzá a 3.4. fejezet 1. gyakorlatában leírt gúlát, a pozíció $(0,0,35)$. Adjuk ki a h parancsot minden élre. A metszésvonalnak a most nem látható szakaszát megnézhetjük, ha elforgatjuk a modellt (r utasítás) a z-tengely körül 40°-kal, és ismét kiadjuk a h parancsot.

4. A közelítő poliéderek adatrendszere

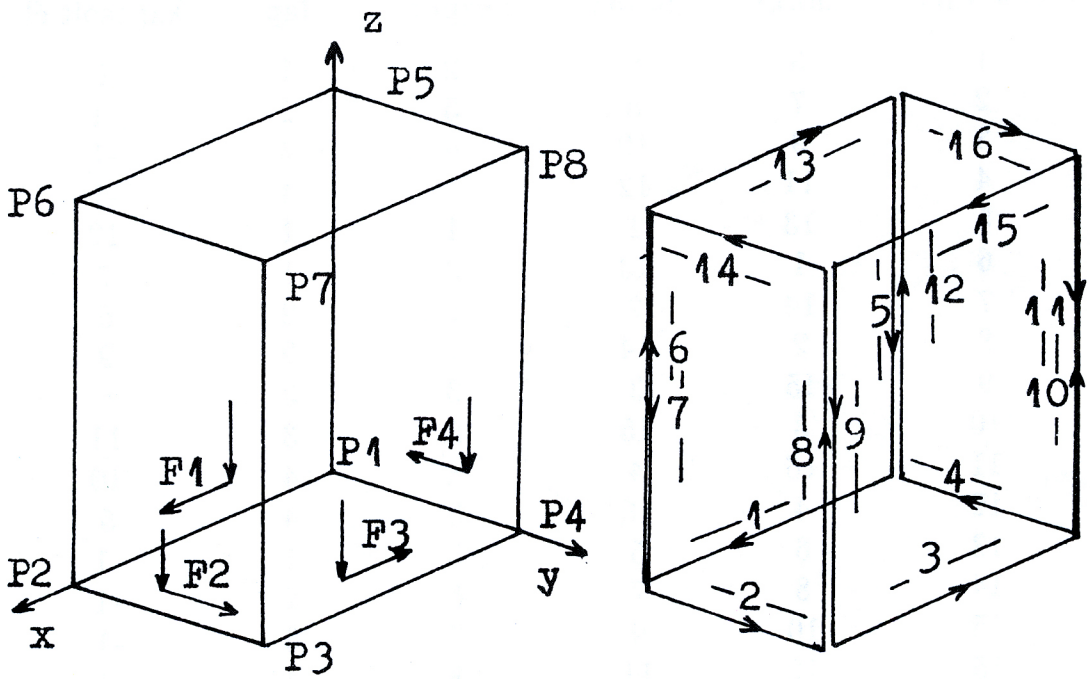
Ha a felületek származtatásánál a meghatározó görbéket (meridiángörbe, vezérgörbe) poligonnak tekintjük, és csak a poligon csúcsaira végezzük el a felületet előállító geometriai műveleteket, akkor nem kapjuk meg a felület minden pontját, csak egy, a felületen elhelyezkedő ponthalmazt vagy szakaszsereget. Nézzük meg részletesen, mit jelent ez a különböző felületekre.

A csavarfelületnek nem állítjuk elő az összes alkotóegyenesét, csak a csavarmozgás egyenlő szögelfordulásaihoz tartozó alkotószakaszokat. Ha ezeknek az azonos csavarvonalon fekvő végpontjait egyenes szakaszokkal összekötjük, akkor egymáshoz csatlakozó térbeli (torz) négyszögeket kapunk. Zérus menetemelkedésnél síknégyszögek keletkeznek, amelyek egy poliéderfelületet alkotnak. A forgásfelületek generálásánál a meridiángörbe adott számú pontját forgatjuk el a tengely körül egyenlő szögértékekkel. Ezáltal minden pont a körpálya helyett szabályos sokszöget ír le. Az a négy pont, amely a meridiánpoligon két szomszédos csúcspontjából két egymás utáni szögelfordulással áll elő, egy síknégyszöget alkot. Ha tehát az egész meridiánpoligonnak az adott szögtartomány egy egyenletes felosztásához tartozó szögértékekkel való elforgatott helyzeteit állítjuk elő, akkor a forgásfelületet egymáshoz csatlakozó síklapokból álló poliéderfelülettel helyettesítjük. Az általános hengerfelület előállításánál a vezérgörbét helyettesítő poligon csúcspontjaiba toljuk el az alkotóvektort, és az alkotóegyeneseknek csak ezen vektor által meghatározott szakaszát tekintjük a felülethez tartozónak. Két szomszédos alkotószakasz a megfelelő végpontjaikat összekötő poligonélekkel együtt egy paralelogrammát alkot. A felületet ezekkel az egymáshoz csatlakozó síklapokkal helyettesítjük. Hasonlóképpen közelítjük az általános kúpfelületnek a vezérgörbe és a csúcspont közé eső darabját egy poliéderfelülettel. Itt az alappoligon élei és a kúp csúcspontja határozzák meg az egymáshoz csatlakozó háromszöglapokat.

A felületeket helyettesítő poliéderfelületek annál jobban közelítik a felület alakját, mennél sűrűbb felosztást veszünk a felületet meghatározó görbéken, ill. szögtartományon. A felosztások sűrítésével, azaz a pont- és lapszám növelésével növekszik a felületet leíró adatmennyiség és a számítási idő is. A felület egyértelmű numerikus leírásához ugyanis a közelítő poliéderfelület minden csúcspontját, élét és lapját meg kell határozni.

Példaképpen felírjuk, hogyan adhatók meg egy négyoldalú hasáb palástjának csúcsai, élei és lapjai. Ez a felület általános hengerfelület, amelynek alapgörbéje egy négyszög, oldalélei pedig az alkotószakaszok.

A csúcsokat megszámozzuk, és mindegyik csúcspontnak felírjuk a koordinátáit (4.1. ábra). Legyen az x-, y- és z-irányú oldalélek hossza rendre a, b és c. Ekkor $P1(0,0,0)$, $P2(a,0,0)$, $P3(a,b,0)$, $P4(0,b,0)$, $P5(0,0,c)$, $P6(a,0,c)$, $P7(a,b,c)$, $P8(0,b,c)$ a csúcsokat meghatározó adatok.



4.1. ábra. Hasáb palástjának adatrendszere

A négy oldallap legyen $F1$, $F2$, $F3$, $F4$. Megszámozzuk az éleket is úgy, hogy minden lapot körbe tudjunk járni egy élláncal, kívülről nézve az óramutató járásával ellentétes irányban. Ehhez az éleket megfelelően irányítjuk, így a függőleges élek kétszer fognak szerepelni egymással ellentétesen irányítva. Minden élhez felírjuk, hogy az őt tartalmazó lapon, azaz élláncban melyik él van előtte és melyik van utána. Az él térbeli helyét pedig úgy határozzuk meg, hogy megadjuk a végpontját, vagyis azt, hogy hányas számú pontba fut be. Ennek megfelelően az 1. élhez tartozó adatok a következők: a láncban előtte van az 5. él, utána a 6., a végpontja a $P1$ csúcspont, és az él az 1. számú lapon van. A kétszeresen szereplő éleket azonosítjuk úgy, hogy egy további adatként felírjuk azt is, hogy a szóban forgó él hányas számú éllel egyezik meg. Az egyszeresen szereplő szélső élekhez -1 -et írunk. Ezáltal elkerüljük azt, hogy az oldaléleket kétszer rajzoljuk meg. Az így előállított adatrendszert kétszeresen kapcsolt élrendszernek nevezzük, és a következő táblázatban írjuk fel:

él száma	előtte	utána	végpont	lap	kapcsolt él
1	5	6	2	1	-1
2	7	8	3	2	-1
3	9	10	4	3	-1
4	11	12	1	4	-1
5	13	1	1	1	12
6	1	13	6	1	7
7	14	2	2	2	6
8	2	14	7	2	9
9	15	3	3	3	8
10	3	15	8	3	11
11	16	4	4	4	10
12	4	16	5	4	5
13	6	5	5	1	-1
14	8	7	6	2	-1
15	10	9	7	3	-1
16	12	11	8	4	-1

Ez az adatrendszer a felület egyértelmű leírásához szükséges adatoknál több adatot tartalmaz. Egy élláncon úgy is végig lehet menni, ha minden élhez pl. csak az utána következőt adjuk meg. Mégis érdemes az előtte álló élt is felvenni az adatok közé, mert így az él megrajzolásához szükséges kezdőpontja gyorsan megtalálható. Egy él kezdőpontja ugyanis a láncban előtte álló él végpontja. Ebből a táblázatból az egy lapon elhelyezkedő élek vagy az egy csúcspontra illeszkedő élek is könnyen összegyűjthetők.

Egy poliéderfelületet leíró adatrendszer természetesen sokféleképpen összeállítható, erre még a 6.2. fejezetben visszatérünk.

Mi van a lemezen ?

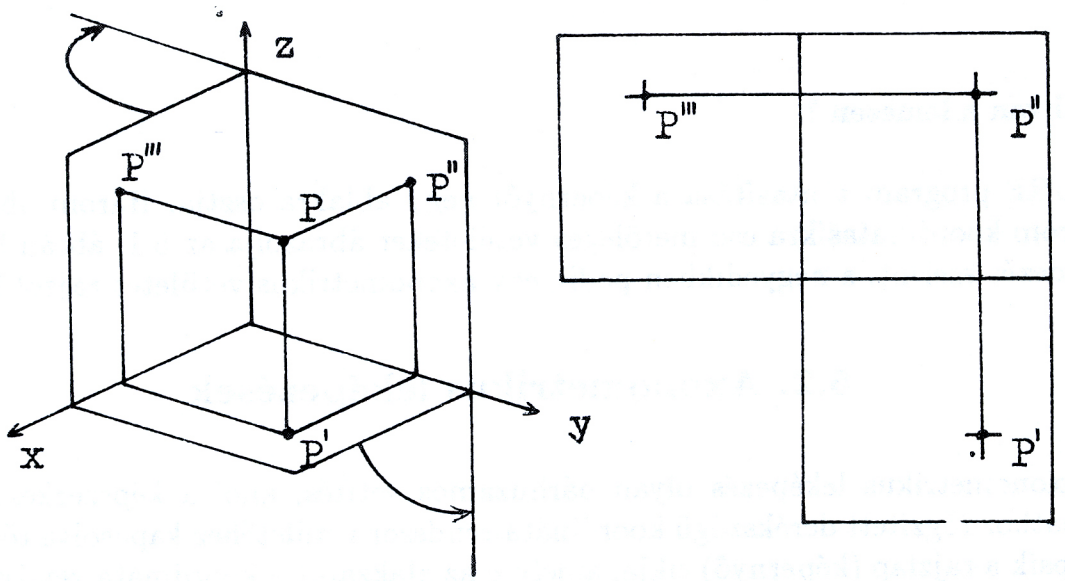
A CAD-iskola SURF programja a különböző típusú felületekhez az itt leírt kétszeresen kapcsolt élrendszert állítja elő. Azokat az éleket, amelyek csak egyszeresen szerepelnek, vagyis a kapcsolt él száma -1, a hidden-line algoritmus (*h* utasítás) határoló (boundary) éleknek nevezi. Határoló élek még azok is, amelyekben egy látható és egy nem látható lap (lásd a 6.4. fejezetben) csatlakozik. Ha tehát egy hengerfelület ábrázolásánál nem akarjuk látni az adatrendszerben szereplő összes élt, adjuk ki a *h* utasítást, és ott csak a határoló éleket rajzoltassuk fel. Ekkor az alkotók közül csak a két szélső alkotó fog megjelenni.

5. A megjelenítés módszerei

5.1. Merőleges nézetek egymásra merőleges képsíkokon

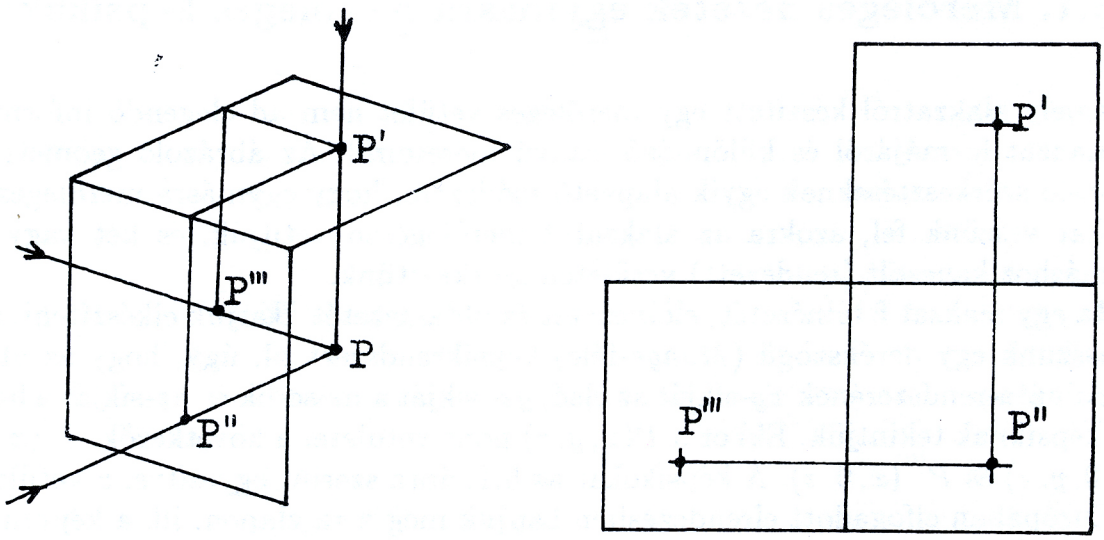
Valamely alakzatról készített egy merőleges vetület nem ad elegendő információt az alakzat formájáról és különböző irányú méreteiről. Az ábrázoló geometriai és műszaki szerkesztéseknek egyik alapvető módszere, hogy egymásra merőleges képsíkokat veszünk fel, azokra az alakzatot merőlegesen vetítjük, és két vagy több egymáshoz kapcsolt (rendezett) vetületen szerkesztünk.

Ha egy alakzat felülnézetét, elölnézetét és oldalnézetét akarjuk elkészíteni, akkor válasszunk egy derékszögű (*Monge*-féle) képsíkrendszert pl. úgy, hogy az alakzat koordináta-rendszerének xy-síkját az első, yz- síkját a második és xz-síkját a harmadik képsíknak tekintjük. Ekkor a $P(x, y, z)$ pont vetületei a következők: $P'(x, y, 0)$, $P''(0, y, z)$ és $P'''(x, 0, z)$. A képsíkokat az 5.1. ábra szerint egyesítve, a vetületeket az Európában elfogadott elrendezésben kapjuk meg a rajzlapon, ill. a képernyőn.



5.1. ábra. Monge-féle derékszögű képsíkrendszer

Az amerikai norma szerint a képsíkokat átlátszónak tekintve készítjük a vetületeket, és a második képsíkba fordítjuk be az elsőt és a harmadikat. A felülnézet tehát az elölnézet felett van, és a balról készített oldalnézet alul (5.2. ábra).



5.2. ábra. Merőleges vetületek elrendezése az amerikai norma szerint

Mi van a lemezen ?

A SURF program *v* utasítása a képernyőt négy ablakra osztja. Három ablakban a három koordinátasíkra eső merőleges vetületeket ábrázolja az 5.1. ábrán látható elrendezés szerint, a negyedikben pedig egy axonometrikus vetületet rajzol fel.

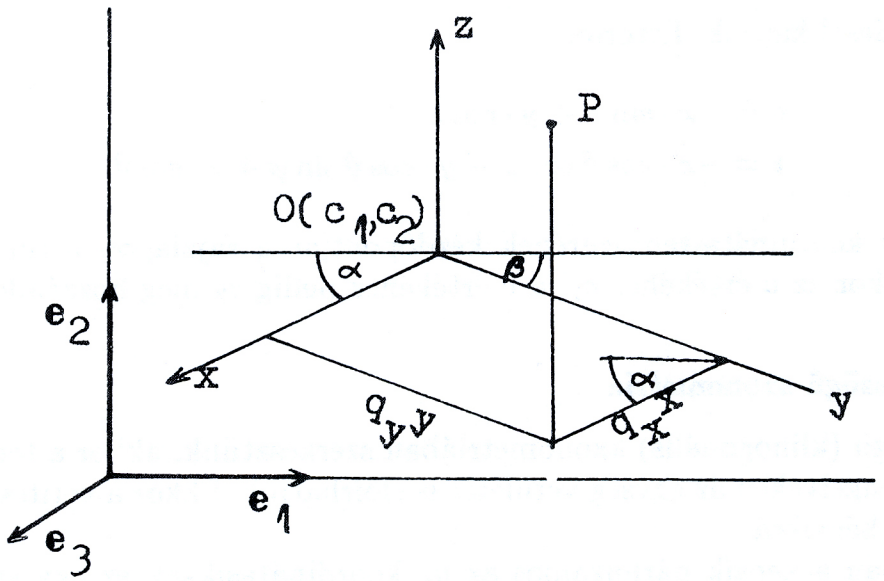
5.2. Axonometrikus leképezések

Az axonometrikus leképezés olyan párhuzamos vetítés, ahol a képszerkesztés az alakzathoz rögzített derékszögű koordináta-rendszer vetületéhez kapcsolva történik. A képsík a rajzlap (képernyő) síkja, amelyre az alakzatot a koordináta-rendszerével együtt vagy merőlegesen, vagy ferdén, de egymással párhuzamos vetítősugarakkal vetítjük. A képet a vetítősugarak képsíkkal alkotott döféspontjai alkotják. A műszaki szerkesztésekben az axonometrikus vetítést a koordinátatengelyek vetületének felrajzolásával szokás megadni. Ezzel határozzuk meg tulajdonképpen az alakzat koordináta-rendszere (az irodalomban előfordul a világkoordináta-rendszer elnevezés is) és a képsík koordináta-rendszere közötti kapcsolatot.

5.2.1. Ortogonális axonometria

Ortogonális axonometrikus vetítés esetén a koordinátatengelyek vetületét úgy kell megadnunk, hogy egymással páronként tompa szöget zárjanak be. Ezek a vetületek azzal a kikötéssel együtt, hogy a vetítés iránya a képsíkra, azaz a rajz síkjára merőleges, meghatározzák a tengelyek egységszakaszainak q_x, q_y, q_z vetületeit, amelyeket tengelyirányú rövidüléseknek nevezünk.

Példánkban az x-tengely vetülete a vízszintessel α szöget, az y-tengely vetülete pedig β szöget alkot, ahol α, β és $\alpha + \beta$ is hegyesszögek. A z-tengely vetülete függőleges (5.3. ábra).



5.3. ábra. Pont ortogonális axonometrikus vetülete

A vetítés numerikus elvégzése azt jelenti, hogy kiszámítjuk a pont vetületének koordinátáit a rajzlaphoz rendelt koordináta-rendszerben. A $P(x, y, z)$ pont vetületének az e_1, e_2 bázisra vonatkozó (u, v) koordinátái

$$u = -q_x \cdot x \cos\alpha + q_y \cdot y \cos\beta + c_1$$
$$v = q_z z - q_x \cdot x \sin\alpha - q_y \cdot y \sin\beta + c_2,$$

ahol a q_x, q_y, q_z rövidülések értékei rendre a következők:

$$q_x = \sqrt{1 + \cot^2\alpha} \;\Big/\; \sqrt{\cot^2\alpha + \cot\alpha\cot\beta}$$

$$q_y = 1 \;\Big/\; \cos\beta\sqrt{1 + \cot\alpha\tan\beta}$$

$$q_z = \sqrt{\cot\alpha\cot\beta - 1} \;\Big/\; \sqrt{\cot\alpha \quad \cot\beta}$$

Ezt a számítást nem részletezzük. A vetítőegyenesek az $\mathbf{e}_3$ bázisvektorral párhuzamosak.

A CAD-rendszerekben az ortogonális axonometrikus leképezés meghatározásának gyakori módja, hogy az alakzat koordináta-rendszerében adjuk meg a vetítési irányt, amely meghatározza a rá merőleges képsíkot. Ahogy azt a 2.5.2. fejezetben leírtuk, a vetítési irányt (az $\mathbf{e}_3$ bázisvektort) a z-tengellyel alkotott ϑ szöggel és az xy-síkra eső merőleges vetületének az x-tengellyel bezárt φ szögével határozzuk meg. A $P(x,y,z)$ pontnak a $P(u,v)$ vetületét a képsíkhoz rendelt $\mathbf{e}_1, \mathbf{e}_2$ bázisra vonatkozóan a

$$V \cdot A^* \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

mátrixszorzással kapjuk. Eszerint

$$u = -x \cdot \sin\varphi + y \cdot \cos\varphi$$
$$v = -x \cdot \cos\vartheta \cos\varphi - y \cdot \cos\vartheta \sin\varphi + z \cdot \sin\vartheta.$$

Ha a képsík koordináta-rendszerének kezdőpontját a rajzlapon a (c_1, c_2) pontba toljuk el, akkor az u értékéhez c_1, a v értékéhez pedig c_2 még hozzáadódik.

5.2.2. Ferdeszögű axonometria

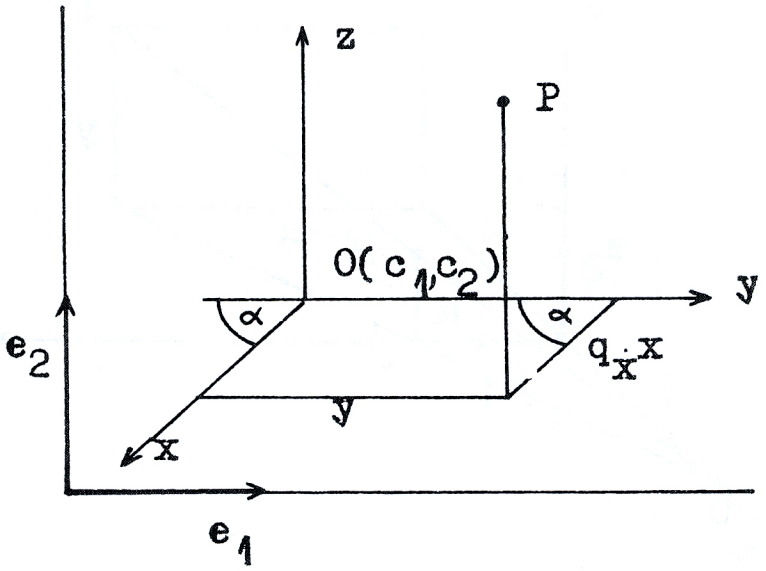
Ha ferdeszögű (klinogonális) axonometriában szerkesztünk, akkor a tengelykereszt képét és a tengelyeken az egység vetületét is előírhatjuk. Ekkor a vetítés iránya nem merőleges a képsíkra.

Példánkban a képsík párhuzamos az yz koordinátasíkkal, ez egy speciális, ún. frontális axonometria. Tehát az y-tengely vetülete vízszintes, a z- tengely vetülete függőleges, az x-tengely vetülete pedig a vízszintessel α szöget zár be. A tengelyirányú rövidülések q_x tetszőleges, $q_y = q_z = 1$ (5.4. ábra).

Az ábrán P jelöli a $P(x,y,z)$ pont axonometrikus vetületét, amelynek az $\mathbf{e}_1$, $\mathbf{e}_2$ bázisra vonatkozó koordinátái u és v. Az ábrából leolvasható, hogy

$$u = y - q_x \cdot x \cos\alpha + c_1$$
$$v = z - q_x \cdot x \sin\alpha + c_2.$$

A vetítési irány ismeretére akkor van szükség, ha a vetületben láthatósági viszonyokat szeretnénk eldönteni. Legyen az $\mathbf{e}_3$ bázisvektor a rajzlap síkjára merőleges, felfelé mutató egységvektor. A rajzlaphoz rendelt bázisban a vetítőegyeneseknek a rajzlap felé mutató irányvektora a $-q_x \cdot \cos\alpha \mathbf{e}_1 - q_x \cdot \sin\alpha \mathbf{e}_2 - \mathbf{e}_3$ vektor.



5.4. ábra. Pont ferdeszögű axonometrikus vetülete

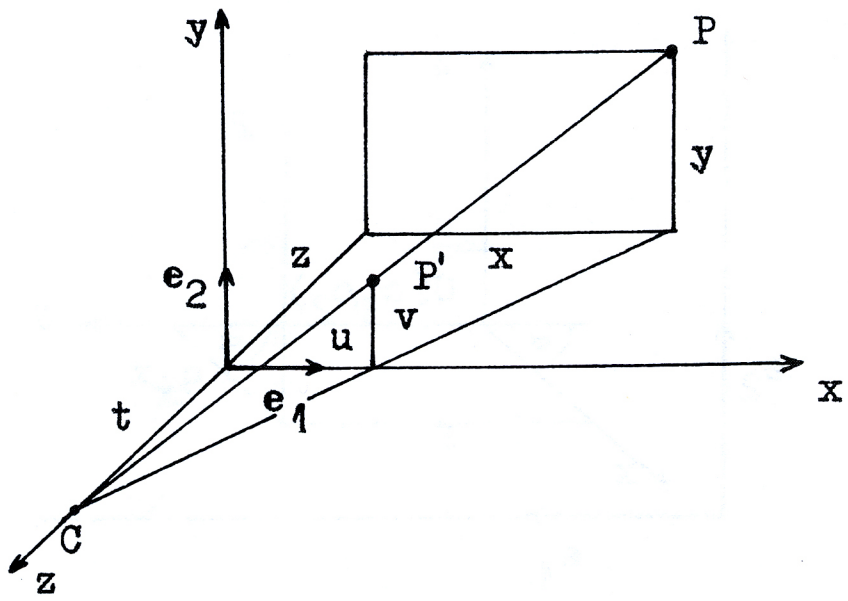
A gyakorlatban több egyéb ferdeszögű axonometrikus leképezés is elterjedt. A vetített pont koordinátái más tengelyválasztás esetén is egyszerűen felírhatók. Mivel ez a vetület kevesebb számítást igényel, mint az ortogonális axonometrikus vetület, gyors szemléltetés céljából gyakran használják.

Mi van a lemezen ?

A SURF program előre beállított vetítési módja ortogonális axonometrikus vetítés. Ugyanezt definiálja az *a* utasítás.

5.3. Centrális projekció

A centrális projekciónál (perspektív leképezésnek is nevezik) abból a helyzetből indulunk ki, hogy a képsík a koordináta-rendszer xy-síkja, a vetítési középpont pedig, amelyen mindegyik vetítőegyenes áthalad, a z-tengelyen helyezkedik el a képsíktól t távolságra. Ezt a helyzetet elérhetjük általános helyzetű képsík esetén, ha a 2.5.2. pontban leírtak szerint egy olyan bázistranszformációt hajtunk végre, amely által az új koordináta-rendszer xy-síkja a képsíkra kerül. Ezután szükség lehet még egy eltolásra, ha a centrum nincs rajta a z-tengelyen. Az 5.5. ábrán már ezt a képsíkhoz rendelt koordináta-rendszert rajzoltuk fel.



5.5. ábra. Pont centrális vetülete

A $P(x, y, z)$ pont centrális leképezéssel kapott vetületét P' jelöli. P'-nek az $\mathbf{e}_1, \mathbf{e}_2$ bázisra vonatkozó (u, v) koordinátáit hasonló háromszögek segítségével számítjuk ki, amely szerint

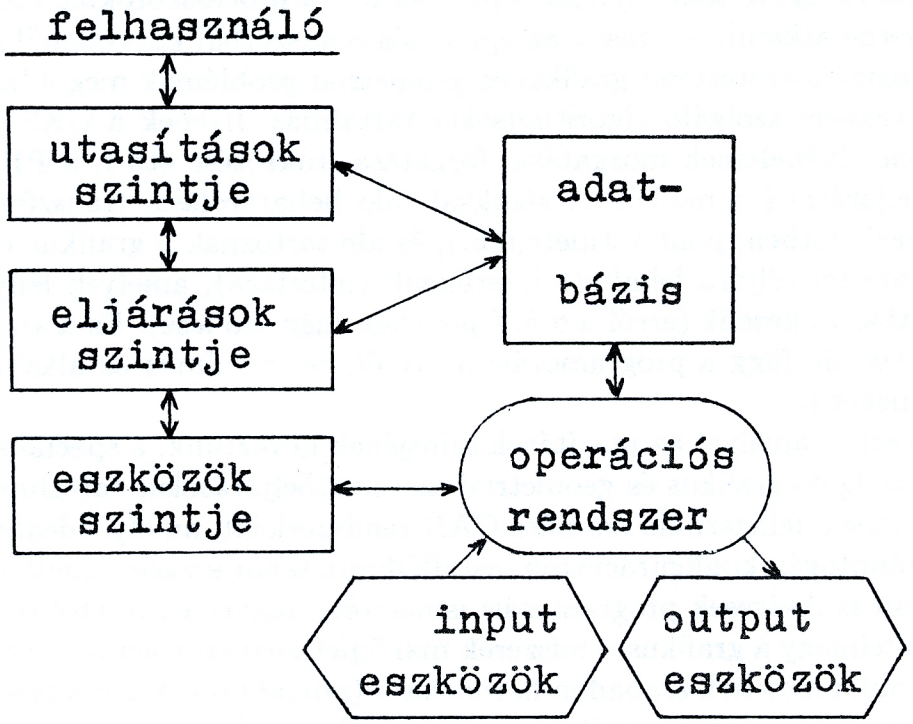
$$u = \frac{t}{t - z}x, \qquad v = \frac{t}{t - z}y.$$

A vetítés irányát minden P pontra a CP egyenes irányvektora adja meg.

6. A CAD-rendszerek felépítése és típusai

Annak érdekében, hogy a mérnöki tervezés különböző műveleteit számítógéppel végezhessük el, a művelet tárgyát olyan rendezett adathalmazzal kell leírni, amely a számítógép számára közvetlenül érthető. Az ilyen módon definiált alakzatot számítógépi modellnek, a meghatározó adathalmazt adatrendszernek vagy adatstruktúrának, a modell előállítását pedig modellezésnek nevezik. Mi csak a modell alakjára vonatkozó geometriai információkat tartalmazó geometriai modellekkel foglalkozunk, műszaki alkalmazásokra a 7. fejezetben térünk ki.

A CAD-rendszerek (Computer Aided Design) a számítógépi geometriai modellezésre szolgáló grafikus rendszerek, amelyek tartalmazzák a számítógéphez csatolt grafikus eszközök működtetéséhez szükséges eljárásokat, a felhasználó által tervezett geometriai alakzatok numerikus leírására és a geometriai manipulációk elvégzésére szolgáló algoritmusokat, az alakzatokról készített képek előállítását és feldolgozását végző utasításokat, továbbá más grafikus vagy gyártó rendszerek számára adatbázisokat készítő eljárásokat (interface).



6.1. ábra. A grafikus rendszer szoftverrétegei

Bármennyire is sokfélék a grafikus rendszerek, szerkezetükben három egymásra épülő réteget különböztetünk meg: eszközszint (device level), az eljárások szintje (procedural level) és az utasítások szintje (functional level) (6.1. ábra).

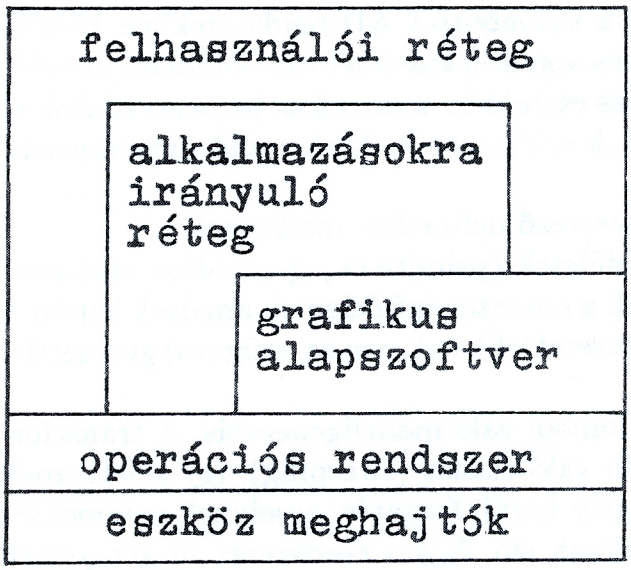
Az eszközszint tartalmazza az input és output eszközöket aktivizáló (driver) és az azokhoz való adatátvitelt végző eljárásokat. Ilyenek az eszközparaméterek standard és aktuális értékeinek lekérdezését végző input eljárások (a képernyő felbontása, a kurzor helye, a lenyomott billentyű stb.), illetve az eszközök állapotának beállítását, mozgatását végző output eljárások (a képernyő és a képpont színe, törlése, a rajzgép tollának kiválasztása és elmozdítása, a kurzor alakja és mozgatása stb.). Tartalmaz továbbá egyszerű rajzoló eljárásokat (karakterek megjelenítése, szakasz és körív megrajzolása, sokszögtartomány kitöltése stb.) és képelemek attributumainak (szín, vonaltípus stb.) beállítását végző eljárásokat. Az eszközszinthez tartozó grafikus eljárásokat tartalmazó programcsomagok adott számítógép-konfigurációkra és programozási nyelvekre készülnek, mint pl. a VDI (Virtual Device Interface) szubrutincsomag IBM-kompatibilis PC-re FORTRAN és PASCAL nyelven. A legelterjedtebb grafikus alapszoftverek a GKS (Graphical Kernel System) és a PHIGS (Programmer's Hierarchical Interactive Graphics System), amelyeket nemzetközi szabványok szerint fejlesztettek ki. Ide tartozik még a felhasználó és a rendszer közötti párbeszéd lebonyolításának céljára fejlesztett ablak- és menükezelő eljárásokat tartalmazó GEM (Graphics Environment Manager) és az X-Windows. A számítógépek fejlődésével egyre több eljárást „építenek be" a processzorokba, ezek a szoftverből hardverbe átkerült eljárások az operációs rendszerből közvetlenül elérhetők.

Az eljárásszint összetettebb grafikai és geometriai problémák megoldására és az adatbázis kezelésére szolgáló algoritmusokat tartalmaz. Ilyenek a GKS szegmenstranszformációi (képelemek mozgatása, forgatása, tükrözése stb.), a PHIGS vágó és leképező eljárásai (a modell-tér síkokkal való behatárolása, transzformációk a háromdimenziós térben, pont vetülete stb.), és ide tartoznak a grafikus rendszerek közötti adatátvitel céljára készített interfészek (interface), amelyek lényegében a grafikus adatbázist kezelik (erről a 6.5. fejezetben még szó lesz). Ez a szoftverréteg eszközfüggetlen, de függ a programozási nyelvtől, és tartalmaz az alkalmazásokra irányuló elemeket is.

A legfelső szint, amelyet az utasítások szintjének neveztünk, a speciális alkalmazási célokra szolgáló grafikus és geometriai szerkesztőeljárásokat tartalmazza. Ez a réteg jelenik meg a felhasználó számára CAD-rendszerként, amely (ideális esetben) különböző számítógép-konfigurációkon installálható, tehát eszközfüggetlen; és használatához nem szükségesek programozási ismeretek, vagyis a nyelvtől is független. Ez a két követelmény a grafikus rendszerek mai fejlettségi szintjén még nem teljesül maradéktalanul, nincsenek szabadon hordozható (portable) CAD-rendszerek, de az alsó két szoftverréteg szabványosítása e cél irányába mutat.

A rétegeknek egymáshoz és az operációs rendszerhez való viszonyáról alkotott másik elképzelést szemléltet a 6.2. ábra. Figyeljük meg, hogy mindhárom rétegnek van érintkezési felülete az operációs rendszerrel és egymással is, ami jól mutatja a hordozhatósági problémákat.

Mi van a lemezen ?

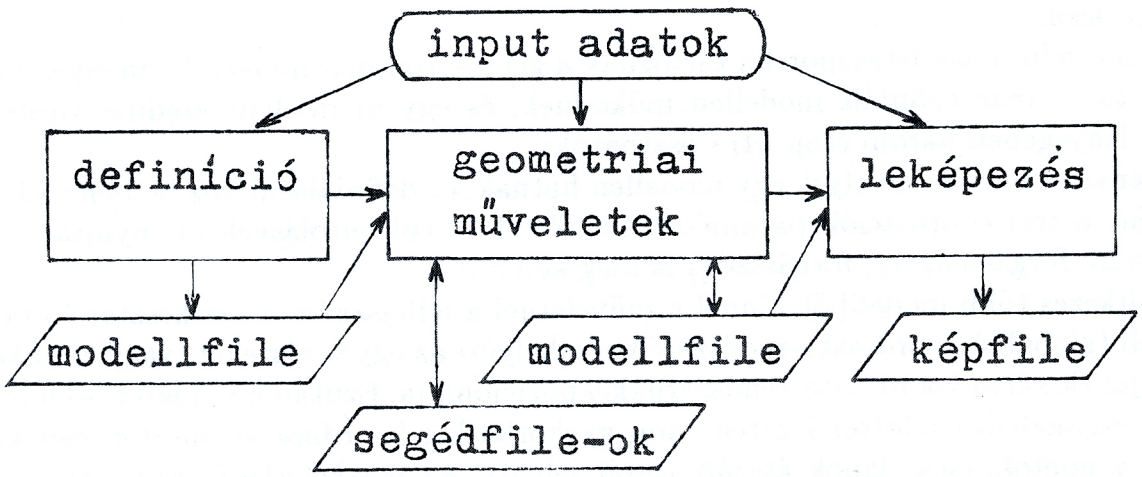


6.2. ábra. Szoftverrétegek a CAD-rendszerben

A CAD-iskola programjai TURBO PASCAL nyelven készültek, amely a Graph Unit-ban saját grafikus eljárásokat tartalmaz. Ezek gyakorlatilag mind az eszközszinthez tartoznak. A program használja a szakaszt rajzoló *line*, a szöveget kiíró *OutTextXY*, továbbá a színeket és vonaltípust beállító eljárásokat. Grafikus output eszközként csak a képernyő aktivizálható a .BGI file-okat használó *InitGraph* eljárással, input eszközként pedig a billentyűzet szerepel (*GetKey, KeyPressed*). A plotter vezérlése közvetlenül a plotter saját utasításaival történik, amelyeket a PLOTS program készít a *.kép* típusú file-okból, és a *plot.dat* file-ba írja ki. Ezt a file-t az operációs rendszerből a *print* utasítással kell a nyomtatóként csatlakoztatott rajzgéphez küldeni. A középső és felső szoftverrétegek eljárásait a CAD-iskola programjai tartalmazzák.

6.1. A számítógépi geometriai modellezés műveletei

A geometriai modellezés műveleteit a grafikus rendszerek segítségével végzett tervező munka fázisai szerint csoportosítjuk (6.3. ábra).



6.3. ábra. A geometriai modellezés műveletei

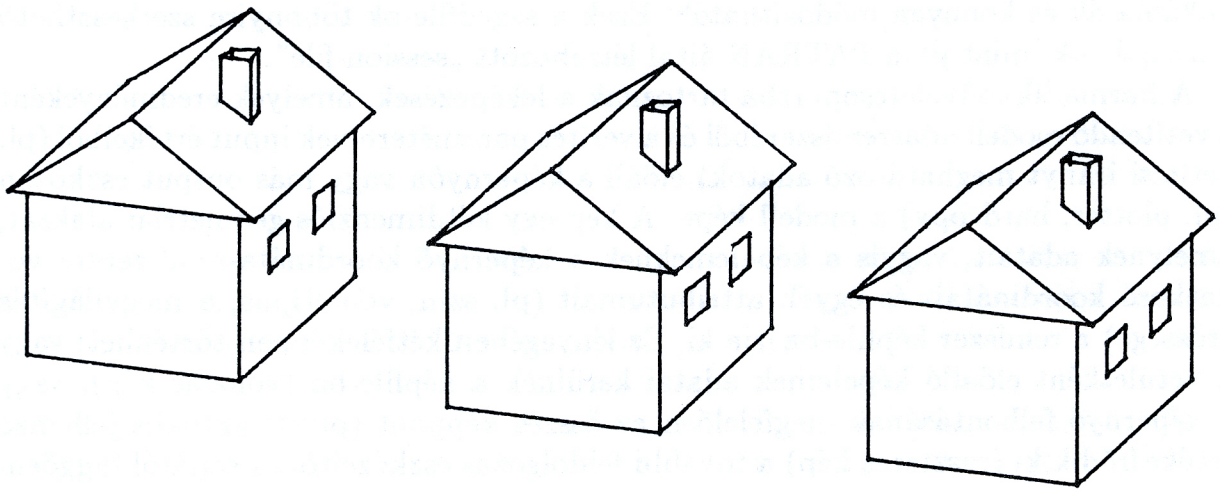
Az első fázis a modellek definiálása. Ennek eredményeként létrejön a modell numerikus leírására szolgáló adatrendszer, amelyet egy név szerint hívható modellfile tartalmaz. A modellek definiálási módja a különböző CAD-rendszerekben külsőre eltér. Egyes esetekben a képernyőn grafikus segédeszközökkel való képmanipulációk szükségesek a modellek generálásához, más esetekben numerikus bemenő adatok és speciális parancsnyelven kiadott utasítások szükségesek. Ezek mind programozástechnikai különbségek.

Tartalmilag különböznek egymástól a következő definiálási módszerek:

- meghatározott alakú alaptestek, ill. felületek (primitívek) generálása szabadon választható méreteik megadásával. Ezek a rendszer építőelemei, amelyek lehetnek egyszerű (pl. hasáb, körhenger) vagy összetettebb (pl. csavar, szögvas) geometriai alakzatok.
- előírt transzformációval valamely síkidomból való modellgenerálás. A transzformáció lehet egyenes vagy görbe mentén való eltolás (sweeping), így keletkeznek a transzlációs felületek, vagy lehet tengely körüli forgatás, amelynek eredményeként forgásfelületek, ill. forgástestek állnak elő. Egyes rendszerek megengednek más generátorokat is, pl. csavarmozgást.
- adott ponthalmazra vagy vonalseregre való felület illesztése, ahol a modell egyértelmű meghatározásához vagy a rendszer, vagy a felhasználó választ peremfeltételeket, amelyek többnyire különböző érintővektorok. Így állnak elő a foltokból (patch) összerakott ún. szabad formájú (free-form) felületek.
- a közvetlenül csúcspontokból, élekből és lapokból való építkezés (*Euler*-operátorok), ami gyakorlatilag igen nehézkes eljárás, és bonyolult alakzatoknál kétséges, hogy sikerül-e hibátlan adatrendszert összerakni. Ezért ez a definiálási módszer nem terjedt el.
- vetületekből való modellrekonstrukció, ami matematikailag általában nem oldható meg, vagy nem adható rá egyértelmű megoldás. Ezért csak térgörbék és meghatározott formájú alakzatok (pl. hasáb az alaplapjával mint felülnézettel és magasságával mint elölnézettel) definiálására léteznek megbízhatóan működő eljárások.

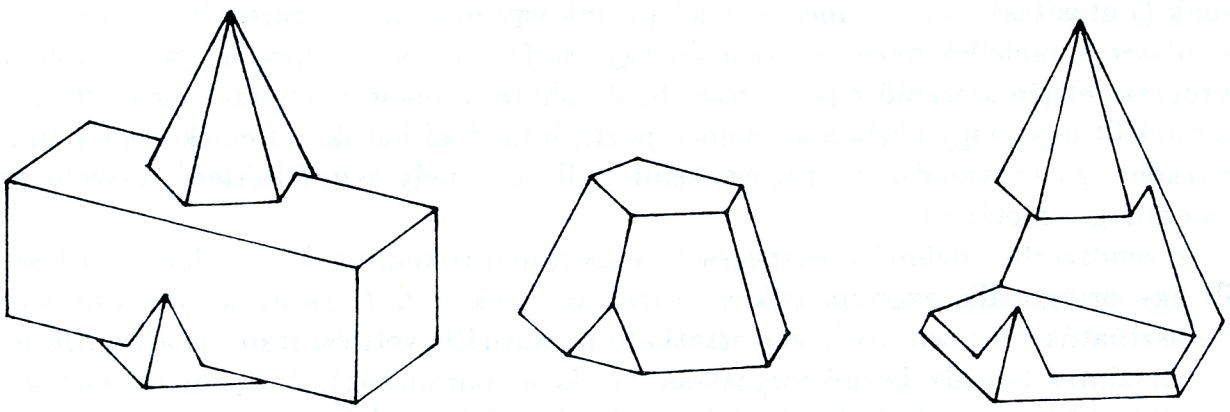
A második műveletcsoportba tartoznak a geometriai manipulációk, amelyek egy vagy több, már definiált modellen működnek, és egy új modellt eredményeznek. Ezek lényegében három csoportra oszthatók:

- transzformációk, amelyek egy modellen hatnak, és definiálásukhoz bemenő adatként a transzformációk paramétereinek értékeit (pl. eltolásvektor, nyújtási tényező, forgástengely, forgásszög) is meg kell adni.
- építkezés több modellből. Ennél a műveletnél a felhasználó a szerkesztendő modell (globális) koordináta-rendszerében elhelyezi az egyes részeket, vagyis azokat saját (lokális) koordináta-rendszerükkel pozicionálja. Ezután a rendszer összefűzi a részmodellek adatrendszereit, ami gyakorlatilag koordinátatranszformációkat és a pontok, élek, lapok átszámozását jelenti. Az így keletkező összetett, nem feltétlenül összefüggő alakzat (6.4. ábra) a további műveletek számára egyetlen modellfile-ként szerepel.



6.4. ábra. Nem összefüggő modell

két vagy több modellen végzett halmazműveletekkel (*Boole*-műveletek) való modellszerkesztés. Ezek a halmazműveletek alakzatok egyesítésének (unió), közös részének (metszet) és két alakzat különbségének meghatározását jelentik (6.5. ábra). A halmazműveletekkel definiált modell adatrendszerének a komponensek adataiból való előállítása nagyon számításigényes művelet a fellépő új csúcspontok, élek és lapok miatt. Minden alakzatra működő általános megoldás nem létezik, de több CAD-rendszer tartalmaz síklapokkal határolt testekre kidolgozott algoritmusokat.



6.5. ábra. Gúla és hasáb uniója, metszete és különbsége

A geometriai manipulációk elvégzésével párhuzamosan célszerű a bemenő adatokat és a műveleteket segédfile-okba kiírni, mert ezáltal a szerkesztések reprodukálhatók és könnyen módosíthatók. Ezek a segédfile-ok többnyire szerkeszthető szövegfile-ok, mint pl. a PATRAN által létrehozott „session file".

A harmadik műveletcsoportba tartoznak a leképezések, amelyek eredményeként a vetítendő modell adatrendszeréből és a vetítés paramétereinek input értékeiből (pl. vetítési irányt meghatározó adatok) előáll a képernyőn vagy más output eszközön (pl. plotter, hardcopy) a modell képe. A kép egy kétdimenziós geometriai alakzat, amelynek adatait, vagyis a képelemeknek a képernyő koordináta-rendszerére vonatkozó koordinátáit és egyéb attributumait (pl. szín, vonaltípus, a megvilágítás erőssége) a rendszer képfile-ba írja ki. Ez lényegében kétféleképpen történhet: vagy a vetületként előálló képelemek adatai kerülnek a képfile-ba (vonalas kép), vagy a képernyő felbontásának megfelelően az összes képpont (pixel) aktuális jellemző értéke íródik ki (raszteres kép) a további feldolgozás eszközeitől és céljától függően.

Mi van a lemezen ?

A modellek definíciói a program elején szerepelnek. A két csavarvonal által meghatározott vonalfelület a programban alapfelületként (primitív) szerepel, amelynek szabadon választható méretei a csavarvonalak adatai, továbbá a paramétertartomány az osztáspontok számával. A hengeres, kúpos és forgásfelületek definiálása a szükséges bemenő adatok beolvasása után a megfelelő algoritmust tartalmazó eljárás (generátor) hívásával történik. Ezek az eljárások egy adott görbe pontjaiból eltolással, centrális hasonlósággal, ill. tengely körüli forgatással állítják elő a felület többi pontját, az éleket és lapokat, tehát a modell adatrendszerét. A definiált modell lesz az aktuális modell mindaddig, amíg újat nem definiálunk, vagy file-ból másikat be nem olvasunk.

A geometriai manipulációkhoz tartozik az aktuális modellen működő eltolás, tengely körüli forgatás és nyújtás a koordinátatengelyek irányában. A modellekből való építkezés kétféleképpen végezhető el: vagy az aktuális modellhez egy másikat csatolunk (*j* utasítás), vagy elemenként felépítünk egy modellkonfigurációt, miközben a rendszer a modellek nevét és pozícióit egy *.mcf* típusú szövegfile-ba írja. Ez a file a program elején szereplő *r* paranccsal beolvasható, amikor a rendszer ismét felépíti a modellt úgy, hogy közben az elemek pozíciói módosíthatók. Az építkezés eredményeként egy adatrendszer, azaz egy felület áll elő, amely nem feltétlenül összefüggő és önmagát metszheti.

A rendszerben definiált leképezések: a koordinátasíkokra való merőleges vetítések és egy ortogonális axonometrikus projekció. Ezek a *t*, *f* és az *a* utasításokkal választhatók ki. Ezen kívül változtatható az aktuális vetítési irány is a függőleges és vízszintes tengely körüli forgatással (*l* és *u* parancsok). A definiált vetítéssel előállított képet a *d* (minden élre) és a *h* (csak a látható élekre) utasításokkal hívott rajzoló algoritmusok készítik el. A *v* utasítás hatására pedig az alakzat felül-, elöl- és oldalnézete, továbbá egy axonometrikus vetülete jelenik meg a négy részre osztott

képernyőn. Az elkészített vetület vonalelemeit a rendszer a *w* utasítás és a hidden-line eljárás megfelelő utasításának hatására egy *.kep* típusú file-ba írja a szakaszok kezdő- és végpontjainak a képernyőhöz rendelt koordináta-rendszerre vonatkozó koordinátáinak felsorolásával. Az így elkészített képfile-ok a PLOTS programmal tovább feldolgozhatók és plotterrel kirajzoltathatók.

6.2. Modellezés diszkrét és analitikus adatrendszerekkel

A CAD-rendszereket többféle szempont szerint osztályozhatjuk. Az általuk definiálható modellek dimenziószáma szerint vannak 2D-s, 2,5D-s és 3D-s rendszerek. A 2D-s rajzoló rendszerekben csak síkbeli alakzatok értelmezhetők. A 2,5D-s rendszerek hengeres alakzatokat képesek kezelni, ahol az összes alkotó iránya megegyezik, ezért a modelleket az alaplapjuk és a magasság számértéke egyértelműen definiálja. A 3D-s rendszerek viszont térbeli struktúrákkal működnek.

Ettől a felosztástól függetlenül a CAD-rendszereket két csoportra lehet osztani aszerint, hogy a modellek leírása diszkrét vagy analitikus adatrendszerrel történik-e. Kétségkívül ez a belső ábrázolási mód a CAD-rendszerek jellemzésének legfontosabb szempontja.

6.2.1. Modellezés diszkrét adatrendszerrel

Diszkrét adatokból áll a poliéderek adatrendszere, amelyben az alakzat minden pontja, éle és lapja fel van sorolva. Egy ilyen adatrendszert már megismertünk a 4. fejezetben. Most bemutatunk egy másikat speciálisan az egységkocka numerikus definiálására.

Numerikus adatok		Topológiai adatok			
csúcs	csúcs koordinátái	él	él csúcsai	lap	lap élei
1	0, 0, 0	1	1, 2	1	1,2,3,4
2	0, 0, 1	2	2, 6	2	5,6,7,8
3	0, 1, 0	3	6, 5	3	1,10,5,9
4	0, 1, 1	4	5, 1	4	3,11,7,12
5	1, 0, 0	5	3, 4	5	4,9,8,12
6	1, 0, 1	6	4, 8	6	1,10,6,11
7	1, 1, 0	7	8, 7		
8	1, 1, 1	8	7, 3		
		9	1, 3		
		10	2, 4		
		11	6, 8		
		12	5, 7		

Ez a táblázat csak a geometriai adatokat tartalmazza, ezen kívül további, nem geometriai adatok is szerepelhetnek az adatrendszerben (pl. szín, vonaltípus, anyagi jellemzők).

Az első oszlopban álló csúcspontok határozzák meg a kocka méretét és helyét, ezek a numerikus adatok. Az összekötési, tartalmazási és szomszédsági viszonyokat meghatározó adatokat nevezzük topológiai adatoknak. A bemutatott adatrendszerben az élek végpontjainak indexe (mutató vagy pointer) és a lapokat határoló élek sorszáma szerepel. A topológiai adatok minden kockára ugyanazok, mert pl. az 1. él mindig az 1. és 2. csúcspontokat köti össze függetlenül attól, hogy mi a kocka térbeli pozíciója. Ezért világos, hogy a kockán végrehajtott térbeli transzformációk csak a numerikus adatokra hatnak. Egy adott kocka számítógéppel való generálása pedig a csúcspontok koordinátáit tartalmazó oszlop kitöltését jelenti, miközben az élek és lapok oszlopát a megfelelő kocka-eljárás már készen tartalmazza.

A különböző CAD-rendszerekben a poliéder adatrendszerek éppen a topológiai adatok szerkezetében különböznek egymástól. A lapokat pl. meg lehet adni csúcspontjainak ciklusával is, de lehet a lapokat az élek oszlopában is feltüntetni. A poliédereken működő algoritmusok hatékonysága nagyrészt attól függ, hogy milyen gyorsan lehet az aktuális adatrendszerből a szükséges topológiai információkat összegyűjteni (pl. mely élek futnak össze egy csúcspontban, vagy melyik élben csatlakozik két szomszédos lap stb.).

Egy egészen másfajta diszkrét adatrendszer az „oktáns-fa" (oct-tree), amellyel nem foglalkozunk részletesen. Lényege az, hogy a térbeli alakzatot kockákból építik fel, amelyeknek méretét - a középsíkokkal mindig nyolc részre vágva - addig finomítják (egészen pixelméretig), amennyire azt az alakzat formája megkívánja. Az összes kockát egy faalakú gráfba rendezik, amelyet azután a különböző szerkesztő és leképező algoritmusok feldolgoznak.

A diszkrét adatrendszerek általában igen terjedelmesek, bizonyos eljárások (pl. hidden-line) programozására nagyon célszerűek, de nem alkalmasak görbevonalú felületek egzakt leírására.

6.2.2. Modellezés analitikus adatrendszerekkel

Analitikus leírási módot, azaz függvényekkel, mégpedig a leggyakrabban spline-függvényekkel való előállítást használnak sima görbék és felületek numerikus definiálására. A spline-technika az 1960-as évek elején jelent meg autó karosszériák és repülőgépek számítógéppel való tervezésére. Azóta nagyon sok CAD-rendszerben ez a geometriai modellezés alapja. Az így előállított felületeket kezdetben szoborszerű (sculptured), később inkább szabad formájú (free-form) felületeknek nevezték.

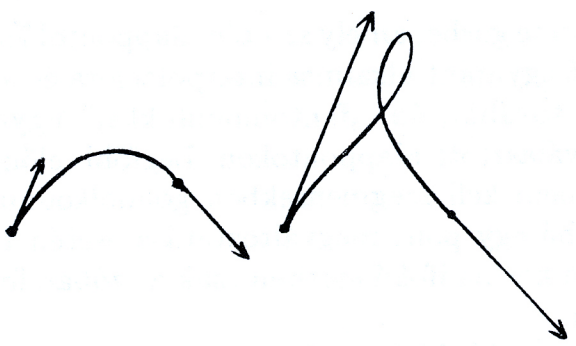
Az adatrendszer az alappontok (kontrollpontok, angolul általában knots) halmazát és a konkrét függvényt tartalmazza, amely az adott ponthalmazból és esetleg még peremfeltételekből a kívánt görbét, ill. felületet generálja. Ez a generálás kétféleképpen történhet. Az interpolációnál a görbe, ill. felület áthalad az adott pontokon, az approximációnál általában nem halad át (esetleg egyiken sem halad át), de bizonyos értelemben közel halad az adott pontokhoz. Mindkét szerkesztési módra

sokféle, a gyakorlatban is jól bevált megoldás született. A modelleket leíró spline-függvényeknek legtöbbször a paraméteres előállítását használják: görbék megadására egyparaméteres, felületek megadására kétparaméteres vektorfüggvényeket. A megfelelő matematikai elmélet a differenciálgeometriához és az approximáció elméletéhez tartozik, itt csak rövid tájékoztatásul sorolunk fel néhány nevezetes példát.

Az *Hermite*-féle (ejtsd: ermit) harmadfokú vektorspline interpolációs feladat megoldására alkalmas. Ha csak két alappontot adunk meg a $\mathbf{p}_0$ és $\mathbf{p}_1$ helyvektorokkal és peremfeltételként e pontokban két vektort, a $\mathbf{p}_0'$ ill. $\mathbf{p}_1'$ vektorokat, akkor az

$$\mathbf{r}(t) = \mathbf{p}_0 \cdot (1-3t^2+2t^3) + \mathbf{p}_1 \cdot (3t^2-2t^3) + \mathbf{p}_0' \cdot (t-2t^2+t^3) + \mathbf{p}_1' \cdot (-t^2+t^3), \quad (0 \leq t \leq 1)$$

vektorfüggvény egy olyan görbét ír le, amelynek kezdőpontja $\mathbf{p}_0$, végpontja $\mathbf{p}_1$ és ezekben a pontokban az érintővektorai $\mathbf{p}_0'$, ill. $\mathbf{p}_1'$. A görbét tekinthetjük egy mozgó pont pályájának, amely a $t = 0$ időpillanatban $\mathbf{p}_0'$ sebességgel indul a $\mathbf{p}_0$ pontból és $\mathbf{p}_1'$ sebességgel érkezik a $\mathbf{p}_1$ pontba. A t paraméter tehát ebben a felfogásban az időt jelenti, és az $\mathbf{r}(t)$ helyvektor adja meg, hogy hol van a pont a t időpillanatban. A görbe alakját a két érintővektor hosszának változtatása úgy befolyásolja, hogy rövidebb vektorokkal laposabb görbét kapunk. Ha a két alappont távolságához képest „viszonylag hosszú" érintőket írunk elő, vagyis a mozgó pont nagy sebességgel futja be a görbét, akkor a görbe hurkolt alakot vehet fel (6.6. ábra).



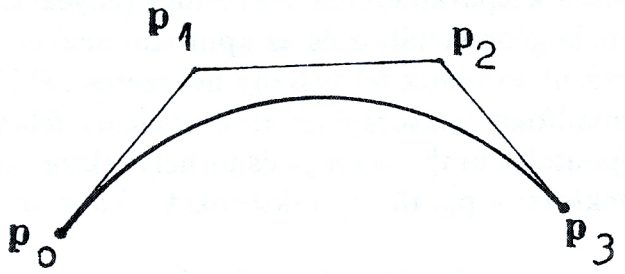
6.6. ábra. Harmadfokú *Hermite*-görbék

Több alappontra illeszkedő köbös *Hermite*-spline meghatározásához az alappontok helyvektorait, továbbá az első és utolsó pontbeli érintővektorokat kell megadni. Ezekből az adatokból ívenként (szegmensenként) előállítható egy olyan görbe, ahol a szegmenseket leíró vektorfüggvény alakja a fentivel megegyezik, és az egész görbe sima, vagyis a görbeívek közös érintővel csatlakoznak az alappontokban, továbbá a görbület is folytonos lesz.

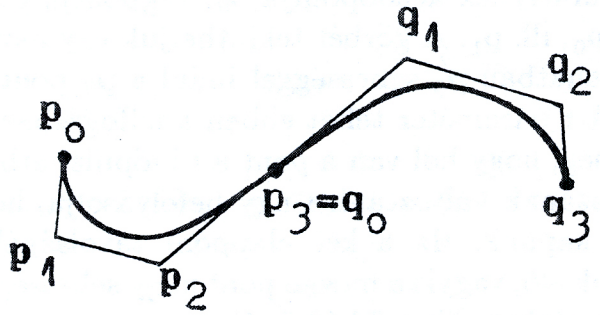
A *Bézier*-féle (ejtsd: bezié) harmadfokú vektorspline az egyik legegyszerűbb megoldás görbe approximációjára. Egy ív meghatározásához négy kontrollpontot kell helyvektoraival megadni. Legyenek ezek $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$. Az

$$\mathbf{r}(t) = (1-t^3)\mathbf{p}_0 + 3t(1-t)^2\mathbf{p}_1 + 3t^2(1-t)\mathbf{p}_2 + t^3\mathbf{p}_3, \quad (0 \leq t \leq 1)$$

vektorfüggvény olyan görbeívet ad meg, amely áthalad az első ($\mathbf{p}_0$) és utolsó ($\mathbf{p}_3$) pontokon, és jól követi a négy pont által meghatározott karakterisztikus poligon alakját úgy, hogy annak első és utolsó éle érinti a görbét (6.7. ábra).



6.7. ábra. Harmadfokú *Bézier*-görbe



6.8. ábra. Összetett *Bézier*-spline

A kontrollpontok függvényszorzóit súlyfüggvényeknek (blending functions) nevezzük.

Több pont megadása esetén úgy állíthatunk elő harmadfokú *Bézier*-ívekből sima görbét, hogy a csatlakozó szegmensek karakterisztikus poligonjainak két szomszédos élét egy egyenesen választjuk (6.8. ábra).

A *Bézier*-spline-ok jól alkalmazhatók görbék számítógépi feldolgozására, mert a meghatározó pontok kényelmesen definiálhatók, a görbe alakja jól elképzelhető, és a polinom függvények könnyen számolhatók. Hátránya, hogy egy pont megváltoztatása után az egész görbét újra kell számítani.

Sokkal rugalmasabban kezelhető a B-spline görbe, amely szintén alappontokkal és súlyfüggvényekkel van definiálva, továbbá egyaránt alkalmas interpolációra és approximációra. Bizonyos módon választott további, ún. „fantompontokkal" ugyanis el lehet érni, hogy a görbe áthaladjon kiválasztott alappontokon. Legfőbb előnyei, hogy akárhány alappontot adunk meg, nem kell szegmensekben gondolkodnunk, mert azokat a függvény generálja, továbbá egy pont megváltoztatása esetén nem kell az egész görbét újra számítani, hanem harmadfokú esetben csak a szóban forgó pont körüli négy paraméterintervallumon.

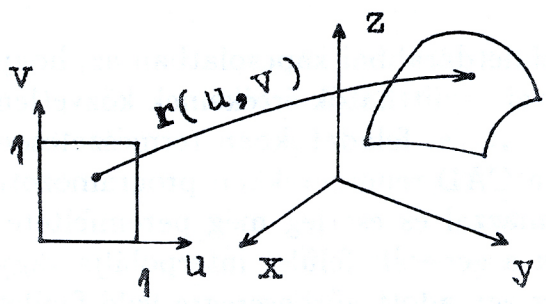
Felületeket kétparaméteres vektorfüggvényekkel lehet leírni. Egy kétparaméteres bikubikus (mindkét változóban harmadfokú) vektorfüggvény általános alakja a következő:

$$\mathbf{r}(u,v) = x(u,v)\mathbf{i} + y(u,v)\mathbf{j} + z(u,v)\mathbf{k},$$

ahol mindegyik koordinátafüggvény egy

$$\begin{aligned} s(u,v) = {} & a_{00} + a_{01}v + a_{02}v^2 + a_{03}v^3 \\ & + a_{10}u + a_{11}uv + a_{12}uv^2 + a_{13}uv^3 \\ & + a_{20}u^2 + a_{21}u^2v + a_{22}u^2v^2 + a_{23}u^2v^3 \\ & + a_{30}u^3 + a_{31}u^3v + a_{32}u^3v^2 + a_{33}u^3v^3 \end{aligned}$$

kétváltozós polinom természetesen általában különböző együtthatókkal.



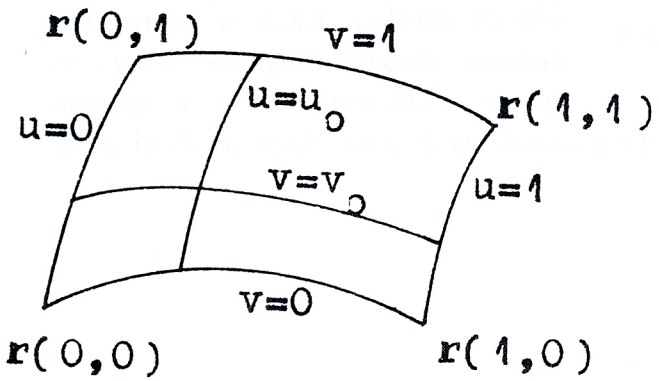
6.9. ábra. Leképezés kétparaméteres vektorfüggvénnyel

Ha a két paraméter csak a $[0,1]$ intervallumon változik, akkor az $\mathbf{r}(u,v)$ vektorfüggvény egy felületdarabot ír le, amely úgy tekinthető mint a paramétersík $[0,1] \times [0,1]$ egységnégyzetének képe az $\mathbf{r}(u,v)$ függvény által definiált leképezés mellett (6.9. ábra).

Az $\mathbf{r}(u,v)$ függvény ugyanis a négyzet minden (u,v) pontjához a térnek az $\mathbf{r}(u,v)$ helyvektorú pontját rendeli. Ezt a felületdarabot az $s(u,v)$ koordinátafüggvények összesen 3×16 együtthatója, vagyis az a_{ij} $(i,j = 0,\ldots,3)$ számértékek egyértelműen meghatározzák, amelyeket a felületdarab algebrai adatainak vagy algebrai együtthatóinak nevezünk.

Két gyakorlati alapkérdés merül fel az analitikusan meghatározott felületekkel kapcsolatban: hogyan ábrázolhatók és milyen módon definiálhatók a számítógép segítségével ?

A felületábrázolásnak jól bevált módszere a paramétervonalak hálózatának megrajzolása. A paramétervonalak az $u = u_0$ és a $v = v_0$ egyeneseknek az $\mathbf{r}(u,v)$ leképezéssel keletkező képei $(u_0,\ v_0)$ állandók. Ezek felületi görbék, amelyeknek egyparaméteres vektoregyenlete $\mathbf{r}(u_0,v)$, ill. $\mathbf{r}(u,v_0)$. Speciálisan a felületdarabot határoló görbék az $\mathbf{r}(u,0)$, $\mathbf{r}(u,1)$ u paramétervonalak és az $\mathbf{r}(0,v)$, $\mathbf{r}(1,v)$ v paramétervonalak (6.10. ábra).

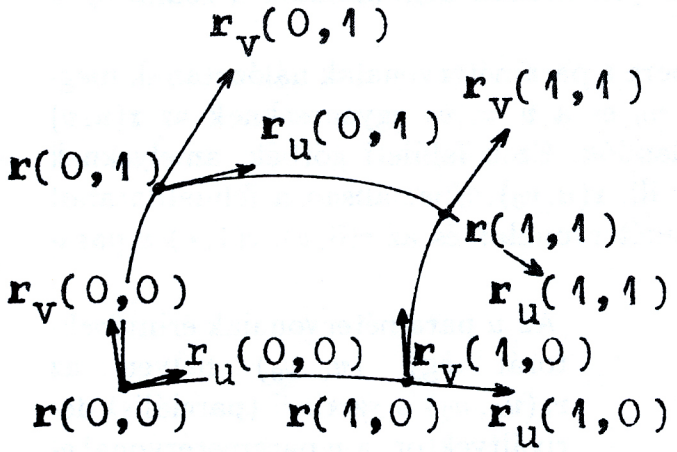


6.10. ábra. Felületdarab paramétervonalai

Az u paramétervonalak érintővektorát egy (u_0,v_0) helyen az $\mathbf{r}_u(u_0,v_0)$ u szerinti (parciális) deriváltvektor, a v paramétervonal érintővektorát pedig az $\mathbf{r}_v(u_0,v_0)$ v szerinti deriváltvektor adja meg. Ezek a vektorok feszítik ki a felület $\mathbf{r}(u_0,v_0)$ pontbeli érintősíkját, amelynek normálvektora tehát az $\mathbf{r}_u \times \mathbf{r}_v$. Bikubikus vektorfüggvény esetén ez a normális és ennek megfelelően az érintősík az egész felületdarabon folytonosan változik. Ilyenkor mondjuk, hogy a felület sima. Ha a $[0,1]$ intervallumon elég sűrűn választjuk az u_0, ill. v_0 értékeket, akkor az ezeknek megfelelő paramétervonalak hálózata jól mutatja a felületdarab alakját.

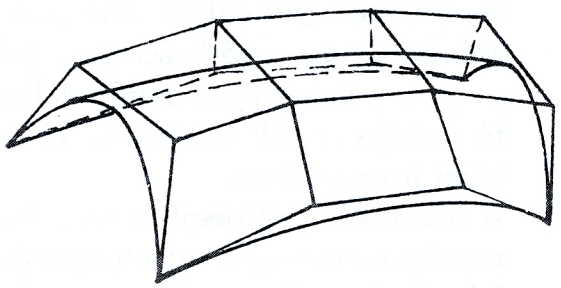
A gyakorlatban használnak más görbehálózatokat is a felületek szemléltetésére, továbbá különböző intervallumfelosztó eljárásokat a felületek háromszöglemezekkel való megközelítésére.
Egy ettől különböző ábrázolási mód a képpontonként előállított raszterkép, amelyről a 6.5. fejezetben lesz szó.

A másik gyakorlati kérdés egy bikubikus felületdarabbal kapcsolatban az, hogy milyen adatokkal célszerű definiálni. Az algebrai együtthatók értékeinek közvetlen megadása lenne egy definiálási mód, ami a tervezési feladatokban természetesen nem célszerű. Ehelyett a felületdarabokat — a CAD-rendszerekben programozott spline-függvényeknek megfelelően — ponthalmazzal és esetleg még peremfeltételekkel határozzák meg. A ponthalmazt azután a generált felület interpolálja vagy approximálja. Ezen kívül szokásos eljárás még egy adott görbeseregre való felület illesztése is. A spline-technikával előállított felületdarabokat foltoknak (patch) nevezik. Az alkalmazásokban előforduló felületek általában nem írhatók le egyetlen vektorfüggvénnyel, ezért azokat egy alkalmas felosztással több, egymáshoz jól (első vagy másodrendben folytonosan) csatlakozó foltból szerkesztik. Röviden megemlítünk három gyakran használt folttípust.

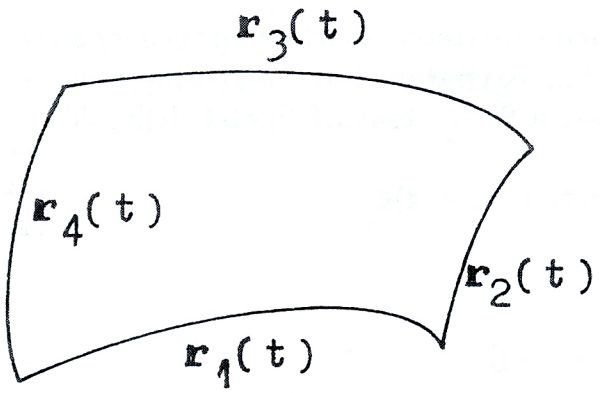


6.11. ábra. A *Ferguson*-folt geometriai adatai

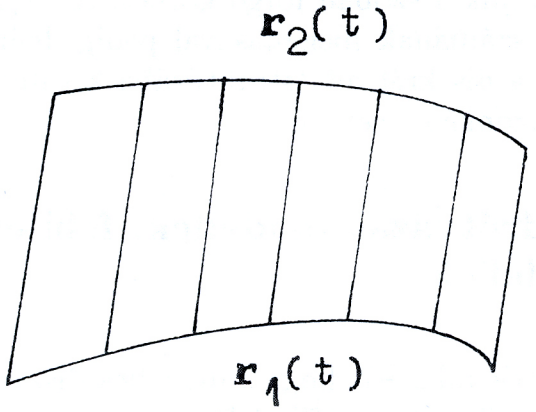
A *Ferguson*-féle foltot 16 ún. geometriai együttható határozza meg, amelyek a következők: a folt négy csúcspontjának helyvektora, a csúcspontokban a határoló görbék érintővektorai (6.11. ábra), továbbá a négy „twisztvektor", amelyek itt speciálisan zérusvektorok, egyébként az $\mathbf{r}_{uv}$ második vegyes deriváltaknak a csúcspontokban felvett értékei. Ezek a geometriai adatok megfelelnek az *Hermite*-spline adatainak, és a *Ferguson*-folt határoló görbéi valóban a megfelelő végpontok és érintővektorok által meghatározott *Hermite*-görbék lesznek.



6.12. ábra. A *Bézier*-foltot meghatározó pontrács



6.13. ábra. Négy görbével adott *Coons*-folt



6.14. ábra. Két görbével adott *Coons*-folt

A *Bézier*-folt geometriai együtthatóit egy 4 × 4 pontból álló pontrács 16 helyvektora alkotja. A *Bézier*-folt csak a négy csúcsponton halad át, a többi pontot approximálja (6.12. ábra).

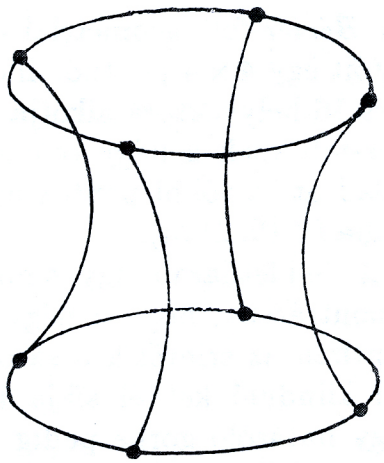
A felületdarab úgy simul bele a pontrácsba, hogy a négy sarokpontban az érintősík a sarokpontból kiinduló két él síkja lesz. A négy határoló görbe pedig a négy keretvonalhoz mint karakterisztikus poligonhoz tartozó *Bézier*-görbe lesz.

A *Coons*-féle foltot négy határoló görbéje határozza meg, amelyek mindegyike harmadfokú polinomiális vektorfüggvénnyel adott (6.13. ábra).

Coons-féle felületdarabnak nevezik azt is, amelyet két, szintén harmadfokú görbeív határoz meg úgy, hogy azok a generált folt szemközti határgörbéi lesznek, és azokat a leíró vektorfüggvény lineárisan interpolálja (6.14. ábra).

A CAD-rendszerekben a megfelelő spline-függvényt előállító algoritmus a geometriai input adatokból minden foltra kiszámítja az algebrai együtthatókat, vagyis a bikubikus vektorfüggvény 3 × 16 együtthatóját, amelyekből a felületi pontok koordinátái közvetlenül számolhatók.

Összetett felületet általában nem kell a felhasználónak foltonként definiálnia. A CAD-rendszerekben programozott eljárások ugyanis sokszor elvégzik a foltokra való felbontást a szükséges peremfeltételek megválasztásával együtt. Gyakori pl. a forgásfelületeknek valamilyen típusú vektorspline függvénnyel való approximálása, ahol a paralelköröket négy negyedívre bontják, és ezeket a 90 fokonként elforgatott meridiángörbékkel együtt választják folthatároknak (6.15. ábra).



6.15. ábra. Négy folttal előállított forgásfelület

Napjainkban gyorsan fejlődik a kétparaméteres B-spline vektorfüggvényekkel való felületgenerálásra szolgáló algoritmusok elmélete. Ezek az eljárások egyszerűen definiálható (általában csak ponthalmazt vagy görbesereget kell megadni) és könnyen manipulálható (lokálisan változtatható) felületeket hoznak létre.
A bikubikus vektorspline-ok alkalmazása mellett gyakori a magasabb fokszámú polinom függvényekkel való modellezés és egyéb típusú spline-ok felhasználása is.

Mi van a lemezen ?

A CAD-iskola térbeli modelljeit a 4. fejezetben ismertetett diszkrét (poliéder) adatrendszer írja le. A modelldefiníciókban azonban egyparaméteres vektorfüggvénnyel meghatározott görbék is előfordulnak. Ilyenek a SURF csavarfelületet definiáló

$$\mathbf{r}(t) = a\cos t\mathbf{i} + b\sin t\mathbf{j} + (ct + d)\mathbf{k}$$

egyenletű csavarvonalai és a CURVE

$$\mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j}$$

egyenletű síkgörbéi. Az a, b, c, d együtthatók megválasztásával, ill. az $x(t)$ és $y(t)$ skalárértékű függvények begépelésével definiáljuk a szóban forgó görbét. A $[t_1, t_2]$ paramétertartománynak és az osztáspontok számának megadásával pedig definiáljuk a görbének azokat a pontjait, amelyek a diszkrét adatstruktúrába kerülnek. Ettől kezdve a görbe a rendszer számára poligonként szerepel.

6.3. Adatrendszerek szerkezete, drótvázas modellek, felület- és testmodellek

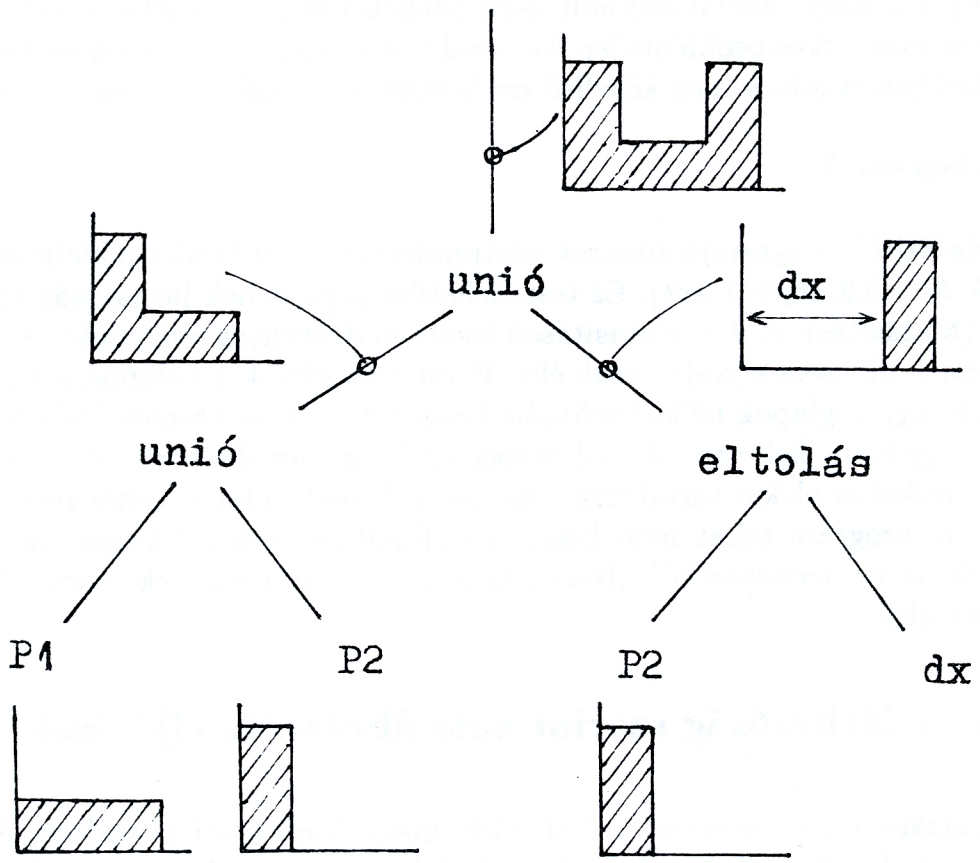
Adatrendszerek szerkezeti felépítése — függetlenül attól, hogy a modellek diszkrét vagy analitikus leírását tartalmazzák — lényegében háromféle lehet.

Drótvázas (wire-frame) vagy más néven élmodelleket generálnak a 2D-s rajzoló rendszerek, a 2,5D-s tervező rendszerek és az 1970-es évek elején fejlesztett 3D-s, főként poliédermodellező rendszerek. Ezek a rendszerek minden modellt az éleivel (egyenes vagy görbe - legtöbbször körív) írnak le, ami térbeli alakzatok esetén természetesen nem kielégítő. Egy kocka 12 élét tartalmazó adatrendszernek ugyanis

egyaránt megfeleltethető egy tömör kocka vagy egy felül nyitott kocka alakú doboz. Emiatt a ma használatos CAD-rendszerekben a drótvázas modell csak munka közbeni segédábrák, ideiglenes vázlatok előállítására szolgál. Az összes él vetületét tartalmazó drótvázas ábrák elkészítéséhez ugyanis elegendő a csúcspontok vetületét meghatározni, ami nagy modellekre is gyorsan elvégezhető. Láthatóság szerint való ábrázolásra ez az adatrendszer nem alkalmas.

Felületmodell áll elő, ha felületi adatstruktúra (boundary representation) írja le a rendszer által generált modell határoló felületét, amely síklapokból áll a poliédermodellek, és foltokból áll a szabad formájú felületek esetén. Ezek az adatrendszerek a felületelemeket meghatározó numerikus adatokat és a szomszédsági viszonyokat meghatározó topológiai adatokat tartalmazzák, és a térbeli alakzatokat mindig egyértelműen meghatározzák. Ezt a 6.2.1. pontban a poliédermodellekre részletesen leírtuk.

Nyilvánvalóan felületi adatrendszerre van szükség pl. alakzatok metszésvonalának meghatározásához vagy a láthatósági viszonyokat feltüntető takart vonalas vetületek elkészítéséhez. A megvilágítás szerint árnyékolt és az anyagminőséget (matt vagy tükröző) is érzékeltető képek generálása is legtöbbször felületi adatstruktúrákon működő algoritmusokkal történik, de léteznek megoldások testmodellekre is.



6.16. ábra. CSG-fa

A testmodelleket vagy más néven térfogatmodelleket (volume-model vagy CSG-representation a Constructive Solid Geometry rövidítéséből) tipikusan olyan rendszerek állítják elő, ahol csak előre definiált primitívekből lehet szerkeszteni. A testmodell egy összetett alakzatot egy speciális szerkezetű gráffal, a CSG-fával ír le, ahol a fa egyélű csúcsaiban a primitívek, a csomópontokban pedig a szerkesztésekhez szükséges geometriai manipulációk, azaz halmazműveletek és transzformációk vannak (6.16. ábra).

Ez a fastruktúra numerikusan tömören leírható, és a modellt mint ponthalmazt egyértelműen állítja elő. A CSG-reprezentáció nagyon célszerű halmazműveletek végrehajtására, de mivel a felületet leíró adatokat közvetlenül nem tartalmazza, nem alkalmas a modell vetületének megrajzolására, csak pontonként számolt, raszteres képek előállítására.

Mivel a különböző szerkezetű adatrendszereken más-más algoritmusok működnek hatékonyan, ezért fejletteb CAD-rendszerek (pl. EUKLID) ún. hibrid modelleket állítanak elő. Ez azt jelenti, hogy az alakzatok definiálásakor létrehozzák az alakzatot reprezentáló felületmodellt és testmodellt is, és a különböző operációkat a megfelelő adatrendszeren hajtják végre. Az egyik adatrendszerről a másikra való áttérés ugyanis matematikailag nem mindig lehetséges, de ha megoldható is, nagyon számításigényes eljárás. A testmodellnek a felületmodellre való konvertálását, vagyis CSG-fával előállított modell felületi adatrendszerének (csúcsok, élek, lapok) kiszámítását (boundary evaluation) poliédermodellekre mindig el lehet végezni, bár közben számos numerikus probléma lép fel. Analitikusan leírt primitívek esetén csak néhány felülettípusra adtak meg közelítő módszereket, általános eljárás nincs.

Mi van a lemezen ?

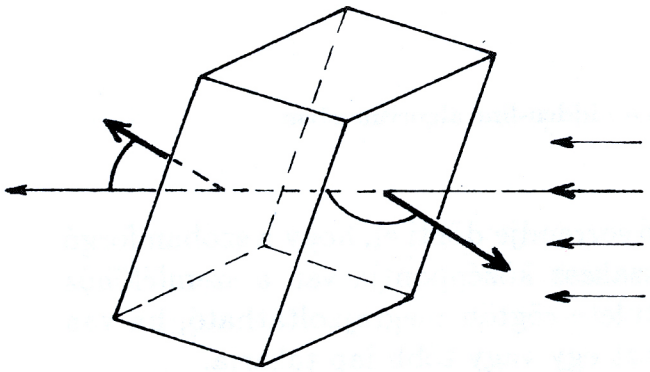
A CAD-iskola SURF programja diszkrét adatrendszerrel leírt felületmodelleket állít elő (lásd a 4. és a 6.2. fejezeteket). Ez teszi lehetővé a modellek láthatóság szerinti ábrázolását (*h* utasítás). A *d* és *v* utasítások azonban drótvázas modelleket ábrázolnak, amelyeknek nincsenek lapjai, csak élei. Ezért nem látjuk a különbséget pl. egy véglapokkal és egy véglapok nélkül definiált henger drótvázas vetületei között. Azt sem látjuk, hogy mely élek vannak elöl és melyek hátul. Az élmodell adatrendszere csak csúcspontokat és éleket tartalmaz, amelyek a felületi adatrendszer részhalmazát képezik. A program tehát nem készít a definiált alakzatról külön élmodellt, hanem a felületi adatrendszerből olvassa ki a drótvázas vetületek elkészítéséhez szükséges adatokat.

6.4. A láthatóság szerint való ábrázolás eljárásai

Egy térbeli alakzatról készített szemléletes kép megkülönbözteti az alakzat látható részeit a nem láthatóktól (pl. a látható élek folytonos vonallal, a nem láthatók szaggatott vonallal vannak ábrázolva), vagy csak a látható éleket tartalmazza. A

gyors számítógépek és nagyfelbontású színes képernyők lehetővé teszik a térbeli alakzatok valósághű, fényképszerű ábrázolását is.

A láthatóság eldöntéséhez minden esetben a modellen (a modelltérben) és a képernyőn (a képtérben) is vizsgálatokat kell elvégezni, de hogy mi látszik, csak a térben dönthető el. A láthatósági problémák eldöntésére nem létezik egy mindenkor érvényes legjobb algoritmus, mert a számítási módszerek függenek egyrészt az adatrendszer típusától (diszkrét vagy analitikus adatok) és szerkezetétől (felületi vagy térfogati), másrészt attól, hogy vonalas vagy raszteres képet állítanak-e elő. A vonalas kép plotterrel megrajzoltatható, a raszteres kép hardcopy elkészítésére alkalmas eszközökkel (pl. ink-jet) nyomtatható ki.



6.17. ábra. Konvex poliéder lapjainak láthatósága

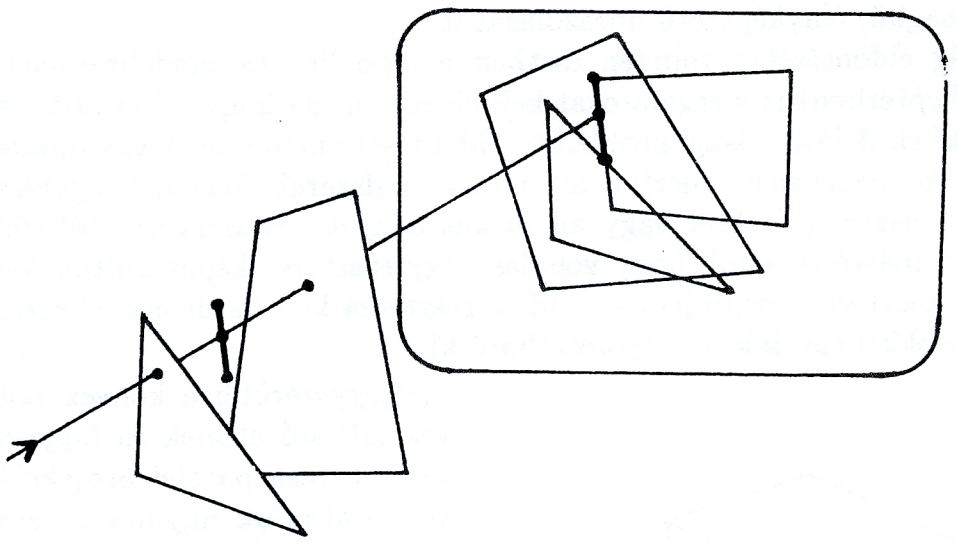
Legegyszerűbb a konvex poliéderek látható éleinek és lapjainak a kiválasztása paralel projekció esetén. Láthatók ugyanis a szemlélő felé forduló lapok élei. Ezeket a lapokat a kifelé irányított normálvektoroknak a vetítősugarakkal bezárt szöge szerint lehet kiválasztani (6.17. ábra).

Ha ez a szög tompaszög, akkor a szóban forgó lap és az élei látszanak, ha hegyesszög, akkor a lap nem látszik, de a látható lapokhoz csatlakozó élei igen.

Ezt a vizsgálatot a nem konvex alakzatok láthatóságának eldöntésére kidolgozott algoritmusok is elvégzik a biztosan nem látható élek kiszűrésére. Görbevonalú felületekkel határolt alakzatok ábrázolásánál gyakori módszer, hogy az alakzatot egy közelítő poliéderrel helyettesítik, és a poliédert ábrázolják.

Összetett alakzatok láthatósági viszonyainak eldöntésére gyakorlatilag mindegyik CAD-rendszer tartalmazza a saját eljárásait. Ezek az eljárások szerkezetükben alapvetően kétfélék lehetnek: modell irányulásúak (modellre orientált) vagy kép irányulásúak.

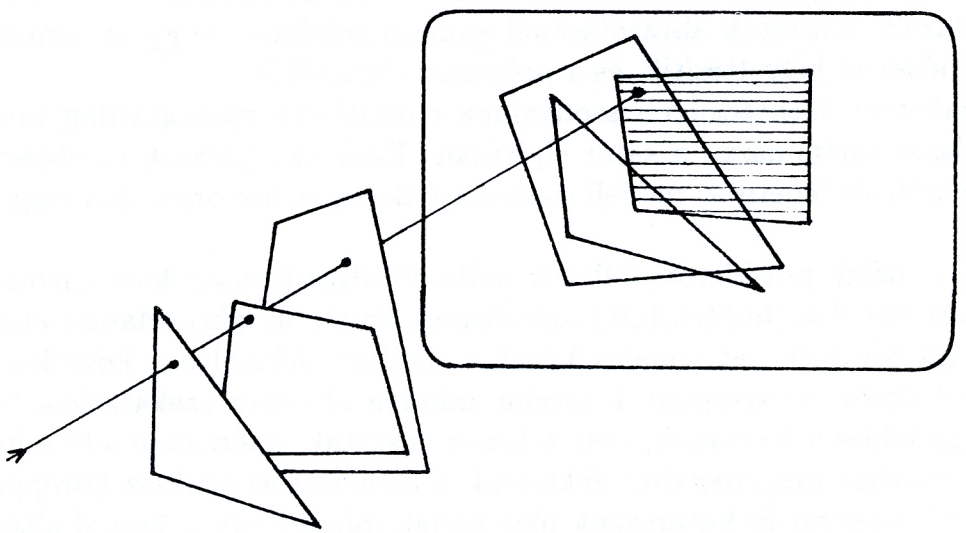
A modell irányulású, poliédermodelleken működő algoritmusok közé tartozik a klasszikus, takart vonalas (hidden-line) algoritmus, amely az ábrázolandó alakzatot az élei szerint vizsgálja, és vonalas képet szolgáltat. Alapelve a következő: a vetületekben keletkező metszéspontok szerint minden él olyan szakaszokra bomlik, amelyek vagy teljesen látszanak, vagy teljesen takartak. Ezért elegendő minden szakasznak egy pontját megvizsgálni. Tekintsük a kiválasztott szakasz középpontján áthaladó vetítősugarat és határozzuk meg annak minden olyan lappal alkotott metszéspontját, amely a kiválasztott szakaszt nem tartalmazza (6.18. ábra).



6.18. ábra. Egy szakasz vizsgálata a hidden-line algoritmusban

Ezeknek a pontoknak a vetítősugáron való sorrendje dönti el, hogy a szóban forgó szakasz látszik-e vagy nem. Ha ugyanis a szakasz középpontja van a szemlélőhöz legközelebb, akkor a szakasz látható, és vetülete rögtön megrajzoltatható, ha van annál közelebbi metszéspont, akkor a szakaszt egy vagy több lap takarja.

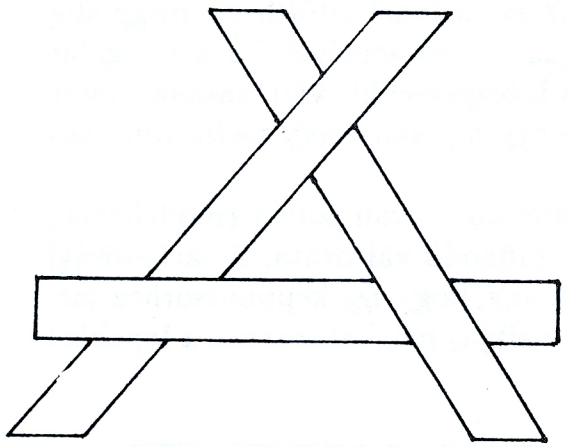
Az ún. festő (painting) algoritmus szintén modell irányulású, laponként vizsgálja az ábrázolandó poliédert és raszteres képet szolgáltat. Az első lépésben sorba rendezi a lapokat a vetítősugarak mentén a képsíktól távolodva (6.19. ábra).



6.19. ábra. Egy lap vizsgálata a festő algoritmusban

Ezután a leghátsó lappal kezdve egyenként vetíti a lapokat és képpontonként (pixel) kitölti a vetületet a laphoz rendelt színnel. Ily módon a szemlélőhöz közeledve a

lapokat ráfesti a mögöttük lévőre. Árnyékolt kép előállításához (shading) minden képpontban meg kell határozni még a képponthoz tartozó fénysugárnak a vetített lappal alkotott metszéspontjához tartozó fényerősséget, és a lap színével együtt azt is a vizsgált képponthoz kell rendelni. A lapokon hátulról előrefelé ily módon sorba haladva a vetület végül úgy lesz színezve, ahogy az a szemlélő felől látszik.

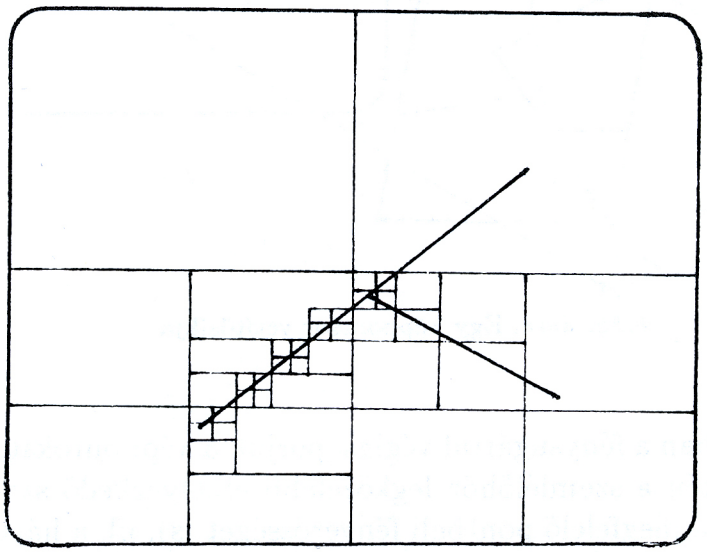


6.20. ábra. Egymást ciklikusan takaró lapok

Mivel az összes lap nem rendezhető sorba, ha köztük olyanok is vannak, amelyek egymást ciklikusan takarják (6.20. ábra) vagy egymást metszik, egy továbbfejlesztett változatban (polygon clipping) minden lapot feldarabolnak a vetületben előálló metszésvonalaknak megfelelően, és minden darabot külön lapként vizsgálnak.

A festő algoritmusnak sok további változata van aszerint, hogy a fénysugarak mentén való számításokat hogyan szervezik, és a megfelelő adatokat hogyan rendezik (pl. z-buffer algoritmus).

A kép irányulású algoritmusok a vetületet a képernyő tartományai vagy képpontsorai szerint állítják elő. Ezek az eljárások a grafikus eszközök fejlődése miatt egyre rövidebb időt igényelnek és egyre inkább elterjednek.

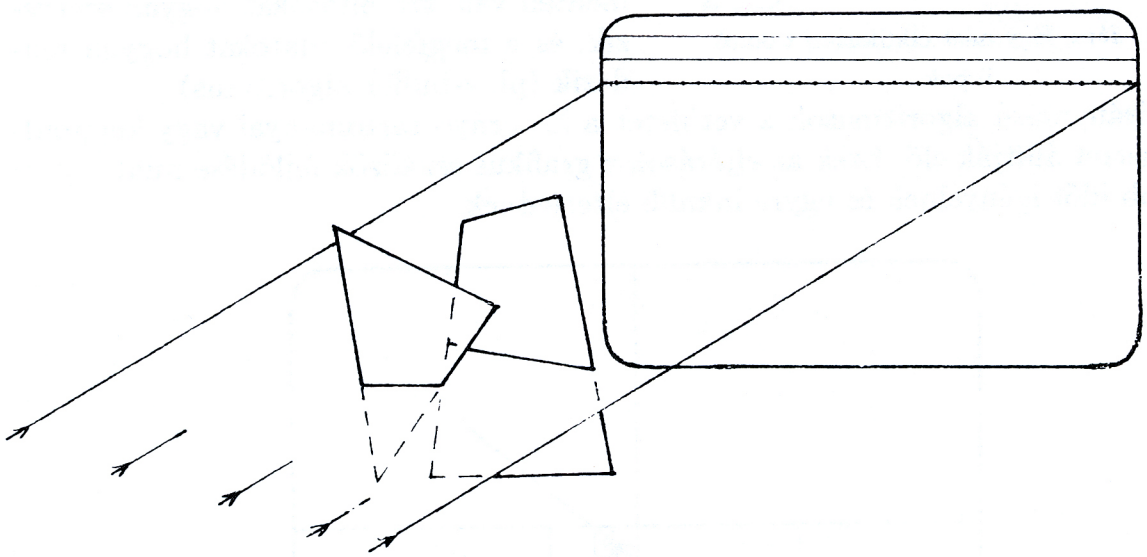


6.21. ábra. *Warnock*-féle ablakfelosztás bal oldalon raszteres képhez, jobb oldalon vonalas képhez

A legrégebbi kép irányulású eljárás a *Warnock*-féle ablakfelosztó algoritmus, amelynek van vonalas képet, ill. raszteres képet előállító változata. Alapelve a következő: a képernyő fokozatos felosztásával olyan ablakokat készítünk, amelyekben

a láthatósági vizsgálat könnyen elvégezhető (6.21. ábra). Vonalas kép elkészítéséhez a felosztást addig finomítjuk, amíg az ablakban csak egy szakasz van, ami egy csúcspont környezetében a képpont méretéig való finomításhoz vezet. Az üres ablakokat el lehet hagyni, a többiben pedig csak az egy szakaszra vonatkozó számításokat kell elvégezni pl. a takart vonalas eljárás szerint. Raszteres kép előállításához az ablakokat addig kell darabolni, amíg azok vagy nem tartalmaznak vonalat, vagy elérik a képpont méretét. Ezután bármelyik ablak egy színnel kitölthető, mégpedig az ablak előtt a szemlélőhöz legközelebb elhelyezkedő lap színével. Ez a vizsgálat hasonlóan végezhető el, mint a festő algoritmus legegyszerűbb változatánál, mert az elkészített ablakok előtt vagy nincs lap, vagy egy lap van, vagy pedig több lap esetén azok egyértelműen rendezhetők.

A pásztázó (scan-line) algoritmus raszteres képet ad, és van poliédermodelleken, ill. másodrendű felületekkel határolt modelleken működő változata. A láthatósági vizsgálat a képernyő képpontsorai szerint történik úgy, hogy egy képpontsorhoz tartozó vetítősíknak (a vetítési irány és a képpontsor síkja) meghatározzuk a lapokkal alkotott metszésvonalait (6.22. ábra).

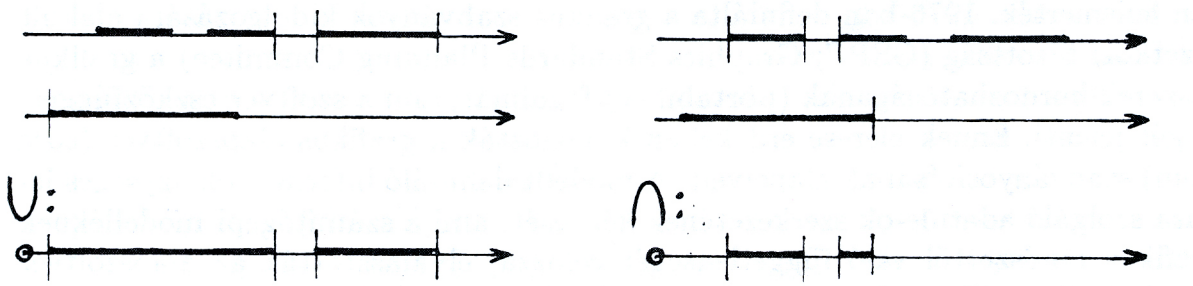


6.22. ábra. Egy képpontsor vetítősíkja

Ezután a vetítősíkban a fénysugárral végigsöpörjük a képpontokat, és mindegyikhez a fénysugár mentén a szemlélőhöz legközelebb elhelyezkedő szakasz színét (és árnyékolt kép esetén a megfelelő pontbeli fényerősséget is), ill. a háttérszínt rendeljük. Másodrendű felületek esetén a vetítősíkban másodrendű görbék adódnak, de a képpontsoron elvégzendő további vizsgálat ugyanaz, mint az előző esetben. Ezeket a számításokat a képernyő minden sorára el kell végezni, ezért a kép előállításához szükséges idő arányos a képernyő képpontjainak számával.

Egy további képernyő irányulású eljárás a CSG-fán működő fénysugárkövető (ray-tracing) algoritmus. Ennél az eljárásnál először az ábrázolandó modell komponenseit elmetszik a képpontokhoz tartozó fénysugarakkal, és minden fénysugárra

feljegyzik azokat az intervallumokat, amelyek valamelyik primitív belsejében vannak. A metszési eljárás függ attól, hogy a primitíveket milyen adatrendszer állítja elő. Görbevonalú felületekkel határolt primitívek esetében közelítő módszerekkel keresik meg a fénysugár felülettel alkotott metszéspontjait. Ezután a CSG-fa feldolgozásával összegzik minden fénysugáron, hogy mely részei haladnak a modellen kívül, ill. annak belsejében, és a megfelelő képponthoz a szemlélőhöz legközelebbi intervallumvégpont színét rendelik. Ez a pont ugyanis a modell felületén a szemlélő felé eső oldalon van (6.23. ábra).



6.23. ábra. Két komponens unióján, ill. metszetén áthaladó fénysugár

A fényképszerű képek generálásánál (rendering) kiszámítják a képpont színét meghatározó felületi pontban a megvilágítás erősségét is, miközben a felület optikai tulajdonságait (fényvisszaverő, fényáteresztő képesség stb.), továbbá a környezetről a szóban forgó pontra eső minden fényhatást (pl. tükröződések) figyelembe vesznek, és a képpont színét ennek megfelelően állítják be.

Mi van a lemezen ?

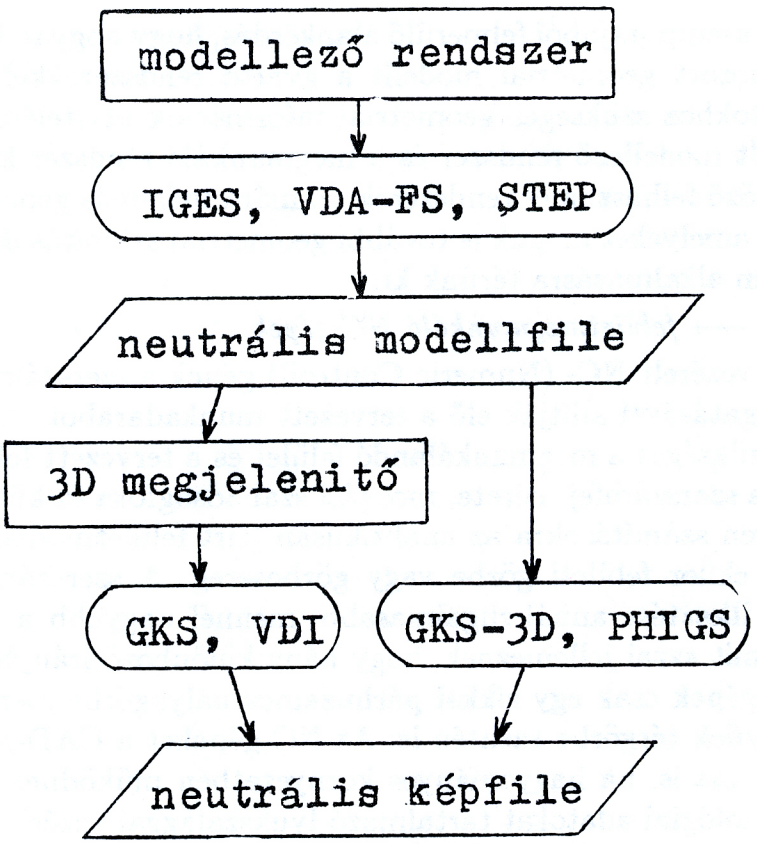
A CAD-iskola hidden-line algoritmusa modellre orientált, vonalas képet szolgáltató eljárás, amely a lapok között fellépő metszésvonalakat is meghatározza. Ezért alkalmas alakzatok áthatásának szemléltetésére is. A számítási eljárás az élek szerint vizsgálja a láthatósági viszonyokat. Mindegyik él vetítősíkjával elmetszi az élt nem tartalmazó lapokat, majd az élnek és az így keletkező metszésvonal szakaszoknak a kölcsönös helyzetét vizsgálja ebben a síkban. Ha az él metszi az egyik ilyen szakaszt, akkor a metszéspont az élnek a szakaszt tartalmazó lappal alkotott döféspontja. A döféspontok összegyűjtése után az áthatási poligon éleire egy hasonló láthatósági vizsgálat következik. Mivel az algoritmus nem vizsgál meg minden speciális esetet, összeeső csúcsok és élek esetében nem keresi meg az áthatási poligont.

6.5. A grafikus rendszerek közötti adatátvitel kérdései

A számítógépi modelleket leíró adatrendszerek különbözősége miatt az egyik CAD-rendszerrel szerkesztett modell adatait egy másik CAD-rendszer általában közvetlenül nem tudja beolvasni és értelmezni, ami egyrészt párhuzamos rendszerfejlesztésekhez vezet, másrészt a rendszer kihasználhatóságát korlátozza. E nehézségek elkerülésére készülnek az adatkonvertáló interfész (interface) programcsomagok, amelyek nemzetközi normáknak (ANSI: American National Standards Institut, ISO: International Organisation for Standardization, DIN: Deutsche Institut für Normung) megfelelő adatrendszerrel írnak le bizonyos számítógépi modelleket. Így ha két CAD-rendszer rendelkezik ugyanazzal az interfésszel, akkor közöttük azok a modellek átvihetők, amelyeket az interfész tartalmaz. Az adatkonvertálás jelentőségét korán felismerték. 1976-ban definiálta a grafikus szabványok kidolgozására alakult nemzetközi bizottság (GSPC: Graphics Standards Planning Commitee) a grafikus rendszerek hordozhatóságának (portability) fogalmát, ami a szoftver eszközfüggetlenségét jelenti. Ennek elérése érdekében kidolgozták a grafikus alapszoftver (core system) szabványosításának alapelveit. A modellt definiáló információk egységes leírására szolgáló adatfile-ok szerkezetének kérdését, ami a számítógépi modelleknek a grafikus rendszertől való függetlenségét célozza, elválasztották az alapszoftver kérdésétől. Ezen elképzelés szerint a modellező rendszer által generált adatrendszert az interfészben rögzített formában egy ún. neutrális modellfile tartalmazza, a képet definiáló grafikus információkat pedig a grafikus alapszoftver segítségével létrehozott neutrális képfile tartalmazza (6.24. ábra).

Elsőként az IGES (Initial Graphics Exchange Specifikation) interfészt fogadták el nemzetközi szabványnak, amely elsősorban síkbeli és egyszerű térbeli drótvázas modellek leírását tartalmazza, továbbá alkalmas nem geometriai információk (pl. szövegek) leírására is. Felépítése meglehetősen rajzra orientált. A VDA-FS (Flächenschnittstelle des Verbandes der deutschen Automobilindustrie) segítségével leírhatók paraméteres polinom függvényekkel megadott görbék és felületek is úgy, hogy minden görbeívhez, ill. felületdarabhoz egy azonosítót, a paraméterintervallumot, a polinom fokszámát és az együtthatóit kell meghatározott formában felírni. A felületek közelítő leírására alkalmas pont-vektor sorozat (felületi pont és felületi normálisok halmaza), továbbá szövegelemek és egyéb nem geometriai (pl. fizikai) információk is definiálhatók ebben az interfészben. A még mindig fejlesztés alatt álló STEP (Standard for the Exchange of Product Model Data) nemcsak geometriai és nem geometriai elemek leírását tartalmazza, hanem topológiai információkat is. Egy formális nyelv (EXPRESS) segítségével összetett modellek is meghatározhatók a már definiált elemek egymáshoz rendelésével.

A gyakorlatban a CAD-rendszerek között, továbbá egyes CAD-rendszerek és felhasználói rendszerek között sokszor speciális, a szóban forgó modellekre fejlesztett adatkonvertáló programokkal végzik el az adatátvitelt. Gyakori megoldás az is, hogy közvetlenül egy széleskörűen elterjedt szerkesztőrendszerhez (pl. AUTOCAD) készítenek interfészt a modelladatok, ill. képfile-ok további feldolgozására.



6.24. ábra. Az interfész helye a grafikus rendszerekben

7. CAD-környezet

Az alkalmazások szempontjából felmerülő alapkérdés, hogy hogyan lehet egy CAD-rendszerrel létrehozott geometriai modellt a gyártó rendszerekkel feldolgozni. A gyártási folyamatokhoz szükséges geometriai információk átvitelére a gyártási folyamatba integrált modellező rendszer és a megmunkáló rendszer közötti interfész szolgál. A különböző felhasználói rendszerek számára más-más geometriai információk szükségesek, amelyeket maguk is további geometriai számításokkal egészítenek ki. Röviden három alkalmazásra térünk ki.

CAD-rendszer ⟶ felületmegmunkáló NC-gépek

A számítógéppel vezérelt NC- (Numeric Controll) gépek a szerszámfejnek a felület mentén való mozgatásával állítják elő a tervezett munkadarabot. A szerszámpálya kiszámításához szükséges a megmunkálandó felület és a tervezett felület numerikus leírása, továbbá a szerszámfej mérete, mozgási szabadságfoka és különböző technológiai adatok. Ilyen számításokra az analitikusan leírt felületmodellek alkalmasak, a szerszámpálya ekkor felületi görbe vagy görbesereg. A szerszámfejnek a munkadarabhoz való illesztése annál rugalmasabb, mennél nagyobb a térbeli mozgási szabadságfoka, amit azzal jellemeznek, hogy hány különböző irányban mozgatható el. A 2-tengelyű gépek csak egy síkkal párhuzamos pályagörbe mentén mozognak, a 3-, 4-, 5-tengelyűek térgörbe mentén is. Az NC-gépeket a CAD-rendszerek megjelenése előtt, ill. ma is, hà hagyományos környezetben működnek, a pályagörbék adatait és a technológiai adatokat tartalmazó lyukszalaggal vezérlik.

CAD-rendszer ⟶ végeselem-rendszerek

Ha egy síkbeli lemezmodell vagy egy térbeli modell belső pontjaiban bizonyos fizikai jellemzőket (hőeloszlás, erőeloszlás, feszültség stb.) kell meghatározni, amelyek adott környezeti behatásokra (melegítés, megterhelés, hajlítás stb.) keletkeznek, akkor célszerű a modellt egyszerű alakú, viszonylag kisméretű, ismert fizikai tulajdonságú elemi részek halmazával helyettesíteni, és a szükséges számításokat ezekre az építőelemekre elvégezni. Ezt a módszert nevezik végeselem módszernek, amellyel egy gyártmány használhatóságát vizsgálják meg. A végeselem analízist vagy egy elkészített mintadarabon, vagy a CAD-rendszerek felhasználásával egy számítógépi modellen végzik el. A CAD-rendszerek integrálásával a gyártmány vizsgálata a következő műveletekből áll: (1) a számítógépi modell előállítása — (2) az elemi részekkel való közelítés (végeselem-háló generálása) — (3) a fizikai behatások, anyagi tulajdonságok, peremfeltételek megadása — (4) a kívánt fizikai értékek kiszámítása minden elemre — (5) az eredmények szemléltetése. Ha a modellező rendszer nem tartalmazza a végeselem-háló generálását,akkor azt egy ún. preprocesszor végzi el, amely a modelladatokat valamilyen interfészen keresztül kapja, és az elemi részek

halmazát adja tovább. A 3. és 4. fázist a végeselem-rendszer végzi el, amely vagy képes az eredmények grafikus megjelenítésére, vagy azt egy ún. posztprocesszor végzi el. A végesèlem-analízis eredményeit szemléltető ábrákon vagy az azonos fizikai értékekhez tartozó pontokat összekötő vonalakat tüntetik fel, vagy a különböző értékekhez tartozó tartományokat különböző színekkel töltik ki (pl. a növekvő feszültségek szerint sárgától pirosig színezve). Összetett modellekre a végeselem-háló elkészítése sokkal gazdaságosabban oldható meg, ha nemcsak a modell adatstruktúrája, hanem szerkesztési módja is ismert. Ezért több olyan CAD-rendszert is kifejlesztettek, amelyek a modell definiálásával egyidejűleg a végeselem-hálót is generálják (ANVIL, ANSYS, PATRAN, NASTRAN, ASKA), sőt bizonyos fizikai jelenségekre vonatkozó végeselem-analízist is el tudnak végezni.

CAD-rendszer ⟶ robotszimuláló rendszerek

Az ipari robotok számítógéppel való megtervezése lényegében geometriai és kinematikai számításokból áll. A munkadarabot és az azt megmunkáló robotot, ill. ezek számítógépi modelljeit egy ún. gyártócellában helyezik el. A munkadarab geometriai adatait, amelyeket egy CAD-rendszer szolgáltat a megfelelő interfészen keresztül, a megmunkálás szimulálásához még további geometriai (pl. forrasztási pontok helye) és gyártástechnológiai adatokkal egészítik ki. A robot mint számítógépi modell leírásához a geometriai adatokon kívül még lehetséges mozgásokat, ill. mozgássorozatokat és technológiai jellemzőket is megadnak. Ha a munkadarab és a robot modellje már a gyártócellában van, a megmunkálási folyamat megtervezése, vagyis a szimulált robot által elvégzendő műveletsor kiszámítása következik. A tervezett mozgások minden időpillanatában meg kell határozni a robot szerszámfejeinek a munkadarabhoz viszonyított kölcsönös térbeli helyzeteit az optimális pályagörbék kialakítására. A robot által végzett mozgások egy részének sorrendjét a robot szerkezete és a gyártási folyamat meghatározza, de az egész kinematikai láncnak egy adott tényező szerint való optimalizálása a robotszimulálás egyik legfontosabb művelete.

A számítógéppel vezérelt gyártási folyamatban részt vevő CAD-rendszer és a rá épülő megmunkáló rendszer, amelyeket összefoglalóan CAM-rendszernek (Computer Aided Manufacturing) neveznek, a termelés nagyon fontos láncszeme, de egy üzem termelésének lebonyolításához még más számítógépi rendszerek is szükségesek. Ilyenek pl. a minőségellenőrző rendszerek, az anyagszükségletet és raktári készletet nyilvántaró rendszerek, a kereskedelmi tényezőket feldolgozó költségtervező rendszerek, a termelési folyamat optimalizálását végző irányító rendszerek stb. E számítógépi rendszerek csak úgy integrálhatók a termelési folyamatba, ha közöttük megfelelő információcsere, azaz adatátvitel működik (pl. a készletfeldolgozó rendszer el tudja olvasni a CAD-rendszer által készített darablistát). Arról eltérőek a vélemények, hogy a különböző számítógépi rendszerek összekapcsolását hogyan célszerű megvalósítani. Egyesek szerint egy központi adatbázist kell létrehozni, amelyhez mindegyik rendszer közvetlenül hozzáfér. Mások a rendszereket egy logikai láncba helyezik el, és a szomszédos rendszereket látják el megfelelő interfésszel, de vannak még egyéb elképzelések is. Ezzel a rövid kitekintéssel csak az volt a célunk, hogy érzékeltetni tudjuk a CAD-rendszerek alkalmazásokban betöltött szerepét.