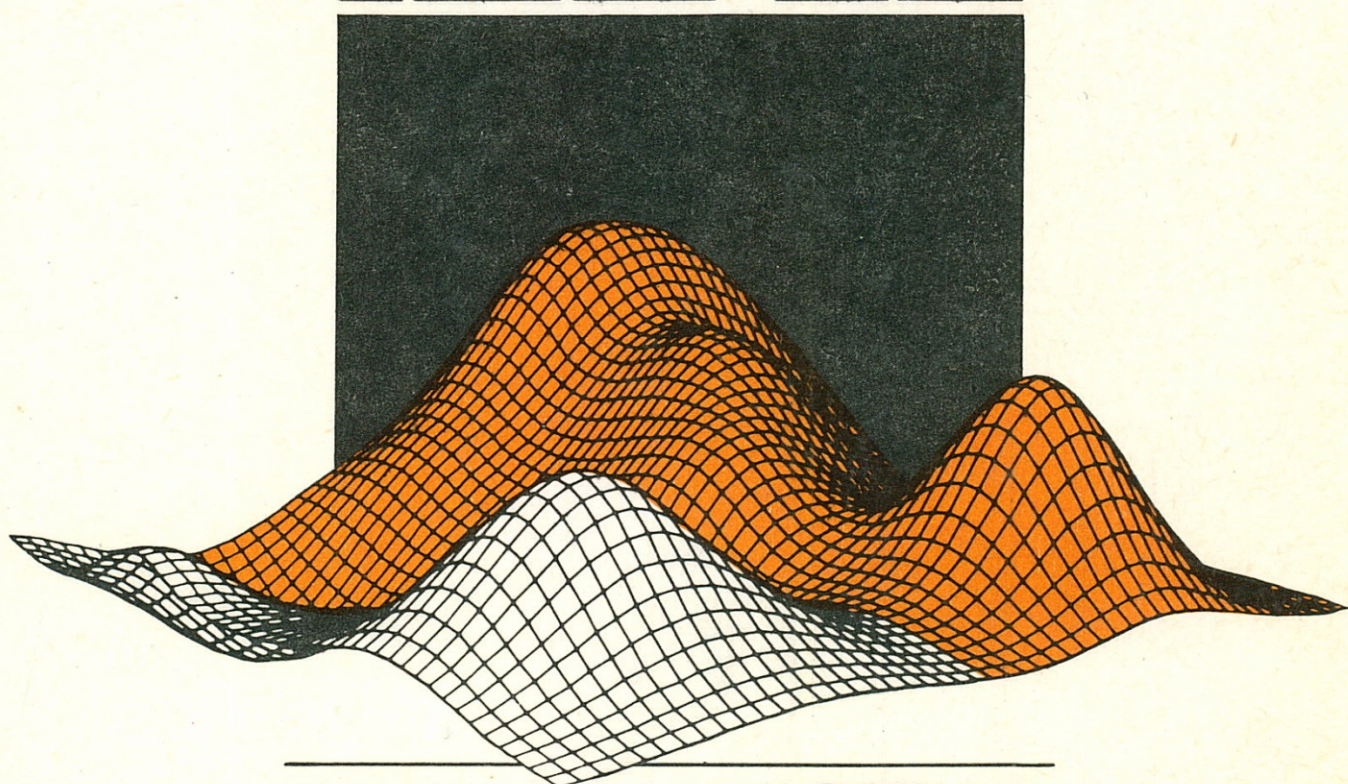
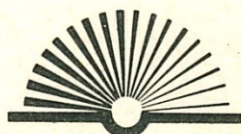

Markus Weber

IBM PC



3-D-Grafika

Elmélet és gyakorlat



MARKUS WEBER

IBM PC 3D-GRAFIKA

Elmélet és gyakorlat

IWT — NOVOTRADE
BUDAPEST, 1987

A könyv eredeti címe: IBM-PC 3D-Grafik. Theorie und Praxis (1985)

Fordította: Manager Informatikai Gmk.

Lektorálta: HONT LÁSZLÓ

A kiadásért felel: RÉNYI GÁBOR, a NOVOTRADE RT. igazgatója

Felelős szerkesztő: SÍK JÚLIA

Műszaki szerkesztő: DÉVÉNYI ERIKA

Szedte: a Neotyp Kiszövetkezet

Készült: a Somogy megyei Nyomdaipari Vállalat kaposvári üzemében

Felelős vezető: MIKE FERENC igazgató

ISBN 963 02 4693 7

Hungarian translation © Manager Informatikai Gmk.

Copyright © 1985 by IWT-Verlag GmbH — Vaterstetten bei München

Minden jog fenntartva. Az IWT cég írásbeli hozzájárulása nélkül tilos a könyvet vagy annak részeit bármilyen eljárással (nyomtatás, fotokópia vagy egyéb technika), elektronikus rendszerek felhasználásával másolni, sokszorosítani, terjeszteni.

TARTALOMJEGYZÉK

ELŐSZÓ	7
Mire alkalmas ez a könyv, és mire nem?	7
Számítógépes grafika — de miért?	11
1. RÉSZ. MATEMATIKAI ALAPOK	17
1.1 VEKTORSZÁMÍTÁS	17
1.1.1 Alapfogalmak	17
1.1.1.1 Vektor és skalár	17
1.1.1.2 Vektorok összeadása és szorzása skalárral	18
1.1.2 Az n -dimenziós tér	20
1.1.2.1 Vektorok lineáris kombinációja	20
1.1.2.2 Vektorkoordináták	22
1.1.2.3 Pontkoordináták	23
1.1.2.4 Példák, alkalmazások	23
1.1.3 Egyenesek és síkok az R^3 -ban	32
1.1.3.1 Paraméteres egyenletek	32
1.1.3.2 Példák, alkalmazások	33
1.1.4 Vektor szorzása vektorral	36
1.1.4.1 A skaláris szorzat	36
1.1.4.2 A vektoriális szorzat	37
1.1.4.3 Példák, alkalmazások	39
1.1.5 Vektortér, bázis, dimenzió	48
1.2 MÁTRIXSZÁMÍTÁS	50
1.2.1 A Gauss-féle eliminációs eljárás	50
1.2.2 Műveletek mátrixokkal	55
1.2.3 Lineáris leképezés	58
1.2.4 Példák egyszerű mátrixműveletekre	65
2. RÉSZ. A GRAFIKA PROGRAMOZÁSÁNAK GYAKORLATA	75
2.1 A GRAFIKA PROGRAMOZÁSA MICROSOFT-BASIC 2.0-BAN	75
2.1.1 A grafikus képernyők felépítése	75
2.1.2 Grafikus utasítások	76
2.1.2.1 A PSET és a PRESET utasítások	76
2.1.2.2 A POINT függvény	77
2.1.2.3 A LINE utasítás	77
2.1.2.4 A PAINT utasítás	78
2.1.2.5 A GET utasítás	79
2.1.2.6 A PUT utasítás	79

2.1.2.7 A BSAVE és a BLOAD utasítások	80
2.1.2.8 A WINDOW utasítás	80
2.2 GEOMETRIAI TRANSZFORMÁCIÓK AZ R^2 -BEN ÉS AZ R^3 -BAN	81
2.2.1 Eltolás	81
2.2.2 Tükrözések	82
2.2.2.1 Középpontos tükrözés	82
2.2.2.2 Tengelyes tükrözés az R^2 -ben	83
2.2.2.3 Síkra vonatkozó tükrözés az R^3 -ban	86
2.2.3 Forgatás	87
2.3 ALAKZATOK GRAFIKUS MEGJELENÍTÉSE	88
2.3.1 Körök	88
2.3.2 Ellipszisek	89
2.3.3 Spirálok	92
2.4 FERDE ÍRÁS	93
2.5 A 3-TEST PROBLÉMA SZIMULÁCIÓJA	96
2.6 ÜZLETI GRAFIKA	101
2.6.1 Oszlopdiagramok	101
2.6.2 Kördiagramok	104
2.7 CAD — SZÁMÍTÓGÉPPÉL TÁMOGATOTT TERVEZÉS	107
2.7.1 A grafikus GET és PUT utasítások	107
2.7.2 Egy egyszerű CAD-program	108
2.8 ROTÁCIÓS MÁTRIXOK	113
2.9 VETÍTÉSI TÍPUSOK	117
2.9.1 Középpontos vetítés	117
2.9.2 Párhuzamos vetítés	119
2.9.3 A rotációs mátrixok és a középpontos vetítés alkalmazásai	120
2.10 3D-GRAFIKA — ELSŐ VÁLTOZAT	125
2.10.1 A háromdimenziós háló	125
2.10.2 A háromdimenziós függvények definíciójához	135
2.10.3 Elevation modell programming (domborzati modellek programozása)	141
2.11 3D-GRAFIKA — MÁSODIK VÁLTOZAT	146
2.12 A HIDDEN-LINE-PROBLÉMA (ELTAKART VONALAK PROBLÉMÁJA)	150
2.13 HÁROMDIMENZIÓS GÖMBÖK	156
2.14 GOMBTÜKRÖZÉS	161
2.15 REAL-TIME — TRÜKKFILM A GET ÉS A PUT UTASÍTÁSOK ALKALMAZÁSÁVAL	165
2.16 VEKTORTÉRTÉKŰ FÜGGVÉNYEK	168
2.16.1 Függvények paraméteres alakja	168
2.16.2 Nemlineáris interpoláció	169

ELŐSZÓ

Mire alkalmas ez a könyv, és mire nem?

A könyv lehetőséget nyújt az Olvasónak arra, hogy válogatott példák tanulmányozásán keresztül ismereteket szerezzen a grafikus ábrázolás, elsősorban az ún. 3D-grafika (a háromdimenziós ábrázolás) elméletéről és gyakorlatáról. Természetesen ez a könyv sem ad valamennyi grafikus ábrázolási feladatra kész megoldást. Ehelyett a számítógépes grafika, mindenekelőtt a 3D-grafika fontosabb alkalmazásait tárgyalja. Bízunk abban, hogy az Olvasó a receptszerű leírások alapján elsajátítja azokat az ismereteket, amelyek képessé teszik saját problémáinak önálló megoldására.

A könyv eredményes tanulmányozásához a számtani alapl műveletek, ill. a MICROSOFT-BASIC programnyelv ismeretén kívül gyakorlatilag semmiféle előismeret nem szükséges. Kívánatos viszont, hogy az Olvasó fogékony legyen az új ismeretek befogadására.

A könyv két, egymástól elkülönülő fő részre tagolódik.

Az első részben kizárólag a grafikus ábrázolás elméletével foglalkozunk. Egyszerű, szemléletes példákon keresztül ismertetjük meg az Olvasót előbb a vektorszámítás, majd a nagy gyakorlati jelentőségű mátrixszámítás alapvető tudnivalóival.

A második részben részletesen foglalkozunk a gyakorlati programozással. Először egyszerű grafikus ábrák készítésére alkalmas példaprogramokat mutatunk be, majd — részben ezekre a példákra épülő — összetettebb programokat, amelyek szemléltetik a számítógépes grafika alkalmazásának széles körű lehetőségeit. Részletesen tárgyaljuk a következő témaköröket:

- egyszerű geometriai alakzatok programozása;
- szabadon választható irányú írásformák;
- rendszerszimulációk;
- üzleti grafika;
- CAD (számítógéppel támogatott tervezés);
- forgatások;
- perspektivikus ábrázolás;
- hálógrafikák;
- a *hidden-line*-probléma (eltakart vonalak ábrázolási problémája);
- gömb és gömbtükrözés;
- *real-time* — trükkfilmek;
- a nemlineáris interpolációs eljárások grafikai alkalmazása.

A programokat az IBM PC, ill. az IBM PC-vel kompatibilis gépeken igen elterjedt MICROSOFT-BASIC 2.0 programnyelven írtuk. A könyv tartalmaz egy fejezetet a MICROSOFT-BASIC 2.0 programnyelv sajátosságairól is, hogy ne zárjuk ki az Olvasók köréből azokat, akik más BASIC-változattal dolgoznak.

E helyen szeretném megköszönni *Herrmann*, *Schnellhardt* és *Tröger* uraknak baráti támogatását és a kellő időben adott ösztönzését a könyv megírására. A kiadónak, különösen *Grohmann* úrnak köszönetemet fejezem ki a könyv kiadásáért, valamint nagy türelméért, amivel a számtalan probléma megoldásában mellettem állt.

*A rend a logika birodalma,
a rendetlenség pedig a fantáziáé.*
George Bernanos

*Négy hét után jön a teoretikus és jelenti:
végre sikerült az elefántot egy gömbbé transzformálnom,
most már elkezdhetünk számolni.*
Erich Sackmann

Számítógépes grafika — de miért?

Számítógépes grafika — jól hangzik, de vajon mire való? Ez a kérdés egyszerűen megválaszolható: a számítógépes grafika segítségével a száraz matematikai tények szemléletes formában jelennek meg a megfigyelő szeme előtt. No, és miért oly előnyös ez? A válasz ismét nagyon egyszerű: mit mondanak pl. az Olvasónak a következő számpárok:

$$\begin{pmatrix} 1.232 \\ 1.866 \end{pmatrix}; \begin{pmatrix} 2.232 \\ 0.134 \end{pmatrix}; \begin{pmatrix} -1.366 \\ 0.366 \end{pmatrix}; \begin{pmatrix} -0.366 \\ -1.366 \end{pmatrix}?$$

Nagy valószínűséggel semmit. Egycsapásra megváltozik azonban a helyzet, ha ezeket a pontokat egy koordináta-rendszerben ábrázoljuk.

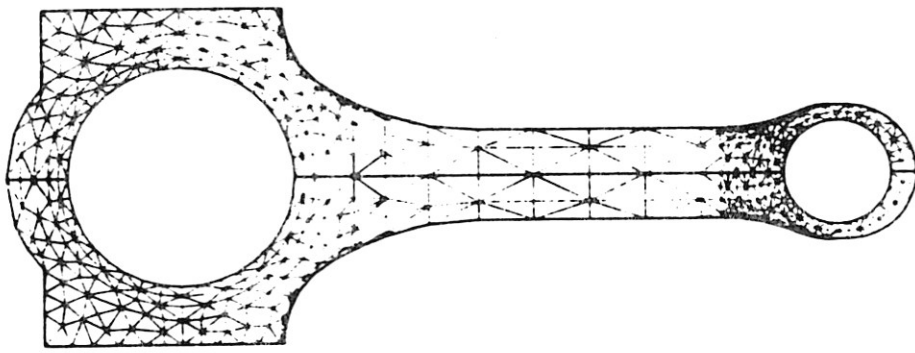
Az emberi agy nyilvánvalóan nem arra van berendezkedve, hogy digitális formájú információkat dolgozzon fel. Azt szoktuk meg, hogy az információkat elsősorban a szem révén, másodsorban pedig a hallás és tapintás útján nyerjük, és ennek megfelelően dolgozzuk fel. Nem véletlenül terjedt el azonban a mondás: *a gyakorlat teszi a mestert*. Némelyek a számokkal való mindennapos tevékenységük során akkora gyakorlatot szereznek, hogy az ilyen jellegű — de nem túl nagy mennyiségű — információkat fejben képesek feldolgozni. De ezek csupán egyedi esetek, azok a kivételek, akik az előbbi szabályt erősítik. A nagy többség (a szerzőt is beleértve) soha nem lesz erre képes. A felmerülő nehézségeket mindazok az Olvasók érezni fogják, akik még soha nem foglalkoztak vektorszámítással. Megfelelő gyakorlat híján kezdetben bizonyára nem lesz könnyű a magyarázatok — különösen pedig a háromdimenziós térbeli gondolatmenetek — követése. De ne rettenjenek meg: végül is egy háromdimenziós világban élünk! Hamarosan kiderül, hogy némi gyakorlat megszerzése után a magyarázatokat már nem is olyan nehéz követni.

A számítógép nem tud három dimenzióban számolni, sőt két dimenzióban sem — a gép „digitalizált” információkhoz kötött. Képes azonban a számítógép arra, hogy ezeket az információkat alkalmas programokkal könnyen áttekinthető rajzokká alakítsa. Ez rendkívüli jelentőségű, mert az egyre nagyobb teljesítményű számítógépekkel grafikus ábrázolást igénylő, bonyolult problémák is megoldhatók. A számítógép nem véletlenül örvend oly nagy népszerűségnek az iparban és a kutatásban. A számítógéppel a kísérletek gyakran egyszerűbben és lényegesen olcsóbban elvégezhetők, esetleg teljesen mellőzhetők is.

Ennek szemléltetésére bemutatjuk az ún. végeselem- (*finite elemente*) módszer ipari alkalmazását egy példán keresztül.

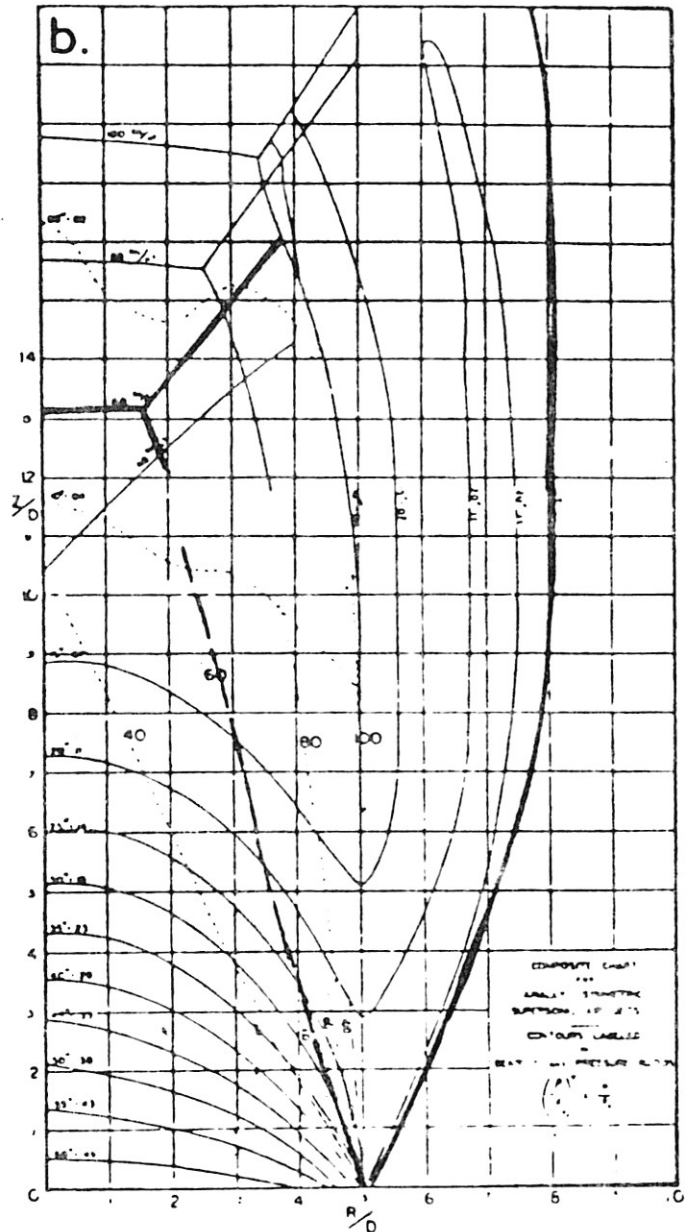
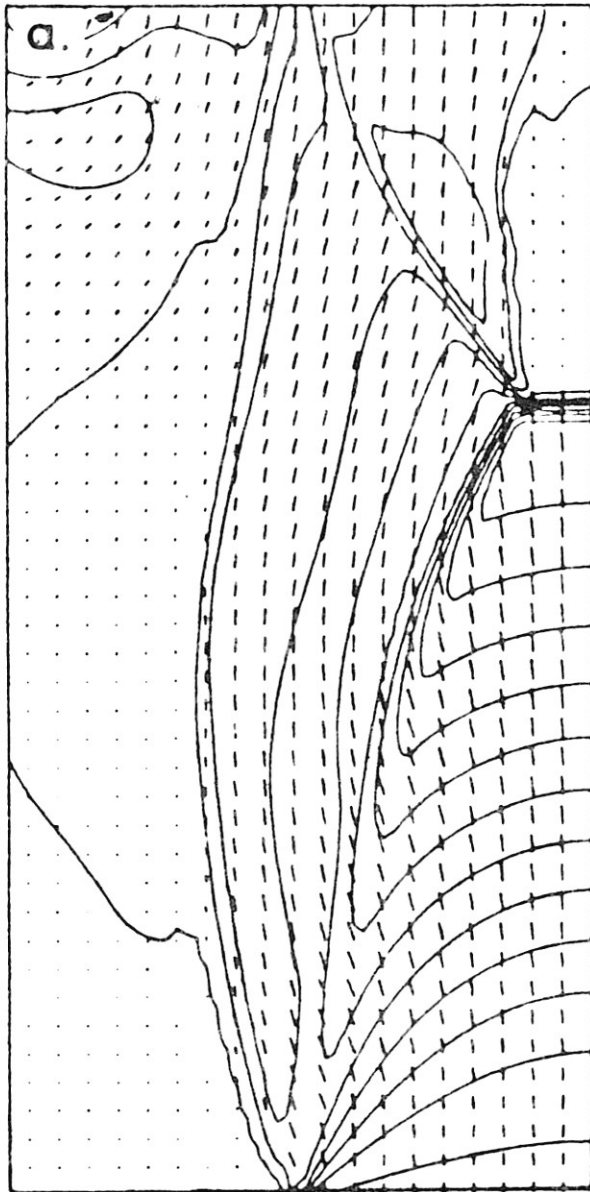
Egy gépjármű-hajtókar „egészének” üzemszerű viselkedését nem tudjuk leírni valamely fizikai törvénnyel. Ezért a számítógépes kísérlet (szimuláció) során a hajtókart véges (*finite*) sok, igen kicsi, ún. elemi részre bontjuk. Ezekre az elemekre jó közelítéssel érvényesek egyszerű fizikai törvényszerűségek, amelyek összességükben már leírják a hajtókar működését (szuperpozícionálás elve).

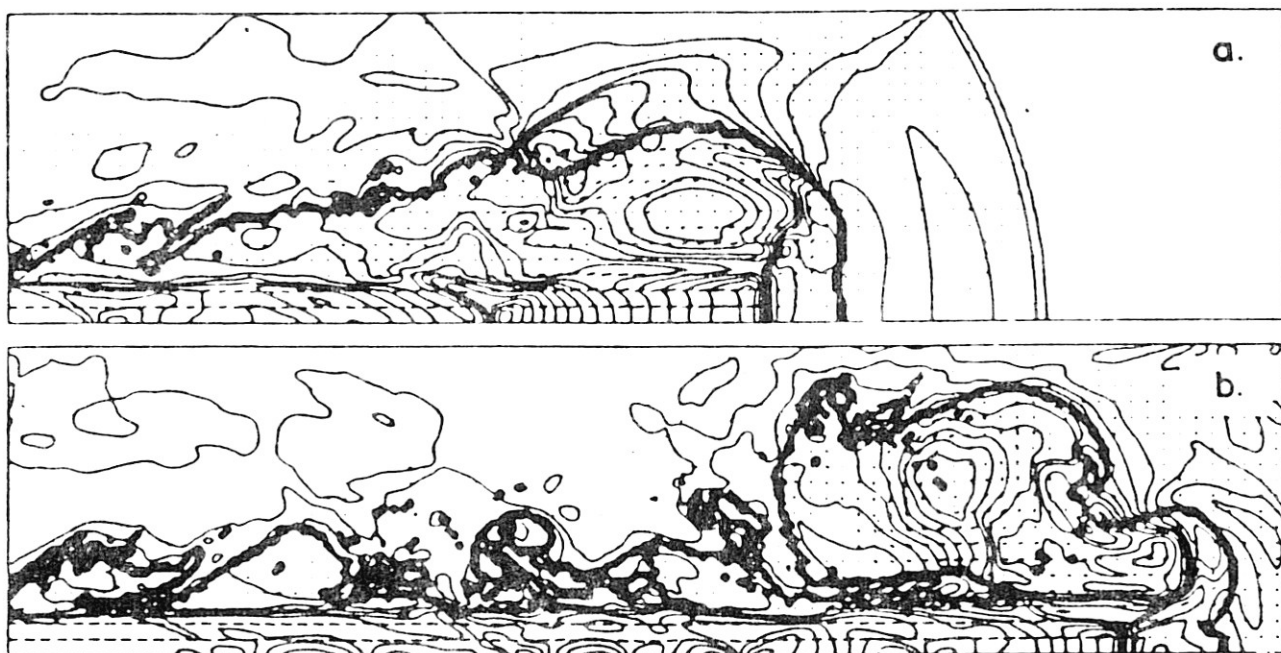
A számítógépes kísérlettel megfigyelhető, hogy hogyan változik a hajtókarban ébredő feszültség eloszlása különböző terhelés hatására, a hajtókar felépítésétől függően. Az információt grafikus úton jelenítjük meg. Ezzel el is jutottunk a számítógépes grafika elsődleges alkalmazásához: nevezetesen az óriási adatmennyiségek feldolgozásához.



Az adatforrások alapvetően kétfélék lehetnek. *Külső adatforrás* a felhasználónak az aktuális számítógépes feldolgozást megelőzően létező adathalmaz. *Belső adatforrás* úgy jön létre, hogy a feldolgozandó adatok maguk is az aktuális számítógépes feldolgozás közben keletkeznek. Számítógépes kísérlettel nyerhető belső adatforrásra mutatnak további példákat a következő ábrák.

Belső adatforrással elsősorban olyan számítógépes kísérletek során találkozunk, mint amelyet az előbbieken is vázoltunk. Az első ábrán egy sugárhajtóműből kiáramló gázsugar elemzési adatait hasonlítjuk össze. A számítógéppel számított (a) és a kísérleti úton nyert eredmény (b) szemmel láthatóan jól egyezik.





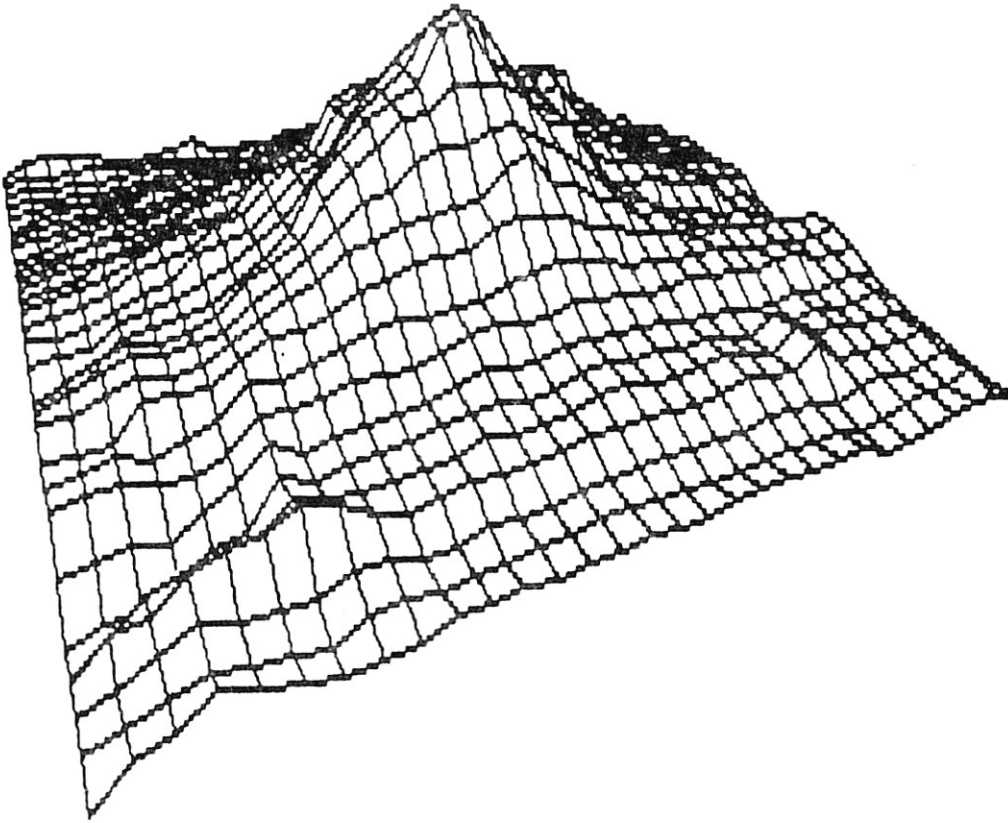
A másik ábra egy jetfejeknek a nálánál tízszer nagyobb anyagsűrűségű közegbe való behatolása során kialakuló két állapotát mutatja. (Jetnek nevezik a fizikában, ill. a csillagászatban azt a sugárzást, ami egy objektumból, pl. egy kozmikus rádióforrásból áramlik ki.) Ebben a számítógépes kísérletben a kis sűrűségű, a hangsebességnél (≈ 340 m/s) háromszor nagyobb sebességgel kilépő gázsugár fejrésznél gombaszerű örvények keletkeznek. Ezeket a számításokat 1980-ban *Michael Norman* és *Karl-Heinz Winkler* a *Garchingi Max Plank Intézet* munkatársai, *Larry Smarr* és *Michael Smith* az *Illionois-i Egyetem* munkatársai végezték el a világ akkoriban leggyorsabb számítógépén, a CRAY-1-en. Az egyébként szükséges 50 óra számítási idő helyett a CRAY-1-en 2-3 órára volt szükség. Figyelembe véve azt a tényt, hogy a CRAY-1 másodpercenként több mint egymillió művelet elvégzésére képes, gyakorlatilag nem lehetett a számítógép ontotta adattömeg grafikus feldolgozásáról lemondani.

A számítógépes grafika egy másik nagy felhasználási területe a külső adatforrások, elsősorban a mérési adatok feldolgozása.

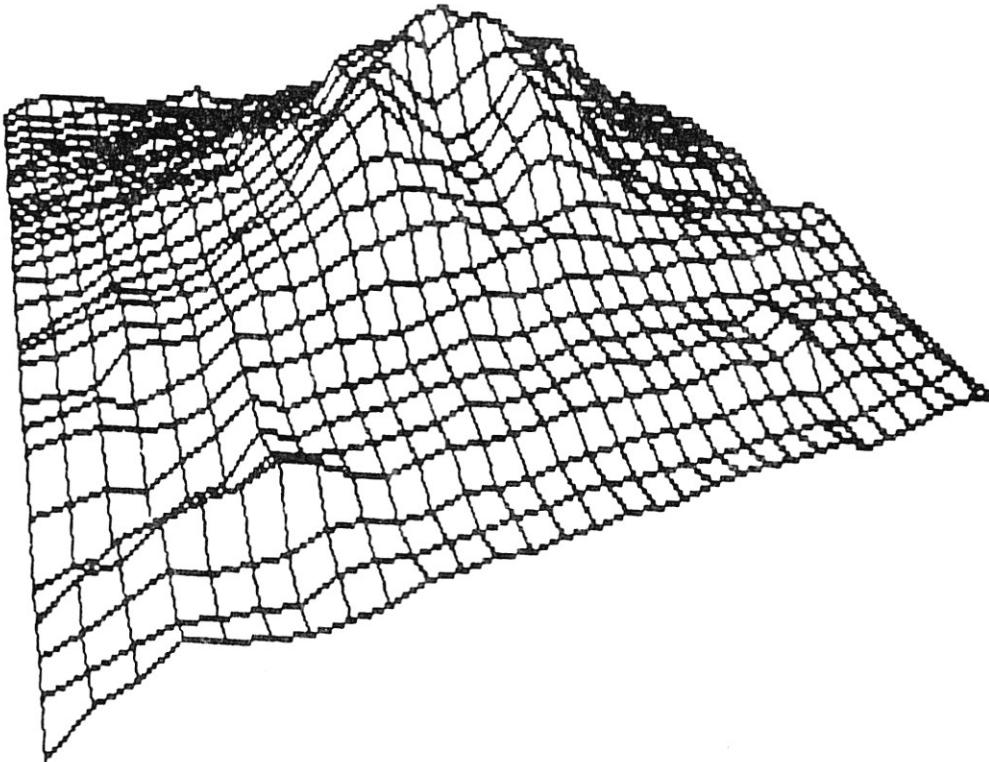
1980. május 18-ig az USA Washington államában található *Szt. Helen* vulkán igazi „képeskönyvbe illő” vulkánnak számított. E napon azonban egy kitörés során a robbanás a vulkán tetejéből 400 m-nyit letépett, mély krátert hagyva vissza. A számítógépes grafikák a vulkánkitörés előtt (a) és után (b) mutatják a hegyet, különböző irányokból. (Ezeket a szerző készítette az *US Geological Survey* adatai alapján.)

A számítógépes grafikák alkalmazására példa a fényképek feldolgozása is, mindenekeelőtt a csillagászatban. Az emberi szem gyakran nagyon megbízhatatlan, mert a kis fényerősségeket a nagyok mellett nem képes felismerni. Ilyenkor az ún. színekódolási (*colour coding*) eljárást alkalmazzák: minden fényerősségértékhez egy meghatározott szint rendelnek hozzá, és így — hála a számítógép lényegesen nagyobb érzékenységének — a korábban láthatatlan struktúrák láthatóvá válnak. (A számítógép természetesen nem a fényerősségeket, hanem azok számszerű értékeit különbözteti meg.) Ez a módszer azért érdekes, mert azokban a hullámhossztartományokban (pl. az infravörös-, valamint a röntgen- és a rádiósugárzás tartományában) is alkalmazható, amelyekben az emberi szem vak.

A következő ábrán látható képek PKS 2208–137, valamint PKS 0548–322 pulzárak intenzitáseloszlását mutatják 1490 MHz frekvencián, azaz a rádióhullámok tartományában.



3D-V2.6 (c) 1984. Objektum: Szt. Helen hegy vulkánkitörés után



(A pulzások sugárzásuk nagyfokú polarizációjával és intenzitásuk erős ingadozásával tűnek ki.)

Ezért volt logikus, hogy a fejlődés a csillagászat hagyományos eszközeitől, a közvetlen megfigyeléstől és a fényképezéstől egyre inkább az új eszköz, a számítógép irányába törekedett.

Miért kellene a viszonylag kevésbé érzékeny fotolemezek nehézkes útját járni, ha a rohamosan fejlődő elektronika és a számítógépek alkalmazásával lényegesen jobb eredményeket célozhatunk meg? A következmény azoknak a chipeknek (áramkörü lapkáknak) a kifejlesztése lett, amelyeket az amerikai megjelölés kezdőbetűi alapján CCD-chipeknek nevezünk (CCD = *charge coupled device*, töltéscsatolt eszköz). Ezek a fényképezési lemezt gyakorlatilag teljesen helyettesíteni tudták. Egy CCD-chip alapjában véve nem más, mint egy túlméretezett szilíciumchip, de rajta a mikrométerű tranzisztorok helyett kicsi, fényérzékeny mezők, az ún. pixelek találhatók (a képelemek számára). Erősen leegyszerűsítve: minden pixel egy fotonszámláló, ami a fotoeffektus alapján működik.

A fényátalakító tulajdonságú anyagok (szilícium) képesek a fotonokat, tehát a fényt közvetlenül elektromos töltésekké alakítani. (Ez a jelenség az alapja annak, hogy a különböző fajta mesterséges holdakat napelemek révén energiával tudjuk ellátni.) Ha egy meghatározott ideig a CCD-chipet sugárzás hatásának tesszük ki, akkor valamennyi pixelben a felhalmozódott töltés (*charge*) arányos lesz a pixelt ért sugárzás erősségével. Ilyen „megvilágítás” után ezeket a töltéseket a számítógép beolvashatja az operatív tárba.

A CCD-chipet nemcsak a digitális kijelzési mód tette közkedveltté. A legújabb CCD-chipek pixelszáma eléri a 640 000-et. Ezek 800×800 elemből állnak és a spektrumnak a kéktől a sárgáig terjedő tartományában a legjobb fotolemezeknél tízszer érzékenyebbek. Ez lehetővé teszi, hogy lényegesen kisebb, de ugyanolyan teljesítményű távcsöveket építhessünk. Ma már a világ egyik legnagyobb távcsövének, a *Mount Palomar* távcsövének 5 m-es tükörét könnyen helyettesítené egy CCD-chippel felszerelt 0,4 m-es tükör.

A számítógépes grafikának a csillagászat különösen kedvelt felhasználási területe. Így határozhatók meg a színeltolódások (*false colour coding*) alapján pl. a sebességeloszlások a galaxisokban. A módszer azon alapul, hogy a mozgásban levő részecske által kibocsátott sugárzásra érvényes a *Doppler*-jelenség. Eszerint a Földtől távolodó részecskék sugárzásának hullámhossza megnő, azaz a spektrum vörös tartománya felé tolódik, ill. a Földhöz közeledőké megrövidül, azaz a kék tartomány felé tolódik el. A fellépő frekvenciaeltolódás mindenesetre oly csekély, hogy az emberi szem nem képes érzékelni. Ezzel szemben könnyű ezt a számítógépnek felismerni és megfelelő diagramon ábrázolni.

1. RÉSZ

MATEMATIKAI ALAPOK

1.1 Vektorszámítás

A következőkben röviden ismertetjük a vektorszámítás alapvető tudnivalóit, mivel ezek az ismeretek a könyvben bemutatott programok megértéséhez nélkülözhetetlenek. Általánosan azt mondhatjuk, hogy a vektorszámítás a matematikai alapja a gyors és pontos számítógépes grafikák készítésének. Első pillantásra úgy tűnhet, hogy e könyv a vektorszámításról, nem pedig a számítógépes grafikáról szól. Valójában a matematika a legalapvetőbb grafikai segédeszköz, és sehol másutt nem tapasztaljuk a matematikai műveletek hatását olyan gyorsan, mint a számítógépes grafikában.

A könyvben bemutatott programokat az Olvasó csekély módosítással saját feladatainak megoldásához is kiválóan felhasználhatja (pl. mérési eredmények grafikus értékelésére). A számítógépes grafikák elsősorban arra valók, hogy az áttekinthetetlen számoszlopokat megfogható, szemléletes formában jelenítsék meg.

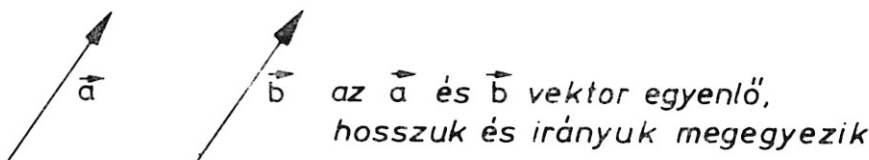
1.1.1 ALAPFOGALMAK

1.1.1.1 Vektor és skalár

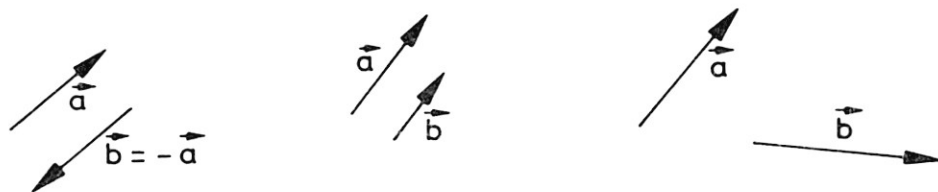
A matematikában megkülönböztethetünk két alapvetően eltérő mennyiséget. Azokat a mennyiségeket, amelyeknek az értéke valós számokkal kifejezhető (pl. tömeg, hőmérséklet, töltés, hosszúság), *skalároknak* nevezzük. Ezekkel szemben azokat a mennyiségeket, amelyek egy szám és a tér egy irányának együttes megadásával jellemezhetünk (pl. sebesség, erő, elektromos és mágneses térerősség), *vektoroknak* nevezzük. Szemléletesen: a vektor irányított szakasz, egy „nyíl” a térben. (A jelölése: \mathbf{a} , \vec{a} , \vec{a} vagy $\overrightarrow{P_1P_2}$, $\overline{P_1P_2}$, ahol P_1 a vektor kezdőpontja, P_2 a végpontja). A szakasz hosszát az \mathbf{a} vektor nagyságának vagy abszolút értékének nevezzük, a jelölése: $|\mathbf{a}|$ vagy a .

Egységvektorok azok a vektorok, amelyek hossza egységnyi, vagyis $|\mathbf{a}| = 1$. A $\mathbf{0}$ *nullvektor* olyan vektor, amelynek kezdő- és végpontja egybeesik; a hossza tehát nulla, aránya nem meghatározott.

Két vektorról akkor mondjuk, hogy egyenlők, ha nagyságuk és irányuk is azonos.

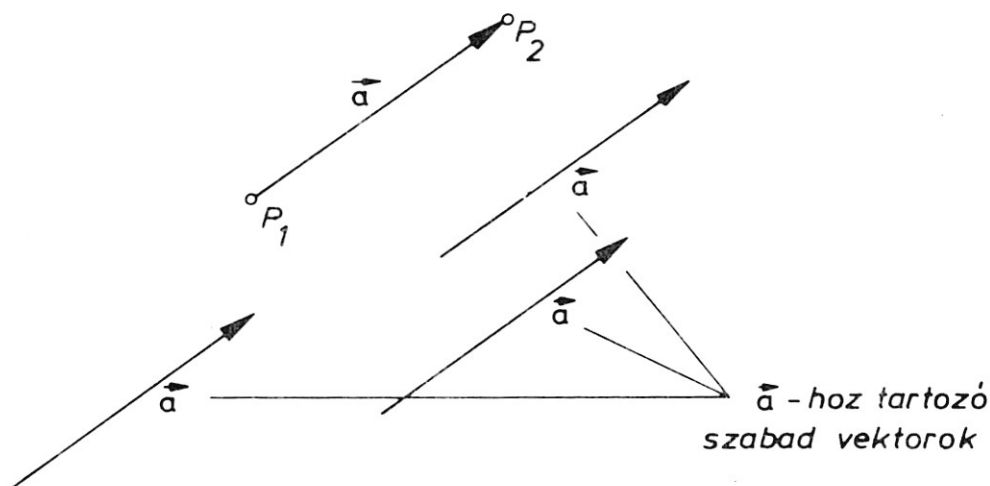


Példák nem egyenlő vektorokra:



A vektoroknak két csoportját különböztetjük meg:

- *szabad vektoroknak* nevezzük azoknak a vektoroknak a halmazát, amelyek egy adott \mathbf{a} vektor párhuzamos eltolásával keletkeznek (azaz valamennyi \mathbf{a} -val egyenlő vektor);
- *\mathbf{a} kötött vektorok (helyvektorok)* meghatározott okok miatt nem tolhatók el, ezek a tér egy meghatározott egyeneséhez vannak rendelve.



A vektorok ezen definíciója szemléletes ugyan, de matematikai értelemben nem kielégítő. Pl. a függvények is felfoghatók vektorokként, de ezzel a problémával a továbbiakban nem foglalkozunk. Figyelmünket csak olyan vektorokra korlátozzuk, amelyekre a megadott definíciója teljesen kielégítő.

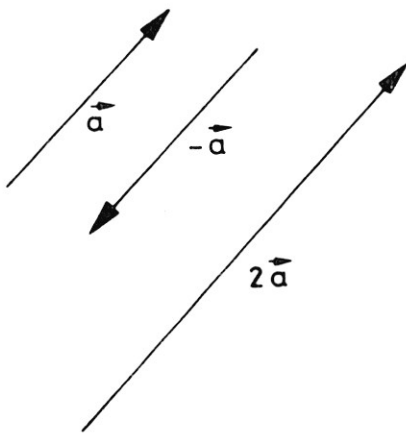
1.1.1.2 Vektorok összeadása és szorzása skalárral

Legyen k egy valós szám (jelölése: $k \in R$, ahol R a valós számok halmaza), és \mathbf{a} egy vektor. Ekkor a $k\mathbf{a}$ szorzat az a vektor, amelynek hosszúsága $|k| \cdot |\mathbf{a}|$, iránya $k > 0$ esetén megegyezik, $k < 0$ esetén pedig ellentétes az \mathbf{a} vektor irányával.

Egy vektort pozitív skalárral szorozva tehát csak a vektor nagysága változik meg, de iránya nem; negatív skalárral való szorzáskor pedig a vektor nagysága és iránya is megváltozik.

Speciálisan $k = -1$ esetén $-1\mathbf{a} = -\mathbf{a}$, vagyis $-\mathbf{a}$ nagysága azonos \mathbf{a} nagyságával, de iránya ellentétes. Abban az esetben, ha $k = 0$, a $k\mathbf{a}$ értéke $0\mathbf{a} = \mathbf{0}$.

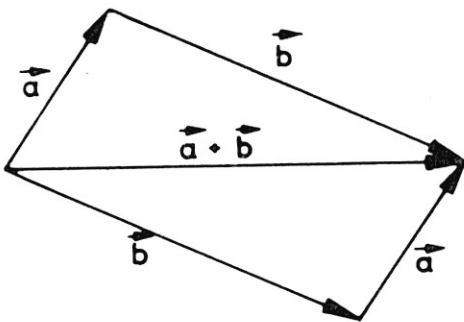
Ha \mathbf{a} és \mathbf{b} két vektor, akkor összegüket a következőképpen értelmezzük: párhuzamos eltolással az \mathbf{a} és \mathbf{b} vektorokat fűzzük egymáshoz, helyezzük át a \mathbf{b} vektor kezdőpontját az \mathbf{a} vektor végpontjához. Az $\mathbf{a} + \mathbf{b}$ összegvektor az a vektor lesz, amelynek kezdőpontja azonos az \mathbf{a} vektor



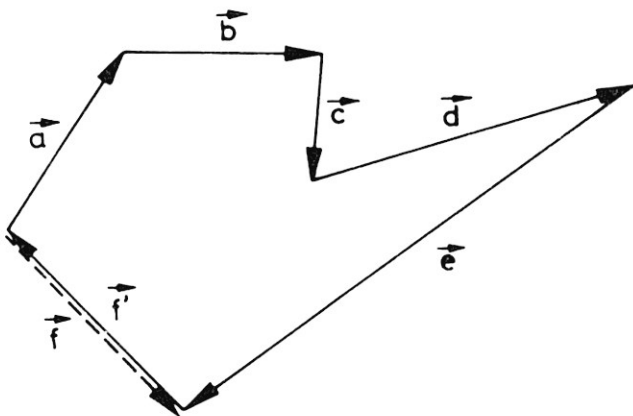
kezdőpontjával, végpontja pedig azonos a \mathbf{b} vektor végpontjával. Mint az a következő rajzon látható, mindegy, hogy a \mathbf{b} vektort fűzzük az \mathbf{a} vektorhoz, vagy az \mathbf{a} -t a \mathbf{b} -hez, azaz:

$$\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}.$$

Ezt a tulajdonságot kommutativitásnak nevezzük.



Több vektor egymáshoz fűzésével — azaz összeadásukkor — irányított töröttvonal keletkezik.



Általánosan az $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots, \mathbf{n}$ vektorok $\mathbf{a} + \mathbf{b} + \mathbf{c} + \dots + \mathbf{n}$ összegét azzal az \mathbf{f} vektorral definiáljuk, amely ellentétes irányú a töröttvonal kezdőpontjába visszatérő, sokszöggé záró \mathbf{f}' vektorral:

$$\mathbf{a} + \mathbf{b} + \mathbf{c} + \dots + \mathbf{n} + \mathbf{f}' = \mathbf{0},$$

ill.

$$\mathbf{a} + \mathbf{b} + \mathbf{c} + \dots + \mathbf{n} = -\mathbf{f}' = \mathbf{f}.$$

Az $\mathbf{a} - \mathbf{b}$ különbséget az \mathbf{a} és a $-\mathbf{b}$ vektorok összegeként értelmezzük:

$$\mathbf{a} - \mathbf{b} = \mathbf{a} + (-\mathbf{b})$$

Általánosan érvényesek a következő műveleti szabályok:

$$\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$$

kommutativitás

$$\mathbf{a} + (\mathbf{b} + \mathbf{c}) = (\mathbf{a} + \mathbf{b}) + \mathbf{c}$$

asszociativitás

$$k(l\mathbf{a}) = (kl)\mathbf{a}$$

$k, l \in R$

$$k(\mathbf{a} + \mathbf{b}) = k\mathbf{a} + k\mathbf{b}$$

disztributivitás

$$k\mathbf{a} = |k| \cdot |\mathbf{a}|$$

$k \in R$

Mint látható, a számok összeadásával és szorzásával kapcsolatos szabályok a vektorokra is érvényesek.

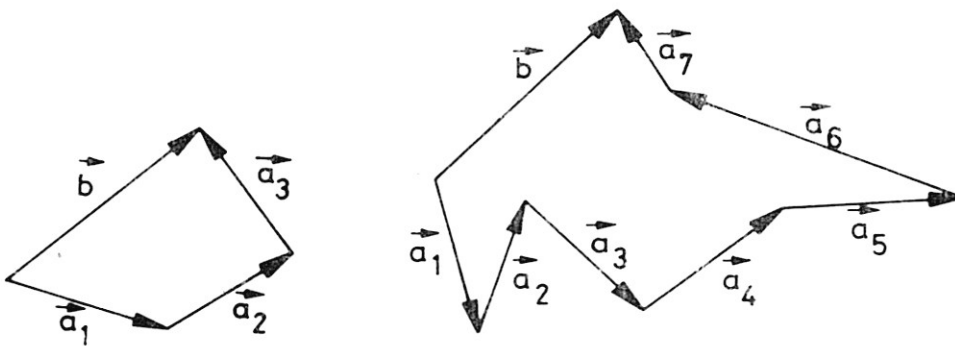
1.1.2 AZ n -DIMENZIÓS TÉR

1.1.2.1 Vektorok lineáris kombinációja

Az n -dimenziós tér fogalmának megértéséhez szükségünk van a vektorokkal kapcsolatos néhány fogalom ismeretére.

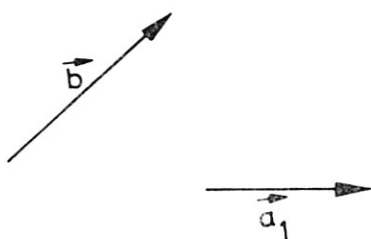
Az $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ vektorok lineáris kombinációjaként a $\mathbf{b} = k_1\mathbf{a}_1 + k_2\mathbf{a}_2 + \dots + k_n\mathbf{a}_n$ vektort értjük, ahol a $k_r, 1 \leq r \leq n$ együtthatók valós számok.

Szemléltessük ezt egy példán:



A \mathbf{b} vektor mindkét esetben az \mathbf{a}_r vektorok lineáris kombinációjaként áll elő. Nyilvánvalóan végtelen sok lehetőség van arra, hogy különböző \mathbf{a}_r vektorok összegeként fejezzük ki a \mathbf{b} vektort. Elsősorban az a kérdés, hogy létezik-e minimális számú \mathbf{a}_r vektor, amelyek lineáris kombinációjaként bármely \mathbf{b} vektor előállítható.

Tekintsünk egy példát:

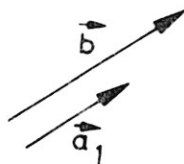


A \mathbf{b} vektor nyilvánvalóan nem fejezhető ki a

$$\mathbf{b} = k\mathbf{a}_1; \quad k \in R$$

formában, mivel az \mathbf{a}_1 és a \mathbf{b} vektorok iránya különböző.

Nézzük a következő példát:



Ebben a példában a \mathbf{b} vektort kifejezhetjük az \mathbf{a}_1 vektor lineáris kombinációjaként, mivel \mathbf{a}_1 és \mathbf{b} vektorok csak nagyságukban különböznek egymástól, irányukban nem. Érvényes tehát:

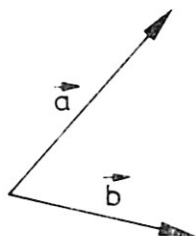
$$\mathbf{b} = k\mathbf{a}_1; \quad k \in \mathbb{R}.$$

Két vektor, \mathbf{a}_1 és \mathbf{a}_2 alkalmasan megválasztott k_1, k_2 együtthatókkal mindig eleget tesz a következő feltételnek:

$$k_1\mathbf{a}_1 + k_2\mathbf{a}_2 = \mathbf{0}.$$

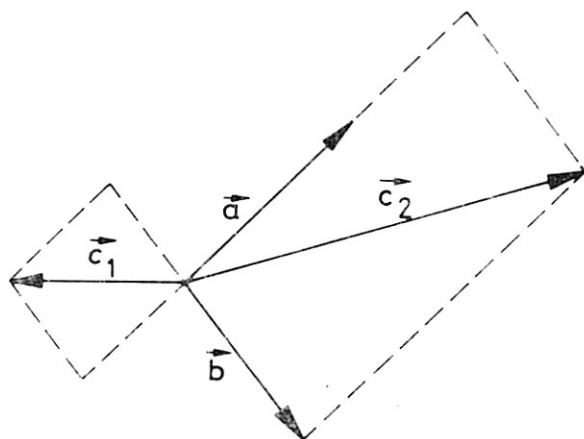
Az \mathbf{a}_1 és \mathbf{a}_2 vektorokat lineárisan függetleneknek mondjuk akkor, ha az egyenletnek nincs a triviálistól (azaz $k_1 = k_2 = 0$ -tól) különböző megoldása.

Nézzünk meg most egy kétdimenziós teret, pl. a papír síkját:



Az \mathbf{a} és \mathbf{b} vektorok lineárisan függetlenek, mivel a feltételt leíró egyenletre csak a $k_1 = k_2 = 0$ együtthatók adnak megoldást.

Felmerül a kérdés, hogy találhatók-e még további vektorok (pl. egy \mathbf{c} vektor), amelyek szintén az \mathbf{a} és \mathbf{b} vektorok síkjában fekszenek, de ezektől lineárisan függetlenek. Az ábrából látható, hogy ez nem lehetséges.



Azt tapasztaltuk tehát, hogy két, lineárisan független vektorral a sík valamennyi más vektora kifejezhető. Általános megfogalmazásban a definíció: *a tér dimenziója egyenlő a tér lineárisan független vektorainak maximálisan lehetséges számával.* (A pontos matematikai bizonyításra — bár nem különösen nehéz — itt nem térünk ki.)

1.1.2.2 Vektorkoordináták

A továbbiakban vizsgálódásunkat a kétdimenziós térre (röviden R^2 ; olvasd R kettő) korlátozzuk, de minden, amit elmondunk, hasonló formában érvényes az R^n -re is.

Legyenek \mathbf{e}_1 és \mathbf{e}_2 az R^2 lineárisan független vektorai. Akkor az R^2 minden \mathbf{b} vektora kifejezhető

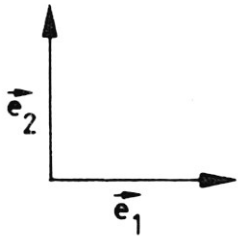
$$\mathbf{b} = b_1\mathbf{e}_1 + b_2\mathbf{e}_2; \quad b_1, b_2 \in R$$

alakban. Adott $\mathbf{e}_1, \mathbf{e}_2$ esetén \mathbf{b} irányát csak a b_1 és b_2 együtthatók határozzák meg. Ezeket a \mathbf{b} vektor ($\mathbf{e}_1, \mathbf{e}_2$) bázisra vonatkoztatott koordinátáinak nevezzük. Szokásos írásmódja:

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \text{vagy} \quad \mathbf{b} = (b_1, b_2).$$

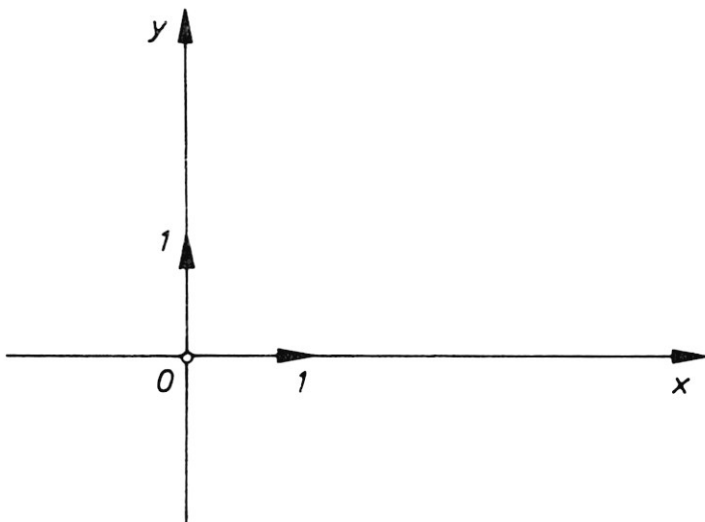
Bázisnak az \mathbf{e}_i lineárisan független vektorok azon minimális halmazát nevezzük, amelyek lineáris kombinációjaként a vizsgált tér valamennyi vektora egyértelműen előállítható.

Egy speciális bázis a következő:



Az \mathbf{e}_1 és az \mathbf{e}_2 egymásra merőleges egységvektorok. Az ilyen bázist ortonormált (merőleges és normált) bázisnak nevezzük.

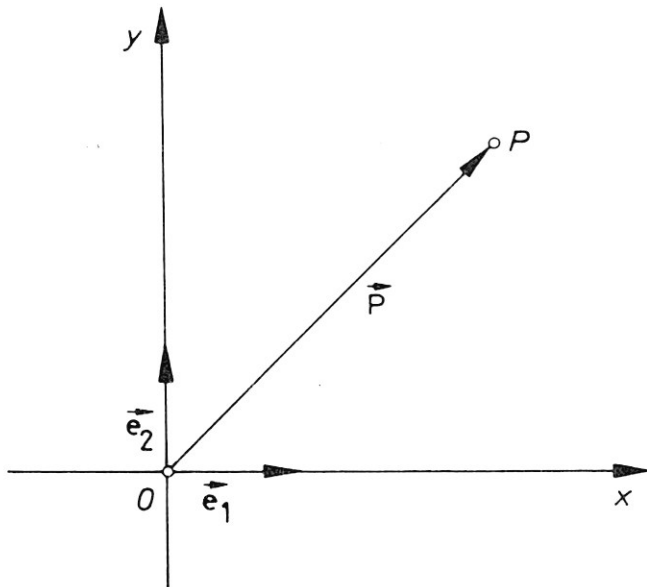
Tekintsük pl. az $(x; y)$ derékszögű koordináta-rendszert! Mindkét tengely előállítható a megfelelő egységvektor és a valós számok szorzataként.



1.1.2.3 Pontkoordináták

Teremtsünk kapcsolatot egy vektor koordinátái és egy pont koordinátái között! Helyezzünk — mint előbb — egy rögzített O pontba két (ill. három, \dots , n), páronként egymásra merőleges (ortogonális) egységvektort. Ekkor egy derékszögű koordináta-rendszert kapunk, amelynek minden P pontjához hozzárendelhetünk egy \vec{OP} vektort. Ezt a \mathbf{P} vektort a P pont helyvektorának nevezzük. Azonnal látható a P pont $(x; y)$ koordinátái és helyvektorának koordinátái közötti kapcsolat:

$$\mathbf{P} = \begin{pmatrix} x \\ y \end{pmatrix}$$



A helyvektor egyértelműen előállítható a pontkoordináták alapján, mert érvényes, hogy:

$$\mathbf{P} = x\mathbf{e}_1 + y\mathbf{e}_2.$$

Elegendő tehát egy pont koordinátáit ismerni, ha a bázisról tudjuk, hogy az standard bázis.

Könnyen belátható: nem feltétlenül szükséges, hogy a \mathbf{b}_1 , \mathbf{b}_2 bázisvektorok egymásra merőlegesek, ill. egységvektorok legyenek, elegendő, ha lineárisan függetlenek egymástól. Erre a tényre később még visszatérünk.

1.1.2.4 Példák, alkalmazások

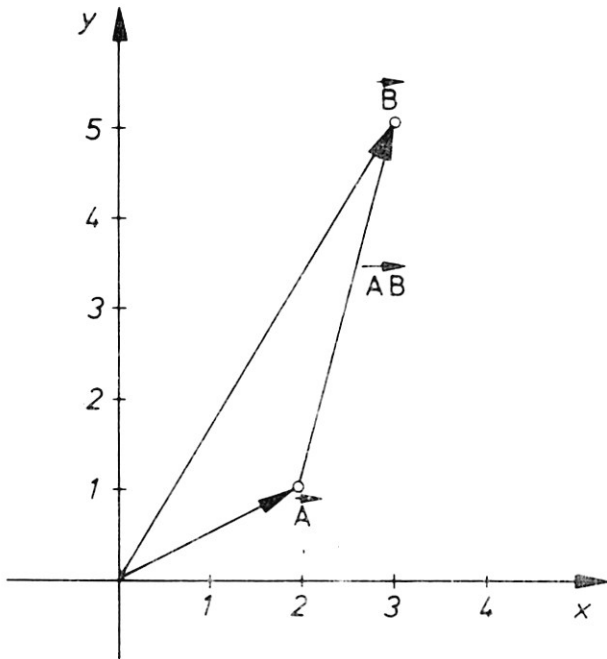
1) Az \mathbf{e}_1 és \mathbf{e}_2 bázisvektorok koordinátákkal felírt alakja:

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Hiszen igaz, hogy:

$$\mathbf{e}_1 = 1\mathbf{e}_1 + 0\mathbf{e}_2; \quad \text{ill. } \mathbf{e}_2 = 0\mathbf{e}_1 + 1\mathbf{e}_2.$$

2) Tekintsük az $\mathbf{A} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ és $\mathbf{B} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$ helyvektorokat!



Hogyan tudjuk az A és a B pontokat összekötő vektort, azaz az \overrightarrow{AB} vektort meghatározni? A problémát egy zárt poligon segítségével oldjuk meg. Érvényesnek kell lenni:

$$\mathbf{A} + \overrightarrow{AB} - \mathbf{B} = \mathbf{0}$$

$$\mathbf{A} + \overrightarrow{AB} = \mathbf{B}$$

$$\overrightarrow{AB} + \mathbf{A} = \mathbf{B}$$

$$\overrightarrow{AB} = \mathbf{B} - \mathbf{A}$$

a \mathbf{B} vektort hozzáadjuk mindkét oldalhoz

kommutativitási szabály

az \mathbf{A} vektort levonjuk mindkét oldalból

Ezzel megkaptuk két pontot összekötő vektor képletét:

$$\overrightarrow{AB} = \mathbf{B} - \mathbf{A}$$

(Vegyük észre, hogy az összekötő vektor = végpont – kezdőpont és nem $\mathbf{A} - \mathbf{B}$, mint azt az \overrightarrow{AB} írásmód sugallja).

Határozzuk meg most az \overrightarrow{AB} vektor koordinátáit!

Ehhez felhasználjuk, hogy ha

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

akkor

$$k\mathbf{a} = k \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} ka_1 \\ ka_2 \end{pmatrix};$$

$$\mathbf{a} + \mathbf{b} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{pmatrix}.$$

Általánosan: ha \mathbf{a} és \mathbf{b} vektorok alakja:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix},$$

akkor

$$\mathbf{a} + \mathbf{b} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{pmatrix}; \quad k\mathbf{a} = \begin{pmatrix} k_{a1} \\ k_{a2} \\ \vdots \\ k_{an} \end{pmatrix}.$$

Az \overrightarrow{AB} vektorra tehát:

$$\overrightarrow{AB} = \mathbf{B} - \mathbf{A} = \mathbf{B} + (-\mathbf{A}) = \begin{pmatrix} 3 \\ 5 \end{pmatrix} + (-1) \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix} + \begin{pmatrix} -2 \\ -1 \end{pmatrix} = \begin{pmatrix} 3-2 \\ 5-1 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}.$$

3) Vizsgáljuk meg néhány esetet a következő vektorok lineáris összefüggésével kapcsolatban:

$$\begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}; \begin{pmatrix} 2 \\ 6 \\ 8 \end{pmatrix}; \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}; \begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

Vegyük észre, hogy

$$2 \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ 8 \end{pmatrix},$$

azaz az első két vektor csak nagyságban különbözik egymástól, az irányuk azonban megegyezik.

A másik két vektorral kicsit nehezebben boldogulunk: a

$$\begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

vektor első koordinátája alapján észrevehetjük, hogy ez a vektor nem állítható elő az

$$\begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} \quad \text{vagy a} \quad \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

vektor skalárszorosaként, mert ehhez igaznak kell lenni a

$$k \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

összefüggésnek.

Ha $k=0$ (az első koordináta miatt), akkor:

$$0 \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

ami ellentmondás!

Azonos megfontolás érvényes a

$$\begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

vektorra is. Vizsgáljuk meg, hogy a

$$\begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

vektor előállítható-e az

$$\begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} \text{ és a } \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

vektorok lineáris kombinációjaként.

Ehhez érvényesnek kell lennie a

$$k_1 \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} + k_2 \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

összefüggésnek.

Kapunk tehát három egyenletet, amelyekben feltétel, hogy k_1 és k_2 kielégítse a következő összefüggéseket:

$$\begin{aligned} (1) \quad k_1 + 3k_2 &= 0; \\ (2) \quad 3k_1 &= 9; \\ (3) \quad 4k_1 + k_2 &= 11. \end{aligned}$$

A (2) egyenlet alapján: $k_1 = 3$.

Ezt behelyettesítve az (1) egyenletbe, adódik:

$$3 + 3k_2 = 0 \Rightarrow k_2 = -1.$$

Ellenőrizzük a megoldást, behelyettesítve az értékeket a (3) egyenletbe:

$$4 \cdot 3 - 1 = 12 - 1 = 11.$$

A $k_1 = 3$ és a $k_2 = -1$ értékek tehát megoldása az egyenletrendszernek. Teljesül tehát, hogy:

$$3 \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

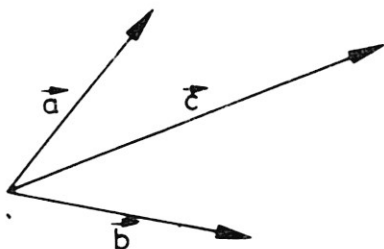
A $\begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$ vektor tehát lineárisan függ az

$$\begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix} \text{ és a } \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$$

vektoroktól.

Ezután bemutatunk még egy módszert, amellyel vektorok lineáris függőségét vizsgálhatjuk. Ez a módszer azonban csak a vektorok lineáris függetlenségének, ill. összefüggésének kimutatására használható anélkül, hogy a lineárisan összefüggő vektorokat egymásból ténylegesen előállítanánk. A módszer megértéséhez szükségünk lesz a következő ismeretekre.

Vegyük a sík 3 vektorát, amelyek legyenek **a**, **b**, **c**.



Mint korábban már megállapítottuk, a kétdimenziós térben nincs kettőnél több lineárisan független vektor, mert bármely harmadik vektor előállítható

$$\mathbf{c} = k\mathbf{a} + m\mathbf{b}$$

formában.

Ez általánosítható: az n -dimenziós térben legfeljebb n számú lineárisan független vektor található, így bármely $n+1$ vektor lineárisan összefügg.

Ha az R^3 -ban — mint a példánkban — adott 4 vektor, akkor előre megmondhatjuk, hogy ezek közül legalább egy lineárisan függ a másik háromtól. Ezért ilyen esetben csak a lehetséges vektorhármasok lineáris függőségét kell megvizsgálni. Ezt a vizsgálatot az ún. determinánsok segítségével végezhetjük el. (Hogy ne vigyük túlzásba az elméletet, itt csak a két-, ill. háromsoros determinánsokkal foglalkozunk.)

Legyenek az R^2 , ill. az R^3 vektorai:

$$\mathbf{a}_2 = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}; \mathbf{b}_2 = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}; \mathbf{a}_3 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}; \mathbf{b}_3 = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}; \mathbf{c}_3 = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}.$$

Akkor definíció szerint a determinánsok:

$$\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1 b_2 - a_2 b_1;$$

ill.

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 b_2 c_3 + b_1 c_2 a_3 + c_1 a_2 b_3 - a_3 b_2 c_1 - b_3 c_2 a_1 - c_3 a_2 b_1.$$

A háromsoros determináns kiszámítását könnyíti meg az ún. *Sarrus*-szabály. A

$$\det \mathbf{A} = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$$

kiszámítására érvényes a következő egyenlőség:

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = \begin{array}{c} + \quad + \quad + \\ \begin{array}{ccc} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{array} \\ - \quad - \quad - \end{array}$$

$\begin{array}{ccc} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{array}$

Összeadjuk a „+”-jellel ellátott egyenesek mentén fekvő együtthatók szorzatait, és kivonjuk a „-”-jellel jelölt egyenesek mentén fekvő együtthatók szorzatait.

Ezek után hogyan vizsgáljuk meg determinánsokkal a vektorok lineáris függőségét?

Érvényes a következő szabály:

ha $\det \mathbf{A} = 0$, akkor a vektorok lineárisan függenek;
 ha $\det \mathbf{A} \neq 0$, akkor a vektorok lineárisan függetlenek.

Ha el akarjuk dönteni, hogy az

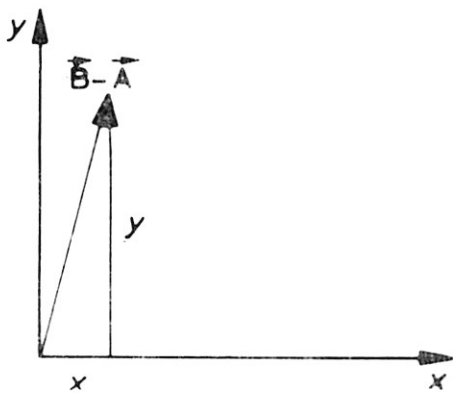
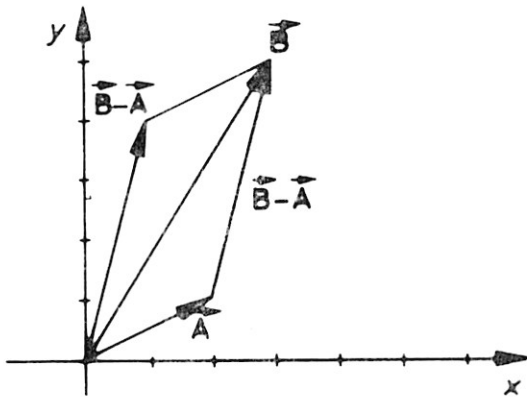
$$\begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}; \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}; \begin{pmatrix} 0 \\ 9 \\ 11 \end{pmatrix}$$

vektorok lineárisan függők-e, akkor meghatározzuk a vektorokból alkotott determináns értékét.

$$\begin{vmatrix} 1 & 3 & 0 \\ 3 & 0 & 9 \\ 4 & 1 & 11 \end{vmatrix} = 0 + (3 \cdot 9 \cdot 4) + 0 - 0 - (1 \cdot 9 \cdot 1) - (11 \cdot 3 \cdot 3) = 108 - 9 - 99 = 0.$$

E vektorok tehát lineárisan összefüggnek. (*Fontos:* ha a tér dimenzióinak száma n , akkor a vizsgálandó vektorok száma is n lehet csak, sem több, sem kevesebb. Tehát 2 az R^2 -ben, 3 az R^3 -ban stb.)

4) A 2) példában megadott A és B pont távolságának meghatározása a feladat. Vegyünk fel egy ortonormált koordináta-rendszert:



A 2) példában meghatároztuk: $\mathbf{B} - \mathbf{A} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$.

A Pitagorasz-tétel alapján:

$$x^2 + y^2 = (|\mathbf{B} - \mathbf{A}|)^2 \Leftrightarrow |\mathbf{B} - \mathbf{A}| = \sqrt{x^2 + y^2}.$$

Továbbá:

$$|\mathbf{B} - \mathbf{A}| = |\overrightarrow{AB}| = \text{az } A \text{ és a } B \text{ pontok távolsága.}$$

Így az $|\overrightarrow{AB}|$ vektor hosszát a koordinátáiból a következő képlettel számíthatjuk ki:

$$|\overrightarrow{AB}| = |\mathbf{B} - \mathbf{A}| = \sqrt{x^2 + y^2}.$$

Mivel

$$x = b_1 - a_1; \quad y = b_2 - a_2,$$

az A és B helyvektorainak koordinátáival:

$$|\overrightarrow{AB}| = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}.$$

A pontok távolsága tehát:

$$\sqrt{1+16} = \sqrt{17}.$$

5) Gyakran szükségünk van arra, hogy egy adott vektorhoz tartozó, az adott vektorral egyirányú egységvektort meghatározzunk. Erre az egységvektorra fennáll, hogy:

$$\mathbf{a}^\circ = \frac{\mathbf{a}}{|\mathbf{a}|}$$

Koordinátákkal kifejezve:

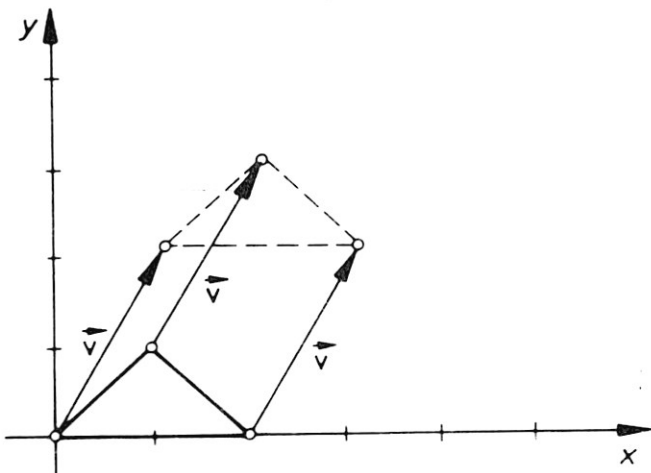
$$\mathbf{a}^\circ = \frac{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}}{\sqrt{a_1^2 + a_2^2 + \dots + a_n^2}}$$

Mit is nyertünk mindezzel? Nagyjából úgy mondhatnánk: nyertük annak a lehetőségét, hogy pontokkal és geometriai alakzatokkal számolni tudjunk. Vagy másképp fogalmazva azt, hogy alkalmazni tudjuk a rendelkezésünkre álló algebrai és analízisbeli eszközöket a geometriában. Ebből származik az *analitikus geometria* kifejezés, amellyel módszeresen először *Descartes* foglalkozott.

Most már képesek vagyunk pl. arra, hogy a

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

háromszöget egy vektor segítségével tetszőleges helyre eltoljuk.



Legyen

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

az eltolási vektor, amellyel a $(0; 0)$ pontot eltoljuk. A háromszög csúcsainak új koordinátáit úgy kapjuk meg, hogy a \mathbf{v} vektor koordinátáit hozzáadjuk a háromszög csúcsainak megfelelő koordinátákhoz:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \begin{pmatrix} 2+v_1 \\ v_2 \end{pmatrix}, \begin{pmatrix} 1+v_1 \\ 1+v_2 \end{pmatrix}.$$

6) Az $\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ vektorok bázist alkotnak az R^2 -ben.

Bizonyítás: nyilvánvaló, hogy e két vektor lineárisan független. Az \mathbf{e}_1 és \mathbf{e}_2 egységvektorok a következő alakban írhatók:

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -\frac{1}{3} \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \frac{2}{3} \begin{pmatrix} 2 \\ 1 \end{pmatrix};$$

$$\mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{2}{3} \begin{pmatrix} 1 \\ 2 \end{pmatrix} - \frac{1}{3} \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Mivel az \mathbf{e}_1 és \mathbf{e}_2 egységvektorok előállíthatók a vizsgált vektorok lineáris kombinációjaként, és mert R^2 valamennyi pontja előállítható az \mathbf{e}_1 és \mathbf{e}_2 egységvektorok segítségével, ezért előállítható az R^2 valamennyi pontja az

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \text{ és a } \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

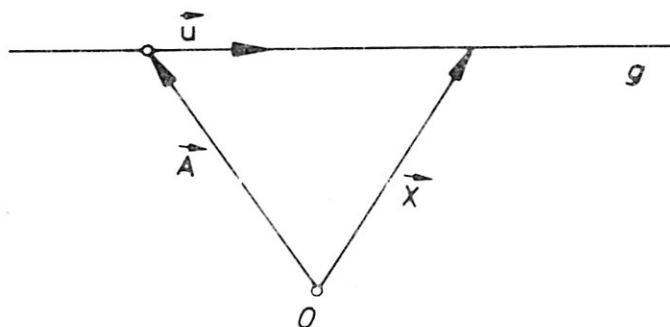
vektorokkal is.

1.1.3 EGYENESEK ÉS SÍKOK AZ R^3 -BAN

Vizsgálatainkat a következőkben az R^3 -ra (a háromdimenziós térre) korlátozzuk, egyrészt azért, mert nagyobb dimenziós számú esetek ritkán fordulnak elő, másrészt, mert az R^2 -re vonatkozó összefüggések az R^3 -beli összefüggések speciális eseteinek tekinthetők.

1.1.3.1 Paraméteres egyenletek

Tekintsük a következő ábrát:



Adott az O kezdőpont, a g egyenes, továbbá a g egyenes egy A pontja és az egyenes u irányvektora. Az egyenes pontjait a következő módon adhatjuk meg:

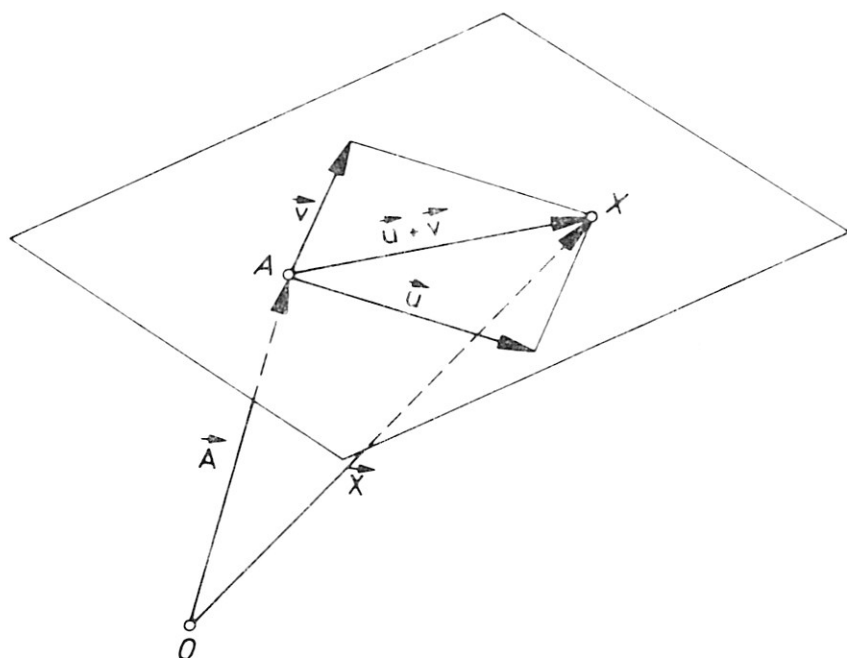
$$g: \mathbf{X} = \mathbf{A} + k\mathbf{u}; \quad k \in R,$$

mert

$$\mathbf{A} + k\mathbf{u} - \mathbf{X} = \mathbf{O}.$$

Az egyenesre megadott összefüggést az egyenes paraméteres egyenletének nevezzük.

Egy sík paraméteres alakjának meghatározásához tekintsük a következő ábrát:



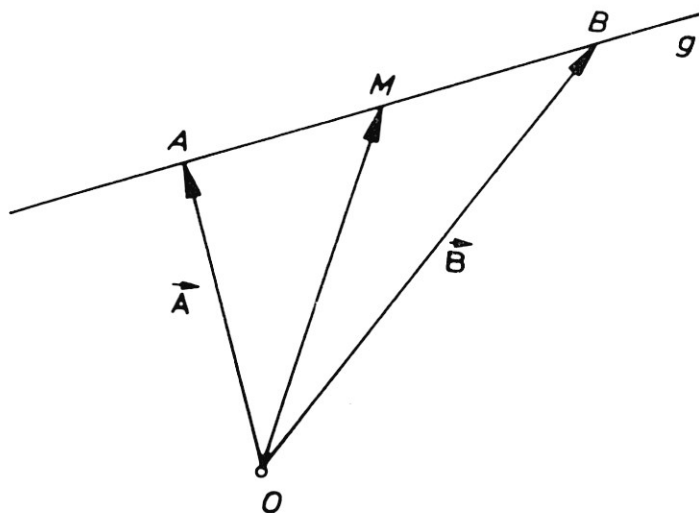
Mint már láttuk, két lineárisan független vektor elégséges ahhoz, hogy az R^2 összes pontját ábrázolhassuk. Ha eltoljuk a térben R^2 -t egy \mathbf{A} vektor mentén, akkor megkapjuk egy sík ábrázolásának általános paraméteres alakját:

$$E: \mathbf{X} = \mathbf{A} + k\mathbf{u} + l\mathbf{v}; \quad k, l \in R.$$

1.1.3.2 Példák, alkalmazások

Az eddigi ismeretek elmélyítésére vizsgáljunk meg néhány feladatot!

1) Egy \overline{AB} szakasz felezőpontja



Ossza fel az M pont az \overline{AB} szakaszt 1 : 1 arányban.
Érvényes tehát:

$$\overline{AM} = \overline{MB}, \quad \text{ill.} \quad |\overrightarrow{AM}| = |\overrightarrow{MB}|$$

Határozzuk meg először az A és B pontokon átmenő egyenes egyenletét!

Az \mathbf{A} helyvektorral és az \overline{AB} irányvektorral:

$$g: \mathbf{X} = \mathbf{A} + k(\mathbf{B} - \mathbf{A}).$$

Az M pont a g egyenes pontja. A hozzárendelt paraméter értéke 0,5, mivel felezi az \overline{AB} szakaszt.
Érvényes tehát:

$$\mathbf{M} = \mathbf{A} + \frac{1}{2}(\mathbf{B} - \mathbf{A}) = \mathbf{A} + \frac{1}{2}\mathbf{B} - \frac{1}{2}\mathbf{A} = \frac{\mathbf{A} + \mathbf{B}}{2}.$$

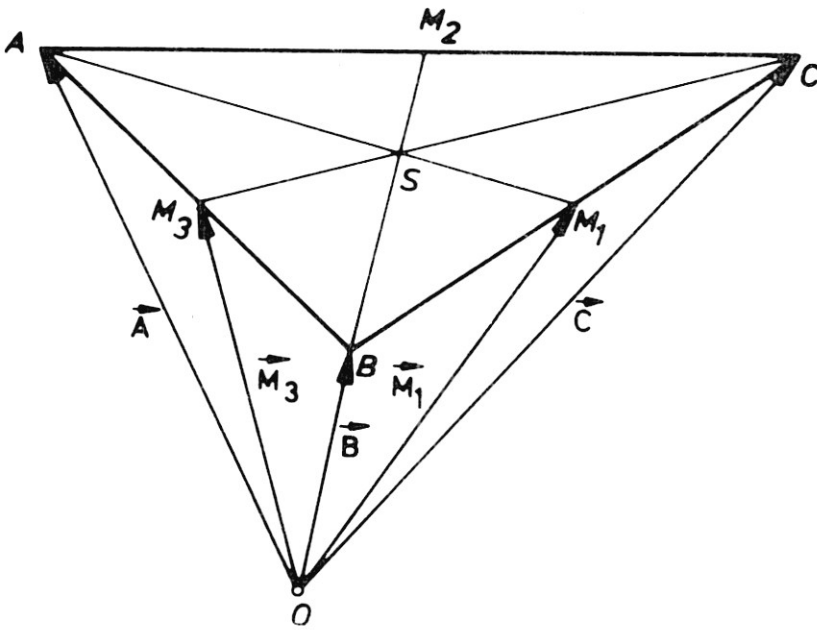
2) A háromszög súlypontja

A geometriából ismert, hogy a háromszög súlypontja a csúcsokat a szemközti oldal felezőpontjával összekötő szakaszok, azaz a súlyvonalak közös metszéspontja. Vegyük kiindulásként az O pontot, amely nem az A , B és C pontok által meghatározott síkban fekszik. Akkor az oldalfelező pontok helyvektorait a következő alakban írhatjuk:

$$\mathbf{M}_1 = \frac{1}{2}(\mathbf{B} + \mathbf{C});$$

$$\mathbf{M}_2 = \frac{1}{2}(\mathbf{C} + \mathbf{A});$$

$$\mathbf{M}_3 = \frac{1}{2}(\mathbf{A} + \mathbf{B}).$$



A súlyvonalak paraméteres alakja:

$$AM_1: \mathbf{X} = \mathbf{A} + r \left(\frac{1}{2}\mathbf{B} + \frac{1}{2}\mathbf{C} - \mathbf{A} \right) = (1-r)\mathbf{A} + \frac{r}{2}\mathbf{B} + \frac{r}{2}\mathbf{C};$$

$$BM_2: \mathbf{X} = \mathbf{B} + s \left(\frac{1}{2}\mathbf{C} + \frac{1}{2}\mathbf{A} - \mathbf{B} \right) = \frac{s}{2}\mathbf{A} + (1-s)\mathbf{B} + \frac{s}{2}\mathbf{C};$$

$$CM_3: \mathbf{X} = \mathbf{C} + t \left(\frac{1}{2}\mathbf{A} + \frac{1}{2}\mathbf{B} - \mathbf{C} \right) = \frac{t}{2}\mathbf{A} + \frac{t}{2}\mathbf{B} + (1-t)\mathbf{C}.$$

Az AM_1 és BM_2 egyenesek S metszéspontjára fennáll, hogy

$$(1-r)\mathbf{A} + \frac{r}{2}\mathbf{B} + \frac{r}{2}\mathbf{C} = \frac{s}{2}\mathbf{A} + (1-s)\mathbf{B} + \frac{s}{2}\mathbf{C} \Leftrightarrow$$

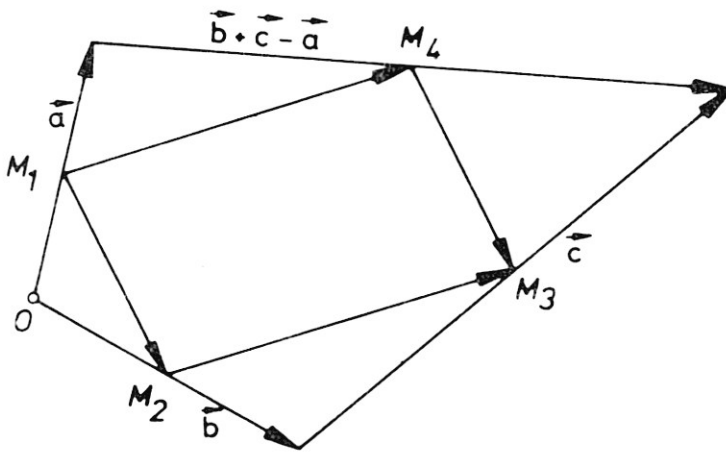
$$\left(1-r-\frac{s}{2}\right)\mathbf{A} + \left(\frac{r}{2}-1+s\right)\mathbf{B} + \left(\frac{r}{2}-\frac{s}{2}\right)\mathbf{C} = \mathbf{0}.$$

Mivel az A , B és C pontok egy háromszög csúcsai, így nem eshetnek egy egyenesre, továbbá az O pontot úgy választottuk meg, hogy ne feködjön a háromszög csúcspontjai által meghatározott síkban, így az \mathbf{A} , \mathbf{B} , \mathbf{C} vektorok lineárisan függetlenek, az \mathbf{A} , \mathbf{B} , \mathbf{C} együtthatóinak mindig 0-nak kell lenni. Így azt kapjuk, hogy $r=s=2/3$. Ugyanígy kiszámítva a BM_2 és a CM_3 egyenesek metszéspontját, adódik, hogy $s=t=2/3$. Mindkét esetben ugyanazt az S pontot kapjuk, a három súlyvonal tehát egy pontban metszi egymást. Az S pontot a háromszög súlypontjának nevezzük, és helyvektora az

$$S = \frac{\mathbf{A} + \mathbf{B} + \mathbf{C}}{3}$$

összefüggés alapján határozható meg.

3) Tetszőleges négyszög oldalfelező pontjai (az R^2 -ben) paralelogrammát alkotnak



Helyezzük a négyszög egyik csúcsát célszerűen a vonatkoztatási rendszer O kezdőpontjába. Ekkor az oldalfelező pontokra a következő egyenleteket kapjuk:

$$\mathbf{M}_1 = \frac{1}{2}\mathbf{a};$$

$$\mathbf{M}_2 = \frac{1}{2}\mathbf{b};$$

$$\mathbf{M}_3 = \mathbf{b} + \frac{1}{2}\mathbf{c};$$

$$\mathbf{M}_4 = \mathbf{a} + \frac{1}{2}(\mathbf{b} + \mathbf{c} - \mathbf{a}).$$

A felezőpontokat összekötő $\overrightarrow{M_1M_2}$ vektorokra adódik:

$$\overrightarrow{M_1M_2} = \frac{1}{2}\mathbf{b} - \frac{1}{2}\mathbf{a} = \frac{1}{2}(\mathbf{b} - \mathbf{a});$$

$$\overrightarrow{M_4M_3} = \mathbf{b} + \frac{1}{2}\mathbf{c} - \mathbf{a} - \frac{1}{2}(\mathbf{b} + \mathbf{c} - \mathbf{a}) = \mathbf{b} + \frac{1}{2}\mathbf{c} - \mathbf{a} - \frac{1}{2}\mathbf{b} - \frac{1}{2}\mathbf{c} + \frac{1}{2}\mathbf{a} = \frac{1}{2}\mathbf{b} - \frac{1}{2}\mathbf{a} = \frac{1}{2}(\mathbf{b} - \mathbf{a});$$

$$\overrightarrow{M_1M_4} = \mathbf{a} + \frac{1}{2}(\mathbf{b} + \mathbf{c} - \mathbf{a}) - \frac{1}{2}\mathbf{a} = \frac{1}{2}\mathbf{a} + \frac{1}{2}\mathbf{b} + \frac{1}{2}\mathbf{c} - \frac{1}{2}\mathbf{a} = \frac{1}{2}\mathbf{b} + \frac{1}{2}\mathbf{c} = \frac{1}{2}(\mathbf{b} + \mathbf{c});$$

$$\overrightarrow{M_2M_3} = \mathbf{b} + \frac{1}{2}\mathbf{c} - \frac{1}{2}\mathbf{b} = \frac{1}{2}\mathbf{b} + \frac{1}{2}\mathbf{c} = \frac{1}{2}(\mathbf{b} + \mathbf{c}).$$

A szemben fekvő M_1M_2 és M_4M_3 , ill. M_1M_4 és M_2M_3 oldalak tehát párhuzamosak és egyenlő hosszúságúak, így valóban paralelogrammát feszítenek ki.

1.1.4 VEKTOR SZORZÁSA VEKTORRAL

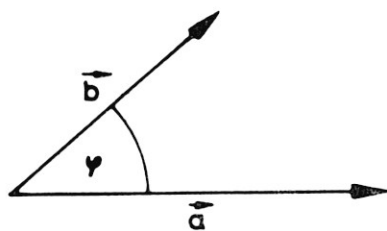
Az eddigiekben a vektorok tisztán additív tulajdonságaival foglalkoztunk. Mint már mondtuk, a vektorok irányított „nyilak”, így az általuk bezárt szögek is meghatározhatók.

1.1.4.1 A skaláris szorzat

A skaláris szorzatot a következőképpen definiáljuk: az \mathbf{a} és \mathbf{b} vektorok skaláris szorzata alatt (jelölése: $\mathbf{a} \cdot \mathbf{b}$ vagy \mathbf{ab}) az

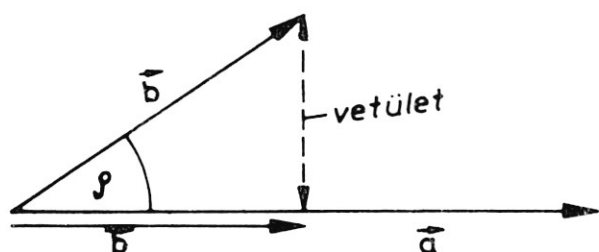
$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| \cdot |\mathbf{b}| \cos \varphi$$

valós számot értjük, ahol φ ($0 \leq \varphi \leq \pi$) az \mathbf{a} és \mathbf{b} vektorok által bezárt szög.



Fontos: a skaláris szorzás eredmények *nem* vektor, hanem egy valós szám, vagyis skalár.

A skaláris szorzás szemléletesen a következő módon értelmezhető: ha az \mathbf{a} és a \mathbf{b} az R^2 vagy az R^3 vektorai, akkor a skaláris szorzat az \mathbf{a} vektor hosszának szorzata az \mathbf{a} -ra vetített \mathbf{b} vektor \mathbf{b}' vetületének hosszával.



Legyen az \mathbf{a} és \mathbf{b} vektor koordinátákkal megadott alakja a következő:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

A vektorok skaláris szorzata:

$$\mathbf{a} \cdot \mathbf{b} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n.$$

A skaláris szorzás tulajdonságai:

$$1) \mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$$

$$2) k(\mathbf{a} \cdot \mathbf{b}) = (k\mathbf{a}) \cdot \mathbf{b} = \mathbf{a} \cdot (k\mathbf{b})$$

$$3) (\mathbf{a} + \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot \mathbf{c} + \mathbf{b} \cdot \mathbf{c}$$

$$4) \mathbf{a} \cdot \mathbf{a} = \mathbf{a}^2 > 0, \text{ ha } \mathbf{a} \neq \mathbf{0}$$

$$\mathbf{a}^2 = |\mathbf{a}|^2$$

kommutatív

valós számmal való

szorzása asszociatív

disztributív

A skaláris szorzás műveletére tehát az aritmetikai alaptörvények mindegyike érvényes.

Fontos: hogyan értelmezzük az

$$(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{c}$$

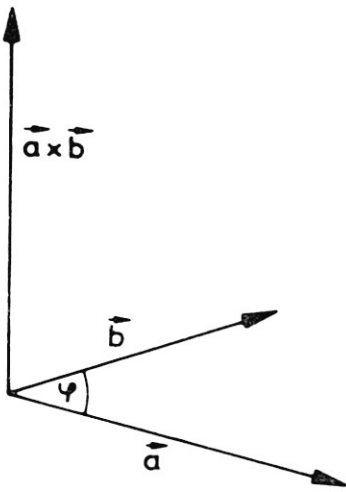
kifejezést? Mivel $\mathbf{a} \cdot \mathbf{b} \in \mathcal{R}$, azaz egy skalár, a \mathbf{c} vektort egy skalárral szorozzuk, nem pedig vektorok közötti skaláris szorzást végzünk.

1.1.4.2 A vektoriális szorzat

A vektoriális szorzatot csak az \mathcal{R}^3 -ban definiáljuk. A skaláris szorzattal szemben a vektoriális szorzás eredménye szintén vektor, amely merőleges az \mathbf{a} és \mathbf{b} vektorok által meghatározott síkra, mégpedig úgy, hogy az \mathbf{a} , a \mathbf{b} és az $\mathbf{a} \times \mathbf{b}$ ún. jobbsodrású rendszert képeznek. Az eredményvektor hossza pedig az \mathbf{a} és \mathbf{b} vektorok hosszának és közbezárt szögük szinuszának szorzata:

$$|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}| \cdot |\mathbf{b}| \sin \varphi.$$

Jobbsodrású rendszerhez úgy jutunk, hogy jobb kezünk hüvelyk-, mutató- és középső ujjait úgy nyújtjuk ki, hogy a középső ujjunk (az $\mathbf{a} \times \mathbf{b}$ vektor) merőleges a hüvelykujj (\mathbf{a} vektor) és a mutatóujj (\mathbf{b} vektor) által meghatározott síkra.



Az $\mathbf{a} \times \mathbf{b}$ szorzat koordinátákkal való írásmódjára érvényes a következő szabály: legyenek

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \text{és} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

az R^3 vektorai, akkor

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix} \\ - \begin{vmatrix} a_1 & b_1 \\ a_3 & b_3 \end{vmatrix} \\ \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \end{pmatrix} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

Az i -edik koordinátát úgy kapjuk meg, hogy az \mathbf{a} és \mathbf{b} vektorok i -edik koordinátájának elhagyásával kapott kétsoros determinánst kiszámítjuk. Fontos, hogy a második koordináta elé mínuszjelet írjunk.

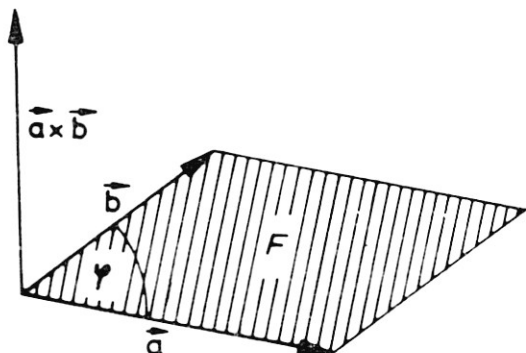
A vektoriális szorzás tulajdonságai:

- 1) $\mathbf{a} \times \mathbf{b} = -(\mathbf{b} \times \mathbf{a})$
- 2) $k(\mathbf{a} \times \mathbf{b}) = (k\mathbf{a}) \times \mathbf{b} = \mathbf{a} \times k\mathbf{b}$
- 3) $\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$

nem kommutatív!
valós számmal való
szorzása asszociatív
disztributív

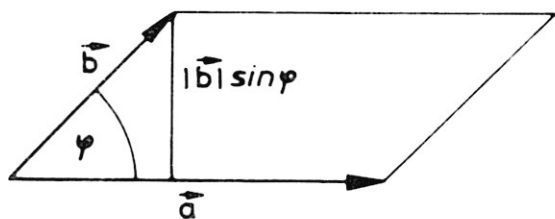
Szemléletesen kifejezve: a vektoriális szorzás eredményvektorának hossza az \mathbf{a} és \mathbf{b} vektorok által kifeszített paralelogramma területével egyenlő:

$$|\mathbf{a} \times \mathbf{b}| = F.$$



Bizonyítás:

$$F = |\mathbf{a}| \cdot |\mathbf{b}| \sin \varphi = |\mathbf{a} \times \mathbf{b}|.$$



A skaláris és vektoriális szorzatra érvényes összefüggések:

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} = 0 &\Rightarrow \mathbf{a} \perp \mathbf{b} \quad (\mathbf{a} \text{ és } \mathbf{b} \text{ merőlegesek}); \\ \mathbf{a} \times \mathbf{b} = \mathbf{0} &\Rightarrow \mathbf{a} \parallel \mathbf{b} \quad (\mathbf{a} \text{ és } \mathbf{b} \text{ párhuzamosak}), \end{aligned}$$

ha $\mathbf{a}, \mathbf{b} \neq \mathbf{0}$

Érvényesek a következő szabályok:

- 1) $(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$
 - 2) $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{a}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{b}(\mathbf{a} \cdot \mathbf{b})$
 - 3) $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) + \mathbf{b} \times (\mathbf{c} \times \mathbf{a}) + \mathbf{c} \times (\mathbf{a} \times \mathbf{b}) = \mathbf{0}$
- Grassmann-azonosság*

1.1.4.3 Példák, alkalmazások

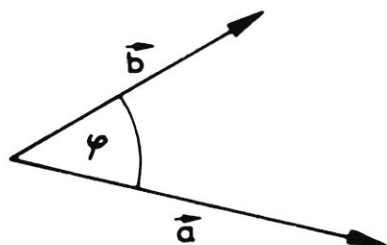
1) Két vektor hajlásszöge

Mint ahogy

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| \cdot |\mathbf{b}| \cos \varphi,$$

ezért

$$\cos \varphi = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| \cdot |\mathbf{b}|}.$$



A képlet alapján meghatározható pl. két, egymást metsző egyenes által bezárt szög. Legyen:

$$g_1: \mathbf{X} = \mathbf{A} + k\mathbf{u};$$

$$g_2: \mathbf{X} = \mathbf{B} + l\mathbf{v},$$

akkor a g_1 és g_2 egyenesek által bezárt δ szögre igaz:

$$\cos \delta = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}| \cdot |\mathbf{v}|}.$$

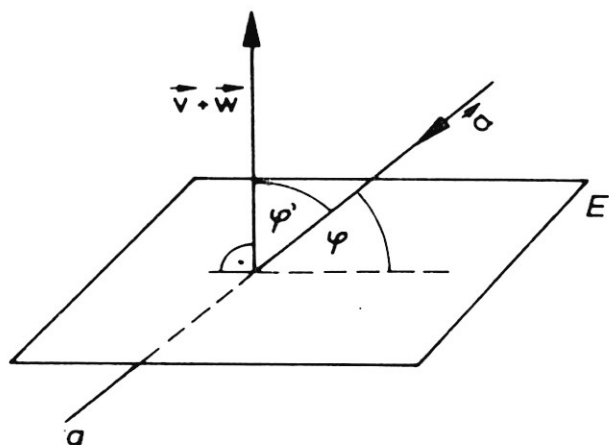
2) Sík és egyenes hajlásszöge

Legyen adott az

$$E: \mathbf{X} = \mathbf{A} + k\mathbf{v} + l\mathbf{w} \text{ sík,}$$

valamint a

$$g: \mathbf{X} = \mathbf{B} + r\mathbf{a} \text{ egyenes.}$$



A $\mathbf{v} \times \mathbf{w}$ szorzat egy olyan vektor, amely a \mathbf{w} és \mathbf{v} vektorok által meghatározott síkra merőleges. A g egyenes és az E sík által bezárt φ szögre érvényes a következő összefüggés:

$$\varphi = 90^\circ - \varphi'.$$

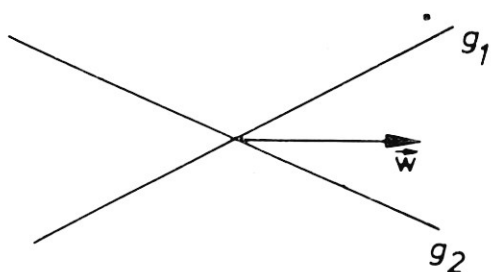
A φ' szög azonban a $\mathbf{v} \times \mathbf{w}$ és \mathbf{a} vektorok skaláris szorzatával egyszerűen meghatározható, mivel

$$\cos \varphi' = \frac{(\mathbf{v} \times \mathbf{w}) \cdot \mathbf{a}}{|\mathbf{v} \times \mathbf{w}| \cdot |\mathbf{a}|}$$

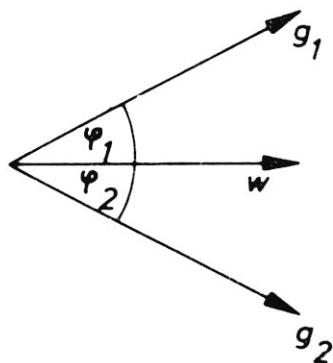
3) Két egyenes szögfelezője

Adottak az egyenesek irányvektorai: \mathbf{a} és \mathbf{b} . Meghatározandók a két egyenes szögfelezőinek irányvektorai,

\mathbf{w}_1 és \mathbf{w}_2 .



Induljunk ki a skaláris szorzatból:



Mivel a \mathbf{w} vektor szögfelezője a g_1 és g_2 egyeneseknek,

$$\varphi_1 = \varphi_2 \Leftrightarrow \cos \varphi_1 = \cos \varphi_2.$$

A skalárszorzat alapján a $\cos \varphi_1 = \cos \varphi_2$ helyett írható:

$$\frac{\mathbf{w} \cdot \mathbf{a}}{|\mathbf{w}| \cdot |\mathbf{a}|} = \frac{\mathbf{w} \cdot \mathbf{b}}{|\mathbf{w}| \cdot |\mathbf{b}|};$$

ill.

$$\frac{\mathbf{w} \cdot \mathbf{a}}{|\mathbf{a}|} = \frac{\mathbf{w} \cdot \mathbf{b}}{|\mathbf{b}|}.$$

Mivel a \mathbf{w} vektor az \mathbf{a} és \mathbf{b} irányvektorok síkjában fekszik, előállítható ezek lineáris kombinációjaként:

$$\mathbf{w} = k\mathbf{a} \pm \mathbf{b}. \quad (*)$$

(Feltehető, hogy a \mathbf{b} együtthatója 1, ill. -1 , ui. a \mathbf{w} irányvektornak csak az iránya lényeges, a hossza érdektelen.)

Így:

$$\frac{(k\mathbf{a} \pm \mathbf{b}) \cdot \mathbf{a}}{|\mathbf{a}|} = \frac{(k\mathbf{a} \pm \mathbf{b}) \cdot \mathbf{b}}{|\mathbf{b}|}$$

$$\Leftrightarrow \frac{k\mathbf{a}^2 \pm \mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|} = \frac{k(\mathbf{a} \cdot \mathbf{b}) \pm \mathbf{b}^2}{|\mathbf{b}|}$$

$$\Leftrightarrow \frac{k\mathbf{a}^2}{|\mathbf{a}|} \pm \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|} = \frac{k(\mathbf{a} \cdot \mathbf{b})}{|\mathbf{b}|} \pm \frac{\mathbf{b}^2}{|\mathbf{b}|}$$

$$\Leftrightarrow k|\mathbf{a}| \pm \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|} = \frac{k(\mathbf{a} \cdot \mathbf{b})}{|\mathbf{b}|} \pm |\mathbf{b}|$$

$$\Leftrightarrow k|\mathbf{a}| - \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{b}|} = |\mathbf{b}| - \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|}$$

$$\Leftrightarrow k \left(\frac{|\mathbf{a}| \cdot |\mathbf{b}| - \mathbf{a} \cdot \mathbf{b}}{|\mathbf{b}|} \right) = \frac{|\mathbf{b}| \cdot |\mathbf{a}| - \mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|}$$

$$\Leftrightarrow k = \frac{|\mathbf{b}|}{|\mathbf{a}|}$$

Ezt behelyettesítve a (*) egyenletbe, a megoldás:

$$\mathbf{w} = \frac{|\mathbf{b}|}{|\mathbf{a}|} \mathbf{a} \pm \mathbf{b}; \quad \frac{1}{|\mathbf{b}|} \cdot \mathbf{w} = \frac{\mathbf{a}}{|\mathbf{a}|} \pm \frac{\mathbf{b}}{|\mathbf{b}|} = \mathbf{a}^\circ \pm \mathbf{b}^\circ.$$

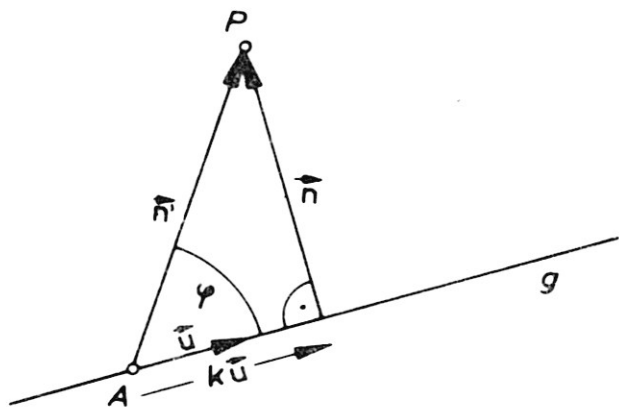
A szögfelező egy irányvektorát tehát az \mathbf{a} és \mathbf{b} vektorhoz tartozó egységvektorok összeadásával, ill. kivonásával kapjuk meg.

4) Pont és egyenes távolsága

Egy egyenesre merőleges vektor segítségével meghatározhatjuk egy pont és egy egyenes távolságát. Legyen adott a P pont, valamint a g egyenes (ami átmegy az A ponton, irányvektora \mathbf{u}).

$$g: \mathbf{X} = \mathbf{A} + k\mathbf{u}$$

Nézzük meg a következő ábrát:



Egy pontnak egy egyenestől mért távolságát úgy kapjuk meg, hogy meghatározzuk a P pontból a g egyenesre bocsátott merőleges g egyenessel való metszéspontjának a P ponttól való távolságát.

Nyilvánvalóan igaz:

$$\mathbf{n}' = k\mathbf{u} + \mathbf{n}.$$

A $k\mathbf{u}$ értékét könnyen kiszámíthatjuk, ha meghatározzuk az \mathbf{n}' vektor vetületét az \mathbf{u} irányvektorra.

Érvényes a következő egyenlőség:

$$\frac{\mathbf{n}' \cdot \mathbf{u}}{|\mathbf{u}|} \mathbf{u}^\circ = k\mathbf{u}$$

(Az előbbi ábráról leolvasható, hogy

$$|k\mathbf{u}| = l \text{ jelölés mellett:}$$

$$l = |\mathbf{n}'| \cos \varphi$$

$$\Leftrightarrow l = |\mathbf{n}'| \frac{|\mathbf{u}|}{|\mathbf{u}|} \cos \varphi = \frac{\mathbf{n}' \cdot \mathbf{u}}{|\mathbf{u}|}$$

$$k\mathbf{u} = l\mathbf{u}^\circ = \frac{l}{|\mathbf{u}|} \mathbf{u}$$

$$k\mathbf{u} = \frac{\mathbf{n}' \cdot \mathbf{u}}{\mathbf{u}} \cdot \frac{\mathbf{u}}{|\mathbf{u}|} = \frac{\mathbf{n}' \cdot \mathbf{u}}{|\mathbf{u}|^2} \mathbf{u} = \frac{\mathbf{n}' \cdot \mathbf{u}}{\mathbf{u}^2} \mathbf{u}.)$$

Behelyettesítés után kapjuk:

$$\mathbf{n}' = \mathbf{n} + \frac{\mathbf{n}' \cdot \mathbf{u}}{u^2} \mathbf{u},$$

amiből

$$\mathbf{n} = \mathbf{n}' - \frac{\mathbf{n}' \cdot \mathbf{u}}{u^2} \mathbf{u}.$$

Egy egyenes és egy pont távolsága tehát a következőképpen számítható ki:

$$d = |\mathbf{n}| = \left| \mathbf{n}' - \frac{\mathbf{n}' \cdot \mathbf{u}}{u^2} \mathbf{u} \right|$$

5) Pont és sík távolsága

Sík egyenletének normálalakja

Sík dőfése egyenessel

Adott az E sík és annak egy A pontja:

$$E: \mathbf{X} = \mathbf{A} + k\mathbf{u} + l\mathbf{v}.$$

A címben jelzett probléma kissé nehezebb az előzőnél. A megoldás azonban egyszerű lesz, ha áttérünk a sík leírásának egy másik módjára.

A sík egyenlete koordinátákra áttérve:

$$\mathbf{X} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + k \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + l \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

Ebből három egyenletet kapunk az \mathbf{X} koordinátáira:

$$(1) \quad x_1 = a_1 + ku_1 + lv_1;$$

$$(2) \quad x_2 = a_2 + ku_2 + lv_2;$$

$$(3) \quad x_3 = a_3 + ku_3 + lv_3.$$

Az egyenletrendszer megoldása a sík egyenletének az ún. nemparaméteres alakját adja.

A továbbiakban tegyük fel, hogy u_1 nem egyenlő 0-val.

Vezessük be a következő rövidítéseket:

$$n_1 = v_3u_2 - v_2u_3;$$

$$n_2 = v_1u_3 - v_3u_1;$$

$$n_3 = v_2u_1 - v_1u_2;$$

$$(2) + \left(-\frac{u_2}{u_1}\right)(1) = (1)'$$

$$\Rightarrow (1)': (x_2 - a_2) - (x_1 - a_1) \left(\frac{u_2}{u_1}\right) = l \left(\frac{n_3}{u_1}\right)$$

$$(3) + \left(-\frac{u_3}{u_1}\right)(1) = (2)'$$

$$\Rightarrow (2)': (x_3 - a_3) - (x_1 - a_1) \left(\frac{u_3}{u_1}\right) = l \left(\frac{-n_2}{u_1}\right)$$

$$(2)' - \left(\frac{-n_2}{n_3}\right)(1)' = (1)''$$

$$(x_3 - a_3) - (x_1 - a_1) \left(\frac{u_3}{u_1}\right) - (x_2 - a_2) \left(\frac{-n_2}{n_3}\right) + (x_1 - a_1) \left(\frac{u_2}{u_1}\right) \left(\frac{-n_2}{n_3}\right) = 0$$

Átrendezve kapjuk:

$$x_3 - x_1 \left(\frac{u_3}{u_1}\right) - x_2 \left(\frac{-n_2}{n_3}\right) + x_1 \left(\frac{u_2}{u_1}\right) \left(\frac{-n_2}{n_3}\right) - C = 0.$$

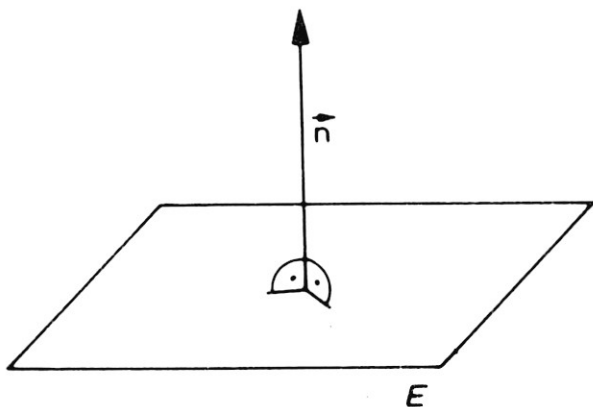
A C konstans — mint látni fogjuk — egyszerűen meghatározható. Hozzuk még a tagokat közös nevezőre és vonjuk össze.

$$x_3 u_1 n_3 + x_2 u_1 n_2 + x_1 u_1 n_1 + C' = 0.$$

u_1 -gyel való osztás után kapjuk:

$$x_1 n_1 + x_2 n_2 + x_3 n_3 + C'' = 0 \quad (*)$$

Ez az eredmény várható volt. Kérdezhetnénk, miért?



Tekintsük az E síkot! Nyilvánvaló, hogy a sík térbeli helyzete leírható csupán a sík egy A pontja és egy, a síkra merőleges vektor segítségével. Ezt a merőleges vektort a sík normálvektorának nevezzük.

Léven közömbös, hogy milyen normálvektort használunk, normálvektora az E síknak pl. az $\mathbf{u} \times \mathbf{v}$ vektor is:

$$\mathbf{u} \times \mathbf{v} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \mathbf{n}.$$

A (*)-gal jelzett egyenlet együtthatóinak összehasonlításával megállapíthatjuk, hogy ezek azonosan az \mathbf{n} komponenseivel.

$$E: n_1 x_1 + n_2 x_2 + n_3 x_3 + n_0 = 0; \quad n_0 = \text{konstans} \quad (**)$$

Az n_0 konstans meghatározható, ha az egyenes A pontját a (**) egyenletbe behelyettesítjük. Igaz:

$$n_1 a_1 + n_2 a_2 + n_3 a_3 + n_0 = 0,$$

amiből

$$n_0 = -(n_1 a_1 + n_2 a_2 + n_3 a_3).$$

A (**) forma a sík egyenletének normálalakja. Mi a haszna mindennek a P pont E síktól való távolságának meghatározásában? Nagyon sok, ui. ha ismerjük a sík egyenletének nemparaméteres alakját, akkor különösen egyszerű módon meghatározhatjuk ennek a síknak és egy egyenesnek a dőléspontját.

A P pontból az E síkra bocsátott merőleges:

$$g: \mathbf{X} = \mathbf{P} + r\mathbf{n}.$$

A g egyenes és az E sík egyenletének egyenlővé tételéből adódik az S dőléspont:

$$\mathbf{P} + r\mathbf{n} = \mathbf{A} + k\mathbf{u} + l\mathbf{v}.$$

Az eredményt máris megkapjuk, csak a (**) egyenletbe \mathbf{X} helyett a g egyenes $\mathbf{P} + r\mathbf{n}$ pontját kell behelyettesítenünk.

Így a behelyettesítés és kis átalakítás után azt kapjuk, hogy

$$r = - \frac{n_1 p_1 + n_2 p_2 + n_3 p_3 + n_0}{n_1^2 + n_2^2 + n_3^2}$$

A merőleges egyenes P -től a dőléspontig tartó szakaszának l hosszára fennáll a következő összefüggés:

$$l = |r| |\mathbf{n}|.$$

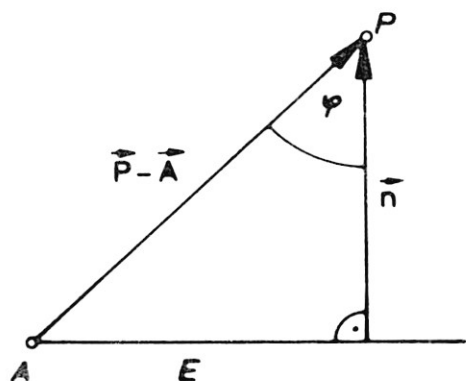
Mivel

$$|\mathbf{n}|^2 = n_1^2 + n_2^2 + n_3^2,$$

igaz, hogy

$$l = \left| \frac{n_1 p_1 + n_2 p_2 + n_3 p_3 + n_0}{\sqrt{n_1^2 + n_2^2 + n_3^2}} \right|. \quad (***)$$

Ez azonban megegyezik a vektorszámítás eredményével, amit úgy kapunk, hogy a $\mathbf{P} - \mathbf{A}$ összekötő vektort a normálvektorra vetítjük. A vetület hossza a P pont és a g egyenes távolsága.



$$\frac{(\mathbf{P} - \mathbf{A}) \cdot \mathbf{n}}{|\mathbf{n}|} = l$$

A (***) egyenlet nem más, mint a sík egyenlet normálalakjának a transzformáltja. Az összefüggést *O. L. Hesse* német matematikus (1811—1874) tiszteletére *Hesse-féle normálalak*-nak nevezzük.

A *Hesse-féle normálalakot* gyakran vektoriálisan írjuk fel:

$$\mathbf{n}_0 \cdot (\mathbf{X} - \mathbf{A}) = 0.$$

(A *Hesse-féle normálalak* egyszerűen adódik a következő gondolatmenet alapján: mivel a sík normálvektora éppen azáltal jellemzi a síkot, hogy annak minden vektorára merőleges, ezért ha a sík egy rögzített pontja A , akkor a sík bármely X pontjára $\mathbf{X} - \mathbf{A} \perp \mathbf{n}_0$, azaz

$$\mathbf{n}_0 \cdot (\mathbf{X} - \mathbf{A}) = 0.)$$

Ha az \mathbf{n}_0 normálvektor előjelét negatívnak választjuk, a *Hesse-féle normálalakból* azt is megtudjuk, hogy a sík melyik oldalán van a pont. A negatív előjel miatt az \mathbf{n}_0 normálvektor iránya mindig olyan, hogy a koordináta-rendszer O kezdőpontjából kifelé mutat.

Ha az

$$\frac{n_1 x_1 + n_2 x_2 + n_3 x_3 + n_0}{-\operatorname{sgn}(n_0) \cdot \sqrt{n_1^2 + n_2^2 + n_3^2}} = d$$

egyenletbe helyettesítjük a P pont koordinátáit, akkor a P pont helyzetétől függ a d előjele:

$d > 0$ a koordináta-rendszer O kezdőpontja és a P pont az E sík két különböző oldalán fekszik;

$d = 0$ $P \in E$ (vagyis a P pont az E síkban van);

$d < 0$ a koordináta-rendszer O kezdőpontja és a P pont az E sík azonos oldalán fekszik.

1.1.5 VEKTORTÉR, BÁZIS, DIMENZIÓ

Az eddigi fogalmak és gondolatok többé-kevésbé szemléletesek voltak, ugyanakkor néhány nagyon fontos fogalmon átsiklottunk. Ennek ellenére már lehet az Olvasónak megközelítő elképzelése arról, hogy szemléletesen mi is egy vektortér. A következő fejezet nagyon formális lesz, mivel a fogalmakat absztrakt módon általánosítjuk. A vektortér, valamint a bázis és a dimenzió fogalmát csak a teljesség kedvéért tárgyaljuk. Ezek ismerete nem előfeltétele annak, hogy a következő részek érthetőek legyenek. Az Olvasó akár a *Gauss*-féle eliminációs (kiküszöbölési) eljárás leírásánál is folytathatja a könyv további tanulmányozását.

A vektortér

Definíció: egy üres V halmazt, amelynek elemeire az összeadás (+) és a valós számokkal való szorzás (*) értelmezett, valós vektortérnek nevezünk, tehát $V = (V, +, *)$. A halmaz elemeit vektoroknak nevezzük. (Szemléletesen kifejezve egy V vektortérre példa az, amit eddig R^n -nel jelöltünk.)

Az összeadás axiómái

$$V_1: \mathbf{a} + \mathbf{b} = \mathbf{c}; \quad \mathbf{a}, \mathbf{b} \in V \Rightarrow \mathbf{a} + \mathbf{b} = \mathbf{c} \in V$$

a V halmaz az összeadásra nézve zárt

(Vegyük az R^n \mathbf{a} és \mathbf{b} két vektorát, akkor az \mathbf{a} és \mathbf{b} vektorok összegének is az R^n -ben kell lenni. Hallgatólagosan eddig ebből a feltételből indultunk ki.)

$$V_2: (\mathbf{a} + \mathbf{b}) + \mathbf{c} = \mathbf{a} + (\mathbf{b} + \mathbf{c}); \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in V$$

asszociativitás

$$V_3: \mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}; \quad \mathbf{a}, \mathbf{b} \in V$$

kommutativitás

(Mint már láttuk, ez a feltétel az eddig vizsgált műveletekre általában teljesült. Volt azonban kivétel is: a vektoriális szorzat.)

$$V_4: \mathbf{a} + \mathbf{0} = \mathbf{a}; \quad \mathbf{a} \in V$$

nullaelem létezése

(A nullaelemet bármely vektorhoz hozzáadva az eredetit kapjuk vissza.)

$$V_5: \mathbf{a} + \mathbf{a}^{-1} = \mathbf{0}$$

inverz elem létezése

(Minden $\mathbf{a} \in V$ -hez tartozik egy \mathbf{a}^{-1} , vagyis az \mathbf{a} -hoz tartozó inverz elem, amelyre a V_5 axióma igaz.)

Vegyük pl. az $\mathbf{a} \in R^n$ vektort! Ennek inverzét $-\mathbf{a}$ -val adhatjuk meg, mert az $\mathbf{a} + (-\mathbf{a}) = \mathbf{0}$ kielégíti a V_5 axiómát. V_5 teljesülése feltételezi a V_4 axióma érvényességét, mivel az axióma nem is volna felírható, ha nem létezne nullaelem. A következmény az lenne, hogy valamennyi

számolási törvény érvénytelenné válna, mert pl. igaz lehetne:

$$\mathbf{a} - \mathbf{a} \neq \mathbf{b} - \mathbf{b}.$$

A szorzás axiómái

$$V_6: k\mathbf{a} = \mathbf{b} \in V; \quad k \in R, \quad \mathbf{a} \in V$$

valós számmal való szorzásra nézve zárt

(Ha $\mathbf{a} \in V$, akkor ennek a vektornak bármilyen többszöröse is eleme a V halmaznak.)

$$V_7: 1\mathbf{a} = \mathbf{a}$$

egységlem létezése a valós számok halmazában

(A vektornak az egységgel való szorzása nem változtatja meg a vektort. Ennek az axiómának számunkra nincs jelentősége, mert mi csak valós számok fölött értelmezett vektorterekkel dolgozunk.)

$$V_8: k(\mathbf{a} + \mathbf{b}) = k\mathbf{a} + k\mathbf{b}$$
$$(k + l)\mathbf{a} = k\mathbf{a} + l\mathbf{a}$$

disztributivitás

Mindezek azt a benyomást kelthetik, hogy formalitások „zagyvalékát” tártuk az Olvasó elé. A tény azonban az, hogy ezen minimális követelmények teljesülése nélkül nem volnának lehetségesek a vektorokkal való műveletek. Az axiómákat akár azonnal el is felejtheti az Olvasó, amennyiben azok a vizsgált vektorterekben mindig teljesülnek. Ez a szakasz talán jelzi azt, hogy az egyszerűnek tűnő ismeretek mögött milyen sok matematika van. (A matematika ui. a konkrét vektorterek — mint amilyen pl. az R^n is — tulajdonságait absztrahálva definiálja a vektortér fogalmát az előbbi axiómákkal. Ezáltal az absztrakt vektortérben igazolt állítások valamennyi konkrét vektortérben érvényesek.)

Bázis és dimenzió

Bázisnak nevezzük a V halmaz lineárisan független vektorainak azon halmazát, amelynek elemeivel alkotott lineáris kombinációk a V halmaz valamennyi elemét előállítják. A lineárisan független bázisvektorok számát *dimenzió*nak nevezzük (jelölése: $\dim V$).

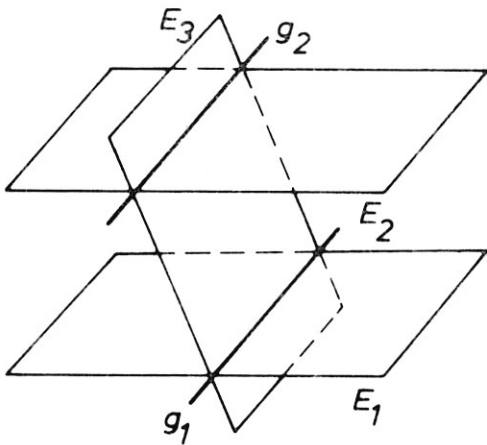
1.2 Mátrixszámítás

1.2.1 A GAUSS-FÉLE ELIMINÁCIÓS ELJÁRÁS

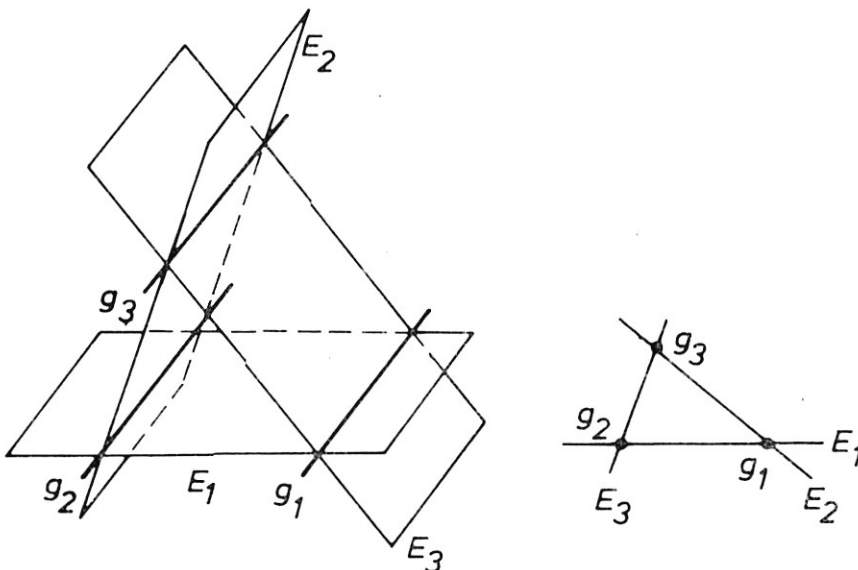
A mátrixokkal való műveletek során rendkívüli fontosságú a Gauss-féle eliminációs eljárás. A módszer ismerete más szempontból is (pl. lineáris egyenletrendszerek általános megoldása) érdeklődésre tarthat számot.

Vizsgáljuk meg a következő problémát: adott az R^3 három síkja: E_1, E_2, E_3 . Nézzük meg, hogy egymáshoz képest milyen helyzetben lehetnek ezek a síkok!

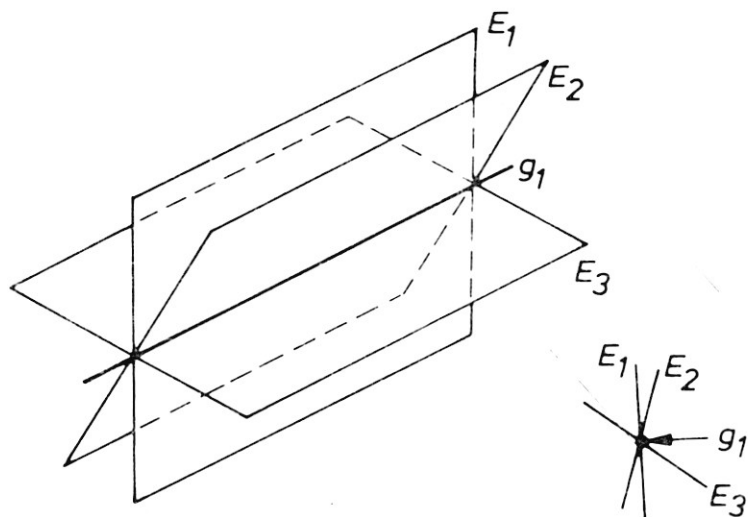
- a) A 3 sík egymással párhuzamos.
- b) 2 sík párhuzamos egymással. Ezt a kettőt metszi a harmadik, a g_1 és a g_2 metszésvonalban.



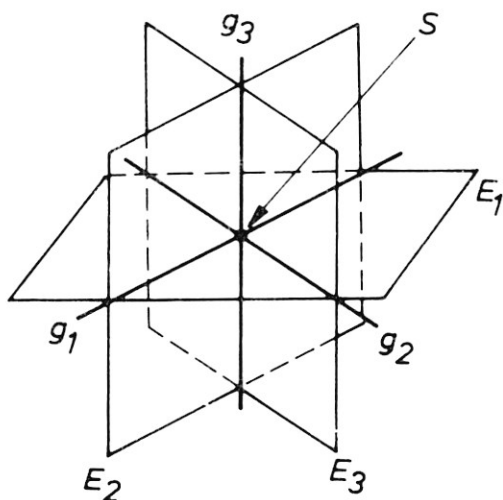
- c) A 3 sík helyzete olyan, hogy egyik sem párhuzamos a másikkal. Ekkor ismét három esetet kell megkülönböztetnünk.
 - c₁) A 3 sík helyzete olyan, hogy bármely kettőt véve, egy-egy metszésvonalban metszik egymást és a metszésvonalak (g_1, g_2, g_3) párhuzamosak egymással.



c₂) A 3 sík egy egyenesre illeszkedik.



c₃) A 3 sík úgy metszi egymást, hogy a g₁, g₂, g₃ egyenesek egy pontban metszik egymást.



Milyen megállapítást tehetünk?

Mint korábban már láttuk, egy sík helyzete jellemezhető a sík normálvektorával. A három sík felírható a következő formában:

$$E_1: a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = 0;$$

$$E_2: a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = 0;$$

$$E_3: a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = 0,$$

ahol az

$$\begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \end{pmatrix}, \begin{pmatrix} a_{21} \\ a_{22} \\ a_{23} \end{pmatrix}, \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} \end{pmatrix}$$

vektorok a 3 sík normálvektorai.

A síkok kölcsönös helyzetéről szerezhethetünk információt, ha ezeket a normálvektorokat lineáris függőség szempontjából megvizsgáljuk. Ha pl. mind a 3 sík párhuzamos egymással, akkor

mindegyik normálvektor előállítható az egyik normálvektor skalárszorosaként, bármelyiket válasszuk is a három közül. A síkok egymáshoz viszonyított helyzetének meghatározása végül is a lineáris egyenletrendszer megoldásához vezet. A következőkben látni fogjuk, hogyan oldható meg egy ilyen egyenletrendszer.

Írjuk fel az együtthatókat téglalap alakú táblázatos formában:

$$\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & -a_{14} \\ a_{21} & a_{22} & a_{23} & -a_{24} \\ a_{31} & a_{32} & a_{33} & -a_{34} \end{array}$$

Egy ilyen táblázatot mátrixnak nevezünk. A mátrix sorait (ill. oszlopait) felfoghatjuk egy vektor koordinátáiként is, így lehet beszélni a mátrix sorairól (oszlopairól), mint sorvektorokról (oszlopvektorokról.)

Az egyenletrendszer megoldásához szükségünk van egy sorra, lehetőség szerint sok nullával. Ehhez átalakítjuk a rendszert.

Legyen $a_{11} \neq 0$.

$$\begin{array}{l} (1) \\ (2) \\ (3) \end{array} \begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & -a_{14} \\ a_{21} & a_{22} & a_{23} & -a_{24} \\ a_{31} & a_{32} & a_{33} & -a_{34} \end{array} \begin{array}{l} \left. \begin{array}{l} \cdot \frac{-a_{21}}{a_{11}} \\ \cdot \frac{-a_{31}}{a_{11}} \end{array} \right\} \\ \leftarrow \\ \leftarrow \end{array}$$

Az eredmény:

$$\begin{array}{l} (1) \\ (2) \\ (3) \end{array} \begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & -a_{14} \\ 0 & a_{22} - \frac{a_{21}a_{12}}{a_{11}} & a_{23} - \frac{a_{13}a_{21}}{a_{11}} & -a_{24} + \frac{a_{14}a_{21}}{a_{11}} \\ 0 & a_{32} - \frac{a_{31}a_{12}}{a_{11}} & a_{33} - \frac{a_{13}a_{31}}{a_{11}} & -a_{34} + \frac{a_{14}a_{31}}{a_{11}} \end{array}$$

Az (1) sort megszoroztuk a $-a_{21}/a_{11}$, ill. $-a_{31}/a_{11}$ valós számokkal és hozzáadtuk a (2), ill. (3) sorokhoz. Kimutatható, hogy ezáltal a rendszer megoldásainak halmaza nem változik. Ezt az eljárást elemi sorátalakításnak nevezzük. A következő sémát kapjuk:

$$\begin{array}{ccc|c} c_{11} & c_{12} & c_{13} & c_{14} \\ 0 & c_{22} & c_{23} & c_{24} \\ 0 & 0 & c_{33} & c_{34} \end{array}$$

Az egyenletrendszert lépésenként — alulról kezdve, felfelé haladva — oldjuk meg ($c_{33}x_3 = c_{34} \Rightarrow x_3 = c_{34}/c_{33} \Rightarrow (2)$ -be stb.).

A megértés megkönnyítésére bemutatunk egy példát.

Adott három sík: E_1, E_2, E_3 .

$$\begin{array}{l} E_1: 2x_1 + x_2 + 3x_3 - 9 = 0 \\ E_2: x_1 - 2x_2 + x_3 + 2 = 0 \\ E_3: 3x_1 + 2x_2 + 2x_3 - 7 = 0 \end{array}$$

Az egyenletrendszert a *Gauss*-féle eliminációs eljárással oldjuk meg.

$$\begin{array}{cccc}
 2 & 1 & 3 & 9 \\
 1 & -2 & 1 & -2 \\
 3 & 2 & 2 & 7 \\
 1 & -2 & 1 & -2 \\
 2 & 1 & 3 & 9 \\
 3 & 2 & 2 & 7 \\
 \hline
 1 & -2 & 1 & -2 \\
 0 & 5 & 1 & 13 \\
 0 & 8 & -1 & 13 \\
 \hline
 (1) & 1 & -2 & 1 & -2 \\
 (2) & 0 & 5 & 1 & 13 \\
 (3) & 0 & 0 & -13/5 & -39/5
 \end{array}$$

vagy az egyenletek felcserelésével:

$$\begin{array}{l}
 (3)\text{-ből } (-13/5)x_3 = -39/5 \quad x_3 = 3 \\
 (2)\text{-ből } 5x_2 + 3 = 13 \quad x_2 = 2 \\
 (1)\text{-ből } x_1 - 4 + 3 = -2 \quad x_1 = -1
 \end{array}$$

Megoldásként tehát a

$$\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}$$

pontot kaptuk, amely mindhárom síkon rajta fekszik.

Mi lenne az eredmény, ha az E_3 síkot a következő módon adnánk meg:

$$E_3: 3x_1 + x_2 + 4x_3 - 7 = 0$$

A *Gauss*-féle eliminációs eljárással kapott eredmény

$$\begin{array}{ccc|c}
 1 & -2 & 1 & -2 \\
 0 & 5 & 1 & 13 \\
 0 & 0 & 0 & 13
 \end{array}$$

szemmel láthatóan a $0 = 13$ ellentmondáshoz vezet.

Vizsgáljuk meg most alaposabban a síkok normálvektorait!

$$\mathbf{n}_1 = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}, \quad \mathbf{n}_2 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, \quad \mathbf{n}_3 = \begin{pmatrix} 3 \\ -1 \\ 4 \end{pmatrix}$$

Nyilvánvalóan igaz az $\mathbf{n}_1 + \mathbf{n}_2 = \mathbf{n}_3$ összefüggés, azaz a normálvektorok nem lineárisan függetlenek.

A $0 = 13$ ellentmondás alapján mondhatjuk, hogy nem létezik olyan pont, amely mindhárom sík közös pontja lenne. Ebből viszont a c) eset alapján az következik, hogy két-két sík metszésvonalai párhuzamosak egymással.

Ezzel szemben, ha a (3) sorban $0=0$ eredményt kapnánk, akkor megoldható lenne az egyenletrendszer, mégpedig végtelen sok megoldással, amit egy k paraméter segítségével írhatunk le.

Igaz lenne:

$$\begin{aligned}x_1 - 2x_2 + x_3 &= -2; \\5x_2 + x_3 &= 13.\end{aligned}$$

Helyettesítsük be x_3 helyett a k paramétert, és akkor kapjuk:

$$5x_2 + k = 13 \Rightarrow x_2 = \frac{13-k}{5}$$

$$x_1 - 2\left(\frac{13-k}{5}\right) + k = -2$$

$$x_1 - \frac{26}{5} + \frac{2k}{5} + k = -2$$

$$x_1 = \frac{16}{5} - \frac{7}{5}k$$

Ezt vektoriálisan felírva:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16/5 \\ 13/5 \\ 0 \end{pmatrix} + k \begin{pmatrix} -7/5 \\ -1/5 \\ 1 \end{pmatrix}$$

Emeljük ki az $1/5$ -öt az irányvektorból, és akkor adódik:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16/5 \\ 13/5 \\ 0 \end{pmatrix} + k' \begin{pmatrix} -7 \\ -1 \\ 5 \end{pmatrix}$$

Ez egy egyenes egyenlete. A három sík egy egyenesben metszi egymást (c_2 -es eset).

Ez a séma általánosítható: ha — mint előbb láttuk — a síkok egyenleteit mátrix alakban írjuk fel, akkor elemi oszlopátalakítással meghatározható a mátrix ún. rangja. Az mindegy, hogy sor-, vagy oszlopátalakítást végzünk, mert a sorvektorok, ill. az oszlopvektorok alapján meghatározott rang egymással egyenlő. Mindenesetre mindig csak egyféle: vagy sor-, vagy oszlopátalakítást alkalmazhatunk.

A mátrix rangját a következőképpen definiáljuk: az A mátrix rangján (r) az egymástól lineárisan független oszlopvektorok maximális számát értjük. Az együtthetők mátrixának rangját r -rel, a kibővített mátrix rangját pedig \tilde{r} -mal jelöljük.

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

$$\tilde{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

Általános érvényes a következő:

r	\tilde{r}	Megoldás
3	3	pont
2	3	párhuzamos egyenesek (2 vagy 3)
2	2	1 közös egyenes
1	2	3 párhuzamos sík, nincs metszéspont
1	1	a 3 sík azonos!
0	1	nem sík!

A mátrix fogalma azáltal válik igazán hasznossá, hogy a mátrixokkal különböző algebrai műveleteket lehet végezni. Ezeket a mátrixműveleteket ismerjük meg a következő pontban.

1.2.2 MŰVELETEK MÁTRIXOKKAL

Az $m \times n$ méretű mátrix m sorba és n oszlopba rendezett elemek táblázata. Az m sorból és n oszlopból álló mátrixok M halmazát a következő módon jelöljük: $M(m \times n, R)$, ahol R a valós számok halmaza.

Legyen A egy $m \times n$ -es mátrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

A táblázat alakja m és n értékétől függ. Az olyan mátrixot, amelynek csak egy sora ($m = 1$) vagy csak egy oszlopa ($n = 1$) van, vektornak (sor-, ill. oszlopmátrixnak) nevezzük.

Ha egy mátrixnak ugyanannyi sora van, mint ahány oszlopa ($m = n$), akkor *négyzetes* vagy *kvadratikus* mátrixról beszélünk. A kvadratikus ($n \times n$ típusú) mátrixok halmazának jelölése: $M(n, R)$.

A *nullmátrix* (vagy *zérusmátrix*) minden eleme 0:

$$\mathbf{0} = \begin{bmatrix} 0 & \dots & 0 \\ \dots & \dots & \dots \\ \dots & 0 & \dots \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{bmatrix}$$

Az *egységmátrix* olyan mátrix, amelynek minden főátlóbeli eleme 1, és minden más helyen 0 áll.

$$E = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & 1 & \dots \\ \dots & \dots & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Egy mátrixot, amely

$$E_n = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & 1 & 0 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix}$$

n

n

típusú, n -ed rendű kvadratikus egységmatrixnak nevezzük.

Mátrixok összeadása

Legyen az A és a B mátrix az $M(m \times n, R)$ halmaz mátrixa. Az A és B mátrixok összegén azt a mátrixot értjük, amelynek elemei rendre az A és B mátrix megfelelő elemeinek összegéből adódnak.

$$A + B = C$$

$$C = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \dots a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} \dots a_{2n} + b_{2n} \\ \vdots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} \dots a_{mn} + b_{mn} \end{bmatrix}$$

Értelemszerűen csak azonos méretű mátrixokat lehet összeadni.

Mátrix szorzása skalárral

Legyen k egy valós szám. Az A mátrix k -szorosán azt a mátrixot értjük, amelynek minden eleme az A mátrix megfelelő elemének k -szorosa.

$$kA = \begin{bmatrix} ka_{11} & ka_{12} \dots ka_{1n} \\ \vdots & \vdots \\ ka_{m1} & ka_{m2} \dots ka_{mn} \end{bmatrix}$$

Mátrixok szorzása

Legyen az A mátrix $A \in M(m \times n, R)$ és a B mátrix $B \in M(n \times p, R)$:

$$A = \begin{bmatrix} a_{11} & a_{12} \dots a_{1n} \\ a_{21} & a_{22} \dots a_{2n} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \dots a_{mn} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \cdots b_{1p} \\ b_{21} & b_{22} \cdots b_{2p} \\ \vdots & \vdots \\ b_{n1} & b_{n2} \cdots b_{np} \end{bmatrix}$$

A szorzatmátrixot a következőképpen értelmezzük:

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} c_{11} & c_{12} \cdots c_{1p} \\ c_{21} & c_{22} \cdots c_{2p} \\ \vdots & \vdots \\ c_{m1} & c_{m2} \cdots c_{mp} \end{bmatrix} \in M(m \times p, R),$$

ahol

$$c_{ik} = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix} \cdot \begin{bmatrix} b_{1k} \\ b_{2k} \\ \vdots \\ b_{nk} \end{bmatrix}$$

Szemléletesen kifejezve: a szorzatmátrix azon elemét, amely az i -edik sor és a k -adik oszlop metszéspontjában van, úgy kapjuk meg, hogy az \mathbf{A} mátrix i -edik sorát skalárisan szorozzuk a \mathbf{B} mátrix k -adik oszlopával.

Ehhez elengedhetetlen feltétel, hogy az \mathbf{A} mátrix oszlopainak száma egyenlő legyen a \mathbf{B} mátrix sorainak számával. Ha tehát az \mathbf{A} mátrix $m \times n$ méretű, akkor a \mathbf{B} mátrixnak $n \times p$ méretűnek kell lennie, és a \mathbf{C} mátrix $m \times p$ méretű lesz.

A mátrixműveletekre érvényes szabályok

$$M_1: (k+l)\mathbf{A} = k\mathbf{A} + l\mathbf{A}$$

$$k(\mathbf{A} + \mathbf{B}) = k\mathbf{A} + k\mathbf{B}$$

$$M_2: (\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C})$$

$$M_3: \mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A} \text{ (általában!)}$$

(A sorok, ill. az oszlopok számára tett feltevések miatt, ha az $\mathbf{A} \cdot \mathbf{B}$ szorzás elvégezhető, akkor a $\mathbf{B} \cdot \mathbf{A}$ általában el se végezhető.)

$$M_4: \text{Két mátrix szorzata akkor is lehet nullamátrix, ha egyik tényezője sem nullamátrix, azaz:}$$

$$\mathbf{A} \neq \mathbf{0}, \mathbf{B} \neq \mathbf{0}, \text{ de } \mathbf{A} \cdot \mathbf{B} = \mathbf{0}$$

Pl.:

$$\begin{bmatrix} 5 & 2 & -2 & 3 \\ 9 & 2 & -3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 2 & 2 & 2 \\ -1 & 3 & -5 \\ 16 & 8 & 24 \\ 8 & 0 & 16 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Következésképpen fennáll, hogy:

$$\mathbf{A} \cdot \mathbf{B} = \mathbf{0}, \quad \mathbf{A} \neq \mathbf{0} \Rightarrow \mathbf{B} = \mathbf{0} \text{ (általában)}$$

$$\mathbf{A} \cdot \mathbf{B} = \mathbf{A} \cdot \mathbf{C}, \quad \mathbf{A} \neq \mathbf{0} \Rightarrow \mathbf{B} = \mathbf{C} \text{ (általában)}$$

Mátrixok transzponálása

A mátrixok halmazán végezhető műveleteknek van egy további esete, amellyel eddig még sehhol sem találkoztunk. Ez az ún. mátrixtranszponálás. Egy A mátrix transzponáltját a t_A szimbólummal jelöljük. Az A mátrix transzponáltját úgy kapjuk meg, hogy a sorvektorokat felcseréljük az oszlopvektorokkal.

$$t \begin{bmatrix} 2 & -1 & 3 & 6 \\ -1 & 0 & 5 & 3 \\ 3 & 5 & -4 & 2 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 3 \\ -1 & 0 & 5 \\ 3 & 5 & -4 \\ 6 & 3 & 2 \end{bmatrix}$$

Kvadratikus mátrixok inverze

A kvadratikus mátrixok fontos tulajdonsága, hogy bizonyos feltételek mellett invertálhatók. Minden reguláris (lineárisan független sorokból álló) A mátrixnak van inverze. Egy négyzetes mátrixot regulárisnak nevezünk akkor, ha a mátrix rangja egyenlő a mátrix rendjével ($r=n$).

Ha a A mátrix az $M(n, R)$ halmaz egy mátrixa, akkor igaznak kell lenni az $r=n$ egyenlőségnek. Ennek magyarázatát majd látni fogjuk. Az A mátrix inverz mátrixának (jelölése: A^{-1}) azt a mátrixot nevezük, amelyre érvényes:

$$\begin{aligned} A \cdot A^{-1} &= E_n && \text{és} \\ A^{-1} \cdot A &= E_n. \end{aligned}$$

Az A mátrix inverzén tehát azt a mátrixot értjük, amellyel az A mátrixot megszorozva (balról vagy jobbról) egységmátrixot kapunk. A művelet kommutatív, ami a mátrixok szorzására általánosan nem érvényes.

Az inverz mátrix kiszámítása előtt ejtsünk még néhány szót a mátrixok szorzásáról!

1.2.3 LINEÁRIS LEKÉPEZÉS

Lineáris leképezésnek nevezünk egy

$$F(V \rightarrow W)$$

függvényt, amelyre tetszőleges $\mathbf{v}, \mathbf{w} \in V$ és $k \in R$ esetén:

$$\begin{aligned} L_1: & F(\mathbf{v} + \mathbf{w}) = F(\mathbf{v}) + F(\mathbf{w}); \\ L_2: & F(k\mathbf{v}) = kF(\mathbf{v}). \end{aligned}$$

A mátrixok szorzásának speciális esete a következő:

$$\begin{bmatrix} a_{11} & a_{12} \cdots a_{1n} \\ a_{21} & a_{22} \cdots a_{2n} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \cdots a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Itt a második mátrix egyetlen oszlopból áll, ez tehát egy vektor. Ha a szorzást elvégezzük, azonnal látható, hogy az eredmény is egy vektor lesz.

$$\begin{bmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & a_{22} \cdots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

A szorzás egyenlete a mátrixok szokásos jelölésével:

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{Y}$$

ahol az \mathbf{X} és az \mathbf{Y} mátrixok vektorok.

A következőkben megmutatjuk, hogy ezzel egy lineáris leképzést definiáltunk.

$$L_1: F(\mathbf{x} + \mathbf{y}) = F(\mathbf{x}) + F(\mathbf{y})$$

$$\begin{aligned} \mathbf{A}(\mathbf{X} + \mathbf{Y}) &= \mathbf{A} \cdot \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix} = \\ &= \begin{bmatrix} a_{11}(x_1 + y_1) + \cdots + a_{1n}(x_n + y_n) \\ \vdots \\ a_{m1}(x_1 + y_1) + \cdots + a_{mn}(x_n + y_n) \end{bmatrix} = \\ &= \begin{bmatrix} a_{11}x_1 + a_{11}y_1 + \cdots + a_{1n}x_n + a_{1n}y_n \\ \vdots \\ a_{m1}x_1 + a_{m1}y_1 + \cdots + a_{mn}x_n + a_{mn}y_n \end{bmatrix} = \mathbf{A} \cdot \mathbf{X} + \mathbf{A} \cdot \mathbf{Y} \end{aligned}$$

$$L_2: kF(\mathbf{x}) = F(k\mathbf{x})$$

$$k(\mathbf{A} \cdot \mathbf{x}) = (k\mathbf{A}) \cdot \mathbf{x} = \mathbf{A} \cdot (k\mathbf{x})$$

Most már a korábban tárgyalt egyenletrendszereket teljes egészében leírhatjuk mátrixokkal is. Az egyenletrendszer a következőképpen néz ki:

$$\mathbf{A} \cdot \mathbf{x} + \mathbf{b} = \mathbf{0}$$

$$\begin{bmatrix} a_{11}x_1 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \end{bmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Ezt az egyenletrendszert mátrixokkal azonban csak egy további feltétel teljesülése esetén tudjuk megoldani.

Ha ui. \mathbf{A} egy n -ed rendű kvadratikus mátrix, akkor érvényesek a következő összefüggések:

$$\begin{aligned} \mathbf{A} \cdot \mathbf{x} + \mathbf{b} &= \mathbf{0} \\ \Leftrightarrow \mathbf{A} \cdot \mathbf{x} &= -\mathbf{b} \\ \Leftrightarrow \mathbf{A}^{-1} \cdot \mathbf{A} \cdot \mathbf{x} &= \mathbf{A}^{-1} \cdot (-\mathbf{b}) \\ \Leftrightarrow \mathbf{E}_n \cdot \mathbf{x} &= \mathbf{A}^{-1} \cdot (-\mathbf{b}) \\ \Leftrightarrow \mathbf{x} &= \mathbf{A}^{-1} \cdot (-\mathbf{b}) \end{aligned}$$

Írjuk a $-\mathbf{b}$ helyett \mathbf{y} -t, akkor azt kapjuk, hogy:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \Leftrightarrow \mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{y}$$

Ezzel egy egyértelműen meghatározott megoldást kapunk, azaz egy pontot. Az \mathbf{A} mátrix rangjának egyenlőnek kell lenni a mátrix n rendjével. Ez a feltétele ui. az \mathbf{A}^{-1} inverz mátrix létezésének.

Ezután már meg tudjuk határozni egy lineáris egyenletrendszer megoldását. Mindenesetre szükségünk van ehhez az \mathbf{A}^{-1} inverz mátrixra. Ennek kiszámítására két elméleti módszer van.

1) Az $\mathbf{A} \cdot \mathbf{x} = \mathbf{E}_n$ meghatározatlan állításából levezetjük a Gauss-féle eliminációs eljárással az

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{E}_n = \mathbf{A}^{-1}$$

alakot. Ezt a módszert mindenekelőtt olyan mátrixok esetében alkalmazzák, amelyek rangjára az $r > 4$ egyenlőtlenség fennáll. Számunkra gyakorlatilag ezért nincs jelentősége.

2) Fennáll a következő egyenlőség:

$$\mathbf{A}^{-1} = \frac{1}{\det \mathbf{A}} \begin{bmatrix} a_{11}^+ & a_{12}^+ & \dots & a_{1n}^+ \\ a_{21}^+ & a_{22}^+ & \dots & a_{2n}^+ \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}^+ & a_{m2}^+ & \dots & a_{mn}^+ \end{bmatrix},$$

ahol az a_{11}^+ az a_{11} -hez tartozó adjungált stb.

Az a_{ik} komponenshez tartozó adjungáltat a következő módon számítjuk ki: töröljük az \mathbf{A} mátrix i -edik sorát és k -adik oszlopát. A maradék számokból alkotott determinánst megszorozzuk a $(-1)^{-(i+k)}$ tényezővel, majd kiszámítjuk a determináns értékét. Tekintettel arra, hogy a determinánsokat csak akkor tudjuk könnyen kiszámítani, ha a determináns legfeljebb 3 soros, nagyobb mátrixok esetében vagy az 1) módszert, vagy valamilyen egyéb numerikus eljárást alkalmazunk.

1. példa: $a_{21}^+ = ?$

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

Az a_{21} elemhez tartozó adjungált:

$$(-1)^{(2+1)} \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} = (-1)(a_{12}a_{33} - a_{32}a_{13})$$

2. példa:

$$\begin{bmatrix} 3 & 2 & 1 \\ 1 & 0 & 2 \\ 4 & 1 & 3 \end{bmatrix}^{-1} = \frac{1}{5} \begin{bmatrix} -2 & 5 & 1 \\ -5 & 5 & 5 \\ 4 & -5 & -2 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} -2 & -5 & 4 \\ 5 & 5 & -5 \\ 1 & 5 & -2 \end{bmatrix}$$

Ha most pl. a

$$\begin{aligned}3x_1 + 2x_2 + x_3 &= 1 \\ x_1 + 2x_3 &= 2 \\ 4x_1 + x_2 + 3x_3 &= 3\end{aligned}$$

egyenletrendszer megoldását kell megtalálnunk, akkor azt a következő módon határozhatjuk meg:

$$\begin{bmatrix} 3 & 2 & 1 \\ 1 & 0 & 2 \\ 4 & 1 & 3 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \frac{1}{5} \begin{bmatrix} -2 & -5 & 4 \\ 5 & 5 & -5 \\ 1 & 5 & -2 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Lineáris egyenletrendszereket nagyon áttekinthető és egyszerű formában tudunk mátrixok segítségével kifejezni.

Az eddig tárgyaltak azt a benyomást keltették bennünk, hogy elsősorban lineáris egyenletrendszerek megoldásával foglalkozunk. Az elmondottakkal az volt a célunk, hogy szemléletessé tegyük az egyszerűbb mátrixműveleteket. Számunkra sokkal fontosabb azonban az, hogy a mátrixok révén olyan segédeszköz került a kezünkbe, amely lehetővé teszi, hogy vektorfüggvényeket definiáljunk.

A következő analógia mindezt világossá teszi:

$$\begin{array}{ll} y = f(x) & \mathbf{y} = F(\mathbf{x}) \\ y = 3x + 1 & \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \end{array}$$

Az első esetben számokhoz számokat, a másodikban vektorokhoz vektorokat rendeltünk.

A mátrix funkciója érthetőbb lesz a következő példa alapján:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix}$$

Legyen $\mathbf{y} = \mathbf{A} \cdot \mathbf{x}$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Helyettesítsük be az egyenlőségbe az $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ standard bázisvektorokat:

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 3 \end{pmatrix}$$

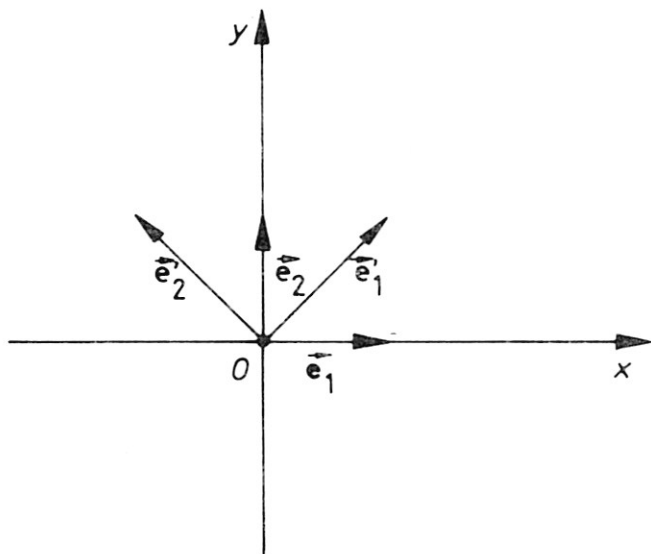
$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 1 \\ 0 & 3 & -2 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ -2 \end{pmatrix}$$

Amint látjuk, a bázisvektorok képei a mátrix oszlopvektorai. A bázisvektorok képei egyben bázist alkotnak az $\mathbf{Y} = \mathbf{A} \cdot \mathbf{X}$ egyenlettel meghatározott képtérben. Ez a mátrixműveletek linearitásából adódik:

$$\mathbf{A}(a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + \dots + a_n\mathbf{e}_n) = \mathbf{A}(a_1\mathbf{e}_1) + \mathbf{A}(a_2\mathbf{e}_2) + \dots + \mathbf{A}(a_n\mathbf{e}_n) = a_1\mathbf{A}(\mathbf{e}_1) + a_2\mathbf{A}(\mathbf{e}_2) + \dots + a_n\mathbf{A}(\mathbf{e}_n).$$

Azaz a tér egy tetszőleges vektorának képe (amelyet előzőleg az $\mathbf{e}_1, \dots, \mathbf{e}_n$ bázisvektorok lineáris kombinációjaként állítottunk elő) előáll a bázisvektorok képeinek lineáris kombinációjaként.

Szeretnénk a mátrixok hasznosságát egy nagyon részletes példával szemléltetni. (Ajánlatos ezt a példát alaposan átgondolni, mert ez az alapja csaknem minden mátrixműveletnek, amelyet később alkalmazni fogunk.)



Adott az R^2 , valamint a két bázis, $\mathbf{e}_1, \mathbf{e}_2$ és $\mathbf{e}'_1, \mathbf{e}'_2$:

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{e}'_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}; \quad \mathbf{e}'_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Jelöljük az \mathbf{e}_1 és \mathbf{e}_2 egységvektorok által meghatározott koordináta-rendszert S -sel, azt pedig, amelyet az \mathbf{e}'_1 és \mathbf{e}'_2 egységvektorok határoznak meg, S' -vel.

Az S' rendszerben \mathbf{e}'_1 és \mathbf{e}'_2 koordinátái:

$$\mathbf{e}'_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \mathbf{e}'_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Az S rendszerbeli koordinátákat — mint látható — így kapjuk meg:

$$\mathbf{e}'_{1S} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{e}'_{2S} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Tehát, ha ismerjük egy pont koordinátáit az S' rendszerben, akkor az

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

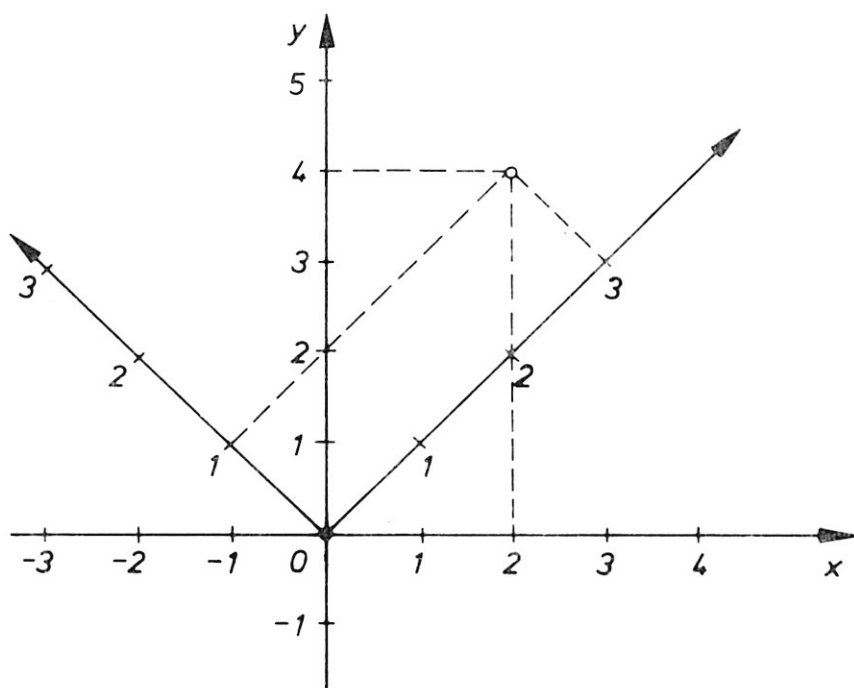
mátrix segítségével az S rendszerbeli koordinátáit is meg tudjuk határozni.

Pl. adott a $P = (3,1)$ pont az S' rendszerben.

A pont S rendszerbeli koordinátáira a $(2,4)$ értéket kapjuk, mert

$$\mathbf{P} = \mathbf{A} \cdot \mathbf{P}' = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}.$$

Az eredmény helyességéről meggyőződhetünk a következő rajz alapján:



Igaz tehát az

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{x}'$$

egyenlőség, amelyben az \mathbf{A} mátrix oszlopait az $\mathbf{e}_1, \mathbf{e}_2$ bázisvektoroknak az S rendszer bázisára vonatkoztatott koordinátái alkotják. Látható, hogy ezt az egyenletet fordítva is fel tudjuk írni, ui. ha az \mathbf{A} mátrix rangja $r=2$, akkor igaz az

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{x}' \Leftrightarrow \mathbf{A}^{-1} \cdot \mathbf{x} = \mathbf{x}'$$

összefüggés.

Az \mathbf{A}^{-1} inverz mátrixot a legegyszerűbben az előbbieken bemutatott 2) módszer szerint határozhatjuk meg.

$$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}^{-1} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Az eredményt ellenőrizzük az

$$\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{E}_n$$

összefüggés felhasználásával.

$$\frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Ha most a $P = (2,4)$ pont koordinátáit meg akarjuk határozni az S' rendszerben, akkor a

$$\mathbf{P}' = \mathbf{A}^{-1} \cdot \mathbf{P}$$

egyenlet alapján

$$\mathbf{P}' = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}.$$

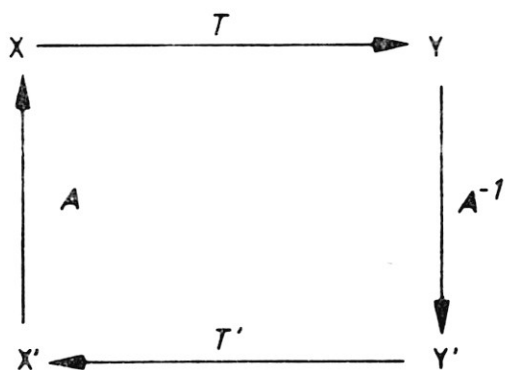
Tehát az \mathbf{A} és az \mathbf{A}^{-1} mátrixok segítségével nagyon egyszerűen át tudunk térni az S rendszerből az S' rendszerbe és viszont.

Tételezzük fel, hogy az

$$\mathbf{y} = \mathbf{T} \cdot \mathbf{x}$$

összefüggéssel megadtuk a \mathbf{T} mátrix leképezését az S rendszerben. Az érdekel most bennünket, hogy hogyan néz ki a \mathbf{T} mátrix, ami az S' rendszerben írja le ugyanezt a leképezést.

Nézzük meg a következő diagramot:



Egy adott \mathbf{x}' pontnál az S' rendszerből először vissza kell térnünk az S rendszerbe, itt a T mátrix leképezését el kell végezni, majd ezt követően ismét át kell térni S rendszerből az S' rendszerbe.

Igaz tehát a következő egyenlőség:

$$\mathbf{y}' = \mathbf{A}^{-1} \cdot \mathbf{T} \cdot \mathbf{A} \cdot \mathbf{x}'$$

Tehát egy olyan képletet kaptunk, amelynek segítségével egy ábrát egy rendszerből egy másik rendszerbe tudunk transzformálni.

1.2.4 PÉLDÁK EGYSZERŰ MÁTRIXMŰVELETEKRE

A következőkben bemutatunk néhány példát mátrixműveletekkel kapcsolatos feladatok megoldására. Azt ajánljuk az Olvasónak, hogy a feladatokat próbálja meg önállóan megoldani és a megoldásokat csak később ellenőrizze.

1. feladat

Az \mathbf{e}_1 és \mathbf{e}_2 egységvektorok az R^2 standard bázisát alkotják.

a) Igaz-e és miért, hogy az

$$\mathbf{e}'_1 = 2\mathbf{e}_1 + \mathbf{e}_2; \quad \mathbf{e}'_2 = -\mathbf{e}_1 + \mathbf{e}_2$$

egyenletekkel egy új bázist definiálunk?

b) Számítsuk ki egy vektor $(x_1; x_2)$ régi koordinátáiból az $(x'_1; x'_2)$ koordinátákat! Végezzük el az átszámítást visszafelé is!

c) Rajzoljuk be egy koordináta-rendszerbe az

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

pontokat és a

$$k \begin{pmatrix} 1 \\ -1 \end{pmatrix}; \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} + k \begin{pmatrix} 1 \\ 2 \end{pmatrix}; \quad k \in R$$

egyeneseket! Az adatokat értelmezzük először régi, majd pedig új koordinátaként!

d) Milyen egyenletek írják le a koordinátatengelyeket a két koordináta-rendszerben?

Megoldás

$$a) \quad \mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{e}'_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}; \quad \mathbf{e}'_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Az \mathbf{e}'_1 és \mathbf{e}'_2 egységvektorok bázist képeznek az R^2 -ben, mivel lineárisan függetlenek és a kétdimenziós tér minden eleme előáll két lineárisan független vektor lineáris kombinációjaként:

b) A transzformációs mátrix oszlopvektorai a bázisvektorok képei. Így:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{x}'$$

$$\mathbf{A}^{-1} = \frac{1}{3} \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}$$

Ellenőrzés:

$$\frac{1}{3} \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

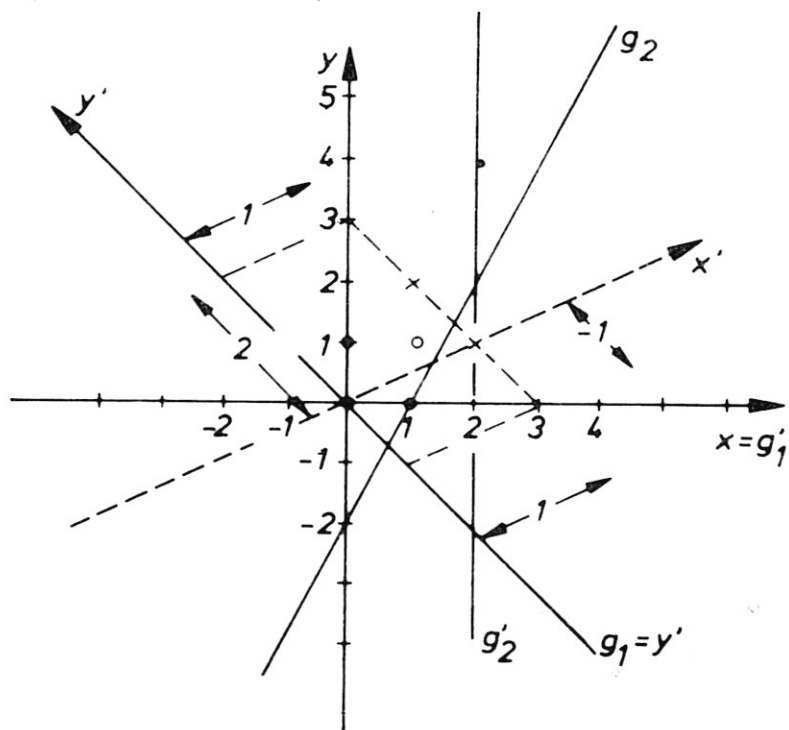
$$\mathbf{x}' = \mathbf{A}^{-1} \cdot \mathbf{x}$$

Igazak tehát a következő egyenletek:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}$$

$$\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

c) Koordináta-rendszernek válasszuk az S rendszert (\circ jelöli az ábrán a régi és \times az új koordinátákat).



$$g_1: \mathbf{x} = k \begin{pmatrix} 1 \\ -1 \end{pmatrix};$$

$$g_2: \mathbf{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + k \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Az új koordináták kiszámítása:

$$\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Az irányvektorok új koordinátái:

$$g_1: \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \Rightarrow g_1': \mathbf{x} = k \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$

$$g_2: \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} \Rightarrow g_2': \mathbf{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} + k \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

d) A koordinátatengelyek az S' rendszerben:

$$x: k \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad y: k \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$x': k \begin{pmatrix} 2 \\ 1 \end{pmatrix}; \quad y': k \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

A koordinátatengelyek az S' rendszerben:

$$\frac{1}{3} \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1/3 \\ -1/3 \end{pmatrix}$$

$$x: k \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\frac{1}{3} \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}$$

$$y: k \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Hasonlóan:

$$x': k \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad y': k \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

2. feladat

Koordináta-transzformációt végeztünk az R^3 -ban a következők szerint:

$$\mathbf{X}' = \mathbf{T} \cdot \mathbf{X} = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 0 \\ -1 & -1 & 1 \end{bmatrix} \cdot \mathbf{X}$$

a) Írjuk fel az inverz transzformációt leíró

$$\mathbf{X} = \mathbf{S} \cdot \mathbf{X}'$$

egyenletet!

b) Mutassuk meg, hogy az $x_1 + x_2 = 0$ sík a második rendszerben koordinátasík lesz és hogy a

$$g: \mathbf{x} = k \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

egyenes pedig a másik rendszer koordinátatengelyévé transzformálódik.

Megoldás

$$a) \quad S = T^{-1} = {}^t \begin{bmatrix} 0 & 1 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Ellenőrzés:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Tehát az inverz transzformáció egyenlete:

$$X = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot X'$$

$$b) \quad E: \quad x_1 + x_2 = 0$$

Meghatározzuk az S' rendszer koordinátatengelyeit az S rendszerben:

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \quad (*)$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Ahhoz, hogy az $(x_1; x_2)$, $(x_2; x_3)$, $(x_1; x_3)$ koordinátasíkok nemparaméteres egyenleteit megkaphassuk, kiszámítjuk e vektorok vektoriális szorzatait (kereszt-szorzatait).

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \times \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}; \quad \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

A koordináta-rendszer $(0; 0; 0)$ kezdőpontjának transzformáltja önmaga, így az mindhárom síknak pontja lesz.

$$E_1: \quad -x_1 - x_2 + x_3 = 0$$

$$E_2: \quad x_1 + x_2 = 0$$

$$E_3: \quad x_1 = 0$$

E_2 az adott sík. A

$$g: \mathbf{x} = k \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

egyenes nyilván éppen az x_1 -tengely, amint az a (*)-gal jelzett előbbi egyenletből látható.

3. feladat

Legyen az A mátrix

$$A = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}, \text{ ahol } \varphi \in R$$

a) Határozzuk meg A inverz mátrixát!

b) Mutassuk meg, hogy a

$$\mathbf{Y} = A \cdot \mathbf{X}$$

egyenlet az \mathbf{X} vektornak az O pont körüli, φ szöggel való elforgatását írja le (a φ szöget ívmértékben számítjuk!)

c) Határozzuk meg az

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} + k \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

egyenes egyenletét a $\varphi = \pi/6$, vagyis 30° -kal elforgatott S' rendszerben.

Megoldás

a) $\det A = \cos^2 \varphi + \sin^2 \varphi = 1$

$$A^{-1} = {}^t \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix}$$

Tehát fennáll az $A^{-1} = {}^t A$ egyenlőség. Azokat a mátrixokat, amelyek ezt a feltételt teljesítik, ortogonális mátrixoknak nevezzük.

b) $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 \cos \varphi - x_2 \sin \varphi \\ x_1 \sin \varphi + x_2 \cos \varphi \end{pmatrix}$$

Az \mathbf{X} és \mathbf{Y} vektorok közötti α szöget skaláris szorzás alapján határozzuk meg.

Fennáll a következő egyenlőség:

$$\cos \alpha = \frac{\mathbf{X} \cdot \mathbf{Y}}{|\mathbf{X}| \cdot |\mathbf{Y}|}$$

$$\cos \alpha = \frac{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \cos \varphi - x_2 \sin \varphi \\ x_1 \sin \varphi + x_2 \cos \varphi \end{pmatrix}}{|\mathbf{X}| \cdot |\mathbf{Y}|}$$

$$\Leftrightarrow \cos \alpha = \frac{x_1^2 \cos \varphi - x_1 x_2 \sin \varphi + x_1 x_2 \sin \varphi + x_2^2 \cos \varphi}{|\mathbf{X}| \cdot |\mathbf{Y}|}$$

$$\Leftrightarrow \cos \alpha = \frac{(x_1^2 + x_2^2) \cos \varphi}{|\mathbf{X}| \cdot |\mathbf{Y}|} = \frac{|\mathbf{X}|^2 \cos \varphi}{|\mathbf{X}| \cdot |\mathbf{Y}|} = \cos \varphi \frac{|\mathbf{X}|}{|\mathbf{Y}|}. \quad (*)$$

Az $|\mathbf{Y}|$ -ra viszont igaz:

$$\begin{aligned} |\mathbf{Y}|^2 &= (x_1 \cos \varphi - x_2 \sin \varphi)^2 + (x_1 \sin \varphi + x_2 \cos \varphi)^2 = \\ &= x_1^2 \cos^2 \varphi - 2x_1 x_2 \cos \varphi \sin \varphi + x_2^2 \sin^2 \varphi + \\ &+ x_1^2 \sin^2 \varphi + 2x_1 x_2 \cos \varphi \sin \varphi + x_2^2 \cos^2 \varphi = \\ &= x_1^2 (\cos^2 \varphi + \sin^2 \varphi) + x_2^2 (\cos^2 \varphi + \sin^2 \varphi) = x_1^2 + x_2^2. \end{aligned}$$

Így tehát írhatjuk:

$$\frac{|\mathbf{Y}|}{|\mathbf{X}|} = 1 \Rightarrow \cos \alpha = \cos \varphi \Rightarrow \alpha = \varphi,$$

mivel csak 180° -nál kisebb szögek jöhetnek szóba.

A (*) egyenlet tehát teljesült, amivel viszont azt mutattuk ki, hogy az \mathbf{A} mátrixszal leírt transzformáció hatása a φ szöggel való elforgatás.

c) Átszámítást végzünk az S rendszerből az S' rendszerbe. Érvényes a következő egyenlet:

$$\mathbf{X}' = \mathbf{A} \cdot \mathbf{X},$$

továbbá a következő egyenlőségek:

$$\cos 30^\circ = \frac{1}{2} \sqrt{3}; \quad \sin 30^\circ = \frac{1}{2}.$$

Fel tudjuk írni tehát az \mathbf{A}^{-1} inverz mátrixot:

$$\mathbf{A}^{-1} = \begin{bmatrix} 0,5 \sqrt{3} & 0,5 \\ -0,5 & 0,5 \sqrt{3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix}.$$

Így azt kapjuk, hogy:

$$\frac{1}{2} \begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{3}+1}{2} \\ \frac{\sqrt{3}-1}{2} \end{pmatrix}$$

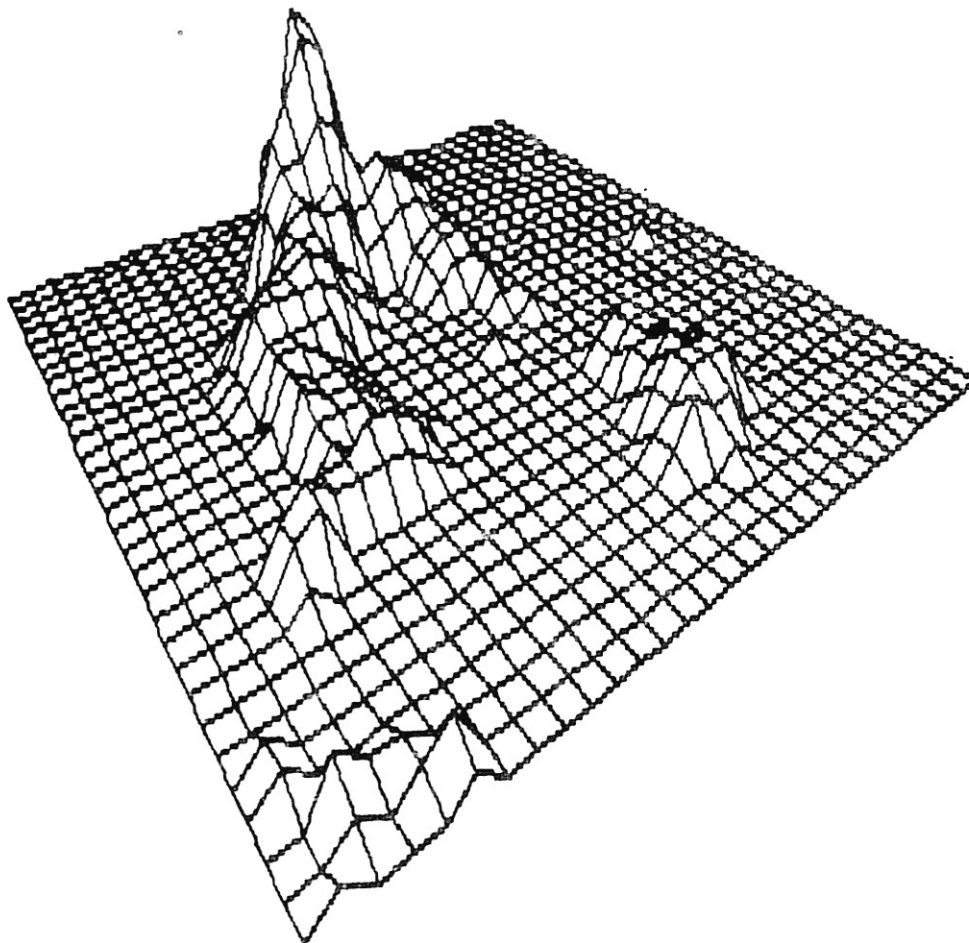
$$\frac{1}{2} \begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix} \cdot \begin{pmatrix} 1 \\ \sqrt{3} \end{pmatrix} = \begin{pmatrix} \sqrt{3} \\ 1 \end{pmatrix}$$

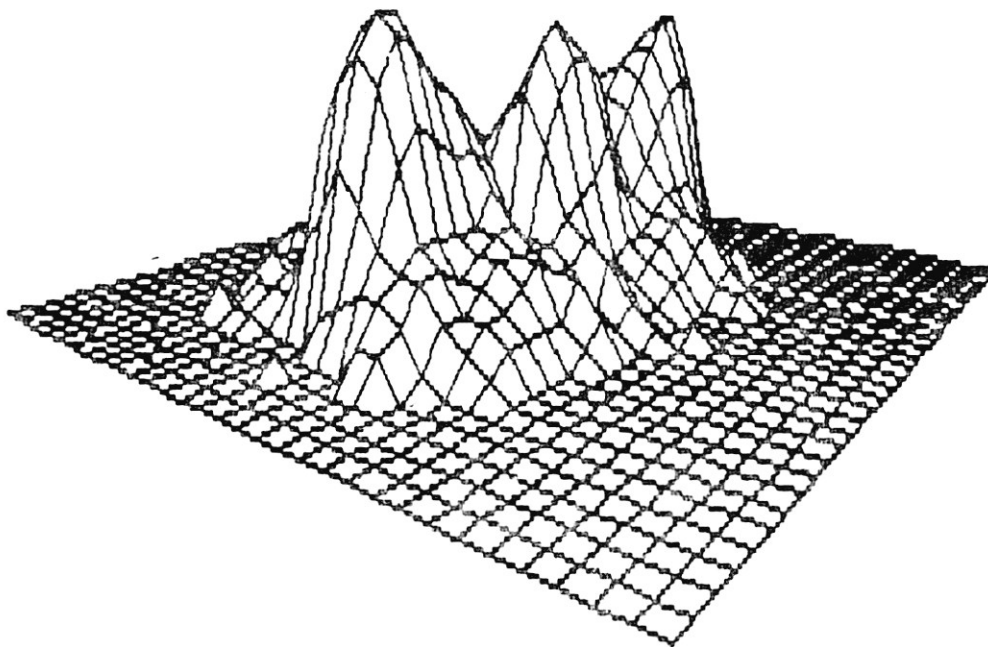
Az egyenes egyenlete tehát az S' rendszerben:

$$\mathbf{x} = \begin{pmatrix} \frac{\sqrt{3}+1}{2} \\ \frac{\sqrt{3}-1}{2} \end{pmatrix} + k \begin{pmatrix} \sqrt{3} \\ 1 \end{pmatrix}.$$

Ezzel megbeszéltük az összes alapvető összefüggést, amelyekre a grafikák összeállításához matematikai szempontból szükségünk van. A *3-test probléma* szimulációjának példáján majd látni fogjuk, hogy ezek az ismeretek egyéb szempontból is hasznosak.

3D-V2.7 (c) 1984. Objektum: PKS 0548-322





2. RÉSZ

A GRAFIKA PROGRAMOZÁSÁNAK GYAKORLATA

2.1 A grafika programozása MICROSOFT-BASIC 2.0-ban

2.1.1 A GRAFIKUS KÉPERNYŐK FELÉPÍTÉSE

A MICROSOFT-BASIC 2.0-ban három különböző képernyő van, az ún. *screen*-ek, amelyek különböző felhasználói célokat szolgálnak.

A screen 0 a standard képernyő. Ebben a képernyő-üzemmódban csak a szokásos íráskép használható, a grafikai utasítások nem alkalmazhatók.

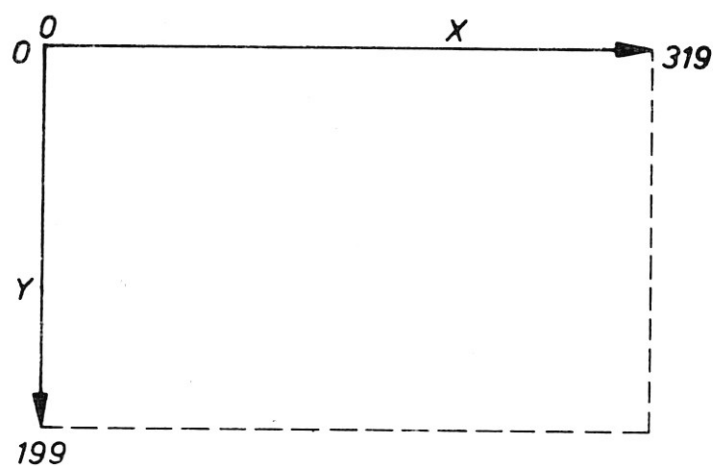
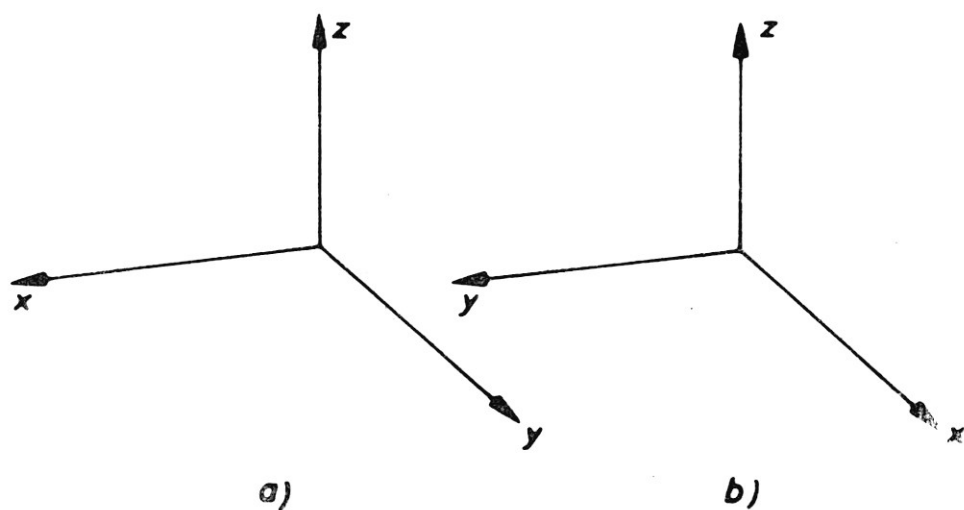
A screen 1 és a screen 2 grafikus képernyők, amelyek egymástól csak a színvariációs lehetőségben és a felbontásban különböznek.

Az első esetben a képernyő felbontása 320×320 pont, amelyek 4 színben jeleníthetők meg. Itt egy pont színének a tárolásához 2 bit-re van szükség. A két bit négyféle kombinációját (00, 01, 10, 11) használjuk fel a megfelelő szín kiválasztására.

A második esetben a képernyő felbontása 640×200 egyedi pont. Ekkor csak két színnel dolgozhatunk, 0 = fekete (a pont nincs kigyújtva) és 1 = fehér (a pont ki van gyújtva). Jóllehet a 2-es képernyő képelemeinek (pixeljeinek) száma kétszerese az 1-es képernyő képelemszámának, de ez nem a felbontás megkétszereződését jelenti, ui. csupán az x -irányú képfelbontás kétszereződött meg, de az y -irányú nem. Ennek a ténynek az ismerete fontos a grafikus ábrázolás szempontjából. Az 1-es képernyő-üzemmódban az x és az y koordinátatengelyek egyenrangúak, vagy más szóval két szomszédos képelem távolsága az x -tengelyen pontosan akkora, mint az y -tengelyen. Ezzel szemben a 2-es képernyő-üzemmódban a képelemek x -irányú távolsága csupán fele akkora, mint az y -irányú távolság. A 2-es képernyő-üzemmódban való grafikaprogramozás esetében ez azt jelenti, hogy amennyiben az Olvasó az egyforma hosszú egységeknek megfelelő x , y -értékekkel számol, úgy az x -értéket meg kell szoroznia 2-vel, ha el akarja kerülni, hogy az ábrázolás az x -irányban egy 0,5-es tényezővel torzuljon.

A következőkben látni fogjuk az x , y , z koordinátatengelyek gyakoribb megjelenéseit, amelyekben az x - és y -tengelyek fel vannak cserélve.

Az a) eset a matematikailag szokásos elrendezés, ezt a tisztán matematikai jellegű levezetések esetén fogjuk alkalmazni. Az 1-es és 2-es képernyő-üzemmódban (a rajzon az 1-es üzemmódban) az x - és y -tengelyek elhelyezkedése a b) esetnek felel meg.



Ezt a második ábrázolási formát mindig akkor választjuk majd, ha a képernyőgrafika speciális problémáiról lesz szó.

A két ábrázolási forma közötti különbség abban nyilvánul meg, hogy az $x - y$ síkot az előbbi esetben alulról, az utóbbiban felülről látjuk. A WINDOW (ablak) utasítás segítségével lehet a rálátást megváltoztatni.

2.1.2 GRAFIKUS UTASÍTÁSOK

A következőkben a grafika programozásának BASIC utasításait fogjuk ismertetni. (Az angolul vagy németül értő Olvasó figyelmébe ajánljuk az IBM PC BASIC 2.0 kézikönyvet. A megfelelő címszavaknál az utasítások részletes ismertetése megtalálható.)

2.1.2.1 A PSET és a PRESET utasítások

A PSET (Point SET) és a PRESET (Point RESET) utasítások arra valók, hogy a képernyő egy meghatározott helyén a pontot megjelenítsük vagy töröljük.

Formája: PSET (x, y), szín
PRESET (x, y), szín

A koordináták két módon adhatók meg: abszolút vagy relatív formában.

A PSET (x, y) a megfelelő képernyő-üzemmódban egy pontot jelenít meg az (x, y) pozícióban (abszolút forma).

A PSET STEP (d_x, d_y) megjelenít egy pontot az utoljára kigyújtott (vagy leoltott) ponthoz képest (d_x, d_y) pozícióban (relatív forma). A PRESET utasításban a koordinátákat hasonlóan adjuk meg.

Példa: PSET (100,100)

Kigyújtja a pontot a (100,100) pozícióban

PSET STEP (20,20)

Pontot ír ki a (100,100) pontra vonatkozó (20,20) pozícióban, tehát a (120,120) pozícióban.

Előfordulhat, hogy a megadott pontok a tényleges képernyőn kívül esnek. Ez mindaddig nem jár semmiféle következménnyel, ameddig a megadott adatok a -32768 -tól a 32768 -ig terjedő tartományban vannak. Ellenkező esetben megjelenik egy OVERFLOW üzenet.

2.1.2.2 A POINT függvény

A POINT függvénnyel lekérdezhető egy képelem színe.

Formája: $v = \text{POINT}(x, y)$

Az 1-es képernyő-üzemmódban a 0, 1, 2 és a 3 felel meg a különböző színeknek, míg a 2-es képernyő-üzemmódban csak a 0 és az 1. A POINT függvénnyel tehát ellenőrizhetjük, hogy egy pont ki van-e gyújtva vagy sem.

2.1.2.3 A LINE utasítás

A LINE utasítás számos lehetőséget rejt magában.

Formája: LINE $(x_1, y_1) - (x_2, y_2)$, szín, BF, mód

a) *Egyenes két pont között*

LINE $(x_1, y_1) - (x_2, y_2)$, szín

Lehetséges rövidítés: LINE $- (x_2, y_2)$, szín

Ekkor a kezdőpont automatikusan az utoljára megjelenített pont. A megjelenítés (ill. törlés) megvalósítható a PSET, a PRESET vagy a LINE utasítás valamelyikével. LINE esetében ez a pont a vonal végpontja.

Ha a színt nem adjuk meg, akkor az 1-es képernyő-üzemmódban automatikusan a 3 értékkel, a 2-es képernyő-üzemmódban pedig az 1 értékkel rajzol.

b) *Két pont közötti vonal törlése*

Analóg a)-val, de a szín = 0 (a 2-es képernyő-üzemmódban), ill. a szín = háttérszín (az 1-es képernyő-üzemmódban) választás szükséges.

c) *Téglalap*

LINE (x_1, y_1) – (x_2, y_2), szín, B

Egy téglalapot rajzol, ahol az (x_1, y_1) és az (x_2, y_2) a szemben fekvő csúcsok.

LINE (x_1, y_1) – (x_2, y_2), szín, BF

„Kitöltött” négyszöget rajzol.

d) *„Line styling”* (szaggatott vonal)

LINE (x_1, y_1) – (x_2, y_2), szín, BF, mód

A mód egy 16-bites kifejezés, azaz a 0 és a 65535 értékek közé eső egész szám. Az értéket decimális vagy hexadecimális (&H...) formában adhatjuk meg.

A mód bitminta szerint szaggatottan rajzolja a vonalat.

Példa:

1, 0, 0, 1/1, 1, 1, 1/1, 0, 0, 1/, 1, 1, 1, 1
 9 F 9 F

Ez a &H9F9F hexadecimális, vagy a
 $9 \cdot 4096 + 15 \cdot 256 + 9 \cdot 16 + 15 \cdot 1 = 40863$
 decimális szám.

A képernyőn megjelenő vonal képe:

—00—00—00—

(0 hiányzó pont, — megjelenített pont).

2.1.2.4 A PAINT utasítás

A PAINT utasítással kitölthetjük a grafikus alakzatot egy meghatározott színnel, vagy egy meghatározott rasztermintával.

Formája: PAINT (x, y), szín (ill. minta)

A PAINT utasítás az alakzatot a megadott színnel (mintával) mindaddig festi, amíg ugyanazt az alapszint találja. Ügyelni kell arra, hogy az (x, y) az alakzaton belül legyen, valamint, hogy az alakzaton ne legyen rés, mert egyébként az utasítás az egész képernyőt befesti. Ha az (x, y) pont már korábban az adott színű volt, úgy a PAINT hatástalan. A szín helyett mintát is meg lehet adni, egy maximálisan 8×8 -as maszkkal. A minta a *line styling*-hez hasonló módon adható meg.

Sor	Byte-minta	Byte-tartalom	
		hexadecimálisan	decimálisan
1	1000 1000	88	136
2	1100 1100	CC	204
3	1110 1110	EE	234
4	1111 1111	FF	255
5	1000 1000	88	136
6	1100 1100	CC	204
7	1110 1110	EE	234
8	1111 1111	FF	255

Ezekből az értékekből egy jelsorozatot (*string* kifejezést) írunk fel:

```
A$ = CHR$ (&H88) + CHR$ (&HCC) + CHR$ (&HEE) + CHR$ (&HFF)
```

```
A$ = A$ + A$
```

vagy

```
A$ = CHR$ (136) + CHR$ (204) + CHR$ (234) + CHR$ (255)
```

```
A$ A$ + A$
```

(Az 5...8 byte értékek megadása tulajdonképpen felesleges, mert azok az 1...4 értékekkel azonosak. Egyébként minden, legfeljebb 8 byte-tal megadott kifejezés megengedett.)

A PAINT utasítás akkor

```
PAINT (x, y), A$
```

alakú.

2.1.2.5 A GET utasítás

A GET utasítással a grafikus ábrát a képernyőről egy tömbbe olvashatjuk.

Formája: GET (x_1, y_1) – (x_2, y_2), tömbnév

Az (x_1, y_1) és (x_2, y_2) a tömbbe olvasandó képet tartalmazó téglalap egymással szemben fekvő csúcsainak koordinátái. A tömbnév egy megfelelő méretű tömbváltozó. A szükséges méret a következőképpen számítható ki (2-es képernyő-üzemmód):

$$4 + \text{INT} ((d_x + 7)/8) * d_y$$

Tehát pl. egy 10×12 -es nagyságú kép esetében a tömb elemeinek száma legalább

$$4 + \text{INT} ((10 + 7)/8 * 12) = 28$$

kell legyen.

A következő sorrendben kell tehát eljárni:

```
DIM A (28)
```

```
GET (x, y) – (x + 10, y + 12), A
```

2.1.2.6 A PUT utasítás

A PUT a GET utasítás fordítottja, így a grafikus ábrát kivihetjük egy tömbből a képernyőre.

Formája: PUT (x, y), tömbnév, tevékenység

Az (x, y) az átviendő ábra bal felső sarka.

A tevékenység lehet:

PSET

átvitel a képernyőre;

PRESET

invertált átvitel a képernyőre;

AND

átvitel csak akkor van, ha az átviendő ábra alatt már létezik kép;

OR

az ábrát, amit átvisz, a képernyőn már meglévő ábrára helyezi;

XOR

az ábrát *exclusive or*-ral viszi át, azaz megjeleníti a pontot, ha az még nem volt kigyűjtva, ill. törli, ha ki volt gyűjtva.

2.1.2.7 A BSAVE és a BLOAD utasítások

A BSAVE és a BLOAD utasítások eredetileg a gépi kódú programok mentésére, ill. betöltésére valók, de velük a képernyőtár lemezen is tárolható.

```
DEF SEG = &HB800
BSAVE „egység: név.kód”, 0, 16200
```

A visszatöltés formája:

```
BLOAD „egység: név.kód”
```

Fontos, hogy BLOAD utasítás végrehajtása előtt ugyanaz a képernyő-üzemmód legyen bekapcsolva, mint ami a megfelelő BSAVE utasítás előtt volt. Így pl. a nagyfelbontású képek visszatöltése előtt a 2-es képernyő-üzemmódot kell beállítani.

2.1.2.8 A WINDOW utasítás

A WINDOW (ablak) utasítás a továbbiakban szereplő minden koordinátát fizikai koordinátaként definiál.

Formája: WINDOW $(x_1, y_1) - (x_2, y_2)$

A WINDOW utasítás segítségével a felhasználó által definiált koordináta-rendszer használható. Pl. az Olvasó meg akarja jeleníteni a képernyőn a $(-10, -10) - (10, 10)$ tartományt. A WINDOW utasítással minden koordinátát ebben a rendszerben adhatunk meg. A $(0, 199)$ pozíció a megadott legkisebb x - és y -értékekhez lesz hozzárendelve, ami példánkban a $(-10, -10)$; a $(319, 0)$, ill. a $(639, 0)$ pozíciók pedig a legnagyobbakhoz, esetünkben a $(10, 10)$ -hez. A pont hozzárendelése tehát megegyeszer megtörténik, azaz $(-10, -10) - (10, 10)$ és a $(10, -10) - (-10, 10)$ tartománymegadási formák egymással ekvivalensek. Minden koordinátaértéket az utasítás átalakít képernyő-koordinátává, és az egész képernyőt a definiált tartománynak tekinti. Másképp fogalmazva: a WINDOW utasítással a felhasználónak a feladathoz illeszkedő koordináta-rendszere automatikusan a képernyő „természetes” koordináta-rendszerébe transzformálódik.

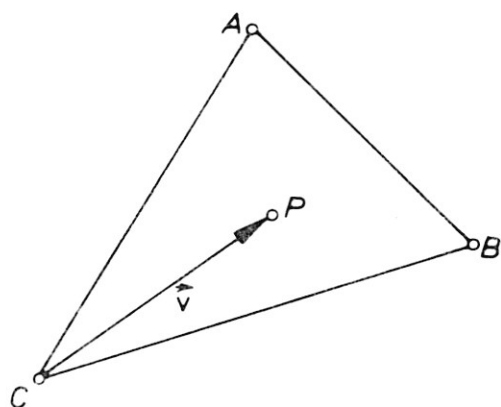
2.2 Geometriai transzformációk az R^2 -ben és az R^3 -ban

A következőkben az R^2 és az R^3 -beli geometriai transzformációkkal (eltolás, tengelyes tükrözés, középpontos tükrözés, forgatás) foglalkozunk. Lehetőség szerint az R^2 - és az R^3 -beli eseteket is megtárgyaljuk, mivel a közöttük levő különbségek nem túl nagyok. Számunkra azonban elsősorban csak a kétdimenziós műveletek érdekesek.

2.2.1 ELTOLÁS

Amint korábban már láttuk, egy véges sok helyvektorral meghatározott alakzatot az eltolási vektor segítségével bármely tetszőleges helyre el tudunk tolni. Mindezt egy R^2 -beli háromszög példáján fogjuk bemutatni. (A megfelelő művelet végrehajtása az R^3 -ban ezzel analóg.)

Adott az ABC háromszög, valamint a P pont.



A háromszöget úgy akarjuk eltolni, hogy a C csúcs a P pontba kerüljön, azaz $C' = P$. A v eltolási vektort ekkor a következőképpen tudjuk meghatározni:

$$\begin{aligned} \mathbf{C} + \mathbf{v} &= \mathbf{C}' \\ \Leftrightarrow \mathbf{v} &= \mathbf{C}' - \mathbf{C} = \mathbf{P} - \mathbf{C} \end{aligned}$$

Az új $A' B' C'$ háromszöget úgy kapjuk meg, hogy a v eltolási vektort hozzáadjuk az A , B , C helyvektorához:

$$\begin{aligned} \mathbf{A}' &= \mathbf{A} + \mathbf{v} = \mathbf{A} + \mathbf{P} - \mathbf{C} \\ \mathbf{B}' &= \mathbf{B} + \mathbf{v} = \mathbf{B} + \mathbf{P} - \mathbf{C} \\ \mathbf{C}' &= \mathbf{P} \end{aligned}$$

A számítógéppel is hasonlóan számíttatjuk ki az eltolt pont koordinátáit.

Legyen pl.:

$$\mathbf{A} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad \text{és} \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} p_1 - c_1 \\ p_2 - c_2 \end{pmatrix}$$

Az új pozícióra a következőt kapjuk:

$$\mathbf{A}' = \mathbf{A} + \mathbf{v} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} a_1 + v_1 \\ a_2 + v_2 \end{pmatrix} = \begin{pmatrix} a'_1 \\ a'_2 \end{pmatrix}$$

Ezzel a programrészlet:

$$\begin{aligned} A1' &= A1 + P1 - C1 \\ A2' &= A2 + P2 - C2 \end{aligned}$$

(A BASIC programban természetesen az $A1'$, $A2'$ helyett más azonosítót, pl. $A1V$ -t és $A2V$ -t kell használnunk, az aposztróf változónévben nem fordulhat elő.)

2.2.2 TÜKRÖZÉSEK

A műveletek itt is analógok az R^2 -ben és az R^3 -ban. Mindenesetre ne tévesszük szem elől, hogy az R^2 -ben csak középpontos és tengelyes tükrözést különböztetünk meg.

Az analóg műveletek:

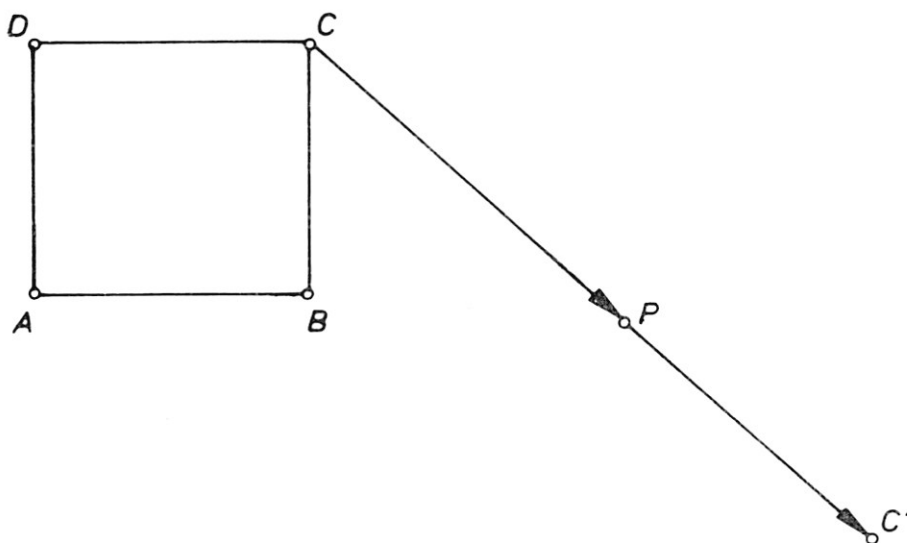
R^2
középpontos tükrözés
tengelyes tükrözés

R^3
középpontos tükrözés
síkra vonatkozó tükrözés

Az R^3 -ban végzett tengelyes tükrözés az R^2 -beli tengelyes és középpontos tükrözésből összetevődő transzformáció, közelebbről nem foglalkozunk vele.

2.2.2.1 Középpontos tükrözés

Példaként tekintsük az R^2 -ben az $ABCD$ négyszöget, amelyet a P pontra kell tükröznünk.



Az új koordinátákat, pl. a C pont koordinátáit a következő módon kapjuk meg: a C és a P pontok meghatároznak egy egyenest. A $\overrightarrow{CP} = \mathbf{P} - \mathbf{C}$ vektor irányvektora az egyenesnek, a P pont pedig az egyenes egy pontja. A C' pontot az jellemzi, hogy a C és a P pontokon áthaladó egyenesen fekszik és a P ponttól mért távolsága egyenlő a P és a C pontok távolságával.

Így a C' pontot nagyon egyszerűen megkapjuk a

$$\mathbf{C}' = \mathbf{P} + \overrightarrow{CP} = \mathbf{P} + \mathbf{P} - \mathbf{C} = 2\mathbf{P} - \mathbf{C}$$

formulával.

Koordinátákkal:

$$\mathbf{C}' = \begin{pmatrix} c'_1 \\ c'_2 \end{pmatrix} = \begin{pmatrix} 2p_1 - c_1 \\ 2p_2 - c_2 \end{pmatrix}.$$

Általánosan tehát:

$$\mathbf{A}' = 2\mathbf{P} - \mathbf{A}$$

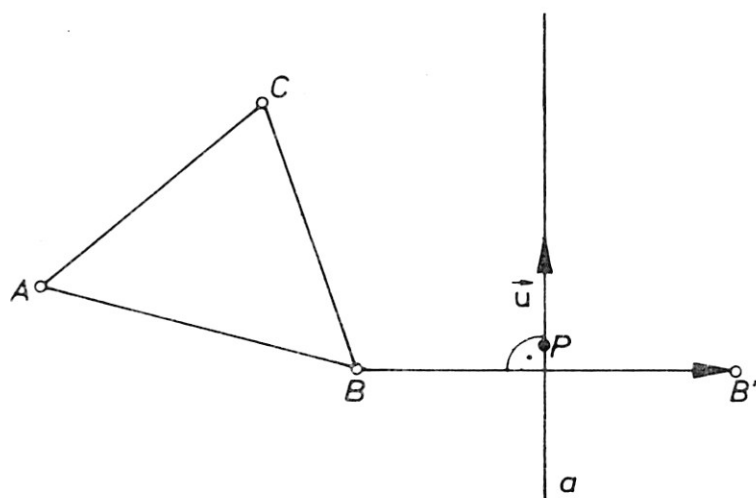
$$\mathbf{B}' = 2\mathbf{P} - \mathbf{B}$$

$$\mathbf{C}' = 2\mathbf{P} - \mathbf{C}$$

$$\mathbf{D}' = 2\mathbf{P} - \mathbf{D}$$

2.2.2.2 Tengelyes tükrözés az R^2 -ben

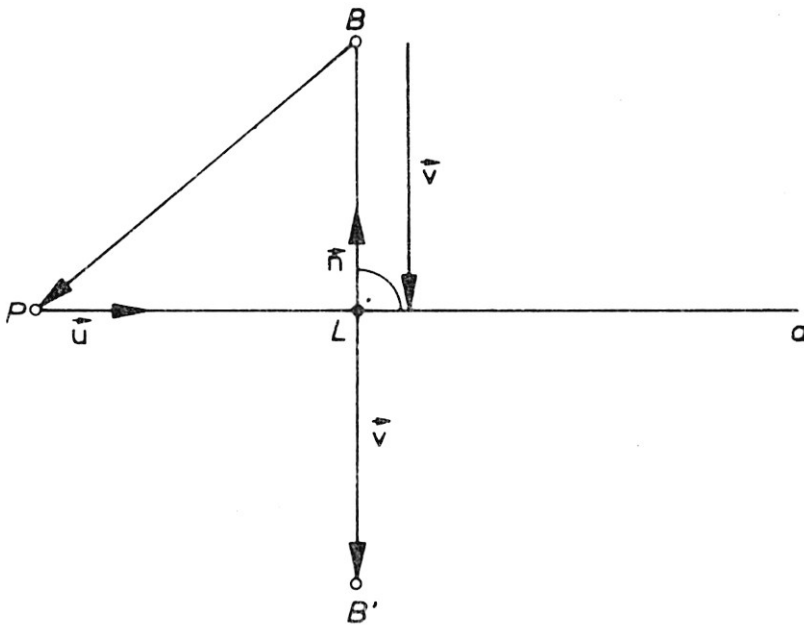
Tekintsük az ABC háromszöget:



A tükrözés tengelye legyen az

$$a: \mathbf{X} = \mathbf{P} + k\mathbf{u}$$

egyenes. Pl. a B pont új koordinátáinak, azaz a B' pont koordinátáinak a megállapítására a következő módon járunk el.



Ha az \mathbf{u} irányvektor koordinátái

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

akkor egy \mathbf{u} -ra merőleges \mathbf{n} normálvektor az

$$\mathbf{n} = \begin{pmatrix} -u_2 \\ u_1 \end{pmatrix},$$

mert igaz az

$$\mathbf{n} \cdot \mathbf{u} = 0$$

egyenlőség.

Ezen kívül az sem okoz nehézséget, hogy a

$$\overrightarrow{BP} = \mathbf{P} - \mathbf{B}$$

vektor meghatározzuk:

$$\overrightarrow{BP} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} p_1 - b_1 \\ p_2 - b_2 \end{pmatrix}.$$

A \overrightarrow{BP} vektornak az \mathbf{n} vektorra való vetítésével a \mathbf{v} vektort kapjuk, amelyre igaz a következő egyenlet:

$$\mathbf{v} = \overrightarrow{BL} = \mathbf{L} - \mathbf{B}$$

A \mathbf{B}' -re vonatkoztatva:

$$\mathbf{B}' = \mathbf{B} + 2\overrightarrow{BL} = \mathbf{B} + 2\mathbf{v}$$

A \mathbf{v} vektor könnyen meghatározható a vetítési képlettel (vö. *Matematikai alapok*). Írható tehát:

$$\mathbf{v} = \frac{\overrightarrow{BP} \cdot \mathbf{n}}{|\mathbf{n}|^2} \mathbf{n} = \frac{\begin{pmatrix} p_1 - b_1 \\ p_2 - b_2 \end{pmatrix} \cdot \begin{pmatrix} -u_2 \\ u_1 \end{pmatrix}}{u_1^2 + u_2^2} \begin{pmatrix} -u_2 \\ u_1 \end{pmatrix}$$

Ebből kapjuk, hogy:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{(p_1 - b_1)(-u_2) + (p_2 - b_2)(u_1)}{u_1^2 + u_2^2} \begin{pmatrix} -u_2 \\ u_1 \end{pmatrix}.$$

Ezek szerint \mathbf{B}' -re érvényes a következő összefüggés:

$$\mathbf{B}' = \mathbf{B} + 2\mathbf{v} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \frac{2l_2u_1 + 2l_1u_2}{u_1^2 + u_2^2} \begin{pmatrix} -u_2 \\ u_1 \end{pmatrix},$$

ahol

$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} = \begin{pmatrix} p_1 - b_1 \\ p_2 - b_2 \end{pmatrix}.$$

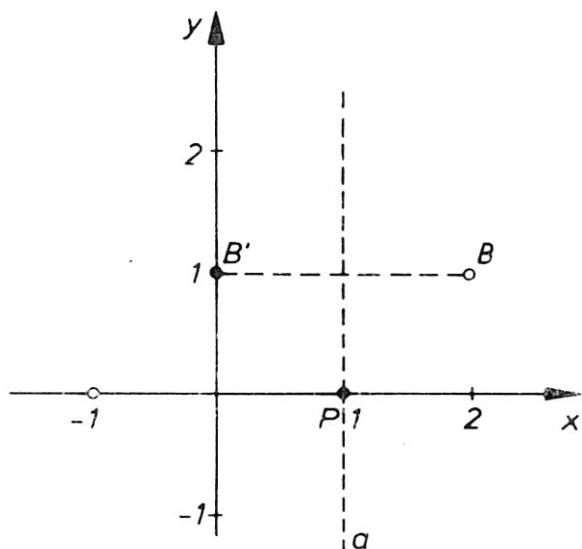
Nézzünk egy példát:

Legyen

$$\mathbf{B} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

és

$$a: \mathbf{X} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + k \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$



Tudjuk, hogy \mathbf{B}' koordinátájának $(0, 1)$ -nek kell lenni. Ellenőrizzük ezt számolással is:

$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} = \begin{pmatrix} p_1 - b_1 \\ p_2 - b_2 \end{pmatrix} = \begin{pmatrix} 1 - 2 \\ 0 - 1 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \frac{2(-1)0 - 2(-1)1}{1} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \begin{pmatrix} -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Szeretnék ezen a helyen a grafikus programozás problémájához egy rövid, általános természetű megjegyzést fűzni. A bonyolult képletek előfordulása szükségessé teszi, hogy ezek helyességét ellenőrizzük, mert felírásuknál, ill. a program kódolásakor hibákat követhetünk el (számolási, gépelési, beolvasási, szintaktikus hibát). Ezért általában kívánatos, hogy a felállított képletet, ill. a megfelelő programszakaszt a teljes program indítása előtt teszteljük egy olyan triviális példával, amelynek a megoldása ismert. Mindig egyszerűbb egy 20 soros, áttekinthető programszakaszban a hibát megkeresni, mint kiküszöbölni ugyanazt a hibát egy 2000 sort tartalmazó programból.

2.2.2.3 Síkra vonatkozó tükrözés az R^3 -ban

Ezt a rövid szakaszt a matematikai érdeklődésű Olvasóknak szánjuk, jelentősége a grafikák feldolgozásában gyakorlatilag nincs.

Adott a B pont, valamint az E sík.

Ha az E síkot paraméteres alakban írjuk fel, akkor normálvektorát az

$$\mathbf{u} \times \mathbf{v} = \mathbf{n}$$

képlettel határozhatjuk meg. Ismert normálvektor esetében azonban ugyanaz a képlet érvényes, mint az R^2 -ben.

$$\mathbf{B}' = \mathbf{B} + 2 \frac{\mathbf{l} \cdot \mathbf{n}}{|\mathbf{n}|^2} \mathbf{n},$$

vagy koordinátákkal:

$$\mathbf{B} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \frac{2(l_1 n_1 + l_2 n_2 + l_3 n_3)}{n_1^2 + n_2^2 + n_3^2} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}.$$

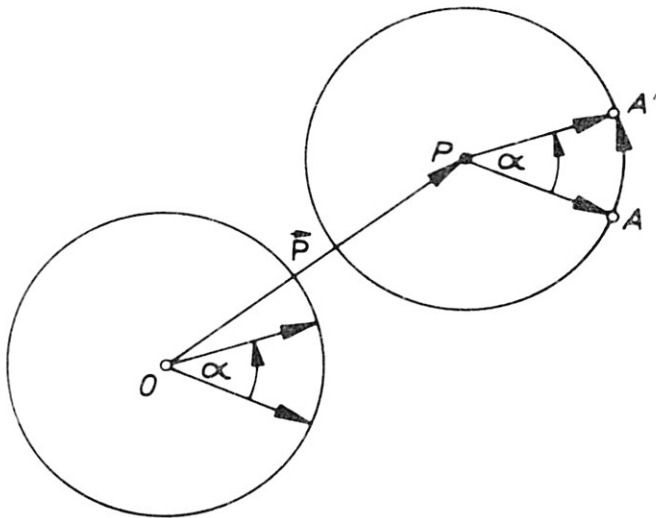
2.2.3 FORGATÁS

A forgatás leírható az ún. rotációs mátrix (a forgatás mátrixa) segítségével. A mátrixképletek levezetésére a rotációs mátrixok fejezetében még alaposabban kitérünk. Ezért itt most csak azt beszéljük meg, hogy hogyan kell egy kétdimenziós rotációs mátrixot kezelni.

Mint már láttuk, a

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

mátrix az \mathbf{A} helyvektort a koordináta-rendszer kezdőpontja körül α szöggel elforgatja. Ha egy A pontot egy tetszőleges P pont körül α szöggel el akarunk forgatni, akkor a következőképpen járunk el.



Meghatározzuk a \overrightarrow{PA} vektort. Az $\mathbf{R}(\alpha)$ rotációs mátrix segítségével elforgatjuk a \overrightarrow{PA} vektort α szöggel és ezáltal megkapjuk a $\overrightarrow{PA'}$ vektort. A \mathbf{P} és a $\overrightarrow{PA'}$ összeadásával az $\mathbf{A'}$ vektort nyerjük.

Tehát tetszőleges A és P körüli α szöggel való A' elforgatottjára:

$$\mathbf{A'} = \mathbf{P} + \mathbf{R}(\alpha) \cdot (\mathbf{A} - \mathbf{P}) = \mathbf{P} + \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot (\mathbf{A} - \mathbf{P}).$$

2.3 Alakzatok grafikus megjelenítése

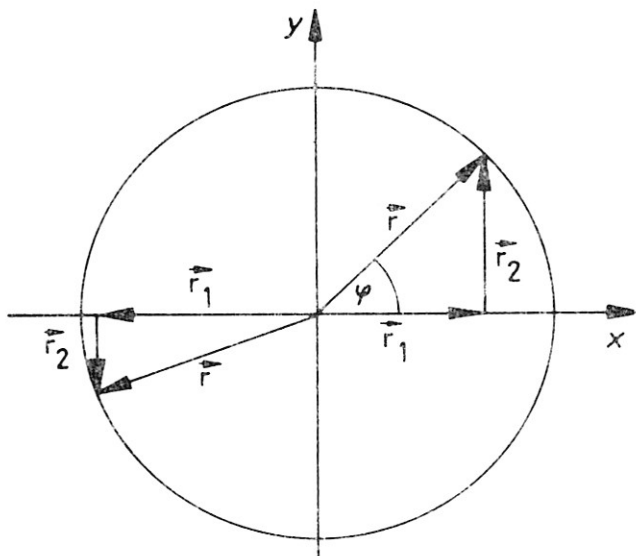
Eddig csak azon töprengtünk, hogy miképp tudunk egyenes szakaszokkal határolt alakzatokat, pl. egyeneseket, háromszögeket stb. vektorokkal ábrázolni. Ez a megközelítési mód azonban körök, ellipszisek, spirálok, gömbök esetében ugyanúgy lehetséges. Mint majd látni fogjuk, elméletileg mindig kétféle ábrázolási módunk van: egy analitikus és egy vektoriális, az *alakzat ún. paraméteres alakja*.

2.3.1 KÖRÖK

Bizonyára sokan ismerik a kör

$$x^2 + y^2 = r^2$$

egyenletét, Nem nehéz felismerni, hogy ez vagy az ebből levezetett összefüggések nehézkesen programozhatók. Egy egyszerűbb előállítás érdekében tekintsük a következő rajzot:



Nyilvánvaló, hogy az \mathbf{r} sugárvektor, amely a kör egy pontjára mutat, mindig felbontható két, egymásra merőleges \mathbf{r}_1 és \mathbf{r}_2 komponensre.

Az \mathbf{r}_1 és \mathbf{r}_2 vektorokra érvényes a következő összefüggés:

$$\mathbf{r}_1 = |\mathbf{r}| \cos \varphi \mathbf{x}^\circ$$

$$\mathbf{r}_2 = |\mathbf{r}| \sin \varphi \mathbf{y}^\circ$$

Ezzel tehát lehetőségünk nyílt arra, hogy a kört nagyon egyszerűen programozhassuk.

Láttuk, hogy az $|\mathbf{r}| \cos \varphi$ az \mathbf{r} x -irányú komponensének, az $|\mathbf{r}| \sin \varphi$ pedig az y -irányú komponensének felel meg, és így az

$$\mathbf{r} = \begin{pmatrix} |\mathbf{r}| \cos \varphi \\ |\mathbf{r}| \sin \varphi \end{pmatrix}$$

egyenlőség igaz.

Ez a kör egyenletének paraméteres alakja.

A paraméter változtatjuk úgy, hogy 0-tól 2π -ig minden értéket felvegyen és mindig csak a sugárvektor végpontját jelenítjük meg. Így egy kört kapunk. Az ennek megfelelő program tehát:

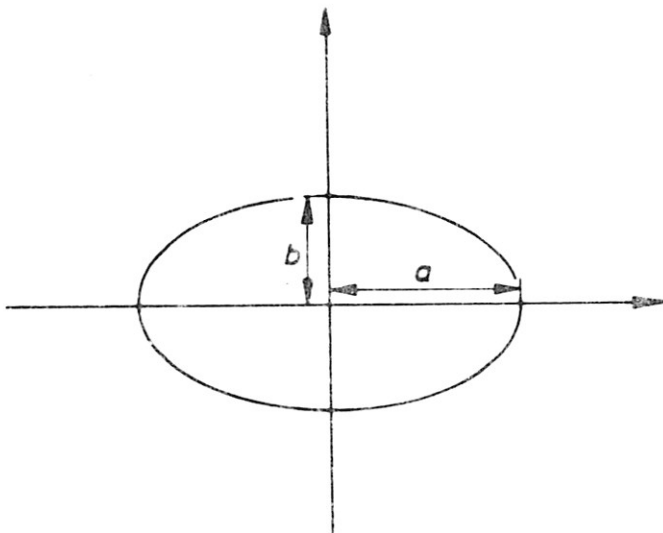
```
CLS: SCREEN 1
R = 10: MX = 160: MY = 100: PI = 3.14159
FOR I = 1 TO 2*PI STEP .1
X = MX + R*COS(I)
Y = MY + R*SIN(I)
PSET X, Y
NEXT
```

A programban MX és MY a kör középpontjának koordinátái. Segítségükkel körünk középpontját és ezzel az egész kört el tudjuk tolni a koordináta-rendszer középpontjától a képernyő tetszőleges pontjába, mert érvényes a kör általános egyenlete:

$$\mathbf{X} = \mathbf{M} + \mathbf{R}$$

2.3.2 ELLIPSZISEK

A kör az ellipszis egy speciális esete. Az ellipszisek két tengelyét különböztetjük meg. A rajzon a a nagy féltengely és b a kis féltengely.



Egy ellipszis létrejöttét úgy képzeljük el, hogy gondolatban egy kör alakú lapot a térben szemmagasságban valamilyen szöggel elfordítunk.

Az ellipszist ezért matematikailag a kör affin képének is nevezzük.*

* Az affinitásnak mint transzformációnak a fenti szemléltetés helyett inkább egy gumilapra rajzolt kör adott irányú „nyújtása” felel meg. (Ford.)

A körhöz hasonlóan, az ellipszis paraméteres meghatározására az

$$\mathbf{r} = \begin{pmatrix} a \sin \varphi \\ b \cos \varphi \end{pmatrix}$$

alakot kapjuk.

Amíg a körnél mindegy, hogy derékszögű koordináta-rendszerünk elhelyezkedése milyen, mivel mindig ugyanazt a kört kapjuk, addig az ellipszis esetében a koordináta-rendszer elforgatása megváltoztatja az ellipszis helyzetét. Erre mégegyszer majd az ellipszis vektoregyenleténél visszatérünk.

Érvényes a következő egyenlőség:

$$\mathbf{r} = a \sin \varphi \mathbf{e}_1 + b \cos \varphi \mathbf{e}_2,$$

ahol \mathbf{e}_1 és \mathbf{e}_2 az $x-y$ koordináta-rendszerünk egységvektorai. Az \mathbf{e}_1 és \mathbf{e}_2 egységvektorokat helyettesíthetjük olyan két egymásra merőleges \mathbf{e}'_1 és \mathbf{e}'_2 egységvektorral, amelyek helyzete az eredeti koordináta-rendszerünkben tetszőleges lehet. Ezáltal lehetővé vált ezen vektorok koordinátaikkal való megadása.

Legyen

$$\mathbf{e}'_1 = \begin{pmatrix} a_x \\ a_y \end{pmatrix},$$

akkor egy erre merőleges egységvektort az

$$\mathbf{e}'_2 = \begin{pmatrix} -a_y \\ a_x \end{pmatrix} = \begin{pmatrix} b_x \\ b_y \end{pmatrix}$$

egyenlőséggel adunk meg.

Ha behelyettesítünk az ellipszis egyenletébe, akkor ezt kapjuk, hogy:

$$\mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = a \sin \varphi \begin{pmatrix} a_x \\ a_y \end{pmatrix} + b \cos \varphi \begin{pmatrix} b_x \\ b_y \end{pmatrix}.$$

Így a két komponens:

$$\begin{aligned} r_1 &= a \sin \varphi a_x + b \cos \varphi b_x; \\ r_2 &= a \sin \varphi a_y + b \cos \varphi b_y. \end{aligned}$$

Ez az ellipszis egyenletének általános alakja.

Koordináta-transzformációként felírva:

$$\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{bmatrix} a_x & b_x \\ a_y & b_y \end{bmatrix} \cdot \begin{pmatrix} a \sin \varphi \\ b \cos \varphi \end{pmatrix}.$$

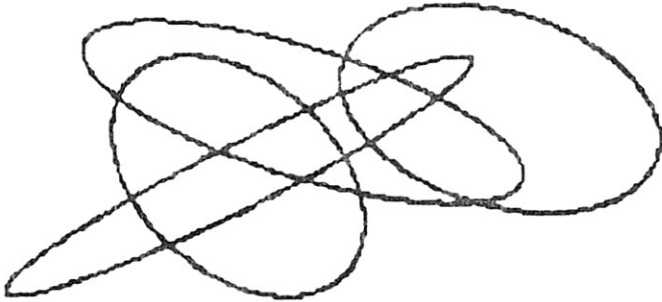
Itt az ellipszis középpontja a $(0; 0)$ pontban van.

Ehhez az

$$\mathbf{M} = \begin{pmatrix} m_x \\ m_y \end{pmatrix}$$

helyvektort hozzáadva, az ellipszis középpontjának az M pontba való eltolását kapjuk (l. a programban).

Az e_1 és e_2 definiálásánál nem szükséges tekintettel lenni arra, hogy itt egységvektorokról van szó, a program ui. a 160-astól a 180-asig terjedő sorokban a vektorokat önállóan átalakítja egységvektorokká.



FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENÜL** el kell olvasni!

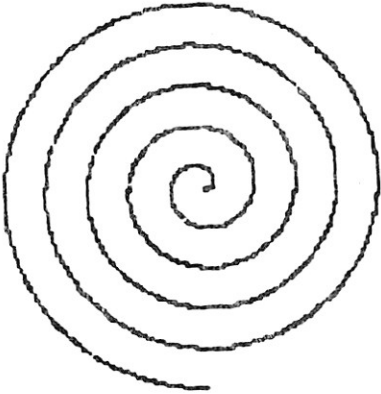
```
50 REM ... P1 ...
100 REM ..... Ellipszisek
110 CLS:SCREEN 2:KEY OFF:PI = 3.16
120 MX = 160:MY = 100
130 A = 70:B = 20
140 AX = 2:AY = 1
150 BX = -AY:BY = AX
160 Z = SQR(AX^2+AY^2)
170 AX = A*AX/Z:AY = A*AY/Z
180 BX = B*BX/Z:BY = B*BY/Z
190 PSET (2*(MX+BX),MY+BY)
200 FOR I = 0 TO 2*PI STEP .03
210 X = MX+SIN(I)*AX+COS(I)*BX
220 Y = MY+SIN(I)*AY+COS(I)*BY
230 LINE -(2*X,Y)
240 NEXT
250 END
```

2.3.3 SPIRÁLOK

A spirálokat a körből is származtathatjuk: a kör sugarát a φ -vel arányosan nagyítjuk, ill. kicsinyítjük, pl.

$$r = \begin{pmatrix} \varphi \cos \varphi \\ \varphi \sin \varphi \end{pmatrix}.$$

Az x - vagy az y -irányú komponens előjelének megváltoztatásával változik a spirál forgásiránya, jobbra csavarodóról átvált balra csavarodóra, és fordítva.

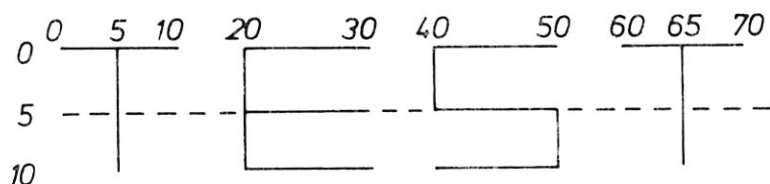


FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENŰL** el kell olvasni!

```
50 REM ... P2 ...
100 REM ..... Spirálok
110 CLS:SCREEN 2:KEY OFF:PI = 3.1419
120 FOR I = 0 TO 10*PI STEP .05
130 X = 320+4*I*SIN(I)
140 Y = 100+2*I*COS(I)
150 LINE -(X,Y)
160 NEXT
```

2.4 Ferde írás

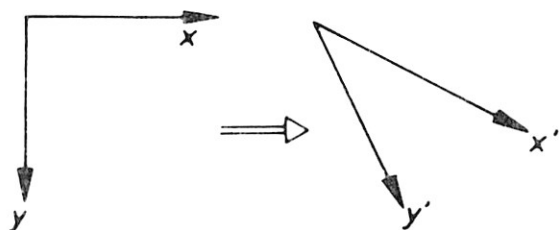
A különböző bázisvektorok R^2 -ben való alkalmazásának különösen szép példája a ferde írás. A TEST szót az élek végpontjainak koordinátaival definiáljuk (vö. 240—270-es sorok).



A vonatkoztatási rendszerünk a képernyő standard koordináta-rendszere.

Vezessük be az \mathbf{e}'_1 és \mathbf{e}'_2 vektorokat mint új bázisvektorokat, így az R^2 egy új koordináta-rendszerét kapjuk.

$$\mathbf{e}'_1 = \begin{pmatrix} a_x \\ a_y \end{pmatrix}; \quad \mathbf{e}'_2 = \begin{pmatrix} b_x \\ b_y \end{pmatrix}.$$



Ha most betűink koordinátáit mint új koordinátákat ($x_L; y_L$) értelmezzük, akkor az

$$\begin{pmatrix} x \\ y \end{pmatrix} = x_L \begin{pmatrix} a_x \\ a_y \end{pmatrix} = y_L \begin{pmatrix} b_x \\ b_y \end{pmatrix}$$

egyenlettel megkapjuk a betűk régi rendszerbeli koordinátáit.

Ez megfelel az

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a_x & b_x \\ a_y & b_y \end{bmatrix} \cdot \begin{pmatrix} x_L \\ y_L \end{pmatrix}$$

mátrixegyenletnek (vö. 170—180-as sorok), hiszen egy koordináta-transzformációról van szó.

A program valójában nem különösen bonyolult. Ha az \mathbf{e}'_1 és \mathbf{e}'_2 vektorokat megfelelően választjuk meg, akkor a TEST szót tetszőleges helyzetben, tetszőleges torzítással írhatjuk fel (pl. jobbról balra, feje tetején állva stb.).

A kezdőpontok, azaz a T betű bal felső sarkát az

$$S = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$$

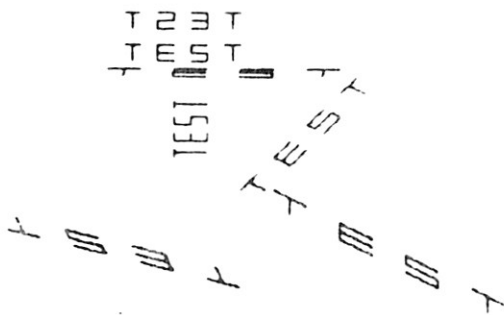
vektor eltolja az S pozícióba.

Javasoljuk az Olvasónak, hogy próbálja ki a megfelelő hatásokat azzal, hogy a bázisvektoroknak más-más értékeket ad meg a 140-es és a 150-es sorokban.

Programleírás

Sor	Magyarázat
130—150	A változók értékének beállítása
160—190	A koordináták beolvasása és transzformáció
200—230	A transzformált betűk rajzolása
240—270	A koordináták DTA-sor formában

FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENÜL** el kell olvasni!



```

50 REM ... P3 ...
100 REM ..... Ferde írás
110 CLS: SCREEN 2
120 DIM X(30),Y(30),XB(30),YB(30)
130 SX = 160:SY = 100
140 AX = 2:AY = 1
150 BX = 0:BY = 1
160 FOR I = 1 TO 21:READ X(I),Y(I)
170 XB(I) = SX+X(I)*AX+Y(I)*BX
180 YB(I) = SY+X(I)*AY+Y(I)*BY
190 NEXT
200 PSET (XB(1),YB(1)):FOR I = 2 TO 4:
    LINE -(XB(I),YB(I)):NEXT
210 PSET (XB(5),YB(5)):FOR I = 6 TO 11:
    LINE -(XB(I),YB(I)):NEXT

```

```
220 PSET (XB(12),YB(12)):FOR I = 13 TO 17:
    LINE -(XB(I),YB(I)):NEXT
230 PSET (XB(18),YB(18)):FOR I = 19 TO 21:
    LINE -(XB(I),YB(I)):NEXT
240 DATA 0,0,10,0,5,0,5,10
250 DATA 30,0,20,0,20,5,30,5,20,5,20,10,30,10
260 DATA 50,0,40,0,40,5,50,5,50,10,40,10
270 DATA 60,0,70,0,65,0,65,10
```

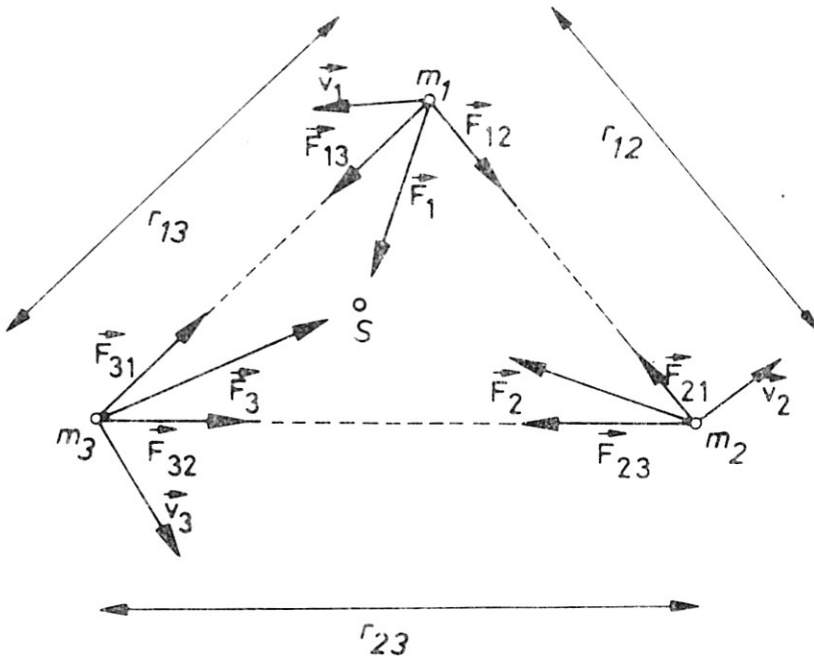
2.5 A 3-TEST probléma szimulációja

A következő program példa arra, hogy az információfeldolgozás milyen szemléletes formája a számítógépes grafika.

A számítógép által kiszámított adatok itt átalakulnak a velük analóg képernyő-pozíciókká.

A program két dimenzióban szimulálja három, különböző tömegű test mozgását, amelyekre — adott kezdeti paraméterek mellett (G a gravitációs állandó; m_1, m_2, m_3 a tömegek; v_1, v_2, v_3 a kezdősebességek; A, B, C a testek kiindulási helyzete) — csak a gravitációs erő hat.

Tekintettel arra, hogy a v sebesség és az F vonzóerő esetében vektoriális mennyiségekről van szó, ezeket a programnak is vektorokként kell kezelni. Egy test mozgásegyenletének felírásához nézzük meg a következő ábrát:



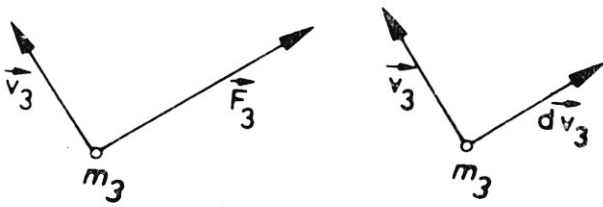
Az egyik tetszőleges testre, pl. az m_3 tömegű testre, egy meghatározott időpontban két erő hat, az F_{31} és az F_{32} , amelyeket az m_1 és az m_2 tömegű testek gravitációs hatása vált ki. A t időpontban az m_3 tömegű testre, tehát az

$$\mathbf{F}_3 = \mathbf{F}_{31} + \mathbf{F}_{32}$$

erő hat, amelynek iránya a rendszer S súlypontjába mutat. Az F_{31} és az F_{32} erőkre érvényes a következő összefüggés:

$$|\mathbf{F}_{31}| = \frac{m_3 m_1 G}{|\mathbf{r}_{31}|^2}; \quad |\mathbf{F}_{32}| = \frac{m_3 m_2 G}{|\mathbf{r}_{32}|^2}$$

Az erők irányát az \mathbf{r}_{31} és az \mathbf{r}_{32} vektorok határozzák meg.



Az m_3 tömegű test \mathbf{v}_3 kezdeti sebessége az \mathbf{F}_3 erő hatására megváltozik. A testnek az \mathbf{F}_3 erő által dt idő alatt létrehozott impulzusa (mozgásmennyisége):

$$d\mathbf{p} = \mathbf{F}_3 dt.$$

A test impulzusváltozására érvényes

$$d\mathbf{p} = m_3 d\mathbf{v}_3$$

összefüggést behelyettesítve az előzőbe:

$$m_3 d\mathbf{v}_3 = \mathbf{F}_3 dt.$$

Ebből a sebességváltozásra azt kapjuk, hogy

$$d\mathbf{v}_3 = \frac{\mathbf{F}_3}{m_3} dt.$$

Ezután lesz szükségünk a szimulációra, mivel eddig számítógép nélkül is eljutottunk, de tovább nem.

A szimulációhoz tételezzük fel, hogy a dt nagyon kicsi. A mi céljainkra megfelel, ha a $dt = 0,1$ s-ot választjuk (fizikai értelemben ez már nagyon kicsi érték). A következő egyenletet kápjuk:

$$d\mathbf{v}_3 = \frac{\mathbf{F}_3}{m_3} \cdot 0,1.$$

Fel kell tételeznünk, hogy a 0,1 s-os időintervallum alatt a gravitációs erő gyakorlatilag nem hat, a test

$$\mathbf{v}'_3 = \mathbf{v}_3 + d\mathbf{v}_3$$

sebességgel mozog, továbbá, hogy a másik két test kölcsönhatásából (vonzásából) származó erők sem lépnek fel.

Ezután a

$$d\mathbf{s}_3 = \mathbf{v}'_3 dt$$

mozgásegyenlettel ki tudjuk számítani az m_3 tömegű test új helyzetét:

$$\mathbf{P}'_3 = \mathbf{P}_3 + d\mathbf{s}_3 = \mathbf{P}_3 + \mathbf{v}'_3 dt.$$

Figyelembe vesszük még azt az esetet is, amikor a testek olyan közel kerülnek egymáshoz, hogy összeütköznek. Feltételezzük, hogy a testek tökéletesen rugalmatlanok és ezáltal egy K' test keletkezik, aminek tömege

$$m' = m_a + m_b$$

és impulzusa

$$\mathbf{p}' = \mathbf{p}_a + \mathbf{p}_b$$

lesz. Így \mathbf{v}' -re felírhatjuk:

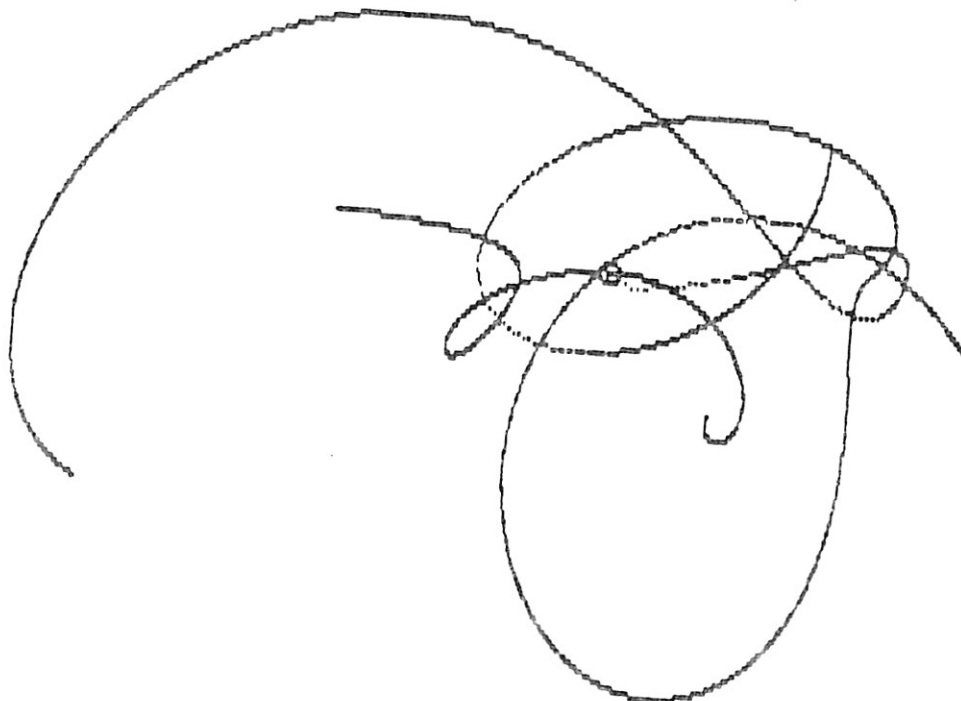
$$\mathbf{v}' = \frac{m_a \mathbf{v}_a + m_b \mathbf{v}_b}{m_a + m_b}.$$

Ha az Olvasó egy kicsit eljátszadzik ezzel a programmal, látni fogja, hogy ezzel a bolygók Kepler-féle mozgása nagyon szépen illusztrálható.

Programleírás

Sor	Magyarázat
120	Távolságfüggvény
130—150	A paraméterek kezdeti értékei
180—220	Az $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ meghatározása és egységvektorokká alakítása
230—250	A mindenkori erőnagyságok meghatározása
260—280	Az erők vektoriális összeadása
290—340	A sebességek vektoriális összeadása
350—370	Az A', B', C' új pozíciók meghatározása
390—450	Az egymáshoz viszonyított távolság ellenőrzése
460—570	Az impulzus- és tömegértékek összeadása ütközés esetén

A 3-test probléma szimulációja $M_1=400, M_2=50, M_3=100, G=3$



FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENŰL** el kell olvasni!

```
50 REM ... P4 ...
100 REM ..... A 3-test problema szimulacioja
110 CLS:SCREEN 2:PRINT"A 3-test problema szimulacioja"
120 DEF FND(DX) = SQR(DX^2+DY^2)
125 FR=3*1E+08
130 AA = 160:A2 = 80:BB = 190:B2 = 140:CC = 280:
    C2 = 60
140 VA(1) = 0:VA(2) = 0:VB(1) = -4.1:VB(2) = -1:
    VC(1) = 0:VC(2) = 1.7
150 M1 = 900:M2 = 50:M3 = 100:G = 6.67*.0000001
160 PRINT"M1= "+STR$(M1)+" :M2= "+STR$(M2)+" :
    M3= "+STR$(M3)+" :G= "+STR$(G)
165 G=FR*G
170 PSET (2*AA,A2):PSET (2*BB,B2):PSET (2*CC,C2)
180 CX = BB-AA:CY = B2-A2:AX = CC-BB:AY = C2-B2:
    BX = CC-AA:BY = C2-A2
190 GOSUB 390
200 C0 = CX/SQR(CX^2+CY^2):C1 = CY/SQR(CX^2+CY^2)
210 B0 = BX/SQR(BX^2+BY^2):B1 = BY/SQR(BX^2+BY^2)
220 A0 = AX/SQR(AX^2+AY^2):A1 = AY/SQR(AX^2+AY^2)
230 FA(1) = G*M1*M2/(CX^2+CY^2):
    FA(2) = G*M1*M3/(BX^2+BY^2)
240 FB(1) = FA(1):FB(2) = G*M2*M3/(AX^2+AY^2)
250 FC(1) = FA(2):FC(2) = FB(2)
260 AR(1) = FA(1)*C0+FA(2)*B0:AR(2) = FA(1)*C1+FA(2)*B1
270 BR(1) = FB(1)*(-C0)+FB(2)*A0:
    BR(2) = FA(1)*(-C1)+FB(2)*A1
280 CR(1) = FC(1)*(-B0)+FC(2)*(-A0):
    CR(2) = FC(1)*(-B1)+FB(2)*(-A1)
290 IF M1 = 0 THEN VA(1) = 0:VA(2) = 0:GOTO 310
300 VA(1) = VA(1)+AR(1)*.1/M1:VA(2) = VA(2)+AR(2)*.1/M1
310 IF M2 = 0 THEN VB(1) = 0:VB(2) = 0:GOTO 330
320 VB(1) = VB(1)+BR(1)*.1/M2:VB(2) = VB(2)+BR(2)*.1/M2
330 IF M3 = 0 THEN VC(1) = 0:VC(2) = 0:GOTO 350
340 VC(1) = VC(1)+CR(1)*.1/M3:VC(2) = VC(2)+CR(2)*.1/M3
350 AA = AA+VA(1)*.1:A2 = A2+VA(2)*.1
360 BB = BB+VB(1)*.1:B2 = B2+VB(2)*.1
370 CC = CC+VC(1)*.1:C2 = C2+VC(2)*.1
380 GOTO 170
390 DX = CX:DY = CY:D1 = FND(DX)-4*(M1 = 0 OR M2 = 0)
400 DX = AX:DY = AY:D2 = FND(DX)-4*(M2 = 0 OR M3 = 0)
410 DX = BX:DY = BY:D3 = FND(DX)-4*(M1 = 0 OR M3 = 0)
420 IF D1>3 AND D2>3 AND D3>3 THEN RETURN
430 IF D1<=3 THEN 460
440 IF D2<=3 THEN 500
```

```

450 IF D3<=3 THEN 540
460 VA(1) = (M1*VA(1)+M2*VB(1))/(M1+M2):VB(1) = 0
470 VA(2) = (M1*VA(2)+M2*VB(2))/(M1+M2):VB(2) = 0
480 M1 = M1+M2:M2 = 0
490 CIRCLE (2*AA,A2),5:RETURN
500 VB(1) = (M3*VC(1)+M2*VB(1))/(M3+M2):VC(1) = 0
510 VB(2) = (M3*VC(2)+M2*VB(2))/(M3+M2):VC(2) = 0
520 M2 = M3+M2:M3 = 0
530 CIRCLE (2*BB,B2),5:RETURN
540 VA(1) = (M3*VC(1)+M1*VA(1))/(M3+M1):VC(1) = 0
550 VA(2) = (M3*VC(2)+M1*VA(2))/(M3+M1):VC(2) = 0
560 M1 = M3+M1:M3 = 0
570 CIRCLE (2*AA,A2),5:RETURN

```

2.6 Üzleti grafika

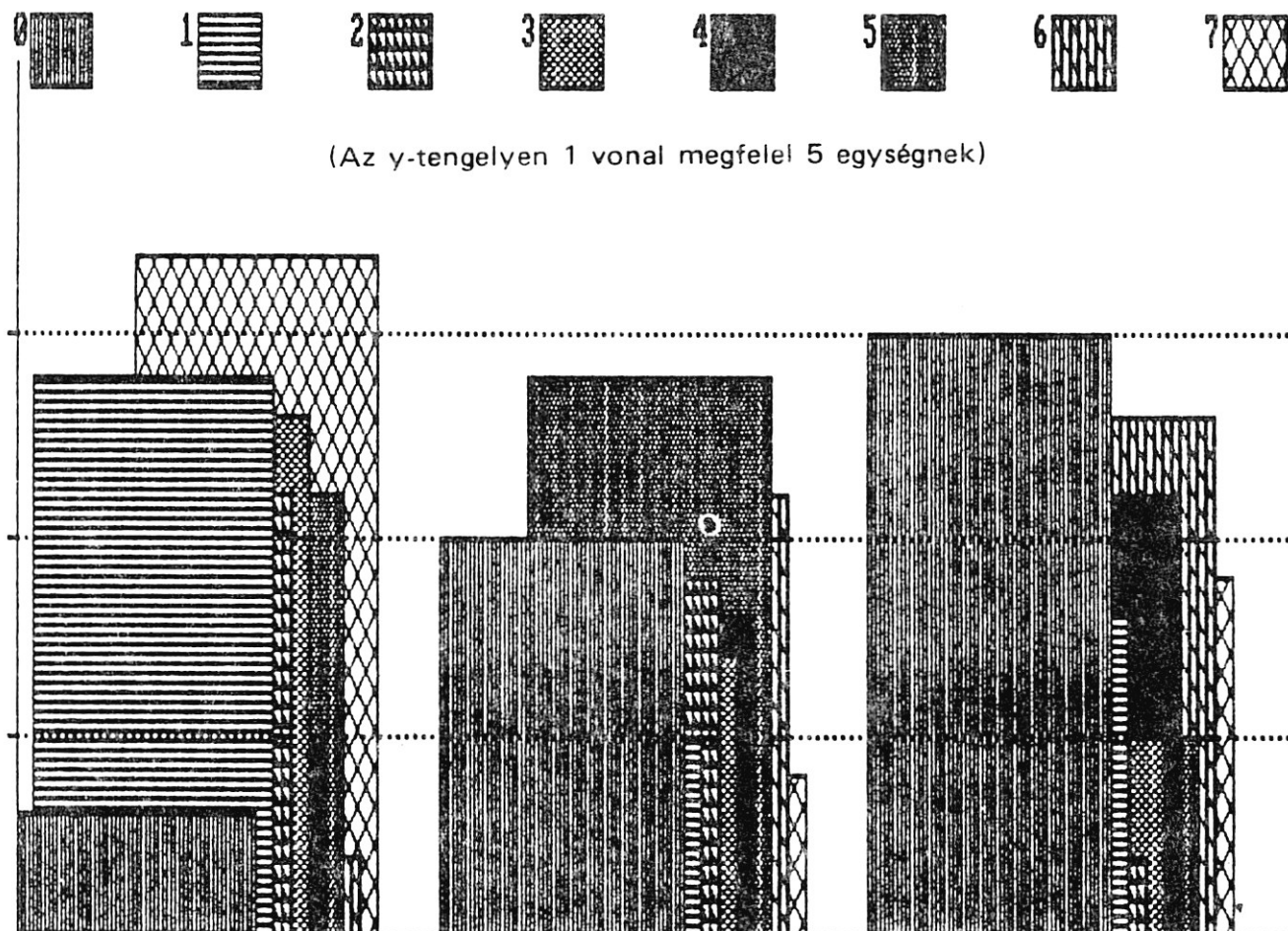
A grafikák két következő fajtája az ún. üzleti grafika témakörébe tartozik és az üzleti adatok grafikus feldolgozásának két leggyakoribb módját képviseli.

2.6.1 OSZLOPDIAGRAMOK

Az oszlopdigramok az adatok feldolgozásának nagyon egyszerű, ugyanakkor szemléletes eszközei. Különösen kétdimenziós adatmezők ábrázolására alkalmasak, pl. arra, hogy összehasonlítsuk több éven keresztül különböző vállalati részlegek üzletmenetét. A következő program ezt szimulálja, a példában 3 év és 8 részleg szerepel.

Az ábrázolás sajátossága, hogy élve a PAINT utasítás adta lehetőségekkel, az oszlopok jól megkülönböztethetők a fekete-fehér képernyőn és az esetleges nyomtatásban egyaránt.

Az adatok a program végén DATA-sorokban vannak megadva. A PAINT utasítás paraméterei a 380—450-es sorokban vannak. A 460—530-as sorokban az ábrázolandó adatok állnak. A programnak a felhasználó egyéni igényei szerinti módosításához fontos a 130-as és a 150-es sor.




```
130 READ M, N: ...
```

ahol M a különböző vállalati részlegek száma;
 N a figyelembe vett évek száma.

Az adatok manuálisan vihetők be, ha végrehajtjuk a következő változtatásokat:

```
130 INPUT „Vállalati részleg, évek”; M, N: ...
150 PRINT „Vállalati részleg”; J; „év”; I: INPUT A(J, I):
```

Természetesen az is lehetséges, hogy az adatokat egy szekvenciális file-ból olvassuk. Akkor a módosítás:

```
125 INPUT „file-név (egység: név.kód)”; NA$
130 OPEN „I”, 1, NA$: INPUT #1, M, N: ...
150 INPUT #1, A(J, I): ...
160 NEXT I: NEXT J: CLOSE 1
```

Fontos, hogy M és N értéke a szekvenciális adatállomány elején álljon. Az y -értékeket a program mindig a legnagyobb fellépő adatérték szerint transzformálja. Ez mindig 140 képelem nagyságú. Az oszlopok szélességét a program úgy határozza meg, hogy azok az x -tengely 10—610 szakaszát egyenlő arányban töltsék ki.

Programleírás

Sor	Magyarázat
120	8 különböző PAINT-minta programozása
130—160	Az adatok beolvasása
180—200	A rendszerkonstansok meghatározása
210—310	Az oszlopok rajzolása
320—370	Feliratozás
380—450	PAINT-adatok
460—570	Adatok

FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENÜL** el kell olvasni!

```
50 REM ... P5 ...
100 REM ..... Oszlopdigram
110 CLS: SCREEN 2: KEY OFF
120 FOR J=0 TO 7: READ Z: A$(J) = "": FOR I=1 TO Z: READ A:
    A$(J) = A$(J) + CHR$(A)
130 NEXT I: NEXT J
135 READ M, N: DIM A(M, N): B=0
140 FOR J=1 TO M: FOR I=1 TO N
150 READ A(J, I): IF A(J, I) > B THEN B = A(J, I)
160 NEXT I: NEXT J
170 LINE (10, 10) - (10, 190): LINE (10, 190) - (610, 190)
```

```

180 E=600/(7*N):BB=4*E:BS=3*E
190 Y0=140/B
200 BY=2*E/(M-1)
210 FOR I=1 TO N:B=190:FOR J=1 TO M
220 XU=10+(J-1)*BY+(I-1)*(BB+BS):X0=XU+BB
230 Y=190-Y0*A(J,I)
240 PSET (X0,190)
250 LINE -(X0,Y):IF J=1 THEN 280
260 FOR K=1 TO J-1:IF POINT (X0-K*BY,Y)=1 THEN
    LINE -(X0-K*BY,Y):GOTO 300
270 NEXT K
280 LINE -(XU,Y)
290 IF Y<B THEN LINE -(XU,B):B=Y
300 PAINT (X0-.5*BY,188),A$((J-1) AND 7)
310 NEXT J:NEXT I
320 FOR I=1 TO 71 STEP 10:LOCATE 1,I:PRINT (I-1)/10
330 XU=(I+1)*8:X0=XU+30:LINE (XU,0)-(X0,15),,B:
    PAINT (XU+2,10),A$((I-1)/10)
340 NEXT
350 LOCATE 4,10:PRINT "Egy vonal az y-tengelyen megfelel
    5 egysegnek"
360 Q=INT(140/(5*Y0)):FOR I=1 TO Q:
    LINE (5,(190-I*5*Y0))-(10,(190-I*5*Y0))
370 LINE (10,(190-I*5*Y0))-(610,(190-I*5*Y0)),,,&H9999:
    NEXT
380 DATA 1,170
390 DATA 2,255,0
400 DATA 4,136,204,238,255
410 DATA 2,204,51
420 DATA 1,255
430 DATA 2,170,85
440 DATA 8,204,204,236,220,204,204,206,205
450 DATA 8,129,66,36,24,24,36,66,129
460 DATA 8,3:REM ..... Adatok
470 DATA 3,10,15
480 DATA 14,5,8
490 DATA 11,9,2
500 DATA 13,7,5
510 DATA 5,14,11
520 DATA 11,8,5,
530 DATA 2,11,13
540 DATA 17,4,9

```

2.6.2 KÖRDIAGRAMOK

A különbség az oszlopdiagram és a kördiagram használata között az, hogy a körmiagramot általában a részek százalékos megoszlásának ábrázolására használják. A következő program egy ilyen rajzot állít elő. (A bemutatott kördiagramokat olykor tortagrafikának is nevezik.) Az értékeket, összesen 14-et (380-as sor), a véletlenszám-generátor segítségével állítjuk elő (180-as sor).

Nem szükséges, hogy az adatok százalékos értékek formájában legyenek, az azonban szükséges, hogy a részek (a példában 14) együttesen a 100%-ot reprezentálják. Az adatokból a program már kiszámolja a százalékos-megoszlást.

Ha az adatokat manuálisan akarjuk bevinni, akkor a programot át kell alakítani a következő módon: a 170-es sorban kitöröljük a

```
READ N : DIM A(N)
```

és a 180-as sorban az

```
A(I) = INT(40*RND (1)) + 10
```

utasításokat.

```
170 INPUT „szám”; N : DIM A(N)
180 FOR I = 1 TO N : INPUT „adat”; A(I) : NEXT
```

Ha az adatok szekvenciális file formában állnak rendelkezésre, akkor gépeljük be a

```
165 INPUT „egység: név.kód”; NA$
170 OPEN „I”, 1, NA$ : INPUT # 1, N : DIM A(N) : . . .
```

utasításokat és a 180-as sorban az $A(I) = \dots$ helyett a

```
180 INPUT # 1, A(I) : B = B + A(I) : NEXT : CLOSE 1
```

utasításokat.

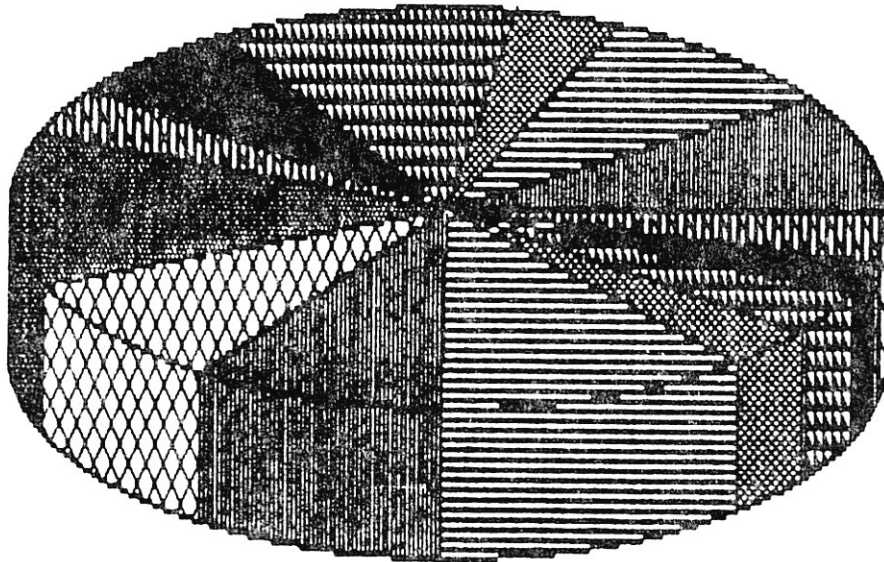
Ügyeljünk arra, hogy a szekvenciális file első adata a vizsgált értékek száma legyen.

Programleírás

Sor	Magyarázat
110	8 különböző PAINT-minta programozása
130—160	A kontúr megrajzolása
170—180	Az adatok beolvasása
190—250	Átszámítás %-értékekre és rajzolás
260—300	Feliratozás
310—370	PAINT-adatok
380	Az adatok száma



Kördiagram



FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENÜL** el kell olvasni!

```

50 REM ... P6 ...
100 REM ..... Kördiagram
110 FOR J=0 TO 7:READ Z:A$(J)="":FOR I=1 TO Z:
  READ A:A$(J)=A$(J)+CHR$(A)
115 NEXT I:NEXT J
120 CLS:SCREEN 2:KEY OFF:RANDOMIZE V
130 CIRCLE (320,100),200,,0,3.14,1/5
140 CIRCLE (320,130),200,,3.14,6.28,1/5
150 LINE (520,100)-(520,130)
160 LINE (120,100)-(120,130):LINE (320,100)-(520,100)
170 READ N:DIM A(N):B=0:C=0:PI=3.14159
180 FOR I=1 TO N:A(I)=INT(40*RND(1))+10:B=B+A(I):NEXT
190 FOR I=1 TO N:C=C+A(I):W=C*2*PI/B
200 X=320+200*COS(W):Y=100-200/5*SIN(W)
210 LINE (320,100)-(X,Y)
220 IF W>PI THEN LINE (X,Y)-(X,Y+30)
230 W=(C-A(I)/2)*2*PI/B:X=320+190*COS(W):
  Y=100-190/5*SIN(W)
240 PAINT (X,Y),A$((I-1) AND 7)
250 NEXT
260 CIRCLE (320,100),200,,3.14,6.28,1/5
270 FOR I=1 TO 71 STEP 10:LOCATE 1,I:PRINT (I-1)/10

```

```
280 XU=(I+1)*8:X0=XU+30:LINE (XU,0)-(X0,15),,B:
  PAINT (XU+2,10),A$((I-1)/10)
290 NEXT
300 LOCATE 4,35:PRINT "Kordiagram"
310 DATA 1,170,2,255,0
320 DATA 2,204,51
330 DATA 4,136,204,238,255
340 DATA 1,255
350 DATA 8,204,204,236,220,204,204,206,205
360 DATA 2,170,85
370 DATA 8,129,66,36,24,24,36,66,129
380 DATA 14:REM ..... Az adatok szama
```

2.7 CAD — számítógéppel támogatott tervezés

Manapság a grafikával kapcsolatos csaknem valamennyi műszaki területen — a racionalizálási intézkedések révén — egyre nagyobb teret nyernek az ún. CAD/CAM technológiák. A CAD/CAM a *Computer Aided Design/Computer Aided Manufacturing* (a számítógéppel támogatott tervezés/számítógépek támogatott gyártás) rövidítése. A szakértők véleménye szerint ezek a technológiák az utóbbi évek legfontosabb fejlesztéseinek minősülnek. A CAD/CAM technológiák alkalmazásával sok — a terv és a gyártás között levő — időt rabló lépés válik feleslegessé, lévén, hogy a feladatokat egyre növekvő mértékben számítógéppel vezérelt gépek veszik át.

2.7.1 A GRAFIKUS GET ÉS PUT UTASÍTÁSOK

A grafikus GET és PUT utasítások funkcióját korábban már elmagyaráztuk. Most e két utasítás speciális felhasználási lehetőségeiről fogunk szólni. Ezek az utasítások lehetővé teszik, hogy egész ábracsoportokat tudjunk a képernyőn eltolni, mégpedig az eltolás vektorának nagyságától függően többé-kevésbé folyamatosan.

A PUT utasítással kapcsolatban lehetséges eljárások egyike az XOR. A hatását itt részletesen megtárgyaljuk.

Az XOR jelölés az *eXclusive OR* (kizáró „vagy”) kifejezésből származik.

Az XOR logikai művelet két byte tartalmát hasonlítja össze bitenként. Az összehasonlítás eredménye egy újabb byte, amely a következő módon áll elő:

- ha az 1-es byte-ban is és a 2-es byte-ban is a bit 1 állapotban van, akkor az összeg byte-ban a bit 0 állapotú (ezért *exclusive*);
- ha az 1-es byte-ban a bit 1 állapotban, a 2-es byte-ban pedig 0 állapotban van, ill. fordítva, akkor az összeg byte-ban a bit állapota 1 lesz;
- ha a bit mindkét byte-ban 0 állapotú, akkor az összeg byte-ban is 0 állapotú lesz.

Egy példa alapján hasonlítsuk össze az XOR (kizáró vagy) hatását a közösleges OR (vagy) hatásával.

OR	XOR
10101010	10101010
+ 11011111	+ 11011111
<hr/>	<hr/>
= 11111111	= 01110101

Milyen előny származik ebből a különleges operációból?

Ehhez tekintsünk még egy példát:

a)	10101010	b)	11101111
	+ 10101010		+ 10101010
	<hr/>		<hr/>
	00000000		01000101
			+ 10101010
			<hr/>
			11101111

Ha ugyanazt a bitsorozatot az XOR-ral adjuk össze (*a*) eset), akkor láthatóan az eredmény egy nulla byte. A PUT utasítást az XOR paraméterrel alkalmazva mindez azt jelenti, hogy törölni tudunk egy ábracsoportot úgy, hogy ugyanazt az ábracsoportot ugyanazon a helyen jelenítjük meg. A PUT utasítással tehát mozgatni tudunk egy ábracsoportot, ha a következő sorrendben járunk el (az ÁCS rövidítés az ábracsoportot jelöli):

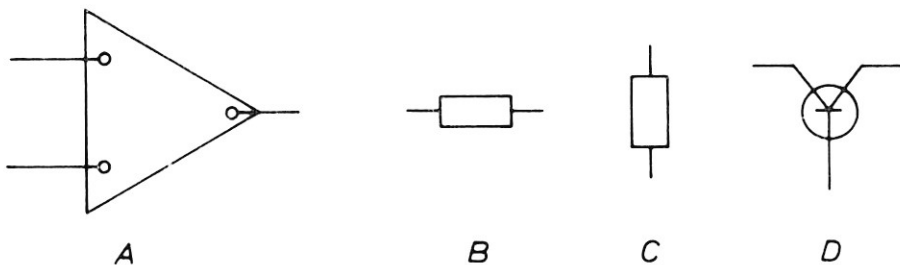
- 1) az ÁCS-ot jelenítsük meg az (x, y) pozícióban;
- 2) számítsuk ki az új (x_1, y_1) pozíciót;
- 3) helyezzük az ÁCS-ot a régi (x, y) pozícióra (törlés);
- 4) jelenítsük meg az ÁCS-ot az új (x_1, y_1) pozícióban (megfelel az 1)-nek).

Mi történik akkor, ha az ÁCS-ot egy már meglévő ábrán mozgatjuk (*b*) eset)? Az első lépésben megjelenítjük az ÁCS-ot az (x, y) pozícióban. Ekkor azok a pontok, amelyek mind az ÁCS-ban, mind a már meglévő ábrában megvannak, invertálódnak, ami most azt jelenti, hogy törlődnek. Ha most a 3) lépésben az ÁCS-ot másodszor is megjelenítjük az (x, y) pozícióban, akkor az ÁCS minden — az első lépésben megjelenített — pontja eltűnik, azok a pontok pedig, amelyek az első lépésben törlődtek, most ismét megjelennek. Az eredeti hátteret tehát egészében visszaállítottuk.

A következő program ennek a technikának egy egyszerű alkalmazási példája.

2.7.2 EGY EGYSZERŰ CAD-PROGRAM

A programmal ismert építőelemekből áramköri kapcsolási rajzot lehet összeállítani. A példához a következő elemeket választjuk:



Az elemek az *A*, *B*, *C*, *D* tömbökben vannak tárolva (a megfelelő dimenzionálás a program kezdete). Előbb az elemeket egyenként megszerkesztjük a képernyőn és a GET utasítással tároljuk. Természetesen tetszőleges számú más elem is definiálható. Törölni nem szabad: az utolsó ábracsoport, amit a QX-ben és a QY-ban tárolunk, az eltolható koordinátatengelyeket határozza meg. Amennyiben további elemeket választunk, a programot módosítani kell. A szükséges változtatások:

- 1) a területnév megfelelő dimenzionálása a program elején;
- 2) a pótlólagos sorok beillesztése ugyanolyan formában, mint amelyek a 390—420, 530—560, 700—730, 790—820 mezőkben már megvannak. A PUT utasítás speciális belső struktúrája sajnos nem enged meg kényelmesebb megoldást.

Funkciók

Indítás után az áramköri elem rajzát tároló tömbnek megfelelő betűt (*A, B, C, D*) nyomjuk meg a billentyűzeten. Az ábra megjelenik a bal felső sarokban. A numerikus blokk 3 nyomógombjával az ábrát a feje tetejére állítva, más szóval az *x*-tengelyre tükrözve kaphatjuk meg, ha a nyomógombot az elem kiválasztása előtt működtetjük. Az elem ekkor a kiválasztás után balra lent jelenik meg. A 9 nyomógommbal állíthatjuk ismét az eredeti állapotot vissza (ez is csak egy új elem kiválasztása előtt lehetséges). Az ábrák a numerikus blokk nyomógombjai segítségével mozgathatók (az 5 befejezi a programot).

Normál beállításban az elemek elmozdulása mindig 5 egységnyi. Ha az inverz ábrázolásra kapcsolunk át, akkor a 7...9 nyomógombok funkciót cserélnek az 1...3 nyomógombokkal. A lépéstávolságot bármikor megváltoztathatjuk, ha az *S* nyomógombot és egy 1 és 9 közötti számot nyomunk meg. Az ábra mozgatásakor ügyeljünk arra, hogy az ki ne csússzon a képernyőről. Ha ez mégis bekövetkezik, akkor az ILLEGAL FUNCTION CALL hibaüzenet jelenik meg a képernyőn.

Az *L* törli a már kiválasztott elemet, ha annak megrajzolása még nem történik meg. A *P* megrajzol egy elemet és várja a felhasználó további parancsát.

A *Q* a kurzort mozgatja, mintha az egy elem lenne (mindaddig hatástalan, amíg egy másik elem van kiválasztva), segítségével két pont között vonalat húzhatunk vagy törölhetjük a vonalat, ill. törölhetjük vele a képernyő egy tartományát.

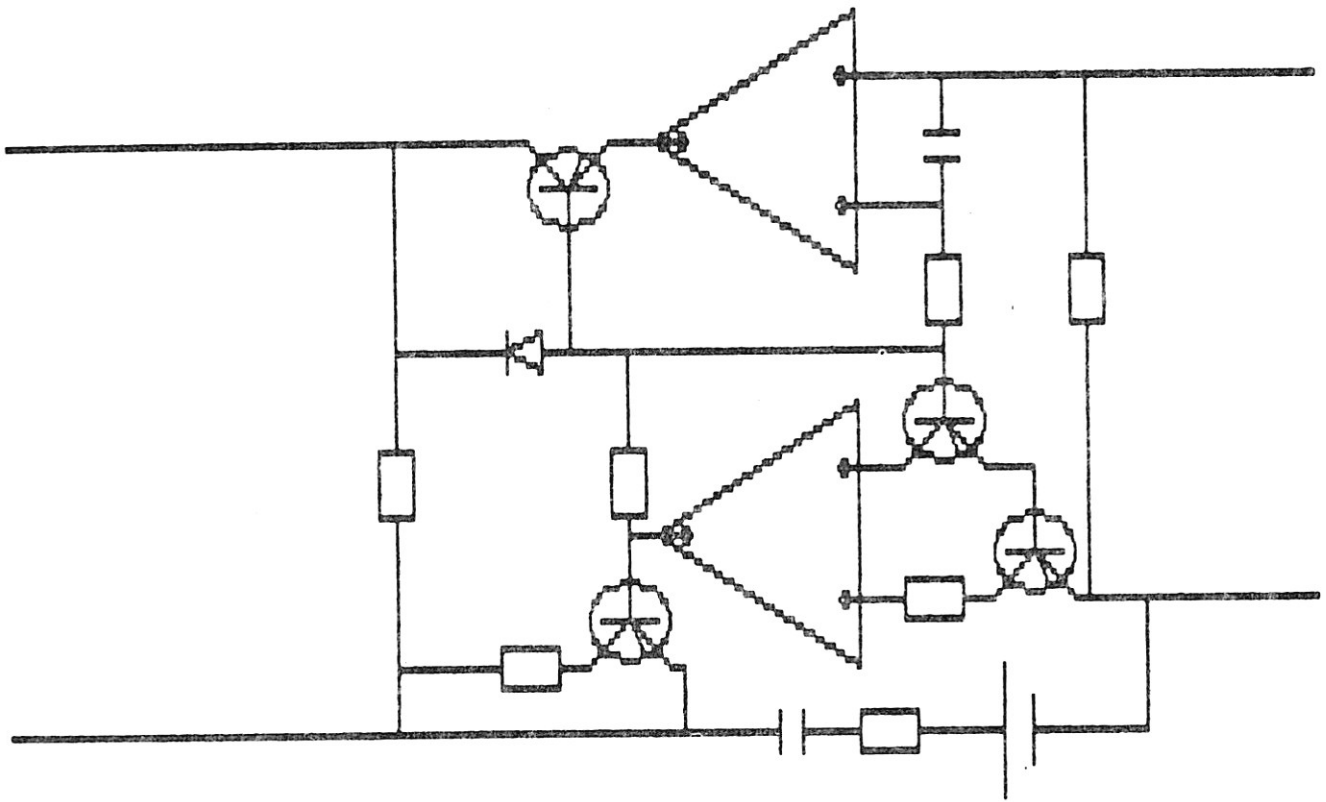
Az *E* a tároló rutint hívja, amely a képernyőt tárolja a *B* egységen, *work. cad* jelöléssel.

A program indításkor megkérdezi, hogy van-e olyan rajz, amelynek feldolgozását tovább kívánjuk folytatni. Ha igen, akkor feltételezi, hogy az *work. cad* néven található. A munka befejezése után vagy egy másik rajz esetleges tárolása előtt az első rajz nevét meg kell változtatni, egyébként az felülíródik.

Áttekintés

Kiválasztás:	a felhasználó által megfelelően definiált, tetszőleges jel
9:	az ábrázolási mód normál beállítása
3:	tükrözött ábrázolás
Vezérlés:	1...9 nyomógombok; 5 = vége
<i>S</i> 1...9:	a léptetés távolsága
<i>L</i> :	grafikus elem kiválasztásának törlése (<i>E</i> esetén hatástalan)
<i>Q</i> :	kurzor
	vonalat húzni: — a kezdőpontnál <i>Q</i>
	— a végpontnál <i>Q</i>
	— vonalat húzni, szököz billentyű
	— a vonal törlése <i>L</i>
	területet törölni: — a bal felső sarok <i>W</i>
	— jobb alsó sarok <i>W</i>
<i>P</i> :	rajzolás
<i>E</i> :	a rajz tárolása a <i>B</i> egységen <i>work. cad</i> néven

Megjegyzés: a következő ábra egy ábrázolási példa, de mint kapcsolási rajz valójában értelmetlen.



FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENÜL** el kell olvasni!

```

50 REM ... P7 ...
100 REM ..... Szamitogepvel tamogatott tervezes
110 CLS:SCREEN 2:DIM A(500),B(30),C(100),D(32),
    QY(90),QX(205):KEY OFF
120 LOCATE 10,20:PRINT "Kerem valtson at a
    NUM.LOCK-ra es kisbetus irasra"
130 FOR I=1 TO 4000:NEXT
140 PSET (10,0):LINE -(10,40):LINE -(80,20):
    LINE -(10,0)
150 LINE (0,10)-(15,10):CIRCLE (15,10),2
160 LINE (0,30)-(15,30):CIRCLE (15,30),2
170 LINE (70,20)-(90,20):CIRCLE (70,20),2
180 GET (0,0)-(90,40),A:CLS
190 LINE (5,0)-(25,6),,B:LINE (0,3)-(5,3):
    LINE (25,3)-(30,3)
200 GET (0,0)-(30,6),B:CLS
210 CIRCLE (19,7),14:LINE (0,0)-(5,0):LINE -(19,7):
    LINE -(33,0):LINE -(38,0)
220 LINE (9,7)-(29,7):LINE (19,7)-(19,19):
    LINE (19,7)-(15,3)
230 GET (0,0)-(38,19),C:CLS
240 LINE (6,0)-(6,2):LINE (0,2)-(12,12),,B:
    LINE (6,12)-(6,14)
250 GET (0,0)-(12,14),D:CLS
260 LINE (0,0)-(639,0):LINE (0,0)-(0,199)

```

```

270 GET (0,0)-(639,0),QY:GET (0,0)-(0,199),QX:CLS
280 LOCATE 5,10:INPUT "Meglevo abrat folytat";Q$
290 IF Q$="N" THEN CLS:GOTO 330
300 PRINT "Lemezegyseg (A/B)"
310 L$=INKEY$:IF L$="" THEN 310
320 A$=L$+":WORK.CAD":BLOAD A$
330 U=10:V=5:X=0:Y=0
340 T$=INKEY$:IF T$="" THEN 340
350 BEEP
360 IF T$="9" THEN SCREEN (0,0)-(639,199):GOTO 340
370 IF T$="3" THEN (0,0)-(639,199):GOTO 340
380 IF T$="5" THEN LOCATE 1,1:END
390 IF T$="A" THEN PUT (X,Y),A:T=1:GOTO 460
400 IF T$="B" THEN PUT (X,Y),B:T=2:GOTO 460
410 IF T$="C" THEN PUT (X,Y),C:T=3:GOTO 460
420 IF T$="D" THEN PUT (X,Y),D:T=4:GOTO 460
430 IF T$="Q" THEN SCREEN (0,0)-(639,199):
X=320:Y=100:PUT (X,0),QX:PUT (0,Y),QY:
T=5:GOTO 460
440 IF T$ = "e" THEN BEEP:GOTO 970
450 GOTO 340
460 A$=INKEY$:IF A$="" THEN 460
470 IF A$="S" THEN BEEP:GOTO 750
480 IF A$="1" THEN BEEP:GOTO 780
490 IF A$="Q" THEN BEEP:GOTO 840
500 IF A$="W" THEN BEEP:GOTO 920
510 IF A$<> "P" THEN 570
520 BEEP:ON T GOTO 530,540,550,560
530 PUT (X,Y),A,OR:GOTO 330
540 PUT (X,Y),B,OR:GOTO 330
550 PUT (X,Y),C,OR:GOTO 330
560 PUT (X,Y),D,OR:GOTO 330
570 W=VAL(A$)
580 ON W GOTO 590,600,610,620,630,640,650,660,670
590 XN=X-U:YN=Y+V:GOTO 680
600 XN=X:YN=Y+V:GOTO 680
610 XN=X+U:YN=Y+V:GOTO 680
620 XN=X-U:YN=Y:GOTO 680
630 XN=X:YN=Y:GOTO 680
640 XN=X+U:YN=Y:GOTO 680
650 XN=X-U:YN=Y-V:GOTO 680
660 XN=X:YN=Y-V:GOTO 680
670 XN=X+U:YN=Y-V
680 XN=-XN*(XN>-1 AND XN<640):YN=-YN*(YN>-1 AND YN<200)
690 ON T GOTO 700,710,720,730,740
700 PUT(X,Y),A:PUT (XN,YN),A:X=XN:Y=YN:GOTO 460
710 PUT(X,Y),B:PUT (XN,YN),B:X=XN:Y=YN:GOTO 460
720 PUT(X,Y),C:PUT (XN,YN),C:X=XN:Y=YN:GOTO 460
730 PUT(X,Y),D:PUT (XN,YN),D:X=XN:Y=YN:GOTO 460

```

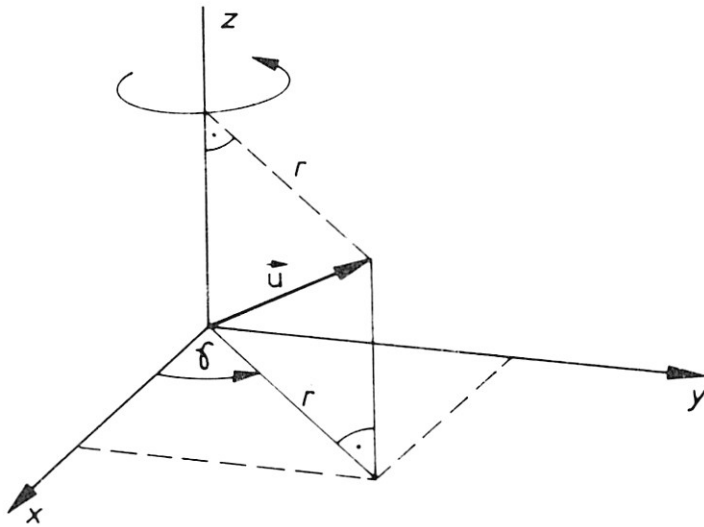
```

740 PUT(X,Ø),QX:PUT (Ø,Y),QY:PUT (XN,Ø),QX:
   PUT (Ø,YN),QY:X=XN:Y=YN:GOTO 46Ø
750 A$=INKEY$:IF A$="" THEN 75Ø
760 S=VAL(A$):IF S=Ø THEN 75Ø
770 U=S:V=.5*S:GOTO 46Ø
780 ON T GOTO 79Ø,80Ø,81Ø,82Ø,83Ø
790 PUT (X,Y),A:GOTO 33Ø
800 PUT (X,Y),B:GOTO 33Ø
810 PUT (X,Y),C:GOTO 33Ø
820 PUT (X,Y),D:GOTO 33Ø
830 PUT (X,Ø),QX:PUT (Ø,Y),QY:GOTO 33Ø
840 QQ=QQ+1:IF QQ=2 THEN QQ=Ø:GOTO 86Ø
850 AX=X:AY=Y:GOTO 46Ø
860 BX=X:BY=Y
870 A$=INKEY$:IF A$="" THEN 87Ø
880 PUT (X,Ø),QX:PUT (Ø,Y),QY
890 BEEP:IF A$="1" THEN LINE(AX,AY)-(BX,XY),Ø:GOTO 91Ø
900 LINE (AX,AY)-(BX,BY)
910 GOTO 33Ø
920 QQ=QQ+1:IF QQ=2 THEN QQ=Ø:GOTO 94Ø
930 AX=X:AY=Y:GOTO 46Ø
940 BX=X:BY=Y
950 PUT (X,Ø),QX:PUT (Ø,Y),QY
960 FOR XX=AX TO BX:LINE (XX,AY)-(XX,BY),Ø:NEXT:
   GOTO 91Ø
970 DEF SEG=&HB8ØØ
980 BSAVE "B:WORK.CAD",Ø,162ØØ
990 CLS:GOTO 34Ø

```

2.8 Rotációs mátrixok

Mint már láttuk, az \mathbf{a} vektort alkalmas mátrix segítségével elforgathatjuk. Ez természetesen háromdimenziós esetben is lehetséges. Ha abból indulunk ki, hogy egy tetszőleges test véges számú helyvektorral leírható, akkor azt a testet egy tetszőlegesen elforgatott helyzetben is tudjuk ábrázolni, csak minden egyes helyvektort el kell forgatnunk a rotációs mátrix (forgatásmátrix) segítségével. A következőkben ilyen mátrixokkal foglalkozunk. Nézzük meg az \mathbf{U} helyvektort egy derékszögű koordináta-rendszerben:



Ezt a vektort szeretnénk a z -tengely körül egy meghatározott γ szöggel elforgatni.

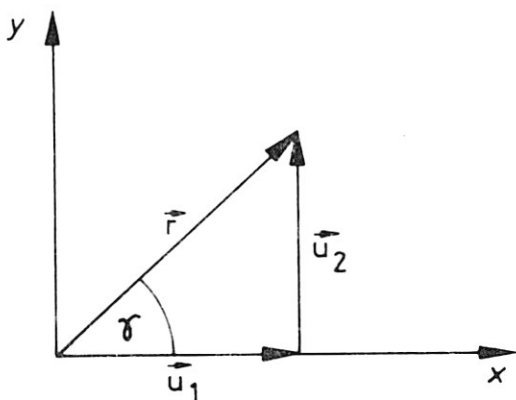
Eközben a z -koordináta és r nem változik meg, csak a szög változik:

$$\gamma' = \gamma + \delta.$$

Az \mathbf{U} vektort koordinátákkal a következő módon írhatjuk fel:

$$\mathbf{U} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} r \cos \gamma \\ r \sin \gamma \\ u_3 \end{pmatrix}$$

Mivel a z -koordináta (u_3) nem változik, figyelmünket az x , y -koordinátákra korlátozhatjuk:



Az \mathbf{r} vektorra érvényes az

$$\mathbf{r} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mathbf{u}_1 + \mathbf{u}_2$$

összefüggés.

Mivel az \mathbf{u}_1 vektor az \mathbf{r} vektor x -tengelyre való vetülete, \mathbf{u}_1 -et a következő formában írhatjuk:

$$\mathbf{u}_1 = \begin{pmatrix} u_1 \\ 0 \end{pmatrix} = \begin{pmatrix} r \cos \gamma \\ 0 \end{pmatrix}.$$

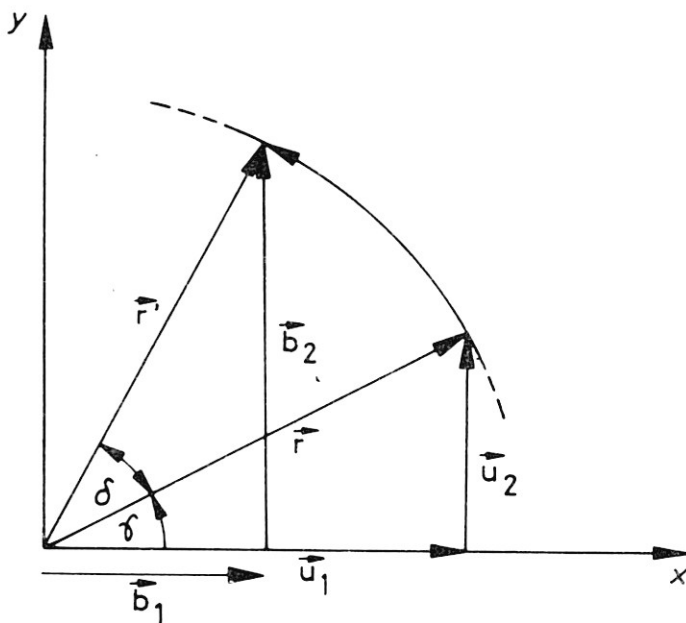
Az \mathbf{u}_2 vektorra ehhez hasonlóan azt kapjuk, hogy

$$\mathbf{u}_2 = \begin{pmatrix} 0 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ r \sin \gamma \end{pmatrix}.$$

Tehát

$$\mathbf{r} = \begin{pmatrix} r \cos \gamma \\ r \sin \gamma \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

Most elforgatjuk az \mathbf{r} vektort δ szöggel:



Az \mathbf{r}' vektorra érvényesek a következő összefüggések:

$$\mathbf{b}_1 = \begin{pmatrix} r \cos (\gamma + \delta) \\ 0 \end{pmatrix} = \begin{pmatrix} r \cos \gamma \cos \delta - r \sin \gamma \sin \delta \\ 0 \end{pmatrix};$$

$$\mathbf{b}_2 = \begin{pmatrix} 0 \\ r \sin (\gamma + \delta) \end{pmatrix} = \begin{pmatrix} 0 \\ r \sin \gamma \cos \delta + r \cos \gamma \sin \delta \end{pmatrix}.$$

Mivel

$$\mathbf{r}' = \mathbf{b}_1 + \mathbf{b}_2,$$

így fennáll az

$$\mathbf{r}' = \begin{pmatrix} r \cos \gamma \cos \delta + r \sin \gamma \sin \delta \\ r \sin \gamma \cos \delta + r \cos \gamma \sin \delta \end{pmatrix}$$

összefüggés. Az

$$\mathbf{r} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} r \cos \gamma \\ r \sin \gamma \end{pmatrix}$$

egyenletből a következő összefüggéseket kapjuk:

$$\cos \gamma = \frac{u_1}{r};$$

$$\sin \gamma = \frac{u_2}{r}.$$

Ezzel a $\cos \gamma$ és a $\sin \gamma$ az \mathbf{r} vektor egyenletéből kiküszöbölhető és azt kapjuk, hogy:

$$\mathbf{r}' = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} u_1 \cos \delta + u_2 \sin \delta \\ u_1 \sin \delta + u_2 \cos \delta \end{pmatrix}$$

Mivel az \mathbf{U} vektor z -komponense nem változott meg, \mathbf{U} -ra felírható:

$$\mathbf{U} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} u_1 \cos \delta - u_2 \sin \delta \\ u_1 \sin \delta + u_2 \cos \delta \\ u_3 \end{pmatrix}. \quad (*)$$

Egy mátrix segítségével az \mathbf{U}' vektor mint az \mathbf{U} vektor transzformáltja a következő formában írható fel:

$$\mathbf{U}' = \mathbf{R} \cdot \mathbf{u}$$

Koordinátákkal kifejezve:

$$\begin{pmatrix} u'_1 \\ u'_2 \\ u'_3 \end{pmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}.$$

Összevetve a (*) egyenlettel, azt kapjuk tehát:

$$\mathbf{R}_\delta = \begin{bmatrix} \cos \delta & -\sin \delta & 0 \\ \sin \delta & \cos \delta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ugyanígy járunk el akkor, ha az \mathbf{U} vektort az x -, ill. az y -tengely körül akarjuk elforgatni. Így kapjuk meg a 3 forgatásmátrixot (a z -tengely körüli elforgatás szögét ezután δ helyett γ -val jelöljük).

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix};$$

$$\mathbf{R}_\beta = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix};$$

$$\mathbf{R}_\gamma = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Mint ez könnyen ellenőrizhető, minden egyes mátrixra igaz az $\mathbf{R}^{-1} = {}^t\mathbf{R}$ összefüggés, azaz a mátrixok ortogonálisak

Szorzással kapjuk, hogy

$$\mathbf{R} = \mathbf{R}_\alpha \cdot \mathbf{R}_\beta \cdot \mathbf{R}_\gamma;$$

$$\mathbf{R} = \begin{bmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma & \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma & -\sin \alpha \cos \beta \\ \sin \alpha \sin \gamma - \cos \alpha \sin \beta \cos \gamma & \sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma & \cos \alpha \cos \beta \end{bmatrix}$$

Ismert α , β , γ esetében nem okoz már problémát, hogy egy testet tetszőlegesen elforgatott állapotban ábrázoljuk, mert minden egyes helyvektorára nézve fennáll az

$$\mathbf{U}' = \mathbf{R} \cdot \mathbf{U}$$

összefüggés. (Transzformációk egymásutánja ui. helyettesíthető ezzel az egyetlen transzformációval, amelynek mátrixa a komponensek mátrixának megfelelő sorrendben vett szorzata. Itt a térbeli forgatást a koordináta-rendszer 3 tengelye körüli egy-egy forgatás egymásutánjaként kezeljük.)

Eközben — mellékesen — a kétdimenzióban való forgatás rotációs mátrixát is meghatároztuk. Hagyjuk el az \mathbf{R} mátrix z -komponensét, azaz a 7. sort és a 3. oszlopot, így megkapjuk az R^2 -beli forgatás \mathbf{R}_γ rotációs mátrixát:

$$\mathbf{R}_\gamma = \begin{bmatrix} \cos \delta & -\sin \delta \\ \sin \delta & \cos \delta \end{bmatrix}.$$

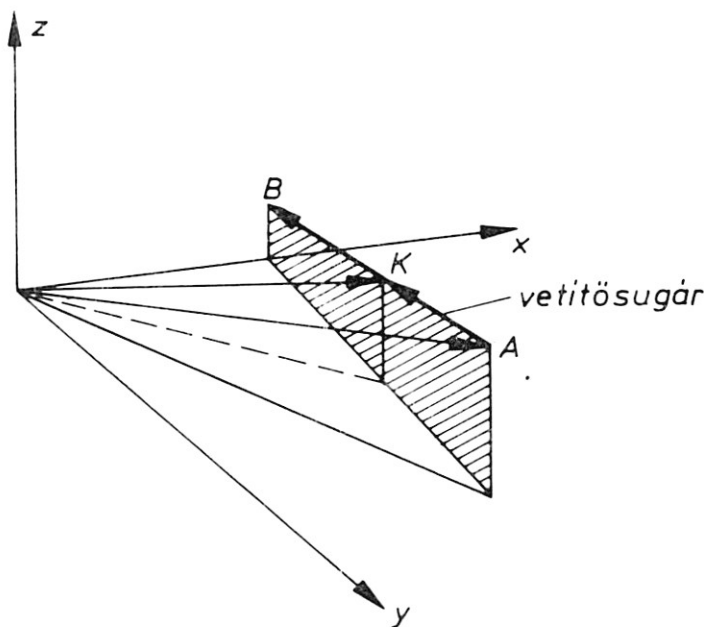
2.9 Vetítési típusok

Még megoldatlan problémánk a test vetítése a képernyőre. A vetítéseknek két alapvetően különböző fajtáját kell megkülönböztetnünk: a középpontos és a párhuzamos vetítést.

Ebben a könyvben gyakorlatilag csak a középpontos vetítést és a párhuzamos vetítés egy speciális esetét tekintjük át. A középpontos vetítés nagyon közel áll az emberi szem látásmódjához, míg a párhuzamos vetítés a vetítés tisztán matematikai formája. A párhuzamos vetítés mégis oly nagy kedveltsége azzal magyarázható, hogy általában kisebb, speciális esetekben pedig számottevően kisebb a számolási igénye, mint a középpontos vetítésé. Elég sok pont vetítésekor ez a számolási időn észrevehető.

2.9.1 KÖZÉPPONTOS VETÍTÉS

Vegyük a test egy K pontját. A középpontos vetítés jellemzője, hogy egy A nézőpontból kiindulva a test K pontját egy meghatározott síkra kivetítjük. Teljesen megfelel valamelyik koordinátasík, mert a rendszer alkalmas elforgatásával bármely adott sík átvihető a vizsgált koordinátasíkba.



A B képpontot úgy kapjuk meg, hogy az egyenest, amit az A és a K pontok határoznak meg, metszük az $x-y$ síkkal. A g egyenes a

$$g: \mathbf{X} = \mathbf{A} + k(\mathbf{K} - \mathbf{A}) = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + k \begin{pmatrix} k_1 - a_1 \\ k_1 - a_2 \\ k_3 - a_3 \end{pmatrix}$$

egyenlettel van megadva.

Az $x-z$ sík nemparaméteres alakja:

$$E: x_2 = 0.$$

(Megjegyzés: $x_1 = x$; $x_2 = y$; $x_3 = z$ helyettesítéssel.)

A metszéspontot úgy kapjuk meg, hogy az x -et behelyettesítjük az E sík egyenletébe.

$$a_2 + k(k_2 - a_2) = 0.$$

Ebből k -ra a következő kifejezést kapjuk:

$$k = \frac{-a_2}{k_2 - a_2}$$

Behelyettesítve az egyenes egyenletébe, megkapjuk a dőféspont koordinátáit:

$$b_1 = a_1 - \frac{a_2}{k_2 - a_2} (k_1 - a_1);$$

$$b_2 = 0;$$

$$b_3 = a_3 - \frac{a_2}{k_2 - a_2} (k_3 - a_3).$$

Átalakítás után:

$$b_1 = \frac{a_1 k_2 - a_2 a_1 - a_2 k_1 + a_2 a_1}{k_2 - a_2} = \frac{a_1 k_2 - a_2 k_1}{k_2 - a_2};$$

$$b_2 = 0;$$

$$b_3 = \frac{a_3 k_2 - a_2 a_3 - a_2 k_3 + a_3 a_2}{k_2 - a_2} = \frac{a_3 k_2 - a_2 k_3}{k_2 - a_2}.$$

Ezzel megkapjuk a vetítés mátrixegyenletét:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \frac{1}{k_2 - a_2} \begin{bmatrix} -a_2 & a_1 & 0 \\ 0 & 0 & 0 \\ 0 & a_3 & -a_2 \end{bmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}.$$

2.9.2 PÁRHUZAMOS VETÍTÉS

A párhuzamos vetítés abban különbözik a középpontos vetítéstől, hogy nem kell minden esetben a $\mathbf{K}-\mathbf{A}$ vetítővektort újból kiszámítanunk, mert mindig ugyanazt az \mathbf{u} vektort használjuk. Egyébként ugyanazok az összefüggések érvényesek, mint a középpontos vetítésre.

$$g: \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{K} + k\mathbf{u} = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix} + k \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

$$E: x_2 = 0$$

Az egyenesnek a síkkal való metszésével ezúttal azt kapjuk, hogy:

$$k_2 + ku_2 = 0,$$

és ebből

$$k = \frac{-k_2}{u_2}.$$

Behelyettesítve az egyenes egyenletébe, adódik:

$$b_1 = k_1 - \frac{k_2}{u_2} u_1;$$

$$b_2 = 0;$$

$$b_3 = k_3 - \frac{k_2}{u_2} u_3.$$

Átalakítás után megkapjuk a mátrixegyenletet:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \frac{1}{u_2} \begin{bmatrix} u_2 & -u_1 & 0 \\ 0 & 0 & 0 \\ 0 & -u_3 & u_3 \end{bmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}.$$

Speciális esetben nagyon egyszerű lesz ez a vetítési képlet, és a számolási igény pontosan nullára zsugorodik. Ha u_i az x_2 -tengely irányvektorát, az

$$\mathbf{u} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

vektort használjuk vetítővektorként, akkor — mint az a behelyettesítéssel könnyen igazolható — azt kapjuk, hogy:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} k_1 \\ 0 \\ k_3 \end{pmatrix}$$

Általánosan érvényes a következő: ha vetítővektorként egy koordinátatengely irányvektorát vagy annak többszörösét alkalmazzuk, akkor egyszerűen azáltal kaphatjuk meg a másik két koordinátatengely síkjában a párhuzamos vetületet, hogy elhagyjuk a pont azon komponenseit, amelyek értéke a koordinátasík nemparaméteres alakjában nulla lesz.

2.9.3 A ROTÁCIÓS MÁTRIXOK ÉS A KÖZÉPPONTOS VETÍTÉS ALKALMAZÁSAI

A következő két program egy kockát, ill. egy házat rajzol meg, tetszőlegesen elforgatott állapotban (l. az ábrákat). A *Kocka forgatása* programban az α , β , γ szögek értékét, amelyekkel a kockát elforgatjuk, közvetlenül is beadhatjuk. A *Ház perspektivikus képe* program esetében ugyanezt a billentyűzet segítségével tudjuk elérni. A program kezeléséhez szükséges tudnivalók a program indításakor automatikusan megjelennek.

FIGYELEM: *a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét FELTÉTLENŰL el kell olvasni!*

```

50 REM ... P8 ...
100 REM ..... Kocka forgatasa
110 DIM M(3,3):M0=30:X0=160:Y0=100:A(1)=.5:
    A(2)=2:A(3)=.8:PI=3.1419
120 CLS:SCREEN 2:PRINT "Kocka forgatasa"
130 INPUT "W1,W2,W3 (FOK)";W1,W2,W3
140 IF W1>=360 THEN W1=W1-(INT(W1/360)*360)
150 IF W2>=360 THEN W2=W2-(INT(W2/360)*360)
160 IF W3>=360 THEN W3=W3-(INT(W3/360)*360)
170 CLS:PRINT "Kocka forgatasa"
180 PRINT "W1=";W1
190 PRINT "W2=";W2
200 PRINT "W3=";W3
210 T=PI/180:W1=W1*T:W2=W2*T:W3=W3*T
220 M(1,1)=COS(W2)*COS(W3)
230 M(2,1)=-COS(W2)*SIN(W3)
240 M(3,1)=SIN(W2)
250 M(1,2)=COS(W1)*SIN(W3)+SIN(W1)*SIN(W2)*COS(W3)
260 M(2,2)=COS(W1)*COS(W3)-SIN(W1)*SIN(W2)*SIN(W3)
270 M(3,2)=-SIN(W1)*COS(W2)
280 M(1,3)=SIN(W1)*SIN(W3)-COS(W1)*SIN(W2)*COS(W3)
290 M(2,3)=SIN(W1)*COS(W3)+COS(W1)*SIN(W2)*SIN(W3)
300 M(3,3)=COS(W1)*COS(W2)
310 FOR I=0 TO 7:READ X(I),Y(I),Z(I):NEXT
320 FOR I=0 TO 7
330 GOSUB 450
340 X1=M0*X(I):Z1=M0*Z(I)
350 XB(I)=X0+X1:YB(I)=Y0+Z1:PSET (2*XB(I),YB(I))
360 NEXT:XB(B)=XB(4):YB(B)=YB(4)

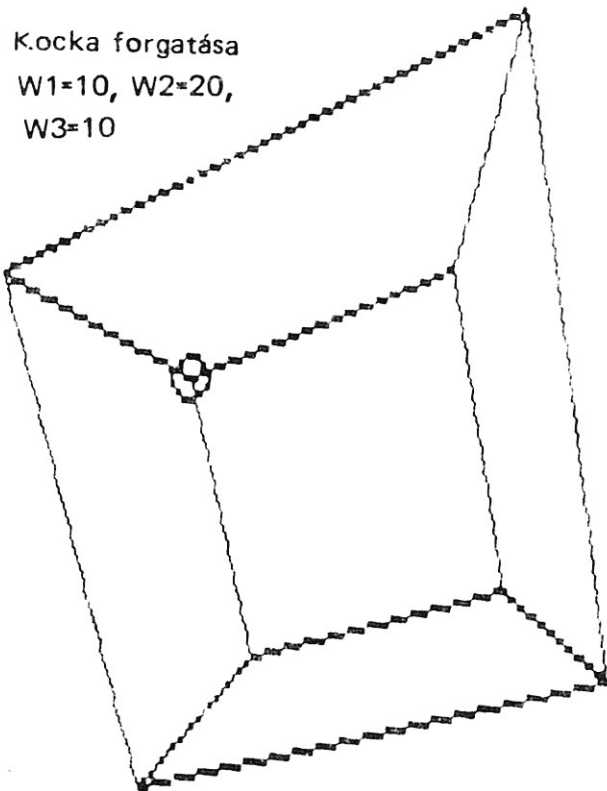
```

```

370 CIRCLE(2*XB(0),YB(0)),6:PSET (2*XB(0),YB(0))
380 FOR I=1 TO 4:LINE -(2*XB(I AND 3),YB(I AND 3)):
NEXT
390 PSET (2*XB(4),YB(4)):FOR I=5 TO 8:
LINE -(2*XB(I),YB(I)):NEXT
400 FOR I=0 TO 4:
LINE (2*XB(I),YB(I))-(2*XB(I+4),YB(I+4)):NEXT
410 DATA 0,0,0,1,0,0,1,1,0,0,1,0,0,0,1,1,0,1,1,1,
1,0,1,1
420 RESTORE
430 A$=INKEY$:IF A$="" THEN 430
440 GOTO 120
450 XA(I)=M(1,1)*X(I)+M(2,1)*Y(I)+M(3,1)*Z(I)
460 YA(I)=M(1,2)*X(I)+M(2,2)*Y(I)+M(3,2)*Z(I)
470 ZA(I)=M(1,3)*X(I)+M(2,3)*Y(I)+M(3,3)*Z(I)
480 IF YA(I)-A(2)=0 THEN T=0:GOTO 500
490 T=YA(I)/(YA(I)-A(2))
500 X(I)=XA(I)-T*(XA(I)-A(1))
510 Z(I)=ZA(I)-T*(ZA(I)-A(3))
520 RETURN

```

Kocka forgatása
W1=10, W2=20,
W3=10



FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét *FELTÉTLENÜL* el kell olvasni!

```

50 REM ... P9 ...
100 REM ..... Haz perspektivikus kepe
110 CLS:SCREEN 0:COLOR 15
120 PRINT "Haz perspektivikus kepenek megrajzolasa"

```

```

130 PRINT "-----"
-----"
140 PRINT :PRINT "A vezerles ismertetese:"
150 PRINT "elmozdulas felfele (U)P"
160 PRINT "elmozdulas lefele (D)OWN"
170 PRINT "elmozdulas balra (1)"
180 PRINT "elmozdulas jobbra (Ø)"
190 PRINT "elfordulas balra (L)EFT"
200 PRINT "elfordulas jobbra (R)IGHT"
210 PRINT :HA FELKESZULT, KEREM NYOMJON MEG
EGY GOMBOT."
220 A$=INKEY$:IF A$="" THEN 220
230 DIM M(3,3):MØ=12:XØ=160:YØ=100:A(1)=2.5:A(2)=10:
A(3)=2.5:PI=3.1419
240 DIM X(46),Y(46),Z(46),XB(46),YB(46),XA(46),YA(46),
ZA(46)
250 CLS:SCREEN 2:PRINT "Haz perspektivikus kepe"
260 INPUT "W1,W2,W3 (FOK)";W1,W2,W3
270 IF W1>=360 THEN W1=W1-(INT(W1/360)*360)
280 IF W2>=360 THEN W2=W2-(INT(W2/360)*360)
290 IF W3>=360 THEN W3=W3-(INT(W3/360)*360)
300 CLS:PRINT "Haz perspektivikus kepe"
310 PRINT "W1=";W1
320 PRINT "W2=";W2
330 PRINT "W3=";W3
340 T=PI/180:W1=W1*T:W2=W2*T:W3=W3*T
350 M(1,1)=COS(W2)*COS(W3)
360 M(2,1)=-COS(W2)*SIN(W3)
370 M(3,1)=SIN(W2)
380 M(1,2)=COS(W1)*SIN(W3)+SIN(W1)*SIN(W2)*COS(W3)
390 M(2,2)=COS(W1)*COS(W3)-SIN(W1)*SIN(W2)*SIN(W3)
400 M(3,2)=-SIN(W1)*COS(W2)
410 M(1,3)=SIN(W1)*SIN(W3)-COS(W1)*SIN(W2)*COS(W3)
420 M(2,3)=SIN(W1)*COS(W3)+COS(W1)*SIN(W2)*SIN(W3)
430 M(3,3)=COS(W1)*COS(W2)
440 FOR I=Ø TO 45:READ X(I),Y(I),Z(I):NEXT
450 FOR I=Ø TO 45
460 GOSUB 630
470 X1=MØ*X(I):Z1=MØ*Z(I)
480 XB(I)=XØ+X1:YB(I)=YØ-Z1:PSET (2*XB(I),YB(I))
490 NEXT
500 PSET (2*XB(Ø),YB(Ø)):FOR I=1 TO 20:
LINE -(2*XB(I),YB(I)):NEXT
510 PSET (2*XB(21),YB(21)):FOR I=22 TO 25:
LINE -(2*XB(I),YB(I)):NEXT
520 PSET (2*XB(26),YB(26)):FOR I=27 TO 40:
LINE -(2*XB(I),YB(I)):NEXT
530 PSET (2*XB(41),YB(41)):FOR I=42 TO 45:
LINE -(2*XB(I),YB(I)):NEXT

```

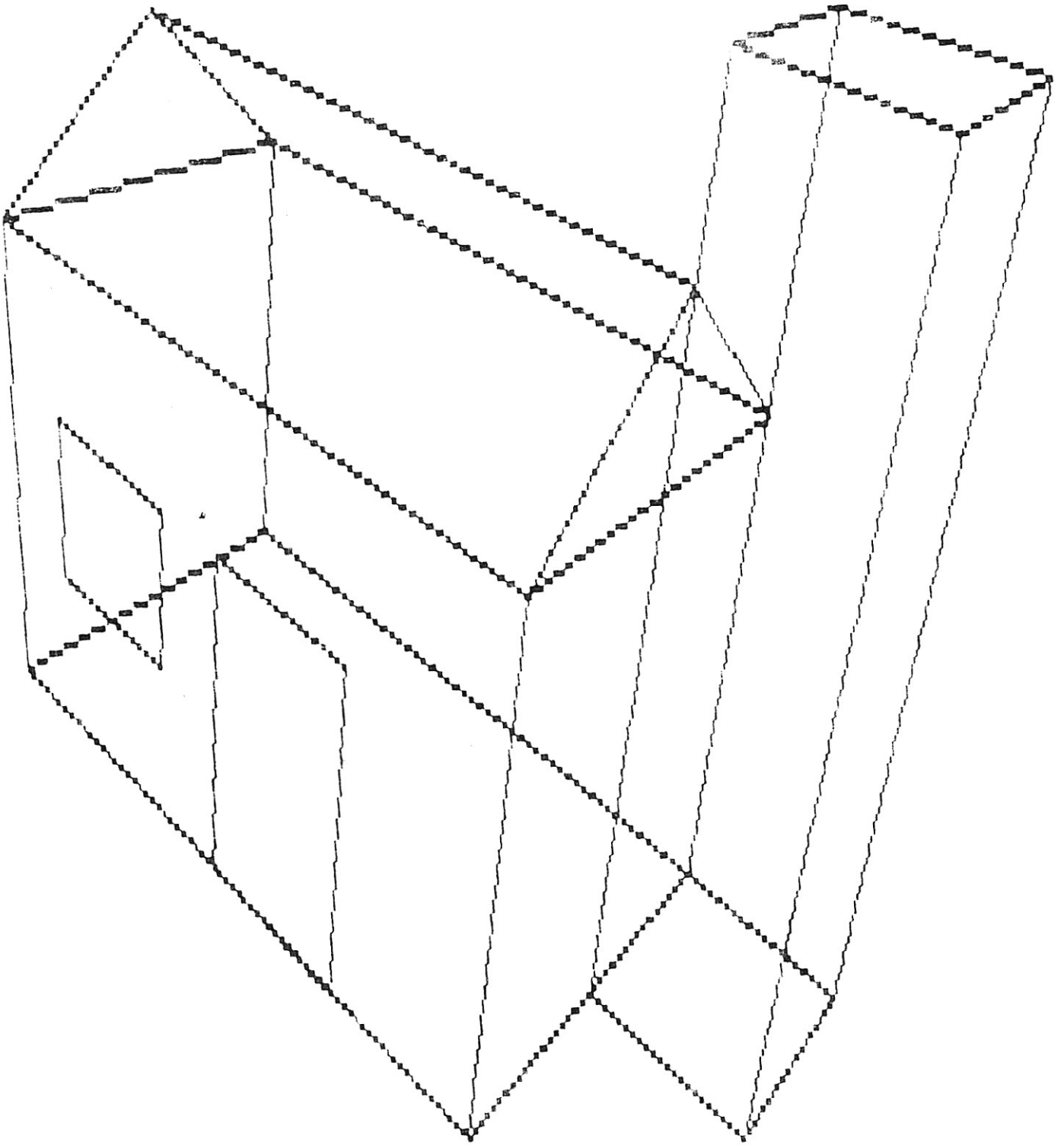
```

540 RESTORE
550 A$=INKEY$: IF A$="" THEN 550
560 IF A$="D" OR A$="d" THEN W1=W1-.1
570 IF A$="U" OR A$="u" THEN W1=W1+.1
580 IF A$="R" OR A$="r" THEN W3=W3-.1
590 IF A$="L" OR A$="l" THEN W2=W2+.1
600 IF A$="0" THEN W2=W2+.1
610 IF A$="1" THEN W2=W2-.1
620 CLS:GOTO 350
630 XA(I)=M(1,1)*X(I)+M(2,1)*Y(I)+M(3,1)*Z(I)
640 YA(I)=M(1,2)*X(I)+M(2,2)*Y(I)+M(3,2)*Z(I)
650 ZA(I)=M(1,3)*X(I)+M(2,3)*Y(I)+M(3,3)*Z(I)
660 IF YA(I)-A(2)=0 THEN T=0:GOTO 680
670 T=YA(I)/(YA(I)-A(2))
680 X(I)=XA(I)-T*(XA(I)-A(1))
690 Z(I)=ZA(I)-T*(ZA(I)-A(3))
700 RETURN
710 DATA 0,0,0
720 DATA 0,2,0
730 DATA 4,2,0
740 DATA 4,0,0
750 DATA 0,0,0
760 DATA 0,0,3
770 DATA 0,2,3
780 DATA 4,2,3
790 DATA 4,0,3
800 DATA 0,0,3
810 DATA 0,1,4
820 DATA 0,2,3,0,2,0,0,2,3
830 DATA 4,2,3,4,2,0,4,2,3
840 DATA 4,1,4
850 DATA 0,1,4
860 DATA 4,1,4
870 DATA 4,0,3
880 DATA 2,2,0
890 DATA 3,2,0
900 DATA 3,2,2
910 DATA 2,2,2
920 DATA 2,2,0
930 DATA 4,0,0
940 DATA 5,0,0
950 DATA 5,1,0
960 DATA 4,1,0
970 DATA 4,1,5
980 DATA 5,1,5,5,1,0,5,1,5
990 DATA 5,0,5,5,0,0,5,0,5
1000 DATA 4,0,5
1010 DATA 4,1,5
1020 DATA 4,0,5

```

```
1030 DATA 4,0,0
1040 DATA .5,2,1
1050 DATA 1.5,2,1
1060 DATA 1.5,2,2
1070 DATA .5,2,2
1080 DATA .5,2,1
```

Ház perspektivikus képe $W1=-22$, $W2=11$, $W3=41$

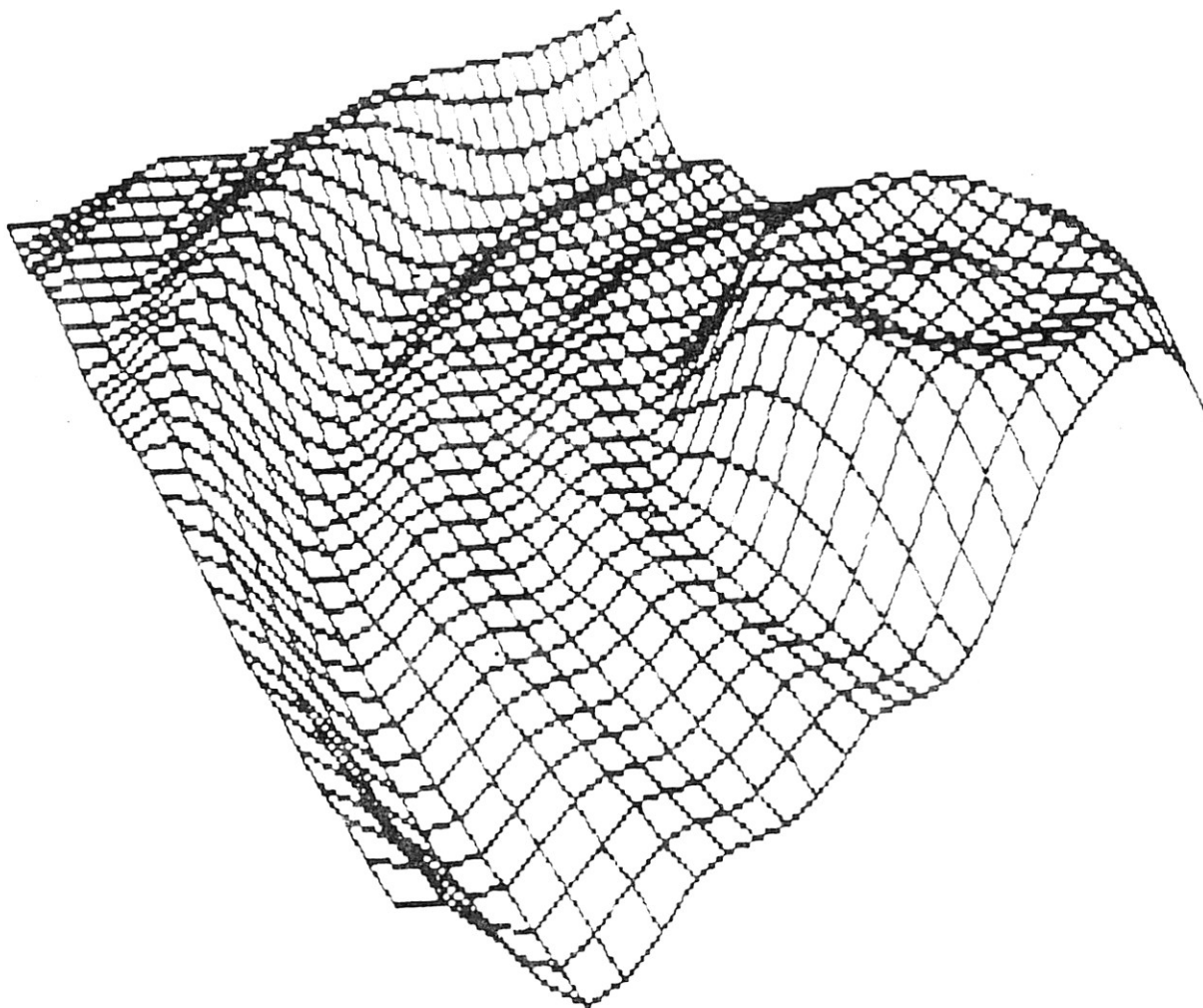


2.10 3D-grafika — első változat

2.10.1 A HÁROMDIMENZIÓS HÁLÓ

Gondolkodjunk most azon, hogyan tudunk egy gyors eljárást kidolgozni olyan 3D-grafikák készítésére, mint amelyek az előző oldalakon láthatók — amennyire ez egyáltalán a BASIC nyelvvel megoldható. Egy nagyon egyszerű, nagyon pontos, de éppen ezért nagyon lassú program található e fejezet végén.

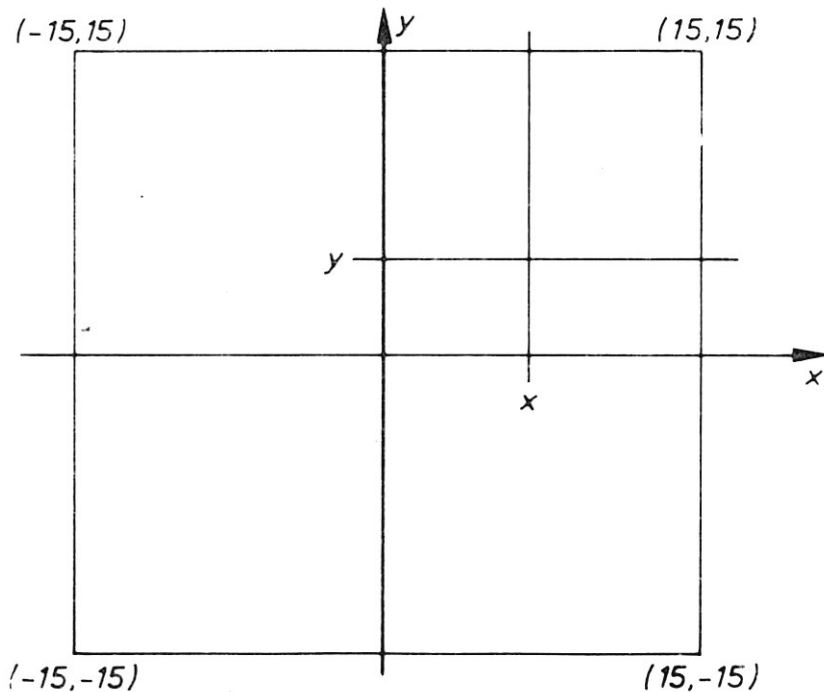
3D-V2.6 (c) 1984. Objektum: példa háromdimenziós hálóra



Felhasználjuk majd azokat az ismereteket, amelyeket a testek térbeli vetítésének és forgatásának tárgyalásakor nyertünk. Egyidejűleg arra is lehetőségünk nyílik, hogy a grafikát különböző látószögből szemlélhessük.

Ha az Olvasó egyszer utána számol, látni fogja, hogy a fejezet végén található — 3D-grafikát előállító program — körülbelül 12 000 egyedi függvényértéket számol ki. Az eljárásnak az az előnye, hogy extrém pontos, de sajnos egyben extrém nagy az időigénye is. Egy másik probléma:

gyakran egyszerűen nem állnak rendelkezésre kielégítően pontos adatok (pl. mérési adatok). A módszer hátrányainak kiküszöbölésére az ötlet a következő: nem számítjuk ki a 12 000 függvényértéket, hanem csupán 961 támpontot (ezek számítását majd látni fogjuk) és interpolációval határozzuk meg a köztük fekvő függvényértékeket.



Kiszámítjuk a függvényértékeket az (x, y) helyen, amelynél az x és az y minden -15 és 15 közé eső egész értéket felvesz. Tehát gyakorlatilag egy raszterrácsot helyezünk a függvényünkre, amikor a $31 \cdot 31 = 961$ függvényértéket kiszámítjuk. A hiányzó függvényértékeket az egyes raszterpontokat összekötő egyenesekkel közelítjük. Mint a háromdimenziós hálórajzon látható, ezzel a módszerrel az adott függvény nagyon jó lineáris közelítését kapjuk. A kis pontatlansággal szemben a mindössze $10 \dots 15$ min időigény áll, ami — ha figyelembe vesszük, hogy a 2.11 alfejezetben bemutatásra kerülő eljárásnak ugyanezen függvény megrajzolásához több mint 3 órára van szüksége — egyértelműen e módszer mellett szól. Ezzel szemben hátrány a nagy tárkapacitás-igény a kivetített pontok tárolására:

- 1) 21×21 mező az x -koordináták;
- 2) 21×21 mező az y -koordináták

számára.

A függvényértékek kiszámításának módja az Olvasóra van bízva. A program összes előnye akkor látszik, ha már kész függvényértékeink vannak. A függvényt definiálhatjuk vagy a program végén (RETURN-nel lezárva), vagy ami még jobb, önálló programmal, ami a függvényértékeket szekvenciális file formájában tárolja lemezen. Az adatokat ez esetben mindig a rajzolóprogram indításakor olvassuk be.

A függvényértékeket kiszámoló és lemezre író program ilyen lehet:

```
CLS: INPUT „lemezegység: név.kód””; NAS
OPEN „0”, 1, NAS
INPUT „osztó”; TE: WRITE # 1, TE
FOR Y = -15 TO 15: FOR X = -15 TO 15
```

FUNKTION Z = F(X, Y)

WRITE #1, Z
NEXT X: NEXT Y
CLOSE 1

Az INPUT utasításban az „osztó” jelentésére később még visszatérünk. Ismert pontok vetítésénél az adatok hívásának időigénye mindig egyenlő, kb. 3 min. Maga a rajzolás további időt, kb. 8 min-t igényel. (Az időadatok gépfüggők azt azonban jól szemléltetik, hogy a módszer megválasztása döntő fontosságú lehet.) Bonyolultabb függvényeknél ez valóban rendkívüli előny, hiszen a függvényértékek kiszámításához szükséges idő már eleve nagy lehet. Ezenkívül sokkal jobban alkalmazható ez az eljárás olyan mérési eredmények feldolgozására, amelyek gyakorlatilag mindig szekvenciális file formájában állnak rendelkezésre. Erre mutatunk be majd egy példát a 2.10.3 pontban.

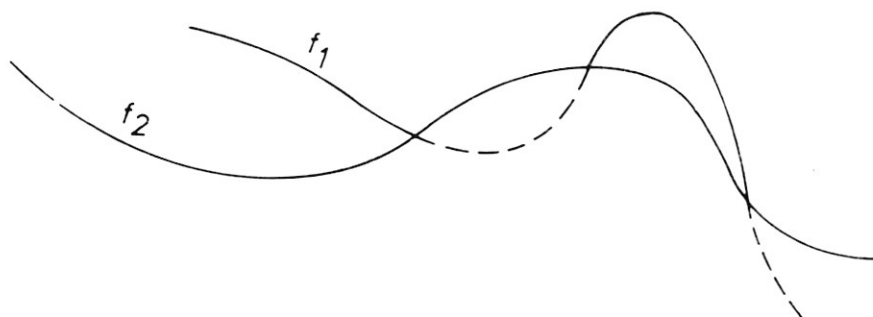
(Megjegyzés: 20-ad részére csökken a számolási idő, ha a BASIC helyett olyan nyelven írjuk a programot, amelyhez fordítóprogram — compiler, nem pedig interpreter — áll rendelkezésre.)

A forgatásról és a vetítésről nincs mit mondani, hiszen ezek a programnak olyan részei, amelyeket már ismerünk. Az y -tengely körüli forgatás esetét a rotációs mátrix tárgyalásakor mellőzzük, mert ez a forgatás az ilyen jellegű háromdimenziós grafikáknál felesleges. Az ily módon egyszerűsített rotációs mátrix a 650—730-as sorokban található:

$$R_{\alpha, \beta} = \begin{bmatrix} \cos \alpha & -\sin \beta & 0 \\ \cos \alpha \sin \beta & \cos \alpha \cos \beta & -\sin \alpha \\ \sin \alpha \sin \beta & \sin \alpha \cos \beta & \cos \alpha \end{bmatrix}$$

Tárgyalásunkat a következőkben a rajzolóprogramra korlátozzuk, mert ebben található a számunkra érdekes programrészek. Az eddig vizsgált forgatásoknál és vetítéseknél mindig lemondunk arról, hogy az eltakart vonalakat eltüntessük, mivel ez általában némi többlétráfordítással jár. (E probléma alapos ismertetését l. később a 2.12 alfejezetben.) A függvények háromdimenziós ábrázolásánál kívánatos az eltakart vonalak eltüntetése, mert egyébként az ábra hamar felismerhetetlenné válik.

Nézzük meg a következő egyszerű rajzot:



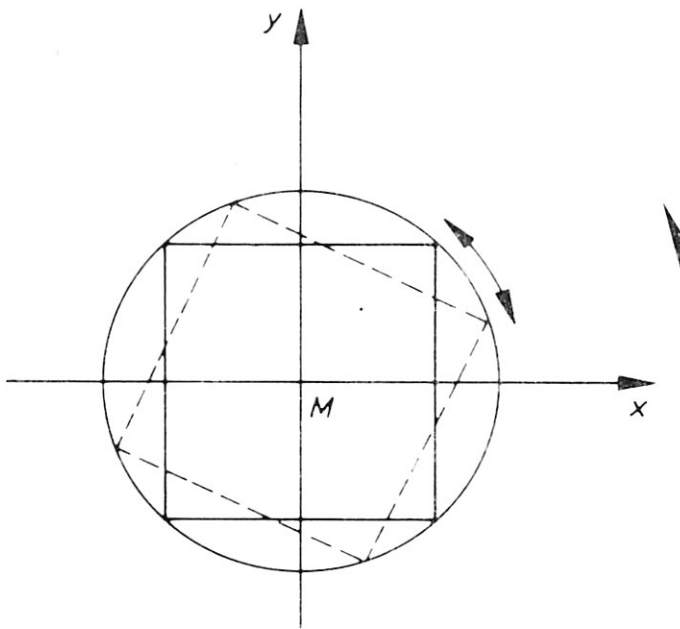
Tételezzük fel, hogy egy térbeli függvény ábrázolásáról van szó, és f_1 az f_2 mögött van. Ekkor f_1 meghatározott részei nem láthatók. A feladatunk az, hogy meghatározzuk az f_1 függvény nem látható részeit. Feltételezzük, hogy az f_2 függvényt már felrajzoltuk és most kezdjük — balról jobbra haladva — az f_1 függvényt rajzolni, meghatározott dx lépéstávolsággal végezve a rajzolást.

A nem láthatóság (eltakartság) kritériumát a következő módon adhatjuk meg:

- 1) $f_1(x) \geq f_2(x)$ és $f_1(x) + dx \geq f_2(x) + dx$;
- 2) $f_1(x) \geq f_2(x)$ és $f_1(x) + dx < f_2(x) + dx$;
- 3) $f_1(x) < f_2(x)$ és $f_1(x) + dx < f_2(x) + dx$;
- 4) $f_1(x) < f_2(x)$ és $f_1(x) + dx \geq f_2(x) + dx$.

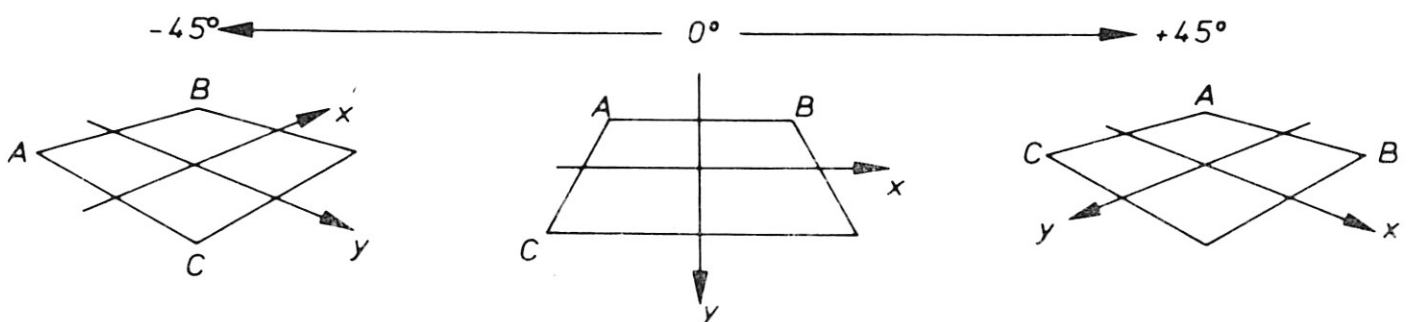
Az első esetben az $(x; x + dx)$ tartományban az f_1 függvény végig látható, mert abból indultunk ki, hogy dx olyan kicsi, hogy a dx tartományban mind az f_1 , mind az f_2 függvény közelítőleg lineáris. A második esetben az $f_1(x)$ látható, ezzel szemben az $f_1(x + dx)$ már nem. Az f_2 és az f_1 az $(x; x + dx)$ tartományban metszik egymást. A harmadik esetben az $(x; x + dx)$ tartományban f_1 nem látható. Végül a negyedik esetben az $f_1(x)$ nem látható, az $f_1(x + dx)$ viszont látható. Arra kell ügyelni, hogy az f_2 függvényt az f_1 előtt rajzoljuk és ekkor az eltakart vonalakat — a takarás kritériuma szerint — egyszerűen nem rajzoljuk meg.

Hogyan néz ki az eljárás programtechnikai megoldása? Mivel négyszög alakú ablakunk középpontja a koordináta-rendszer $(0,0)$ pontjával azonos, így annak helyzetét nem változtatja meg a forgatás.



Ha először csak a z -tengely körüli, $+45^\circ$ és -45° közötti forgatást engedjük meg, akkor kiindulhatunk a következő rajzból (a korlátozás az eltakart vonalak láthatatlanná tétele miatt szükséges):

$$\begin{aligned} A &= (-15, -15) \\ B &= (15, -15) \\ C &= (-15, 15) \end{aligned}$$

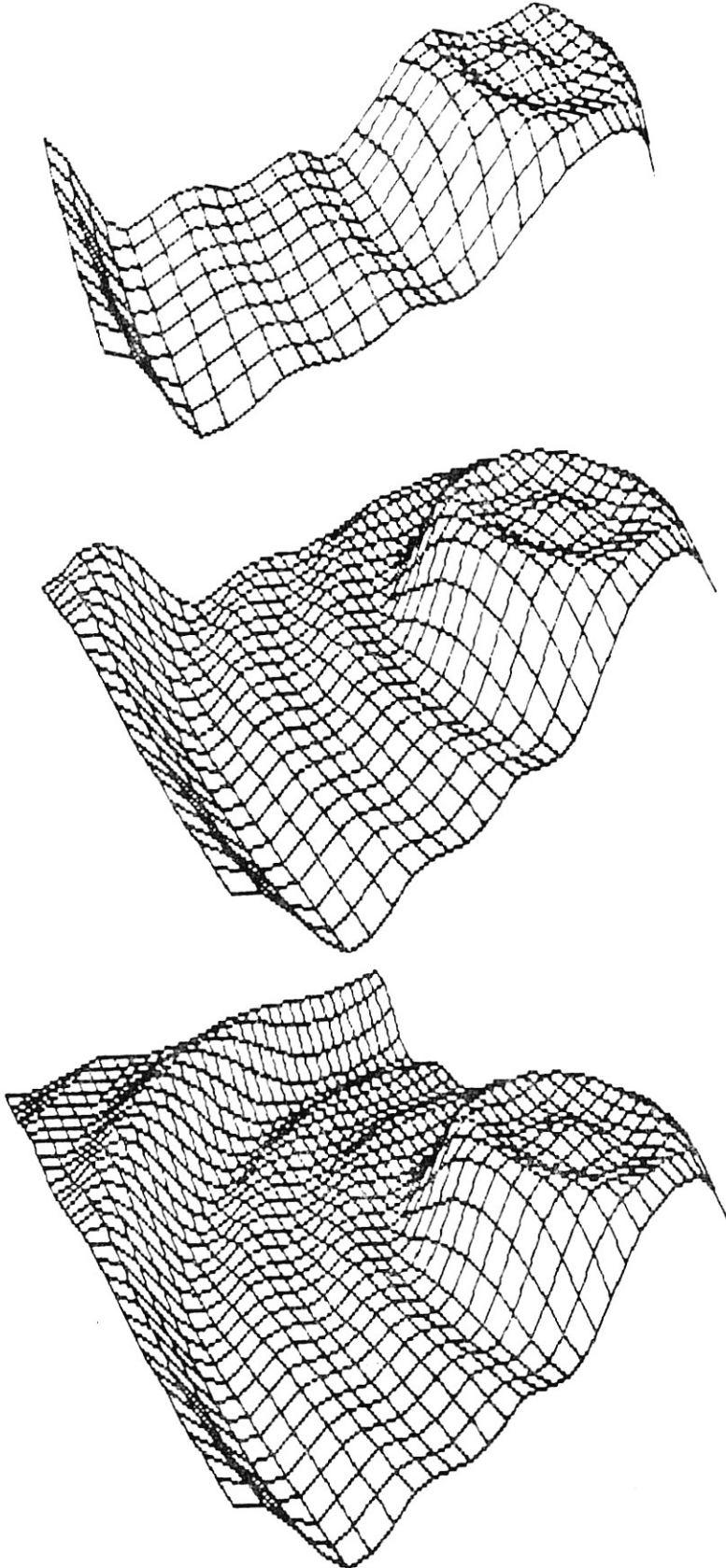


Különböztessünk meg két esetet: $\varphi > 0$ és $\varphi \leq 0$.

A $\varphi \leq 0$ esetben a vonalakat a következő sorrendben rajzoljuk meg: először az y -értékek fússanak végig a $+15$ -től a -15 -ig terjedő tartományon és azután rajzoljuk meg az egyes y -értékekhez tartozó, a -15 -től a $+15$ -ig változó x -értékek vonalát.

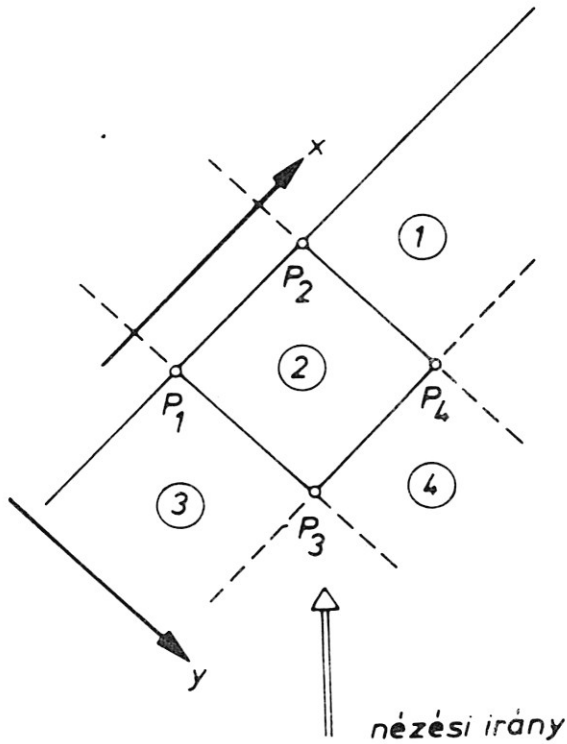
A jobb érthetőség kedvéért nézzük meg a következő képet, ahol egy ábra felrajzolása során készített három pillanatfelvétel látható.

3D-V2.6 (c) 1984. Objektum: példa 3D-grafika készítésére



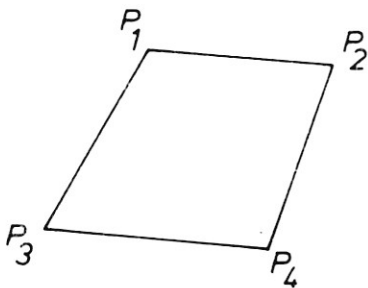
Térjünk rá a rajzolósi folyamatra!

A hálónak mindig egy négyzetét rajzoljuk meg. Tekintsük a következő ábrát:



A rácsot felülről nézve látjuk, a rács sarokpontjait jelölje P_1, P_2, P_3, P_4 . Ekkor a P_1 és P_3 pontok egybeesnek az előzőleg megrajzolt (3) négyzet P_2 és P_4 pontjaival. Hasonlóan egybeesnek a P_3 és a P_4 pontok a (4) négyzet P_1 és P_2 pontjaival. A P_1P_3 és a P_3P_4 vonal tehát már meg is van rajzolva.

Megrajzolásra vár még a P_1P_2 és a P_2P_4 vonal. Ezeket a vonalakat a program a 870—1050-es, ill. az 1060—1140-es sorokban rajzolja. A két rész gyakorlatilag azonos, ezért tárgyalásunkat korlátozhatjuk csupán az egyikre. Az 1030-as és az 1040-es sorban mindig azt határozzuk meg, hogy a vetítésben melyik x -érték a kisebb. Fontos, hogy előbb a P_2 és a P_4 pontot kössük össze, mert egyébként a következő fordulhat elő: tételezzük fel, hogy az ábrát a z -tengely körül csak egy nagyon kis szöggel forgattuk el, mégpedig bal oldali irányba. Ekkor egy négyzetet a perspektíva következtében a következőképpen látunk:



Ha a P_1P_2 vonalat rajzoljuk meg előbb, akkor ez azt jelenti, hogy a P_4P_2 vonal jóval a P_1P_2 vonal alá esik, a programnak tehát ezt úgy kell tekinteni, mintha egy eltakart vonal lenne. Ha előbb a P_4P_2 vonalat rajzoljuk meg, akkor ez a probléma nem fordulhat elő.

Az eltakart vonalák láthatatlanná tétele az 1350—1530-as sorokban történik. Az előbbieken leírt láthatatlansági kritériumok szerint járunk el.

A $\varphi > 0$ esetben az eljárás analóg az előzővel. A vonalakat itt a következő sorrendben rajzoljuk meg: az y -értékek végigfutnak $+15$ -től -15 -ig és minden y -értékhez megrajzoljuk a $+15$ -től a $-$ -ig terjedő x -értékekhez tartozó vonalát.

A rajzolóprogram a $\varphi \leq 0$ esetre is alkalmazható, azonban szükséges, hogy a P_1 és P_4 változókat más sorrendben definiáljuk (1070—1170-es sorok).

Az eljárást eddig $\pm 45^\circ$ -nál nem nagyobb szögű elforgatásra korlátoztuk. A korlátok közé eső tartomány megfelel egy negyedfordulatnak. A korlátozás az eltakart vonalak láthatatlanná tétele miatt volt szükséges. Minden forgatás azonban kezelhető úgy, hogy az legfeljebb $\pm 45^\circ$ -os forgatásnak feleljen meg, ha előzetesen a pontokat alkalmas módon transzformáljuk (830—870-es sorok). Szóljunk még az ún. „osztó”-ról, amelynek a szekvenciális adatfile elején kell lenni. A grafikus ábra legnagyobb koordinátájú pontjai sem eshetnek a képernyőn kívül, ezért a legnagyobb koordinátaértékek sem haladhatják meg a 8,5-et. Az osztót tehát ennek megfelelően úgy kell megválasztani, hogy teljesüljön ez a feltétel.

Programleírás

Sor	Magyarázat
200—410	Instrukciók a 3D-V2.6 kezelésére
430—480	Lekérdezés, hogy az adatok vetítésére kész állapotban rendelkezésre állnak-e, vagy a program végén még ki kell azokat számítani
490	A képet tartalmazó adatállomány neve
500—570	Definíciók
580	Lekérdezés, hogy kell-e az ábrát lemezen tárolni
630—640	Programkezdet
650—730	Rotációs mátrix
740—880	A z -koordináta meghatározása (szekvenciális file vagy számítás), vetítés, tárolás
900—950	Az első vonal megrajzolása
970—1050	Rajzolórutin, forgatási szög ≤ 0
1060—1140	Rajzolórutin, forgatási szög > 0
1150—1180	Akusztikus visszajelzés, adott esetben tárolás lemezen
1200—1260	Az új forgatási szög redukciója -45° és $+45^\circ$ közötti forgatásra
1270—1340	Szubrutinok: forgatás és vetítés
1350—1530	Szubrutinok: vonal rajzolása

FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENÜL** el kell olvasni!

```

50 REM ... P10 ...
100 CLS:SCREEN 0:KEY OFF:COLOR 15
110 PRINT "3D-V2.6  MICROSOFT-BASIC 2.0-ban az
      IBM PC-kompatibilis szamitogepekre"
120 PRINT "=====
      ====="
130 PRINT "                ----- (C) 1984 szerzo:
      M.Weber"

```

```

140 PRINT "          ====="
150 PRINT :PRINT "(1)";:COLOR 7:
    PRINT "Instrukciok a 3D-V2.6 kezelesehez"
160 COLOR 15:PRINT "(2)";:COLOR 7:PRINT "START"
170 COLOR 15:PRINT :PRINT "Kerem valasszon"
180 A$=INKEY$:IF A$="" THEN 180
190 IF A$="2" THEN 430
200 CLS:COLOR 15:PRINT "3D-V2.6  MICROSOFT-BASIC
    2.0-ban az IBM PC kompatibilis szamitogepekre"
210 PRINT "=====
    ====="
220 PRINT "          ----- (C) 1984 szerzo:
    M.Weber - Instrukciok
230 PRINT "          =====
    ====="
240 PRINT :PRINT "3D-V2.6";:COLOR 7:PRINT "Ugyanugy
    alkalmas ketvaltozos fuggvenyek szemleltetesere,"
250 PRINT "mint meresi adatok grafikus abrazolasara"
    ;:COLOR 15:PRINT "mindig elonyos"
260 PRINT "az adatokat "
270 PRINT "szekvencialis file-ban elokesziteni, akkor"
280 PRINT "a program futasa gyorsabb"
290 PRINT "Amennyiben fuggvenyt akar abrazolni,
    akkor a fuggvenyertekeket szamito"
300 PRINT "programszakaszt a program vegehez is kap-
    csolhatja":COLOR 7
310 PRINT "Ez esetben irja be ";:COLOR 15:
    PRINT "SOR 1540";: COLOR 7:PRINT "es a
    fuggvenyt a kovetkezo alakban:"
320 PRINT "Z(I)= ..... Ugyeljen ra,hogy csak a ";:
    COLOR 15:PRINT "Maximalis ertek<8.5";:COLOR 7:
    PRINT "ad hasznalhato eredmenyt."
330 PRINT "----- A definiciot
    RETURN-nel zarja le"
340 PRINT "A program a mindenkori objektumot akkor
    abrazolja< ha on ezt ohajtja< es eloszor"
350 PRINT "a standard értelmezésben, a
    W1=-.4   W3=-.62 parameterekkel"
370 PRINT "Ha befejezte, a program akusztikus
    visszajelzést ad"
380 PRINT "Ekkor a W1 es W2 forgatasi szögeket szaba-
    don megvalaszthatja. De ugyeljen arra, hogy"
390 PRINT "W1 csak -.45 es 0 kozott ad hasznalhato"
400 PRINT "eredmenyeket"
410 COLOR 15:PRINT "Kesz?"
420 A$=INKEY$:IF A$="" THEN 420
430 CLS:PRINT "Az adatok ---- szekvencialis file-ban
    ----- vannak?"
440 A$=INKEY$:IF A$="" THEN 440

```

```

450 IF A$="N" THEN 490
460 INPUT "Lemezegyseg (A:/B:)";N1$
470 INPUT "cod jel ";C0$:DIR$=N1$+"*."+C0$:
LOCATE 15,1:PRINT "Kovetkezo file-ok allnak
rendelkezesre:": FILES DIR$
480 LOCATE 4,1:PRINT "nev-----":LOCATE 4,5:
INPUT N2$:N$=N1$+N2$+C0$
490 INPUT "Az abra neve -----";NA$:
IF LEN (NA$)>48 THEN PRINT "tul hosszu:GOTO 490
500 DIM XB(30,30),YB(30,30),X(4),Y(4),Q(640)
510 FOR I=0 TO 639:Q(I)=199:NEXT I
520 DEF FNG(X)=Y(U)+(X-X(U))*(Y(0)-Y(U))/(X(0)-X(U))
530 DEF FNB(X)=Y1+(X-X1)*(Y2-Y1)/(X2-X1)
540 A(1)=0:A(2)=40:A(3)=1:PI=3.1419:W3=-.62:
W1=-.4:TS=0
550 PRINT "Standard abrazolas (I/N)"
560 A$=INKEY$:IF A$="" THEN 560
570 IF A$="N" THEN 1180
580 PRINT "---- Kell tarolni a grafikát?"
590 A$=INKEY$:IF A$="" THEN 590
600 IF A$="N" THEN 630
610 INPUT "(lemzegyseg):file-nev";SP$:SP$=SP$+".GRA"
620 DEF SEG=&HB800
630 CLS:SCREEN 2:
PRINT " 3D-V2.6 (C)1984 Objectum:"
640 LOCATE 1,33+INT(.5*(48-LEN(NA$))):PRINT NA$
650 M(1,1)=COS(W3)
660 M(2,1)=-SIN(W3)
670 M(3,1)=0
680 M(1,2)=COS(W1)*SIN(W3)
690 M(2,2)=COS(W1)*COS(W3)
700 M(3,3)=-SIN(W1)
710 M(1,3)=SIN(W1)*SIN(W3)
720 M(2,3)=SIN(W1)*COS(W3)
730 M(3,3)=COS(W1)
740 M=PI/15:AA=0
750 IF N$="" THEN 770
760 OPEN "I",1,N$:INPUT #1,OSZTO
770 FOR Y=-15 TO 15:FOR X=-15 TO 15
780 X(1)=X:Y(1)=Y
790 AA=AA+1:LOCATE 1,1:PRINT AA
800 IF N$="" THEN GOSUB 1500:GOTO 820
810 INPUT #1,Z(1):Z(1)/OSZTO
820 GOSUB 1270
830 IF TS=0 THEN XB(X+15,Y+15)=INT(320+10*XA(1)):
YB(X+15,Y+15)=INT(80-7*ZA(1))
840 IF TS=1 THEN XB(15-Y,X+15)=INT(320+10*XA(1)):
YB(15-Y,X+15)=INT(80-7*ZA(1))
850 IF TS=2 THEN XB(15-X,15-Y)=INT(320+10*XA(1)):
YB(15-X,15-Y)=INT(80-7*ZA(1))

```



```

860 IF TS=3 THEN XB(Y+15,15-X)=INT(320+10*XA(1)):
    YB(Y+15,15-X)=INT(80-7*ZA(1))
870 IF TS=4 THEN XB(15-X,15-Y)=INT(320+10*XA(1)):
    YB(15-X,15-Y)=INT(80-7*ZA(1))
880 NEXT:NEXT:CLOSE 1
890 LOCATE S=30:FOR I=1 TO 30
900 S=30:FOR R=1 TO 30
910 X1=INT(XB(R-1,S)):X2=INT(XB(R,S))
920 Y1=INT(YB(R-1,S)):Y2=INT(YB(R,S))
930 FOR X=X1 TO X2:Q(X)=FNB(X):NEXT X
940 LINE (X1,Y1)-(X2,Y2)
950 NEXT R
960 IF WT>0 THEN 1060
970 FOR S=29 TO 0 STEP -1:FOR R=1 TO 30
980 X(1)=INT(XB(R-1,S)):X(2)=INT(XB(R,S))
990 Y(1)=INT(YB(R-1,S)):Y(2)=INT(YB(R,S))
1000 X(3)=INT(XB(R-1,S+1)):X(4)=INT(XB(R,S+1))
1010 Y(3)=INT(YB(R-1,S+1)):Y(4)=INT(YB(R,S+1))
1020 IF R=1 THEN U=-1*(X(1)<=X(3))-3*(X(3)<=X(1)):
    O=4-U:GOSUB 1350
1030 U=-2*(X(2)<=X(4))-4*(X(4)<X(2)):O=6-U:GOSUB 1350
1040 U=-1*(X(1)<=X(2))-2*(X(2)<X(1)):O=3-U:GOSUB 1350
1050 NEXT R:NEXT S:GOTO 1150
1060 FOR S=29 TO 0 STEP-1:FOR R=30 TO 1 STEP-1
1070 X(2)=INT(XB(R-1,S)):X(1)=INT(XB(R,S))
1080 Y(2)=INT(YB(R-1,S)):Y(1)=INT(YB(R,S))
1090 X(4)=INT(XB(R-1,S+1)):X(3)=INT(XB(R,S+1))
1100 Y(4)=INT(YB(R-1,S+1)):Y(3)=INT(YB(R,S+1))
1110 IF R=30 THEN U=-1*(X(1)<=X(3))-3*(X(3)<X(1)):
    O=4-U:GOSUB 1350
1120 U=-2*(X(2)<=X(4))-4*(X(4)<X(2)):O=6-U:GOSUB 1350
1130 U=-1*(X(1)<=X(2))-2*(X(2)<X(1)):O=3-U:GOSUB 1350
1140 NEXT R:NEXT S
1150 BEEP:IF SP$="" THEN 1170
1160 BSAVE SP$,0,16200:SP$=""
1170 A$=INKEY$:IF A$="" THEN 1170
1180 PRINT :PRINT "W1,W3";W1,W3:BEEP:CLS:RESTORE
1190 FOR I=0 TO 639:Q(I)=199:NEXT I
1200 IF W3>=0 THEN W3=W3-2*PI*INT(.5*W3/PI):
    IF W3>PI THEN W3=-2*PI*W3
1210 IF W3<0 THEN W3=W3+2*PI*INT(-.5*W3/PI):
    IF W3<-PI THEN W3=2*PI+W3
1220 IF W3>=-PI/4 AND W3<PI/4 THEN WT=W3:TS=0:GOTO 580
1230 IF W3>=PI/4 AND W3<3*PI/P THEN WT=W3-PI/2:
    TS=1:GOTO 580
1240 IF W3>=3*PI/4 THEN WT=W3-PI:TS=2:GOTO 580
1250 IF W3>=-3*PI/4 AND W3<-PI/4 THEN WT=W3+PI/2:
    TS=3:GOTO 580
1260 IF W3<=-3*PI/4 THEN WT=W3+PI:TS=4:GOTO 580

```

```

127Ø XA(1)=M(1,1)*X(1)+M(2,1)*Y(1)+M(3,1)*Z(1)
128Ø YA(1)=M(1,2)*X(1)+M(2,2)*Y(1)+M(3,2)*Z(1)
129Ø ZA(1)=M(1,3)*X(1)+M(2,3)*Y(1)+M(3,3)*Z(1)
130Ø IF YA(1)-A(2)=Ø THEN T=Ø:GOTO 132Ø
131Ø T=YA(1)/(YA(1)-A(2))
132Ø XA(1)=XA(1)-T*(XA(1)-A(1))
133Ø ZA(1)=ZA(1)-T*(ZA(1)-A(3))
134Ø RETURN
135Ø IF INT(X(U))=INT(X(O)) THEN 146Ø
136Ø IF Y(U)<=Q(X(U)) THEN AX=X(U):AY=Y(U):A=Ø:
    GOTO 138Ø
137Ø A=1
138Ø FOR X=X(U) TO X(O):G=FNG(X)
139Ø IF G<=Q(X) AND A=Ø THEN T=Q(X):Q(X)=G:GOTO 143Ø
140Ø IF G<=Q(X) AND A=1 THEN GOSUB 150Ø:AX=SX:AY=SY:
    T=Q(X):Q(X)=G:A=Ø:GOTO 143Ø
141Ø IF IF A=Ø THEN GOSUB 150Ø:LINE (AX,AY)-(SX,SY)
142Ø A=1:T=Q(X)
143Ø NEXT X
144Ø IF A=Ø THEN LINE (X(O),Y(O))-(AX,AY)
145Ø RETURN
146Ø IF Y(U)<Q(X(U)) THEN
    LINE (X(U),Q<X(U))-(X(U),Y(U)):Q(X(U))=Y(U)
147Ø IF Y(O)<Q(X(U)) THEN
    LINE (X(U),Q<X(U))-(X(U),Y(O)):Q(X(U))=Y(O)
148Ø IF R=1 AND U=1 THEN LINE (X(1),Y(1))-(X(3),Y(3))
149Ø RETURN
150Ø PAR=(T-FNG(X-1))/(G-FNG(X-1)+T-Q(X))
151Ø SX=X-1+PAR
152Ø SY=T+PAR*(Q(X)-T)
153Ø RETURN

```

2.10.2 A HÁROMDIMENZIÓS FÜGGVÉNYEK DEFINÍCIÓJÁHOZ

Különböző lehetőségeink vannak a háromdimenziós függvények definiálására. A háromdimenziós függvény megnevezés voltaképpen hibás, mert a három dimenzió azt jelenti, hogy mind a 3 térváltozó szerepel a függvényben. Tehát az $f(x, y, z) = \dots$ típusú függvényekkel kellene foglalkoznunk. Azok a függvények azonban, amelyekkel mi foglalkozunk, $f(x, y) = \dots$ alakúak.

A definíció legegyszerűbb formája az, amikor egyváltozós függvényeket kapcsolunk össze:

Példák:

```

cos(x) + sin(y)
(1 - cos(x)) * (1 - cos(y))
exp(-x) + exp(-y)

```

Másik lehetőség: két változót azáltal kapcsolunk össze, hogy azokat a függvény argumentumában együtt szerepeltetjük.

$$\begin{aligned} & \sin(x + y) \\ & 1 - \cos(x * y) \end{aligned}$$

Nagyon gyakran alkalmazott megoldás a forgásszimmetrikus függvények felhasználása. Így pl. ha a $\sin(x)$ függvényt — azt is kikötve, hogy az x csak pozitív értéket vehet föl — a koordináta-rendszer középpontja körül körbeforgatjuk, akkor egy körhullámot kapunk.

A körbeforgatást a következő trükkel oldhatjuk meg: a függvény változója ne x vagy y legyen, hanem egy rácspont, és a forgatási középpont közötti távolságot válasszuk változónak. Jelöljük ezután p -vel ezt a távolságot, amelyet úgy kapunk meg, hogy meghatározzuk a rácspontból $(x; y)$ a forgási középpontba $(r_1; r_2)$ mutató vektor nagyságát, azaz abszolút értékét:

$$p = \sqrt{(x - r_1)^2 + (y - r_2)^2}$$

(Itt és a továbbiakban a függvény grafikonját a z -tengely, ill. az (r_1, r_2) -ben az $x - y$ síkra emelt merőleges körül forgatjuk meg.)

Ezek után a következő módon definiálhatunk egy körhullámot:

$$\begin{aligned} m &= \pi/15 \\ p &= \sqrt{x^2 + y^2} \\ z &= 1 - \cos(5mp) \end{aligned}$$

Nagyon érdekes effektusokat érhetünk el, ha felhasználjuk a burkológörbéket, a forgásszimmetrikus függvények és más függvények összekapcsolásával nyert lehetőségeket stb.

Vizsgáljuk meg közelebbről a következő oldalakon bemutatott példákat!

1. példa

A megrajzolt függvény:

$$z = 0.5 * (1 - \cos(5 * m * x * y/15))$$

Mivel az x és y változók által felvett értékek a -15 -től $+15$ -ig terjedő tartomány értékei, a legnagyobb változóértékre az $xy = \pm 225$ adódik.

Mivel $m = \pi/15$, ezért a koszinuszfüggvény argumentuma 0 -tól $\pm 5\pi$ -ig változik, így az ábrán $\pm 2,5$ periódus, kétszer $2,5$ hullám jelenik meg.

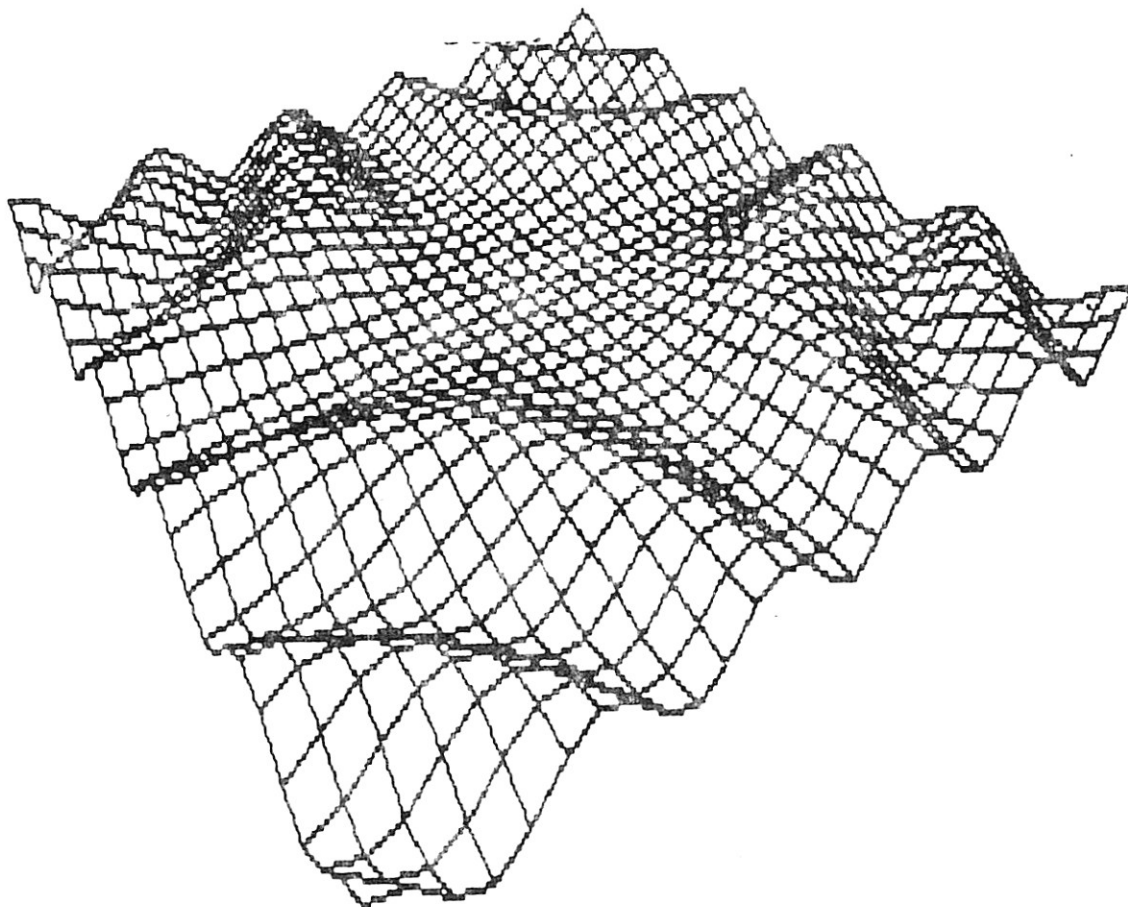
2. példa

A képen a

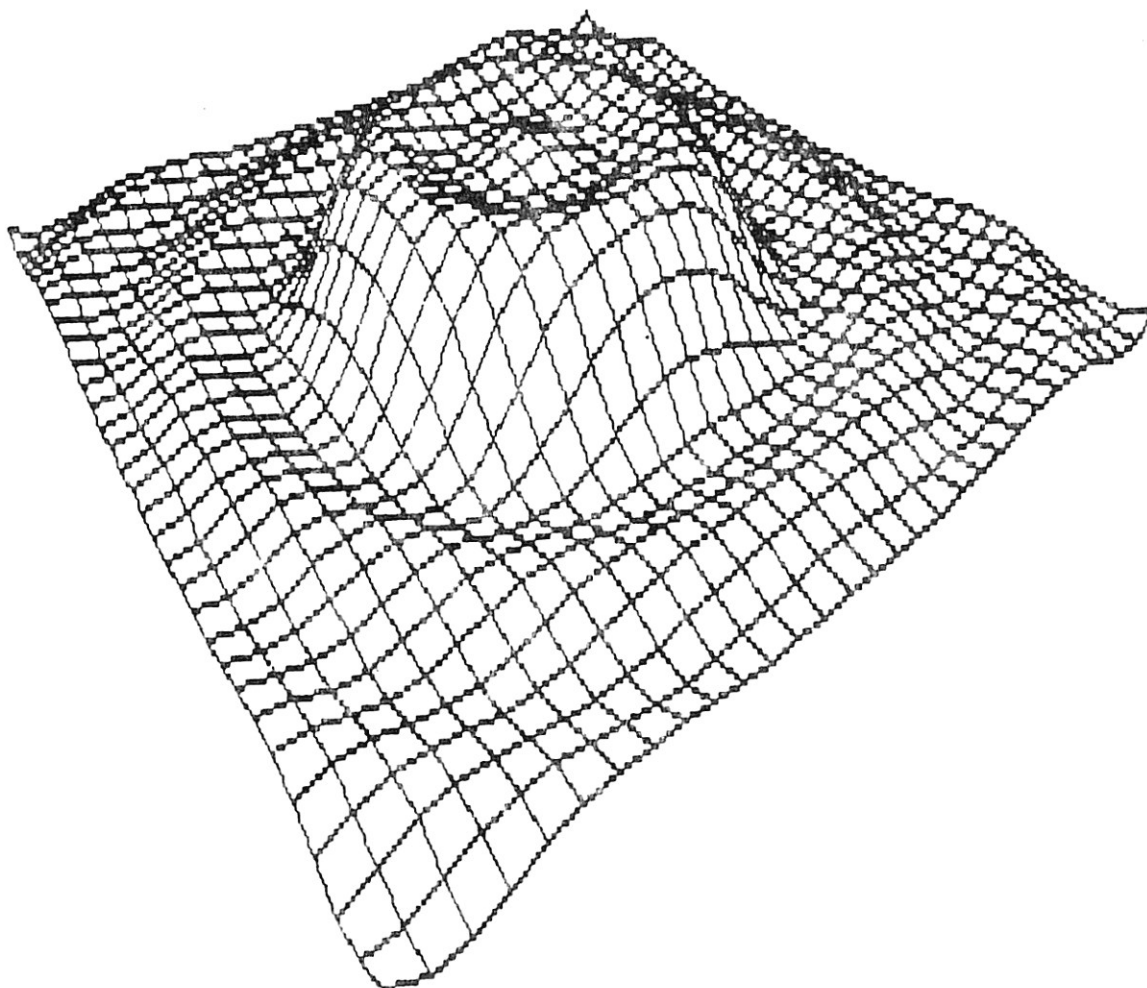
$$\begin{aligned} p &= \text{sqr}(x \wedge 2 + y \wedge 2) \\ z &= 1.4 + 2 * (\cos(m * p) - \cos(3 * m * p))/3 + \cos(5 * m * p)/5 \end{aligned}$$

függvényt ábrázoltuk.

3D-V2.6 (c) 1984. Objektum: 1. példa



3D-V2.6 (c) 1984. Objektum: 2. példa



Az első sor a függvény $(0; 0)$ pontra vonatkozó szimmetriáját fejezi ki. A második sorban felírt függvény egy négyszöghullám koszinuszfüggvénnyel való 3. közelítése (a négyszöghullám ún. *Fourier-féle sorfejtésének* egy szelete).

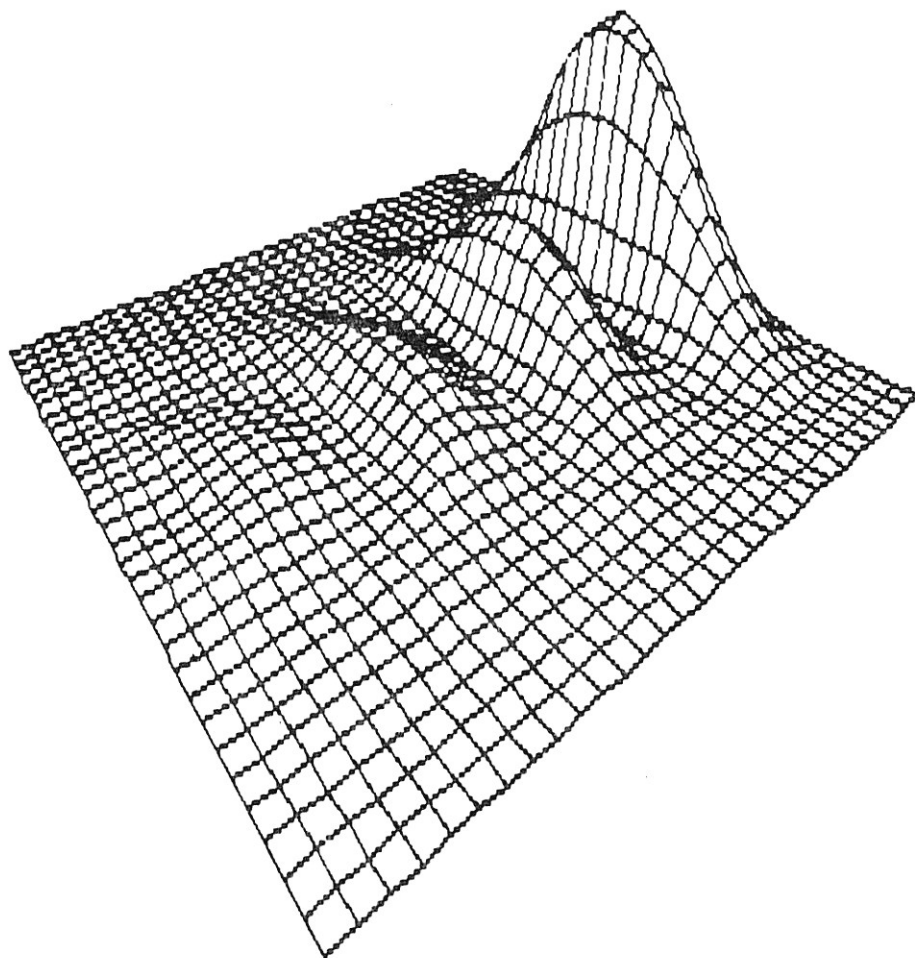
3. példa

A következő függvényt írjuk fel:

$$p = \text{sqr}((x + 13)^2 + y^2) : \text{OSZTO} = 1.224$$

$$z = (1 - \cos(4 * m * p)) * \exp(x/8) * .25 * (1 + \cos(m * y)) \wedge 2 * (1/\text{OSZTO})$$

3D-V2.6 (c) 1984. Objektum: 3. példa

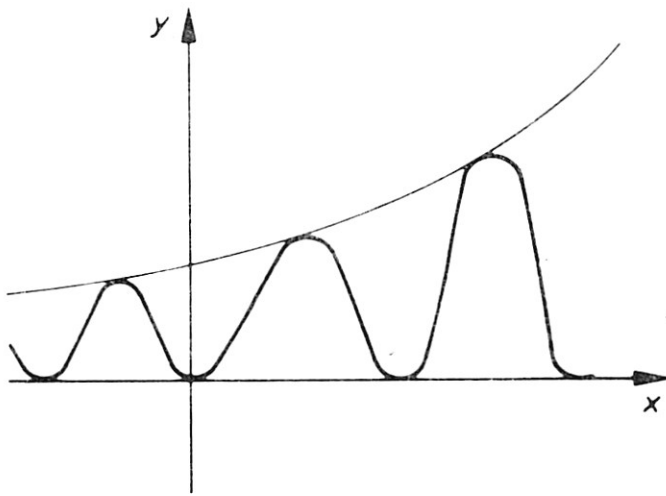


Azáltal, hogy a p értékét a

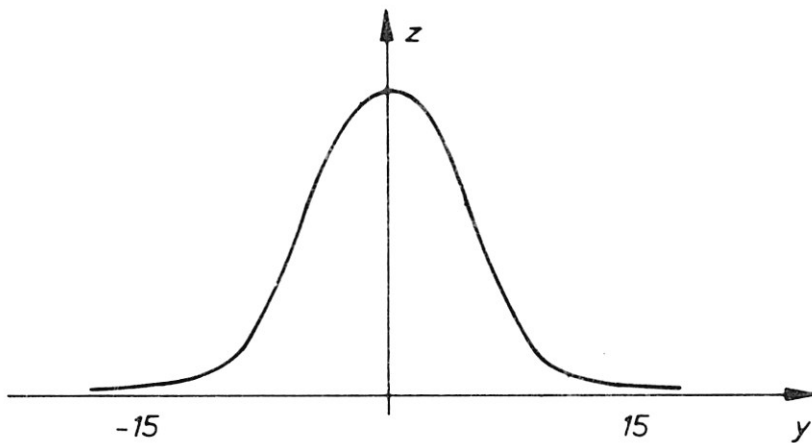
$$p = \sqrt{(x + 13)^2 + y^2}$$

formában adtuk meg, a függvénynek nem a $(0; 0)$, hanem a $(-13; 0)$ pont körüli forgatását írtuk elő. Az $(1 - \cos(4 * m * p))$ függvény által meghatározott körhullámnak a $(-13; 3)$ x -tartományában pontosan két periódusa van.

A két burkológörbével megváltoztathatjuk az elérhető legnagyobb magasságot, így a függvény menetét. Az $\exp(x/8)$ tényezővel való szorzással a maximumok balról jobbra irányú exponenciális görbét kapjuk. A görbét az $\exp(x/8)$ írja le.



Az $(1 - \cos(m * y))^2$ kifejezés által leírt burkológörbe biztosítja az ábránkat alkotó görbék y -irányú magasságának korlátosságát. A burkológörbe alakját a következő rajzon láthatjuk:



4. példa

Az ábrázolt függvény már lényegesen komplikáltabb:

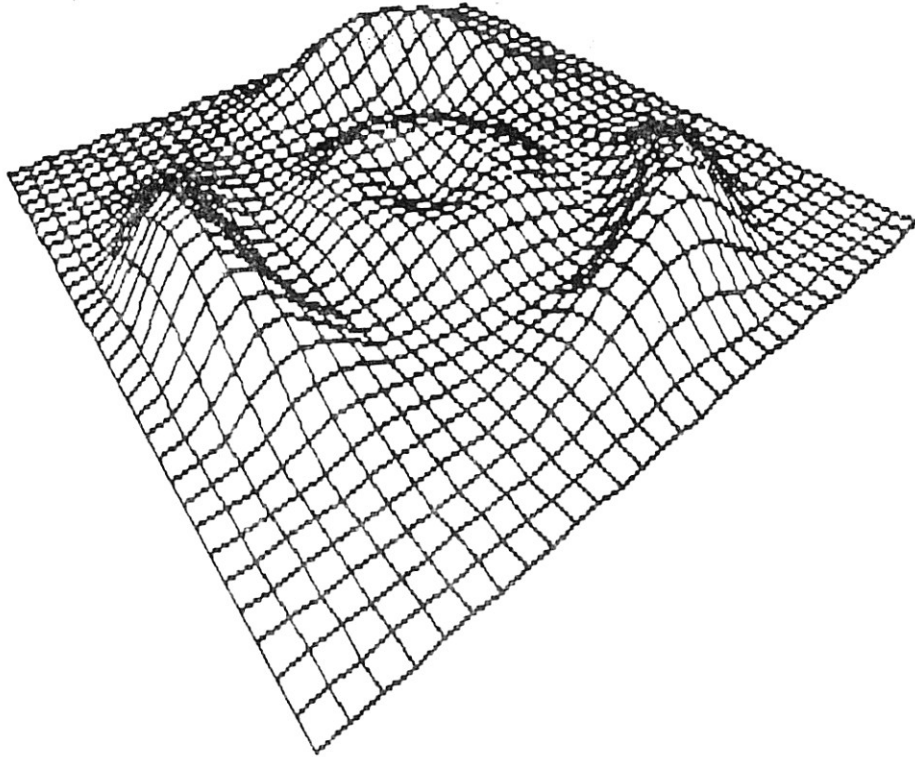
```

p = sqrt(x ^ 2 + y ^ 2): if x = 0 then phi = pi/2 -
    - pi * (y < 0): goto (*)
phi = atan(y/x) - pi * (x < 0)
(*) z = .7 * (1 - cos(4 * m * p)) * (p <= 15)
z(1) = -.8 * (1 - cos(3 * phi)) * z * (p > 7.5) - .5 * z * (p <= 5)

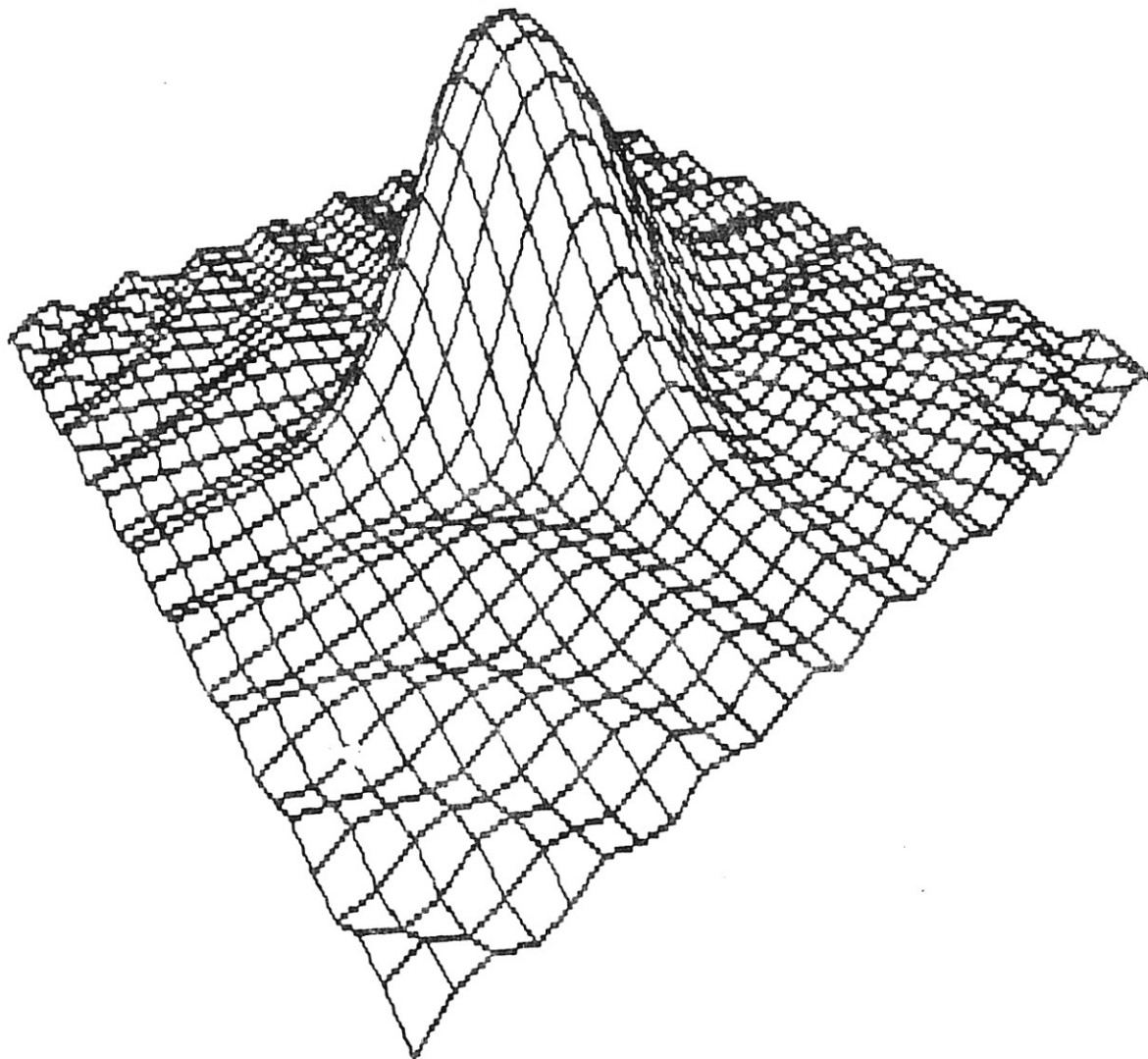
```

Alapfüggvényünk a $(0; 0)$ pont körüli körhullám $(*)$ sor). A második körgyűrűt — a koordináta-rendszerbeli helyzetének függvényében — megváltoztatjuk. Ehhez a változáshoz meghatározzuk azt a φ szöveget, amelyet a p helyvektora a pozitív x -tengellyel zár be (első és második sor). A φ meghatározása csak az arctg függvénnyel lehetséges, lévén ez az egyetlen inverz trigonometriai függvény, ami a BASIC-ben definiálva van. Mivel az arctg függvény csak a $-\pi/2$ és $\pi/2$ értékek között értelmezett, az $x < 0$ esetben π -t kell hozzáadnunk. Ezt érjük el a $-\pi * (x < 0)$ kifejezéssel. Az $(x > 0)$ formájú kifejezés szintaktikailag hibásnak tűnhet. Ez valójában egy *Bool*-féle logikai érték. Hatása az, hogy amikor az $(x < 0)$ igaz, akkor helyébe a

3D-V2.6 (c) 1984. Objektum: 4. példa



3D-V2.6 (c) 1984. Objektum: 5. példa



– 1 értéket helyettesíti, egyébként pedig a 0 értéket. A $-\pi * (x < 0)$ kifejezéssel tehát azt kapjuk, hogy

π : ha $x < 0$ és

0: ha $x \geq 0$

Hasonló módon tudjuk szabályozni a körhullám megváltoztatásának mértékét (határait) a $p \leq 7.5$ tartományban, ill. magának a körhullámnak a méretét a $p \leq 7.5$ tartományban. Az $(1 - \cos(3 * w))$ kifejezéssel érjük el, hogy a $p > 7.5$ tartományban három maximumot (ill. minimumot) vegyen fel a körhullámunk. Az egyes maximumhelyek — egymáshoz viszonyítva — pontosan 120° -os szöveget zárnak be.

5. példa

Ez a függvény arra mutat be példát, hogy hogyan kell két függvényt szuperponálni.

$$p = \text{sqr}(x \wedge 2 + y \wedge 2)$$

$$z1 = .5 * (1 + \cos(m * p)) \wedge 3$$

$$z2 = (1 + \cos(9 * m * x * y/15))/2$$

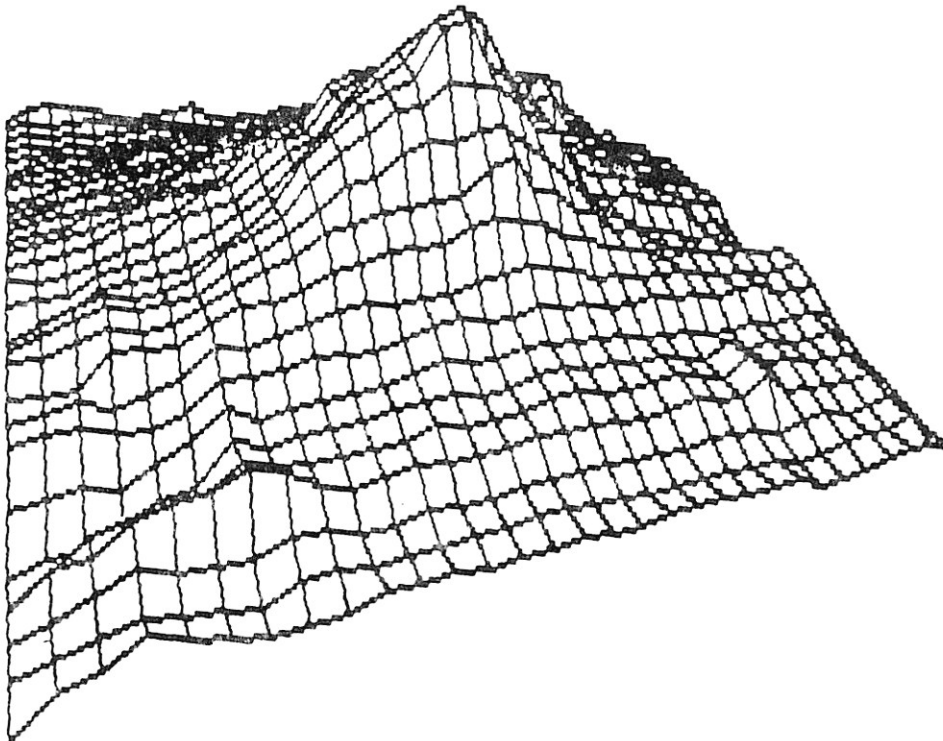
$$z(1) = z1 * z2$$

2.10.3 ELEVATION MODEL PROGRAMMING (DOMBORZATI MODELLEK PROGRAMOZÁSA)

Az *elevation modell programming* az amerikai angol nyelv kifejezése, ami arra utal, hogy számítógép segítségével elkészíthetjük a földrajzi tájak háromdimenziós modelljeit.

A domborzati formákat a térképeken szintvonalakkal ábrázolják. Ezek felismerése meglehetősen nehéz, különösen a nem gyakorlott szemlélő számára. A háromdimenziós modellek előnye, hogy ezeken számos domborzati forma többnyire nagyon jól látható.

3D-V2.6 (c) 1984, Objektum: Szt. Helen hegy



A modellek készítése során nem maga a program az elsődleges probléma, az ui. már rendelkezésünkre áll. Lényegesen nagyobb gondot okoz a megfelelő kiindulási adatok biztosítása, mivel a megfelelő adatokat elsősorban a szintvonalas térképek szolgáltatják.

Példaként bemutatunk egy programot, amely a *U. S. Geological Survey*-től származik és a *St. Helen*-vulkán 1980-as kitörésének adatait dolgozza fel. Az adatokat legkényelmesebben egy szekvenciális file-ban tárolhatjuk. Az adatokat ekkor a program elején olvassuk be. A következő oldalakon láthatjuk a kitörés előtti és utáni adatokat, olyan sorrendbe, mint ahogyan azoknak a szekvenciális file-ban meg kell jelenniük.

```

Ø REM ... P11 ...
1 REM ..... Adatok:/a Szt.Helen hegy a vulkankitores
                                elott/
2 DATA 8,1Ø,1Ø,9,8,6,5,6,6,7,4,3,3,3,4,4,4,4,4,3,2,3,2,
        2,2,4,7,6,6,6,3
3 DATA 8,9,8,8,8,7,6,7,6,5,4,4,5,5,5,6,6,6,6,7,7,6,4,3,
        5,6,8,9,9,8,7
4 DATA 8,8,8,8,7,7,7,7,6,5,5,5,5,6,7,6,6,6,7,7,8,6,5,6,
        7,8,9,9,9,9,7
5 DATA 8,8,8,8,7,7,7,7,7,7,8,7,6,9,9,9,9,7,7,8,8,8,8,8,
        8,9,1Ø,1Ø,7,5,4
6 DATA 8,9,9,8,8,8,8,9,1Ø,1Ø,9,9,9,1Ø,12,11,11,1Ø,9,1Ø,
        11,11,1Ø,1Ø,9,9,9,8,6,5,5
7 DATA 8,9,1Ø,1Ø,9,9,1Ø,1Ø,11,11,1Ø,1Ø,1Ø,12,13,13,12,
        12,1Ø,11,12,12,11,1Ø,9,9,8,7,6,5,5
8 DATA 9,1Ø,1Ø,1Ø,1Ø,1Ø,12,11,12,12,12,11,12,14,14,13,
        13,13,12,12,12,12,11,1Ø,9,8,8,6,6,7,5
9 DATA 1Ø,1Ø,1Ø,11,11,11,11,12,13,13,13,13,14,14,13,12,
        12,1Ø,9,12,13,14,12,11,11,1Ø,9,6,7,7,5
1Ø DATA 1Ø,1Ø,11,11,12,12,12,14,14,14,14,15,15,14,13,
        13,12,11,11,14,15,15,14,12,12,11,9,7,8,8,8
11 DATA 1Ø,11,11,12,13,14,15,16,16,16,17,17,16,16,15,
        14,14,13,11,15,16,16,15,15,13,13,11,9,9,9,8
12 DATA 12,12,13,14,14,14,15,17,19,2Ø,2Ø,22,23,21,19,
        18,17,16,17,17,17,17,18,18,17,16,14,14,13,11,
        1Ø,1Ø
13 DATA 12,13,14,14,15,15,16,19,19,2Ø,23,23,25,25,25,
        26,24,24,22,2Ø,19,19,19,18,17,16,15,14,13,11,
        11
14 DATA 13,14,14,16,16,17,19,19,19,23,26,28,3Ø,32,32,
        31,29,3Ø,29,28,27,26,28,28,23,19,18,15,13,13,
        11
15 DATA 14,15,15,16,18,19,2Ø,2Ø,22,27,28,3Ø,36,38,39,
        39,38,38,37,34,32,32,34,32,21,19,18,16,14,14,
        13
16 DATA 14,15,15,16,18,19,2Ø,22,23,27,3Ø,31,37,39,41,
        41,41,42,41,37,36,32,33,27,22,19,18,16,16,14,
        13

```

17 DATA 15,14,14,16,17,18,21,22,23,26,30,32,37,42,46,
49,51,50,45,37,38,31,28,27,23,20,19,16,16,15,
13

18 DATA 14,14,14,17,17,18,19,21,23,26,30,35,40,43,46,
52,54,52,45,39,36,31,27,25,22,20,19,17,16,15,
13

19 DATA 9,13;14,17,17,18,18,21,23,26,33,37,41,43,45,53,
54,46,43,37,35,30,25,23,22,21,19,17,15,14,13

20 DATA 14,16,18,19,18,18,19,23,25,27,32,36,39,42,42,
46,45,44,41,36,35,28,23,22,22,20,19,17,13,13,10

21 DATA 16,17,19,20,19,19,20,23,25,27,30,33,34,37,39,
38,40,40,36,33,30,26,23,21,18,17,16,14,13,11,9

22 DATA 18,17,19,20,19,19,21,23,25,26,29,30,31,31,32,
31,36,37,31,30,26,25,21,21,18,16,15,13,13,12,11

23 DATA 16,15,15,20,19,19,21,23,23,24,26,26,28,27,27,
29,30,29,28,25,24,22,21,20,18,18,17,16,16,15,15

24 DATA 16,16,17,15,16,17,21,20,21,22,23,23,22,22,22,
23,24,24,23,22,20,20,19,18,18,17,16,16,15,14,15

25 DATA 16,17,17,15,14,16,20,19,19,20,19,19,20,21,21,
20,20,21,20,19,18,18,18,17,17,16,16,16,14,14,14

26 DATA 11,12,12,10,13,15,19,16,17,17,17,17,18,19,18,
17,18,18,17,17,16,15,15,14,13,13,13,12,13,13,11

27 DATA 6,8,7,10,12,12,18,16,16,16,16,17,17,17,16,16,
17,17,16,16,15,15,14,13,15,13,13,12,12,12,10

28 DATA 7,10,14,15,15,15,19,17,16,16,16,16,17,16,16,
16,17,17,16,15,15,16,14,13,16,14,13,12,12,11,9

29 DATA 11,15,15,19,20,20,22,20,17,15,16,16,15,15,15,
16,15,15,15,14,14,15,14,13,16,12,12,11,11,11,9

30 DATA 12,13,15,17,15,14,14,15,15,13,13,14,13,13,13,
13,13,13,12,12,12,13,12,11,10,11,11,10,10,11,8

31 DATA 12,13,14,15,14,13,12,14,14,13,13,13,13,13,13,
13,12,12,11,11,11,11,10,10,9,9,10,10,10,11,8

32 DATA 10,14,15,18,16,14,13,13,13,14,14,14,13,13,13,
13,12,12,12,12,12,11,11,11,9,9,9,10,10,10,8

Ø REM ... P12 ...

1 REM Adatok:/a Szt.Helen hegy a vulkankitores
utan/

2 DATA 8,10,10,9,8,6,5,6,6,7,4,3,3,3,4,4,4,4,4,3,2,3,
2,2,2,4,7,6,6,6,3

3 DATA 8,9,8,8,8,7,6,7,6,5,4,4,5,5,5,6,6,6,6,7,7,6,4,
3,5,6,8,9,9,8,7

4 DATA 8,8,8,8,7,7,7,7,6,5,5,5,5,6,7,6,6,6,7,7,8,6,5,
6,7,8,9,9,9,9,7

5 DATA 8,8,8,8,7,7,7,7,7,7,8,7,6,9,9,9,9,7,7,8,8,8,8,
8,8,9,10,10,7,5,4

6 DATA 8,9,9,8,8,8,8,9,10,10,9,9,9,10,12,11,11,10,9,10,
11,11,10,10,9,9,9,8,6,5,5

7 DATA 8,9,10,10,9,9,10,10,11,11,10,10,10,12,13,13,12,
12,10,11,12,12,11,10,9,9,8,7,6,5,5
8 DATA 9,10,10,10,10,10,12,11,12,12,12,11,12,14,14,13,
13,13,12,12,12,12,11,10,9,8,8,6,6,7,5
9 DATA 10,10,10,11,11,11,11,12,13,13,13,13,14,14,13,12,
12,10,9,12,13,14,12,11,11,10,9,6,7,7,5
10 DATA 10,10,11,11,12,12,12,14,14,14,14,15,15,14,13,
13,12,11,11,14,15,15,14,12,12,11,9,7,8,8,8
11 DATA 10,11,11,12,13,14,15,16,16,16,17,17,16,16,15,
14,14,13,11,15,16,16,15,15,13,13,11,9,9,9,8
12 DATA 12,12,13,14,14,14,15,17,19,20,20,22,23,21,19,
18,17,16,17,17,17,18,18,17,16,14,14,13,11,10,10
13 DATA 12,13,14,14,15,15,16,19,19,20,23,23,25,25,25,
26,24,24,22,20,19,19,19,18,17,16,15,14,13,11,11
14 DATA 13,14,14,16,16,17,19,19,19,23,26,28,30,32,32,
31,29,30,29,28,27,26,28,28,23,19,18,15,13,13,11
15 DATA 14,15,15,16,18,19,20,20,22,27,28,30,36,38,39,
39,38,38,37,34,32,32,34,32,21,19,18,16,14,14,13
16 DATA 14,15,15,16,18,19,20,22,23,27,30,31,37,39,41,
44,45,43,41,38,36,32,33,27,22,19,18,16,16,14,13
17 DATA 15,14,14,16,17,18,21,22,23,26,30,32,37,40,44,
40,43,45,44,40,38,31,28,27,23,20,19,16,16,15,13
18 DATA 14,14,14,17,17,18,19,21,23,26,30,35,40,41,37,
33,36,40,41,40,36,31,27,25,22,20,19,17,16,15,13
19 DATA 9,13,14,17,17,18,18,21,23,26,33,37,41,38,33,
29,29,33,40,39,35,30,25,23,22,21,19,17,15,14,13
20 DATA 14,16,18,19,18,18,19,23,25,27,32,36,40,35,22,
28,28,33,40,36,35,28,23,22,22,20,19,17,13,13,10
21 DATA 16,17,19,20,19,19,20,23,25,27,30,33,34,32,31,
27,27,32,36,33,30,26,23,21,18,17,16,14,13,11,9
22 DATA 18,17,19,20,19,19,21,23,25,26,29,30,31,30,28,
26,25,30,31,29,26,25,21,21,18,16,15,13,13,12,11
23 DATA 16,15,15,20,19,19,21,23,23,24,26,26,28,27,27,
24,24,24,25,25,24,22,21,20,18,18,17,16,16,15,15
24 DATA 16,16,17,15,16,17,21,20,21,22,23,23,22,22,22,
23,24,24,23,22,20,20,19,18,18,17,16,16,15,14,15
25 DATA 16,17,17,15,14,16,20,19,19,20,19,19,20,21,21,
20,20,21,20,19,18,18,18,17,17,16,16,16,14,14,14
26 DATA 11,12,12,10,13,15,19,16,17,17,17,17,18,19,18,
17,18,18,17,17,16,15,15,14,13,13,13,12,13,13,11
27 DATA 6,8,7,10,12,12,18,16,16,16,16,17,17,17,16,16,
17,17,16,16,15,15,14,13,15,13,13,12,12,12,10
28 DATA 7,10,14,15,15,15,19,17,16,16,16,16,17,16,16,
16,17,17,16,15,15,16,14,13,16,14,13,12,12,11,9
29 DATA 11,15,15,19,20,20,22,20,17,15,16,16,15,15,15,
16,15,15,15,14,14,15,14,13,16,12,12,11,11,11,9

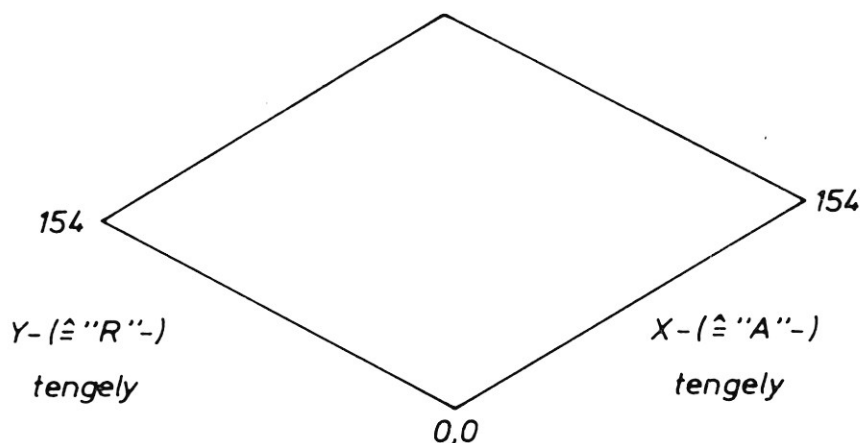
30 DATA 12,13,15,17,15,14,14,15,15,13,13,14,13,13,13,
13,13,13,12,12,12,13,12,11,10,11,11,10,10,11,8
31 DATA 12,13,14,15,14,13,12,14,14,13,13,13,13,13,13,
13,12,12,11,11,11,11,10,10,9,9,10,10,10,11,8
32 DATA 10,14,15,18,16,14,13,13,13,14,14,14,13,13,13,
13,12,12,12,12,12,11,11,11,9,9,9,10,10,10,8

2.11 3D-grafika — második változat

Következő programunk eredménye egy nagy képfelbontással készített háromdimenziós grafikon.

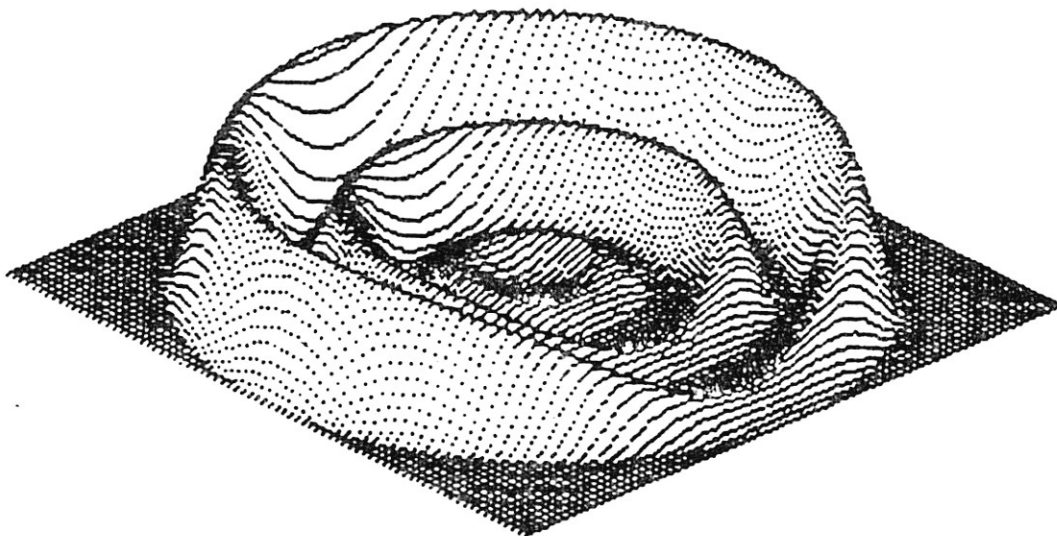
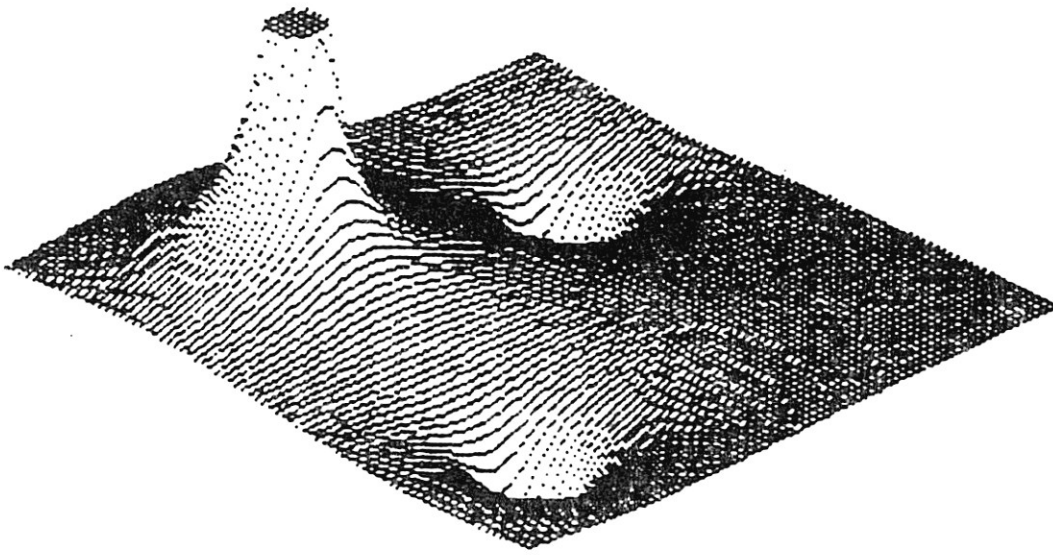
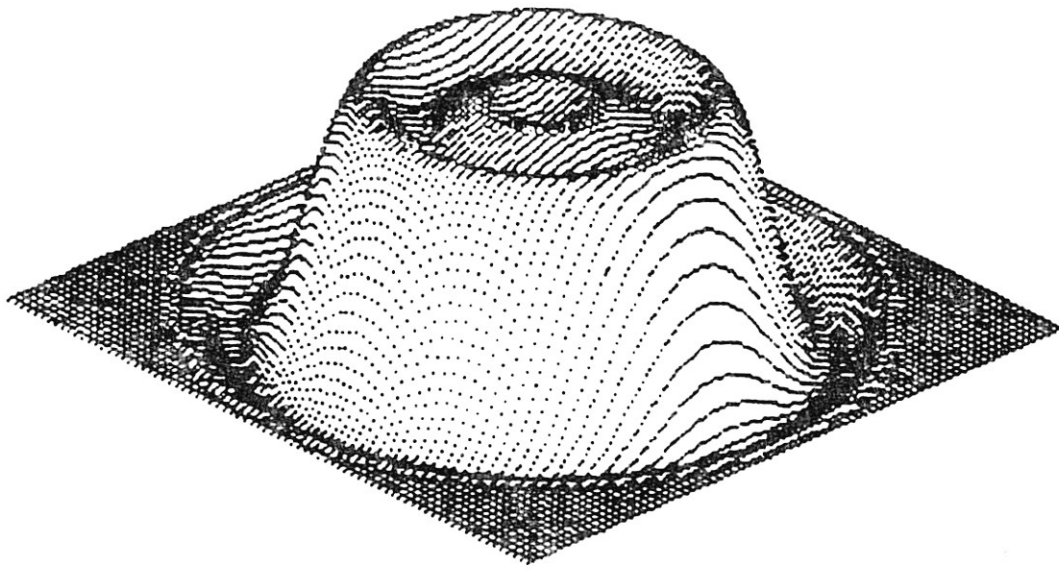
A kép minősége a jellegzetes fény-árnyék hatás alapján válik meggyőzővé. Az eljárás lényegesen egyszerűbb, mint az előzőek voltak, de — és ez a legnagyobb hátránya a programnak — összehasonlíthatatlanul több időt igényel. Ha abból indulunk ki, hogy egészen egyszerű függvényt akarunk ábrázolni, akkor meglepő, hogy ehhez átlagosan 2...6 órára (!) van szükség. Ezalatt a program 11 858 pontot számol ki. Az előbb elmondottak miatt az eljárást csak röviden ismertetjük.

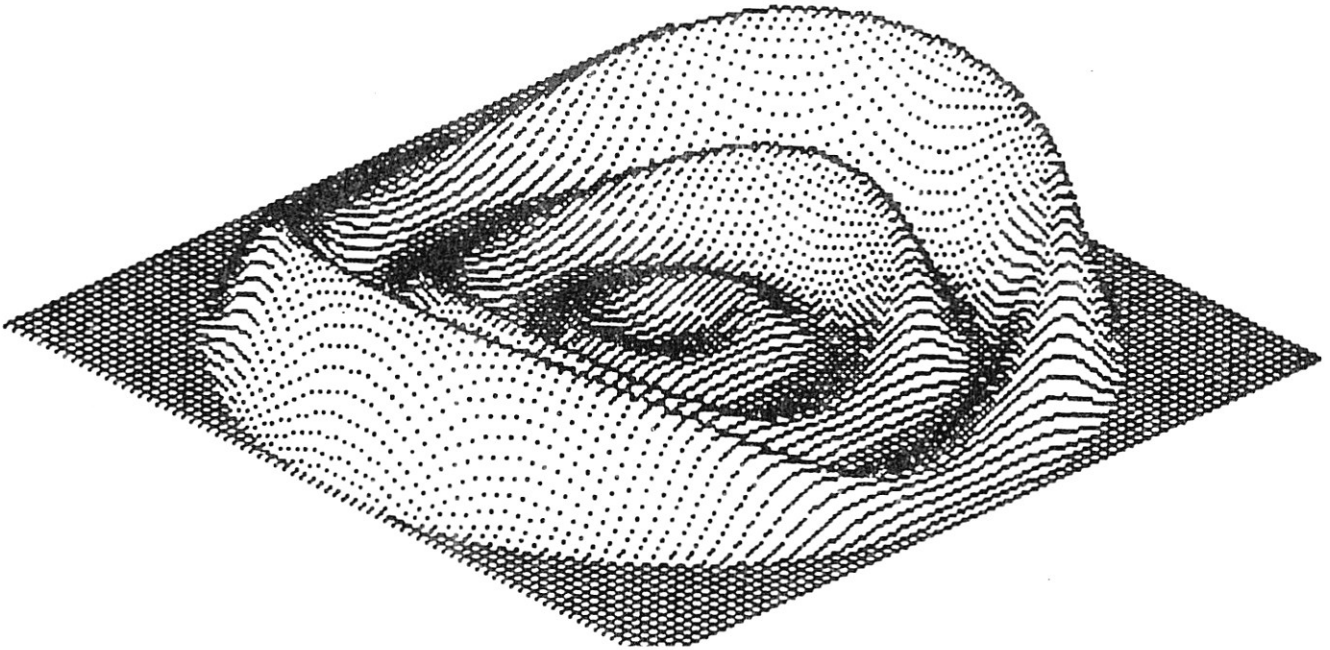
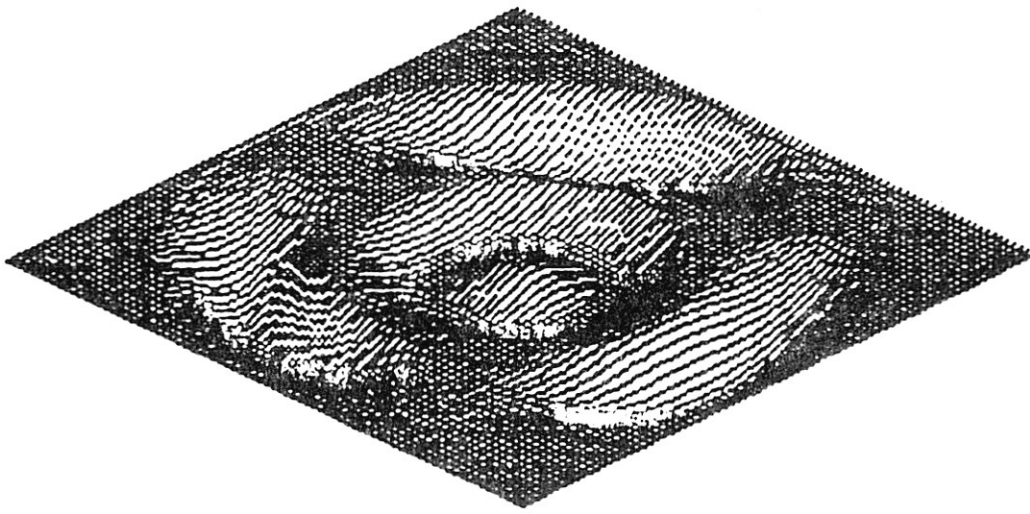
Szóljunk néhány szót az alapnégyszög jellemzőiről! Ez esetben párhuzamos vetítésről van szó, az alapsíkot felülről, 30° -os szögben nézzük. A program a következő séma alapján számol:



A középpontban $A = 77$ és $R = 77$ (vö. 150—250-es sorok). Fontos a 180-as sorban levő 10-es szorzótényező, ami az amplitúdót növeli. Enélkül ui. a kép alig lenne látható.

Az amplitúdó megnövelése azért szükséges, mert a program vetítési képlet nélkül számol. Erről egy korábbi fejezetben már futólag volt szó. A párhuzamos vetítés ui. egy lineáris leképezés, a középpontos vetítés viszont nem az. Éppen ezért teljesen elegendő, ha az eredeti síkunk bázisvektorainak x' , y' , z' képeit határozzuk meg, mivel ezek képterünk bázisát képezik. Mivel a függvényeket az A és az X speciális képernyőváltozóktól függően definiáltuk, szükséges, hogy bizonyos nagyítást végezzünk, pl. a 10-es szorzótényező segítségével. A pontos érték $10\sqrt{2}\cos 30^\circ = 12,2$ lenne. A következő oldalakon néhány példát láthatunk az ezzel az eljárással készített függvényábrázolásra.





Programleírás

Sor	Megyarázat
110—120	Konstansok meghatározása
130	Pontok megjelenítése az eltakart vonalak lezárására a képernyő alsó szélén
150—200	A függvényértékek kiszámítása
210	A láthatóság tesztelése
220—250	Ha látható, akkor a pont megjelenítése
260	Következő sor
290—300	Vége

```

50 REM ... P13 ...
100 REM ..... 3D- a masodik változat
110 DIM Y(320):PI=3.1419
120 R=0:A=160:D=190:F=PI/.77
130 FOR S=1 TO 320:Y(S)=199:NEXT
140 CLS:SCREEN 1

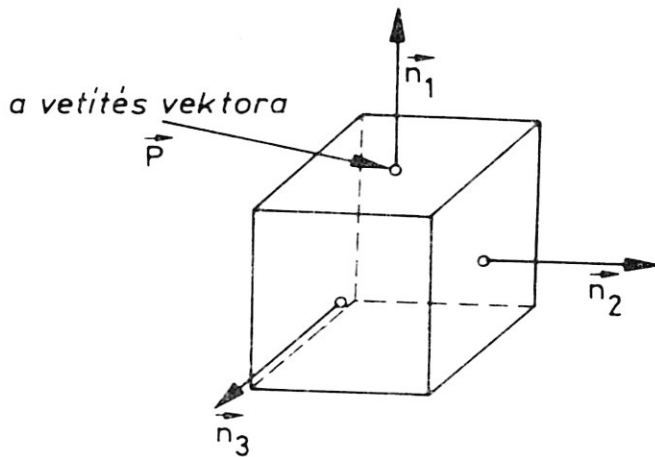
```

```
150 FOR X=1 TO A+154
160 L=D-(1/2)*(X-A)
170 P=SQR((((X-A)/100)-.77)^2)+(((R/100)-.77)^2))
180 Y=10*(1-COS(6*F*P))*(P<=.76)
190 Y=Y*(1.2+COS(4.078*((X-A)/100)))
200 Y=L+Y
210 IF Y>Y(X) THEN 240
220 Y(X) = Y
230 PSET (X,Y)
240 IF X= A+154 THEN 260
250 NEXT
260 R=R+2:A=160-R:D=D-1
270 IF A<6 THEN 290
280 GOTO 150
290 A$=INKEY$:IF A$="" THEN 290
300 END
```


2.12 A *Hidden-line*-probléma (eltakart vonalak problémája)

Tekintsünk egy sík lapok által határolt konvex testet, pl. egy kockát. (Konvex a test akkor, ha bármely két pontját összekötő szakaszt tartalmazza.) Forgassuk el a kockánkat a koordináta-tengelyek körül, tetszőleges $(\varphi_x, \varphi_y, \varphi_z)$ szöggel. Azt akarjuk megállapítani, hogy a kocka felszínének mely lapjai láthatóak és melyek nem. Feltételünk értelmében a felszín sík lapokból áll. Mint már láttuk, egy ilyen lap térbeli helyzetét a normálvektora írja le. Ha valamennyi sík normálvektorát egyöntetűen vagy a test középpontja felé, vagy azzal éppen ellentétesen irányítjuk, akkor a normálvektorok segítségével meg tudjuk állapítani, hogy egy lap látható vagy sem. Ha párhuzamos vetítést alkalmazunk, akkor a láthatóság kritériumaként elegendő a normálvektort vizsgálni. Középpontos vetítésnél szükségünk van még az adott lap súlypontjára is.

A következőkben egy párhuzamos vetítést vizsgálunk meg.



Íranyítsuk a normálvektorokat kifelé! A lap láthatóságának egy feltételét nyerjük azáltal, hogy a normálvektor és a vetítővektor által bezárt szöget vizsgáljuk.

Igaz ui., hogy:

ha $\varphi < 90^\circ$, akkor a lap nem látható;

ha $\varphi \geq 90^\circ$, akkor a lap látható.

A φ szög könnyen meghatározható a vetítővektor és a normálvektor skaláris szorzásával:

$$\cos \varphi = \frac{\mathbf{p} \cdot \mathbf{n}}{|\mathbf{p}| \cdot |\mathbf{n}|}.$$

Tekintettel arra, hogy a $[0; \pi]$ tartományban monoton csökkenő a koszinuszfüggvény, a feltételt így is felírhatjuk:

ha $\cos \varphi > 0$ ($\varphi < 90^\circ$), akkor a lap nem látható;
 ha $\cos \varphi \leq 0$ ($\varphi \geq 90^\circ$), akkor a lap látható.

A láthatóság e feltétele általánosítható: érvényes minden olyan testre, amelynek felszínét síklapok alkotják.

Ha bonyolultabb felépítésű testet akarunk megrajzolni, akkor a megadott feltétel teljesülésének figyelése önmagában nem elégséges. Lehetséges ui., hogy a test egy látható része eltakarja a test másik — a feltétel szerint egyébként látható — részét.

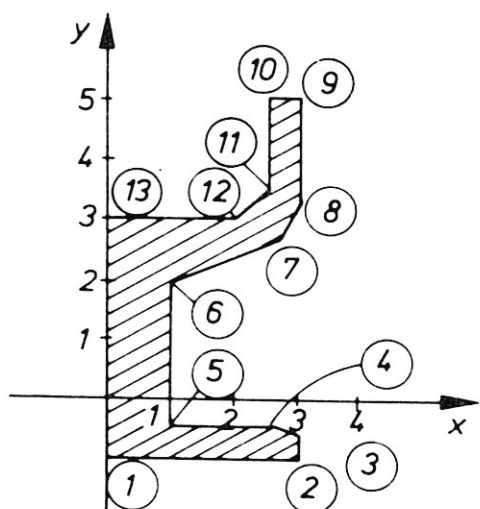
Ilyen esetben a következő elvet kell alkalmazni:

- 1) teljesül az előbb megadott feltétel;
- 2) a test távolabbi részeit eltakarhatják a közelebbi részek.

Általában nem egyszerű a második feltétel figyelése. Erre általános érvényű megoldást úgy kapunk, hogy az összes lapot mélység szerint csoportosítjuk és a rajzoláskor e szerint a csoportosítás szerint járunk el.

A testek egy speciális csoportjára az elmondottak viszonylag könnyen alkalmazhatók. Ez a csoport a forgástestek csoportja, amely egy további feltételnek is eleget tesz.

Példa: azt a testet ábrázoljuk, amely az alábbi alakzatnak az y -tengely körüli forgatása révén keletkezik.



Sorszám	Koordináták
1	(0,1; -1)
2	(3; -1)
3	(3; -0,6)
4	(2,5; -0,5)
5	(1; -0,5)
6	(1; 2)
7	(2,7; 2,7)
8	(3; 3,2)
9	(3; 5)
10	(2,5; 5)
11	(2,5; 3,5)
12	(2; 3)
13	(0,1; 3)

A forgatással egy poharat, kupát, kelyhet — bárminek nevezhetjük a testet — nyertünk. Nevezzük a testet pohárnak! Ez a pohár a testek azon halmazába tartozik, amelyek teljesítik a következő mellékfeltételt: a sarokpontok y -koordinátáinak — a pontokat az előbbi módon sorbarendevezve — csak egy szélsőértéke van. (Ez a szélsőérték esetleg több szomszédos pontban is elérhető, ezek y -koordinátája tehát azonos.) Esetünkben a szélsőértéket a 9-es, ill. a 10-es pont képviseli.

Matematikailag felírva:

$$y_{k-1} \leq y_k \quad \text{valamennyi } k \leq m\text{-re};$$

$$y_k \leq y_{k-1} \quad \text{valamennyi } k > m\text{-re}.$$

A következő program ezt a pohár alakú testet állítja elő.

Az eltakart vonalak problémáját a következő módon oldjuk meg: abból indulunk ki, hogy az y -értékek először növekednek. Ezután meg kell határozni azt, hogy az y -értékek a maximum után melyik pontnál érik el a minimumukat és mi ennek a pontnak az y -értéke. A legnagyobb y -értékű ponttól kezdve visszafelé megyünk, mégpedig mindaddig, amíg el nem érünk egy y_n minimumnál kisebb y -értéket.

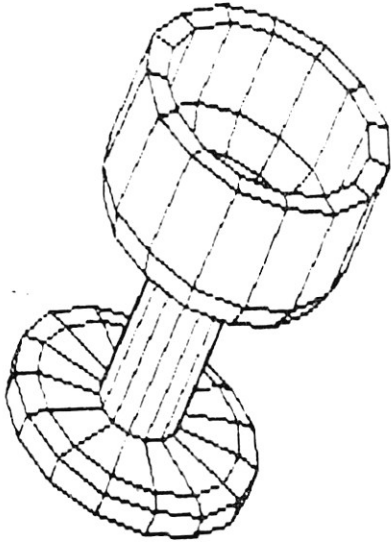
Ezen y -érték által meghatározott P_T pontnál választóvonalat kell húzni (példánkban ez a 7-es pont). Azokat az értékeket, amelyek indexe kisebb, mint ez az érték, nyugodtan megrajzolhatjuk.

Ha fennáll az $x_n < x_T$ egyenlőtlenség, akkor az $(x_n; y_n)$ ponttól az $(x_T; y_T)$ pont felé, azaz visszafelé rajzolunk, különben szabadon rajzolhatunk tovább. Azt a tényt, hogy az elől álló felületrészek a mögöttük levőket eltakarhatják, úgy vesszük figyelembe, hogy a megrajzolandó négyszögek területéről — a PAINT utasítás segítségével — a bennük fekvő vonalakat elhagyjuk.

Programleírás

Sor	Magyarázat
120—130	Függvénydefiníciók
170—220	A forgatási szög és a megrajzolandó felületek számának megadása
250—280	Fejléc
300—380	Rotációs mátrix
400—430	Szubrutinok: forgatás és vetítés
440—490	A pontok forgatása és vetítése
500—710	Szubrutin: megrajzolja a felületet, ha az látható, és a belül fekvő, eltakart vonalakat eltünteti
720—770	Szubrutin: a normálvektor és a vetítővektor vektoriális szorzata és a közöttük levő szög-meghatározása
780—990	A test szétbontása két, külön megrajzolandó részre

Megjegyzés a programhoz: a géptől függő számolási pontatlanság miatt nincs mindig egyenlőségjel a matematikai elmélet és a gyakorlati alkalmazás között. A gépben ábrázolt számok a rendszertől függően pontatlanok lehetnek. Ez — ebben a programban — azt okozhatja, hogy azoknál a felületeknél, amelyek a látható és láthatatlan részek határára esnek, a PAINT utasítás kezdőpontja a kitöltendő tartományon kívül fekszik. Ezáltal viszont a PAINT az egész képernyőt kitölti. Ha ez az eset előfordulna, akkor a 630-as sorban levő összehasonlítási értéket ($CO > -.05$) kis mértékben csökkenteni kell, pl. a $-.052$ értékre. Az adott érték azonban általában megfelelő.



FIGYELEM: a program oem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENŰL** el kell olvasni!

```

50 REM ... P14 ...
100 REM ..... Pohar grafikaja
110 CLS:SCREEN 2:KEY OFF
120 DEF FNG(X)=YU+(X-XU)*(YO-YU)/(XO-XU)
130 DEF FNB (X,Y,Z)=SQR(X^2+Y^2+Z^2)
140 DIM M(3,3):M0=20*X0=160:Y0=100:PI=3.141592
150 PRINT "POHAR"
160 READ M
170 INPUT "W1,W2,W3 (+/- 90 FOK), SZAMA (>2)";W1,W2,
W3,N:N=INT(N)
180 IF ABS(W1)>90 OR ABS(W2)>90 OR ABS(W3)>90 OR N<3
THEN PRINT "Input hiba":GOTO 170
190 IF N>20 THEN PRINT "Ha a szam >20, akkor a szerzo
a sikert nem garantálja " ELSE 230
200 PRINT "megszakitas (a) mehet tovabb (w)"
210 A$=INKEY$:IF A$="" THEN 210
220 IF A$="A" THEN 170
230 DIM XB(M,N),YB(M,N),XA(M,N),YA(M,N),ZA(M,N)
240 DIM R(M),Y(M)
250 CLS:PRINT "POHAR"
260 PRINT "W1=";W1
270 PRINT "W2=";W2
280 PRINT "W3=";W3
290 T=PI/180:W1=W1*T:W2=W2*T:W3=W3*T
300 M(1,1)=COS(W2)*COS(W3)
310 M(2,1)=-COS(W2)*SIN(W3)
320 M(3,1)=SIN(W2)
330 M(1,2)=COS(W1)*SIN(W3)+SIN(W1)*SIN(W2)*COS(W3)
340 M(2,2)=COS(W1)*COS(W3)-SIN(W1)*SIN(W2)*SIN(W3)

```

```

350 M(3,2)=-SIN(W1)*COS(W2)
360 M(1,3)=SIN(W1)*SIN(W3)-COS(W1)*SIN(W2)*COS(W3)
370 M(2,3)=SIN(W1)*COS(W3)+COS(W1)*SIN(W2)*SIN(W3)
380 M(3,3)=COS(W1)*COS(W2)
390 GOTO 440
400 XA(I,E)=M(1,1)*X1+M(2,1)*Y1+M(3,1)*Z1
410 YA(I,E)=M(1,2)*X1+M(2,2)*Y1+M(3,2)*Z1
420 ZA(I,E)=M(1,3)*X1+M(2,3)*Y1+M(3,3)*Z1
430 RETURN
440 FOR I=1 TO M:READ R(I),Y(I):R(I)=R(I)/1.5
450 FOR E=0 TO N:X=E*2*PI/N
460 X1=R(I)*COS(X):Y1=R(I)*SIN(X):Z1=Y(I)
470 GOSUB 400
480 XB(I,E)=320+2*M0*XA(I,E):YB(I,E)=120-M0*ZA(I,E)
490 NEXT E:NEXT I:RESTORE:GOTO 780
500 FOR I=KEZDET TO BEFEJEZES STEP LEPES:R(I)=R(I)/1.5
510 FOR E=0 TO N
520 IF E=0 THEN 710
530 IF I=1 THEN 710
540 GOSUB 720:IF C0>0 THEN 710
550 AB=0
560 LINE(XB(I,E),YB(I,E))-(XB(I-1,E-1),YB(I-1,E-1)),0
570 LINE(XB(I-1,E),YB(I-1,E))-(XB(I,E-1),YB(I,E-1)),0
580 XP1=(XB(I,E)+XB(I-1,E-1))/2:
    YP1=(YB(I,E)+YB(I-1,E-1))/2
590 XP2=(XB(I-1,E)+XB(I,E-1))/2:
    YP2=(YB(I-1,E)+YB(I,E-1))/2
600 XP=(XP1+XP2)/2:YP=(YP1+YP2)/2:PRESET (XP,YP)
610 LINE(XB(I,E),YB(I,E))-(XB(I,E-1),YB(I,E-1)):
    LINE-(XB(I-1,E-1),YB(I-1,E-1))
620 LINE-(XB(I-1,E),YB(I-1,E)):LINE-(XB(I,E),YB(I,E))
630 IF CO>-.05 OR POINT (XP,YP)=1 THEN 700
640 PAINT (XP,YP)
650 LINE(XB(I,E),YB(I,E))-(XB(I,E-1),YB(I,E-1)),0:
    LINE-(XB(I-1,E-1),YB(I-1,E-1)),0
660 LINE-(XB(I-1,E),YB(I-1,E)),0:
    LINE-(XB(I,E),YB(I,E)),0
670 PAINT (XP,YP),0
680 LINE(XB(I,E),YB(I,E))-(XB(I,E-1),YB(I,E-1)):
    LINE-(XB(I-1,E-1),YB(I-1,E-1))
690 LINE-(XB(I-1,E),YB(I-1,E)):LINE-(XB(I,E),YB(I,E))
700 IF AB=0 THEN AB=1:GOTO 560
710 NEXT E:NEXT I:AB=0:RETURN
720 AX=XA(I,E)-XA(I,E-1):AY=YA(I,E)-YA(I,E-1):
    AZ=ZA(I,E)-ZA(I,E-1)
730 BX=XA(I-1,E-1)-XA(I,E-1):BY=YA(I-1,E-1)-YA(I,E-1):
    BZ=ZA(I-1,E-1)-ZA(I,E-1)
740 NX=AY*BZ-AZ*BY:NY=AZ*BX-AX*BZ:NZ=AX*BY-AY*BX
750 BN=FNB(NX,NY,NZ)

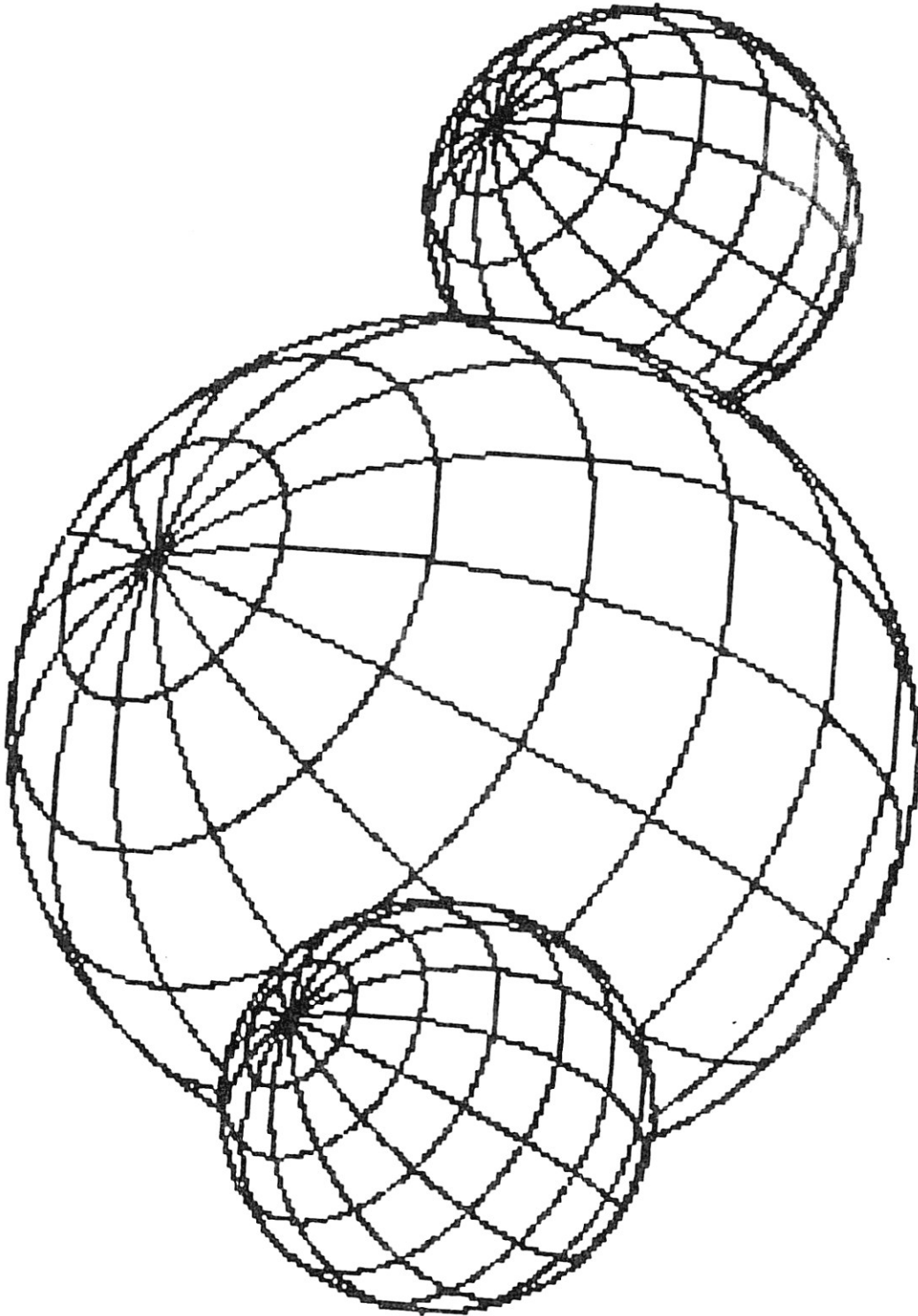
```

```

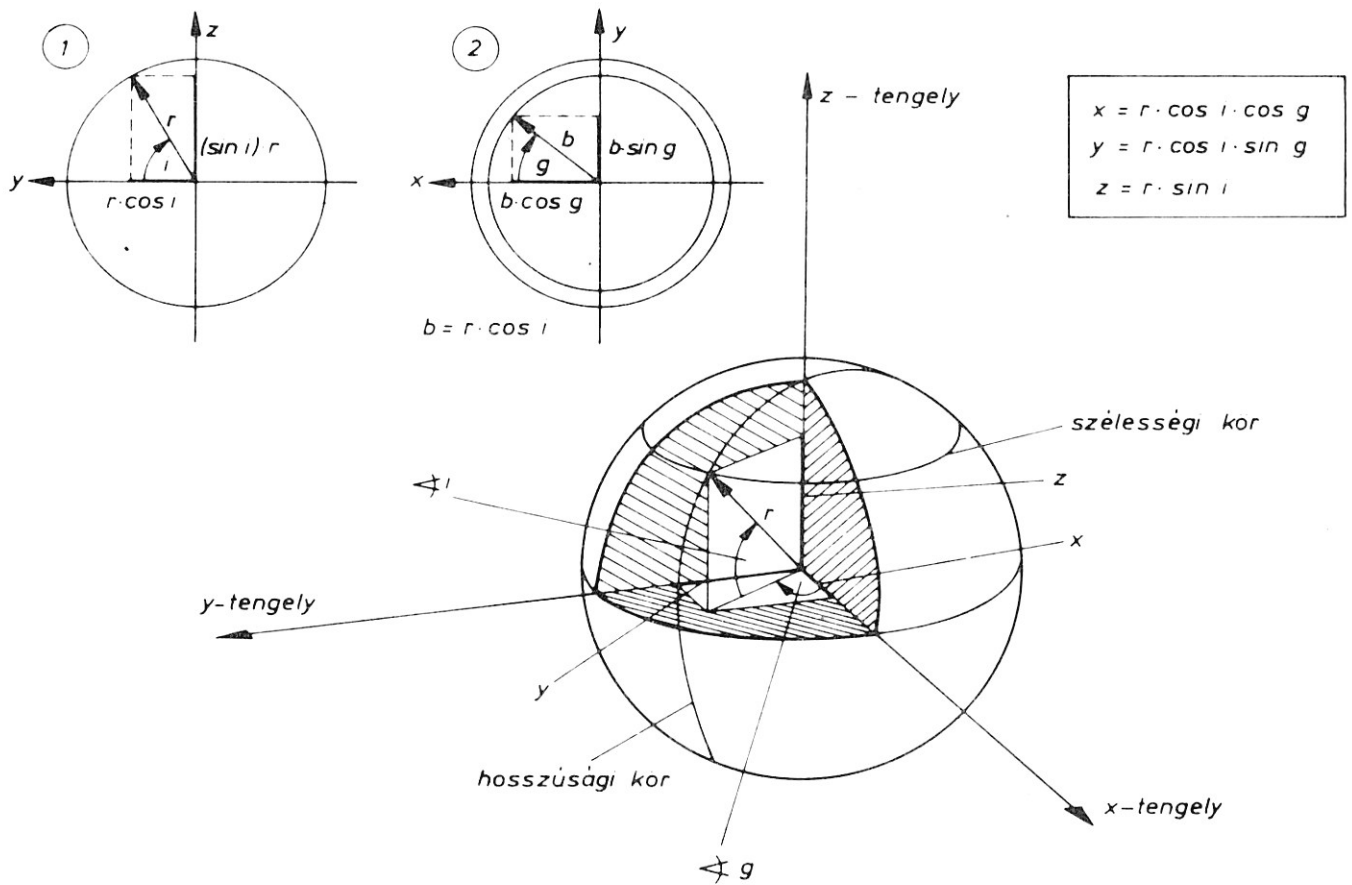
760 CO=NY/BN
770 RETURN
780 READ M
790 FOR R=1 TO M:READ R(R),Y(R):NEXT
800 BEGINN=1:FOR R=2 TO M
810 IF Y(R)>=Y(R-1) THEN 980
820 IF R=M THEN 840
830 IF Y(R+1)<Y(R) THEN 980
840 U=Y(R):A=R-1:O=U
850 IF Y(A)>U THEN O=-O*(Y(A)<=O)-Y(A)*(Y(A)>O):A=A-1:
    IF A=KEZDET-1 THEN 860 ELSE 850
860 START=A
870 A=R
880 IF A=M THEN 900
890 IF Y(A)<O THEN A=A+1:GOTO 880
900 ENDE=A
910 IF W1>0 THEN 950
920 KEZDET=ELEJE : BEFEJEZES=START:LEPES=1:GOSUB 500
930 KEZDET=VEGE:BEFEJEZES=START+1:LEPES=-1:GOSUB 500
940 ELEJE=VEGE+1:R=ELEJE:GOTO 980
950 KEZDET=VEGE:BEFEJEZES=START+1:LEPES=-1:GOSUB 500
960 KEZDET=START:BEFEJEZES=ELEJE:LEPES=-1:GOSUB 500
970 ELEJE=VEGE+1:R=ELEJE
980 NEXT
990 END
1000 DATA 13,.1,-1,3,-.6,2.5,-.5,1,-.5,1,2,2,7,2.7,3,
    3.2,3,5,2.5,5,2.5,3.5,2,3,.1,3

```

2.13 Háromdimenziós gömbök



Ahhoz, hogy a gömböt három dimenzióban ábrázolni tudjuk, át kell gondolni, hogy miképp írható le egyáltalán egy gömb az R^3 -ban. A gömb azon pontok mértani helye, amelyek meghatározott r távolságra vannak egy adott ponttól, pl. a koordináta-rendszer O pontjától. Más megközelítésben: ha az r rádiuszvektor az O pont körül forog, akkor a vektor csúcsa mindig egy gömbön mozog. Ezt szemlélteti a következő ábra:



Ha egy szélességi kört akarunk leírni, akkor a z -koordinátát állandónak kell választani. Az (1) jelű ábra alapján:

$$z_r = r \sin(i).$$

A gömb egy pontját vektoriálisan a következő összefüggéssel adhatjuk meg:

$$\mathbf{r} = \mathbf{x}_r + \mathbf{y}_r + \mathbf{z}_r.$$

Az $\mathbf{x}_r + \mathbf{y}_r$ vektor az $x-y$ síkban fekszik, így írható:

$$|\mathbf{x}_r + \mathbf{y}_r| = |r \cos(i)|.$$

A (2) jelű ábra alapján x_r - és y_r -re írható:

$$\begin{aligned} x_r &= r \cos(i) \cos(g); \\ y_r &= r \cos(i) \sin(g). \end{aligned}$$

Ezzel meghatároztuk a gömb egy $\mathbf{P} = (r, i, g)$ pontjának az ún. polárkoordinátáit. A polárkoordinátákra a következő egyenletet írhatjuk fel:

$$\mathbf{P} = (r, i, g) = \begin{matrix} x_r & r \cos(i) \cos(g) \\ y_r & r \cos(i) \sin(g) \\ z_r & r \sin(i) \end{matrix}$$

Térjünk vissza a szélességi körhöz: a szélességi kört úgy tudjuk leírni, hogy i értékét állandónak választjuk, a g pedig befutja a 0-tól 2π -ig terjedő intervallumot.

A hosszúsági kör ehhez hasonlóan írható le, de ekkor a g állandó és i értéke változik 0-tól 2π -ig.

Ha a gömb egy $\mathbf{P} = (x_r, y_r, z_r)$ pontját koordinátaival megadtuk, akkor ezt a \mathbf{P} vektort tetszőlegesen el tudjuk forgatni a rotációs mátrix segítségével, Ha ezt az elforgatást a gömb minden pontjára elvégezzük, akkor ezzel az egész gömböt forgatjuk el.

A képernyőn való ábrázoláshoz célszerűen a párhuzamos vetítést használjuk fel, mégpedig az y -tengellyel párhuzamosan, az $x-z$ síkra vetítve. Így a gömb felszínének egy tetszőleges pontja akkor nem lesz látható, ha az y_r koordinátája negatív. Ezzel egy nagyon egyszerű módszer van a kezünkben arra, hogy az eltakart vonalakat felismerjük és láthatatlanná tegyük.

A vetítési képletre — mint már annak tárgyalásakor végiggondoltuk — ezúttal nincs szükségünk, ui. az x_r és a z_r komponenseket egyszerűen mint kétdimenziós képernyő-koordinátákat értelmezzük. Ezekben a megfontolásokon alapul a következő programunk is.

A bemutatott eljárás sokoldalúan használható. Ha kellő számú koordinátát adunk meg a gépnek, akkor a földgömböt tetszőleges pozícióból megnézhetjük. Az eljárással egyszerűbb molekulák — mint pl. H_2O vízmolekula vagy az NH_3 ammóniamolekula — modelljét is előállíthatjuk.

Programleírás

Sor	Magyarázat
120—130	A forgatás szöge
140—210	Rotációs mátrix
230—260	A gömb kontúrja
270—360	Hosszúsági körök
370—460	Szélességi körök

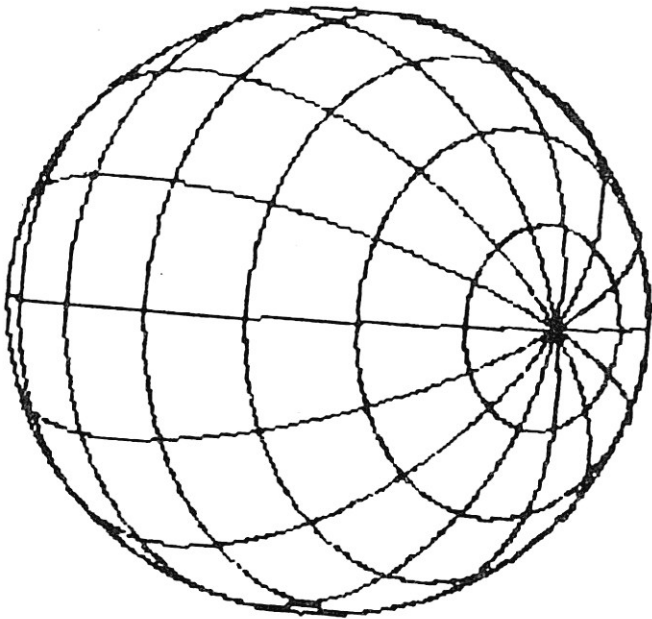
FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét **FELTÉTLENÜL** el kell olvasni!

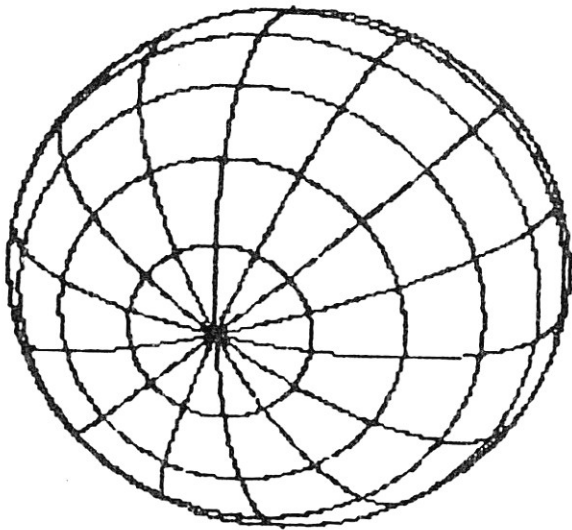
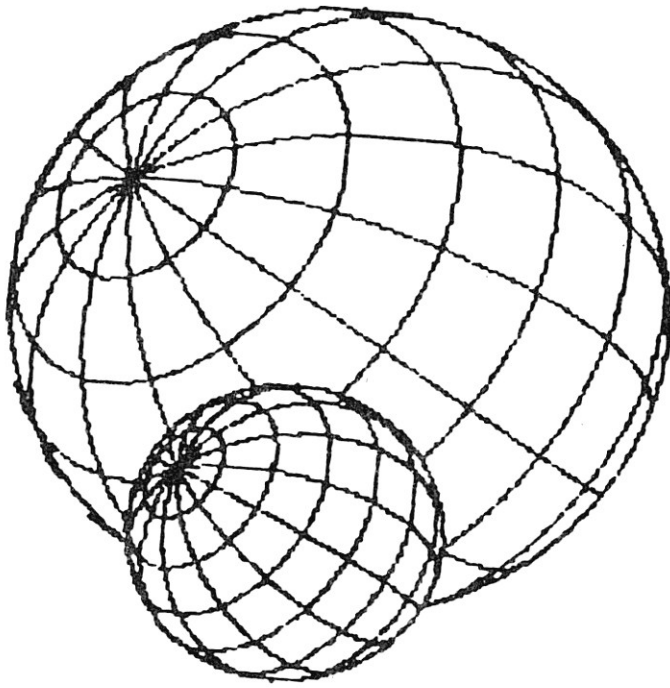
```
50 REM ... P15 ...
100 REM .....Gombok haromdimenzios abrazolasa
110 CLS:SCREEN 2:KEY OFF:PI=3.1415
120 INPUT "W1,W2,W3,R";W1,W2,W3,R:CLS
130 W1=W1*PI/180:W2=W2*PI/180:W3=W3*PI/180
140 M(1,1)=COS(W2)*COS(W3)
150 M(2,1)=-COS(W2)*SIN(W3)
160 M(3,1)=SIN(W2)
170 M(1,2)=COS(W1)*SIN(W3)+SIN(W1)*SIN(W2)*COS(W3)
180 M(2,2)=COS(W1)*COS(W3)-SIN(W1)*SIN(W2)*SIN(W3)
190 M(3,2)=-SIN(W1)*COS(W2)
200 M(1,3)=SIN(W1)*SIN(W3)-COS(W1)*SIN(W2)*COS(W3)
210 M(2,3)=SIN(W1)*COS(W3)+COS(W1)*SIN(W2)*SIN(W3)
220 M(3,3)=COS(W1)*COS(W2)
230 PSET (320+2*R,100)
240 FOR I=0 TO 2*PI STEP .04
250 X=320+2*R*COS(I):Y=100-R*SIN(I)
```

```

260 LINE-(X,Y):NEXT
270 FOR G=0 TO PI-.2 STEP PI/7:
    FOR I=0 TO 2*PI STEP .03
280 Z=SIN(I):X=COS(I)*COS(G):Y=COS(I)*SIN(G)
290 XA=M(1,1)*X+M(2,1)*Y+M(3,1)*Z
300 YA=M(1,2)*X+M(2,2)*Y+M(3,2)*Z
310 ZA=M(1,3)*X+M(2,3)*Y+M(3,3)*Z
320 IF YA<0 THEN AA=1:GOTO 360
330 XB=320+2*R*XA:YB=100+R*ZA
340 IF I=0 OR AA=1 THEN AA=0:PSET (XB,YB):GOTO 360
350 LINE -(XB,YB)
360 NEXT:NEXT
370 FOR I=(-PI/2+PI/9) TO (PI/2-PI/9) STEP PI/9:
    FOR G=0 TO 2*PI STEP .03
380 Z=SIN(I):X=COS(I)*COS(G):Y=COS(I)*SIN(G)
390 M(1,1)*X+M(2,1)*Y+M(3,1)*Z

```

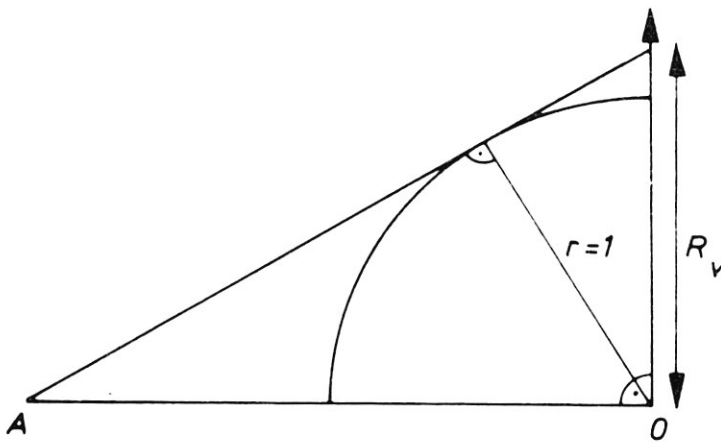




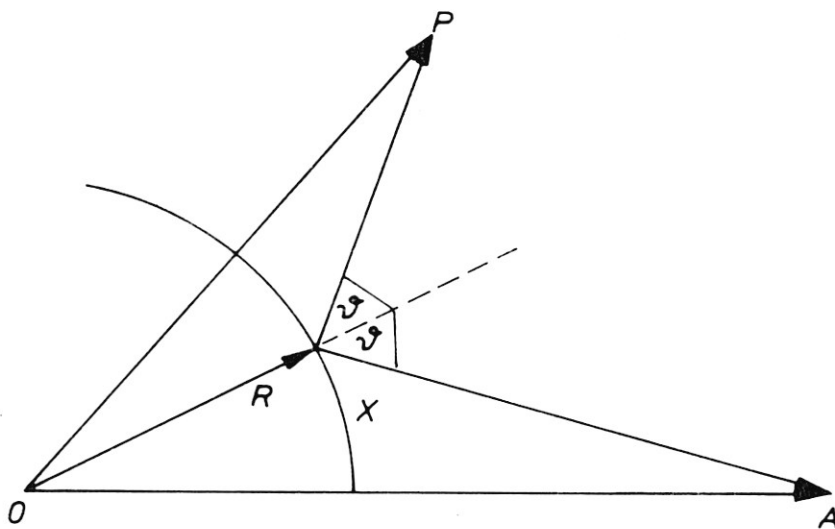
2.14 Gömbtükrözés

A vektorszámítás nagyon szép alkalmazása az az eljárás, amikor olyan tükörképet szerkesztünk meg, mint amit egy szappanbuborékon vagy fényes vasgolyón látunk.

Induljunk ki a következő feltételekből: a szemünk helyzete változatlan marad az $\mathbf{A} = (0; 10; 0)$ pozícióban, amíg a tükörképet nézzük, továbbá, hogy olyan gömbön levő tükörképet nézzük, amelynek sugara egységnyi ($r = 1$), középpontja pedig a koordináta-rendszer középpontjában van. A szemünk elé táruló képet ismét a középpontos vetítés képletével határozhatjuk meg úgy, hogy a látósugarat metsszük az $x-z$ síkkal. Mint az ábrából látható, ekkor a gömb sugara virtuális és a virtuális sugár $R_v = (1; 1 - 1; |\mathbf{A}|)$ koordinátákkal jellemezhető.



Mielőtt meghatározzuk egy tárgy gömbön látható tükörképét, térjünk ki egy egyszerűbb problémára!



Adott a K kör az R^2 -ben: a kör sugara 1 és középpontja a $(0; 0)$ pontban van. Továbbá adott a $\mathbf{P} = (p_1; p_2)$ pont, ahol $p_2 \geq 0$, ill. az $\mathbf{A} = (a_1; a_2)$ pont, ahol $a_2 = 0$.

Határozzuk meg azt az $\mathbf{X} = (x_1; x_2)$ pontot, amely kielégíti az

$$x_1^2 + x_2^2 = 1$$

egyenletet, azaz rajta van a K körön, és amelynek az x_1 koordinátájára fennáll az $x_1 \geq 0$ egyenlőtlenség, továbbá eleget tesz az alábbi egyenlőségeknek:

$$\frac{\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}{r_p} = \frac{\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}{r_a},$$

ahol

$$r_p = \sqrt{p_1^2 + p_2^2}; \quad r_a = \sqrt{a_1^2 + a_2^2}$$

Figyelembe véve a P és az A pontra megadott feltételeket, a következő egyenletrendszert állíthatjuk fel:

$$x_1 = \sqrt{\frac{B^2}{A^2 + B^2}}; \quad x_2 = \sqrt{\frac{A^2}{A^2 + B^2}},$$

ahol

$$A = p_1 r_a - a_1 r_p; \quad B = p_2 r_a - a_2 r_p.$$

Meg kell még vizsgálnunk az $A^2 + B^2 = 0$ esetet. Ha figyelembe vesszük, hogy $r_a = a_1$ és $a_2 = 0$, ill. $p_2 \geq 0$, akkor behelyettesítés után az $r_p = p_1$ összefüggést kapjuk. Ez viszont azt jelenti, hogy $p_2 = 0$, amiből pedig az következik, hogy a P pontnak az x -tengelyen kell fekdnie.

Ezzel a kör egy olyan pontját határoztuk meg, amely az A pontban levő szem számára a P pont tükörképékként jelenik meg, ui. ebben a pontban teljesül, hogy a beesési szög egyenlő a visszaverődési szöggel.

Ezt az egyszerű megoldást most már felhasználhatjuk gömbtükrözési problémánk megoldásában is. Tételezzük fel, hogy szemünk az A pontból nézi a P pont tükörképét, ami a kiindulási gömbünkön tükröződik. A P pontnak a gömbön látható tükörképe az X pont, amelyre nyilván érvényes a következő állítás: X abban a síkban fekszik, amelyet a \mathbf{P} és az \mathbf{A} vektorok feszítenek ki. Ezzel viszont tükrözési problémánkat az előbbiekre vezettük vissza. Határozzunk meg most egy olyan $(x; y)$ koordináta-rendszert, amelyben P és A a megadott feltételeket kielégítik. Az x -tengely legyen az \mathbf{A} vektor. Szerkesszünk egy erre merőleges \mathbf{Y} vektort, amely a \mathbf{P} és az \mathbf{A} vektorok által kifeszített síkban fekszik. Ezt az \mathbf{Y} vektort az

$$\mathbf{Y} = (\mathbf{A} \times \mathbf{P}) \times \mathbf{A}$$

vektoriális szorzásokkal nyerjük. Képezzünk most az \mathbf{X} és \mathbf{Y} irányú egységvektorokat! Ezekkel az egységvektorokkal egy olyan kétdimenziós koordináta-rendszert definiáltunk, amelynek irányvektorai \mathbf{X}° és \mathbf{Y}° .

Koordináta-transzformációt alkalmazva:

$$\begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{bmatrix} x_1^0 & x_2^0 & x_3^0 \\ y_1^0 & y_2^0 & y_3^0 \end{bmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

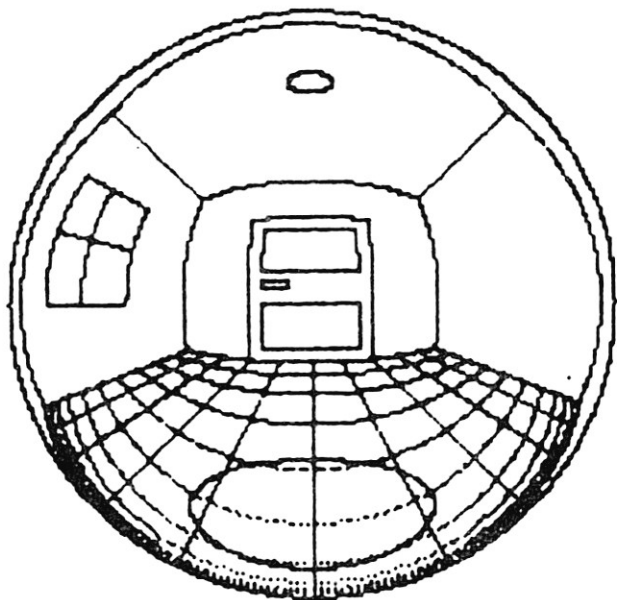
az A és a P pontot az új koordináta-rendszerünkbe transzformálhatjuk át.

A bemutatott eljárással már kiszámíthatjuk az X' tükörképpontot az S' rendszerben is. Az X' pontot inverz koordináta-transzformációval visszatranszformáljuk az S rendszerbe és ezzel megkapjuk a gömbön levő pontot.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} x_1^0 & y_1^0 \\ x_2^0 & y_2^0 \\ x_3^0 & y_3^0 \end{bmatrix} \cdot \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}$$

Természetesen azokat a pontokat, amelyek a gömb mögött vannak, nem láthatjuk. Ezért a $p_2 < 0$ esetben először is meghatározzuk a PA egyenesnek az $x-z$ síkkal való dőléspontját. Ha ennek a pontnak az O ponttól mért távolsága kisebb, mint az R_v virtuális sugár, akkor ezt a pontot már el is felejthetjük, ui. nem fog látszani.

Példaként a következő ábrán egy olyan vasgolyót láthatunk, ami szabadon lebeg egy szoba közepén. A képen a szoba mennyezetén levő lámpa mellett az az árnyék is látszik, amit a szoba közepén lebegő golyó a padlóra vet.



Programleírás

Sor	Magyarázat
110	Konstansok
120—180	A gömb kontúrja
200—230	Ellenőrzés, hogy a pont látszik-e
240—310	Koordináta-transzformáció
320—350	A pont tükörképének kiszámítása
360—380	Koordináták visszatranszformálása
390—440	A pont vetítése
450—470	A környezet definiálása

FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét
FELTÉTLENŰL el kell olvasni!

```
50 REM ... P16 ...
100 CLS:SCREEN 2:KEY OFF:MX=160:MY=100:M0=50:PI=3.1419
110 A1=0:A2=10:A3=0
120 RA=SQR(A1^2+A2^2+A3^2)
130 X1=A1/RA:X2=A2/RA:X3=A3/RA
140 RV=1/SQR(1-1/RA^2)
150 PSET (2*(MX+M0*RV),MY)
160 FOR I=0 TO 2*PI STEP .04
170 X=2*(MX+M0*RV*COS(I)):Y=MY+M0*RV*SIN(I)
180 LINE -(X,Y):NEXT
190 GOTO 450
200 IF P2-A2=0 THEN T=0:GOTO 220
210 T=P2/(P2-A2)
220 GX=P1-T*(P1-A1):GY=P3-T*(P3-A3)
230 IF (GX^2+GY^2)<RV^2 AND P2<0 THEN 440
240 B1=A3*(A3*P1-A1*P3)-A2*(A1*P2-A2*P1)
250 B2=A1*(A1*P2-A2*P1)-A3*(A2*P3-A3*P2)
260 B3=A2*(A2*P3-A3*P2)-A1*(A3*P1-A1*P3)
270 BB=SQR(B1^2+B2^2+B3^2)
280 Y1=B1/BB:Y2=B2/BB:Y3=B3/BB
290 PX=P1*X1+P2*X2+P3*X3
300 PY=P1*Y1+P2*Y2+P3*Y3
310 AX=RA:AY=0
320 RP=SQR(P1^2+P2^2+P3^2)
330 A=PX*RA-AX*RP:B=PY*RA-AY*RP
340 IF A^2+B^2=0 THEN GX=0:GY=0:GOTO 430
350 SX=SQR(B^2/(A^2+B^2)):SY=SQR(A^2/(A^2+B^2))
360 G1=SX*X1+SY*Y1
370 G2=SX*X2+SY*Y2
380 G3=SX*X3+SY*Y3
390 IF G2-A2=0 THEN T=0:GOTO 410
400 T=G2/(G2-A2)
410 GX=G1-T*(G1-A1)
420 GY=G3-T*(G3-A3)
430 PSET (2*(MX+M0*GX),MY-M0*GY)
440 RETURN
450 P3=-2:FOR P1=-3 TO 3 STEP .5:
    FOR P2=-3 TO 3 STEP .15:GOSUB 200:NEXT:NEXT
460 P3=-2:FOR P2=-3 TO 3 STEP .5:
    FOR P1=-3 TO 3 STEP .05:GOSUB 200:NEXT:NEXT
470 END
```

2.15 *Real-time* — trükkfilm a GET és a PUT utasítások alkalmazásával

A GET és a PUT utasítások alkalmazásával lehetőségünk nyílik arra, hogy pl. térhatású mozgásokkal ún. *real-time* (valós idejű) trükkfilmet készítsünk. Nézzünk erre egy példát: gömböt rajzoló programunkkal elkészítünk 10 képet egy gömbről, mégpedig úgy, hogy minden kép az előzőhöz képest 1° -kal el legyen fordulva. A 10 képet a GET utasítással tároljuk az operatív tárban. Az 1° -os folytonos elfordítást azért választottuk, mert a rajzon a hosszúsági köröket 10° -onként rajzoltuk meg. Ha most a PUT utasítással a 10 képet gyorsan egymás után ugyanazon a helyen megjelenítjük, akkor az a benyomásunk alakul ki, hogy a gömb forog. A kép kissé villódzó ugyan, de ha a tizedik kép után rögtön előlről kezdjük az elsővel, akkor folyamatos forgást látunk. A gömb sugara nem lehet nagyobb, mint 65 egység, mert egyébként nem volna elegendő a 64 kbyte-os BASIC-terület.

A következő program egy bolygó kísérőholdjainak mozgását szimulálja. A központi bolygó képét a gömböt rajzoló programmal állítottuk elő. A keringési pálya síkját 20° -os szögből látjuk. Ha megadjuk a legbelső hold keringési idejét, akkor a többi hold keringési ideje a legbelső hold keringését leíró szinusz-, ill. koszinuszfüggvény argumentumának egy konstanssal való osztásával számítható ki. Ha az r pályasugarat szabadon választjuk meg, akkor a keringési pálya ellipszisének az a nagy féltengelye a képernyőn r , a kis féltengelye pedig $b = \sin 20^\circ$ lesz.

A legbelső holdat a központi bolygó időnként takarja. Ha a bolygó sugarát 25 egységben határozzuk meg, akkor a takarásra a következő feltételt adhatjuk:

```
d = sqr ((x(hold 1) - 320) ^ 2 + (y(hold 1) - 100) ^ 2)
if d < 25 and y < 100 then ...eltakarás
```



Sor	Magyarázat
120—200	Az 1...5 holdak megszerkesztése
210	A gömböt rajzoló programmal korábban előállított, majd lemezen tárolt bolygó képének betöltése
220—250	A háttér csillagai
260—560	1...5 holdak

FIGYELEM: *a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét FELTÉTLENÜL el kell olvasni!*

```

50 REM ... P17 ...
100 REM ..... Bolygorendszer
110 DIM M1(34),M2(34),M3(34),M4(34),M5(34):MX=155:
    MY=95:PI=3.14159
120 FOR I=1 TO 5
130 CLS:LOCATE 1,1:PRINT RIGHT$(STR$(I),1)
140 CIRCLE (13,5),5:PAINT(13,5)
150 IF I=1 THEN GET(0,0)-(18,10),M1
160 IF I=2 THEN GET(0,0)-(18,10),M2
170 IF I=3 THEN GET(0,0)-(18,10),M3
180 IF I=4 THEN GET(0,0)-(18,10),M4
190 IF I=5 THEN GET(0,0)-(18,10),M5
200 NEXT
210 BLOAD "B:Bolygo.PLS":RANDOMIZE V
220 FOR I=1 TO 200:X=319*RND(1):Y=199*RND(1)
230 IF SQR((X-160)^2+(Y-100)^2)<25 THEN 250
240 PSET (2*X,Y)
250 NEXT
260 XA1=MX:YA1=MY+20:PUT(2*XA1,YA1),M1
270 XA2=MX:YA2=MY+29:PUT(2*XA2,YA2),M2
280 XA3=MX:YA3=MY+33:PUT(2*XA3,YA3),M3
290 XA4=MX:YA4=MY+41:PUT(2*XA4,YA4),M4
300 XA5=MX:YA5=MY+51:PUT(2*XA5,YA5),M5
310 FOR I=0 TO 40*PI STEP .5
320 X1=MX+60*SIN(I/2)
330 Y1=MY+20*COS(I/2)
340 D=SQR((X1-MX)^2+(Y1-MY)^2)
350 IF D<27 AND A=1 THEN 390
360 IF D<27 AND Y1<MY THEN PUT (2*XA1,YA1),M1:A=1:
    GOTO 390
370 IF A=1 THEN A=0:PUT (2*X1,Y1),M1:GOTO 390
380 PUT (2*XA1,YA1),M1:PUT (2*X1,Y1),M1
390 XA1=X1:YA1=Y1
400 X2=MX+85*SIN(I/3)
410 Y2=MY+29*COS(I/3)
420 PUT (2*XA2,YA2),M2:PUT (2*X2,Y2),M2

```

```
430 XA2=X2:YA2=Y2
440 X3=MX+95*SIN(I/3.5)
450 Y3=MY+33*COS(I/3.5)
460 PUT (2*XA3, YA3), M3: PUT (2*X3, Y3), M3
470 XA3=X3:YA3=Y3
480 X4=MX+120*SIN(I/6)
490 Y4=MY+41*COS(I/6)
500 PUT (2*XA4, YA4), M4: PUT (2*X4, Y4), M4
510 XA4=X4:YA4=Y4
520 X5=MX+150*SIN(I/8)
530 Y5=MY+51*COS(I/8)
540 PUT (2*XA5, YA5), M5: PUT (2*X5, Y5), M5
550 XA5=X5:YA5=Y5
560 NEXT
```

2.16 Vektorértékű függvények

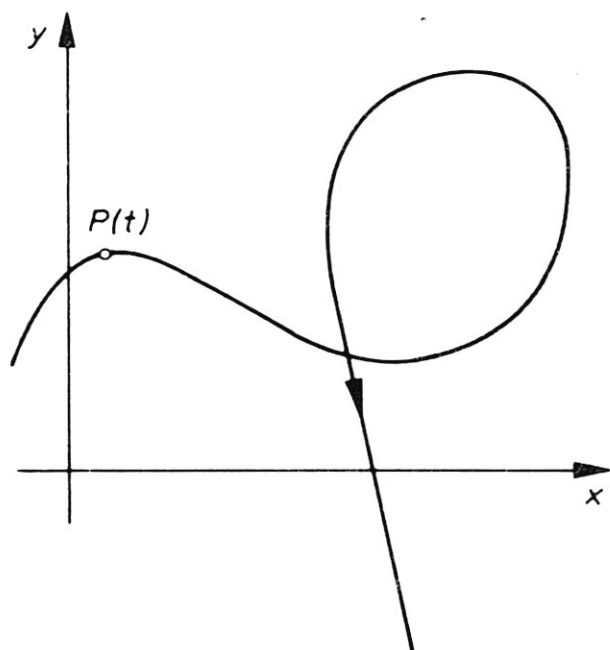
A következőkben azzal fogunk foglalkozni, hogy miképp lehet egy tetszőleges térgörbét leírni. Megvizsgáljuk azt is, hogy a csak néhány érbeli ponttal megadott görbe által meghatározott töröttvonal — nemlineáris interpolációval — hogyan számítható. Ebben az alfejezetben meglehetősen sok, nem éppen könnyű matematikát kell alkalmaznunk. Ha az Olvasónak a matematikai alapokra nincs égetően szüksége, javasoljuk, ugorja át az interpolációs görbék elméleti levezetését. A programmal jól lehet dolgozni akkor is, ha az elméleti alapok hiányoznak. A térgörbe paraméteres alakjának a megértése azonban feltétel, ezért a program bevezetőjét alaposan el kell olvasni!

2.16.1 FÜGGVÉNYEK PARAMÉTERES ALAKJA

A paraméteres alakot már akkor megismertük, amikor a körök és az ellipszisek programozásával foglalkoztunk. A függvény paraméteres alakja egy görbefogalom, ami a kinematikából származik. A $P(t)$ függvény a P pont térbeli mozgását írja le, megadva a P pont helyét az idő függvényében, azaz a t időpillanatban. A

$$t \rightarrow \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}.$$

összefüggés alapján pl. leírható a következő függvény:

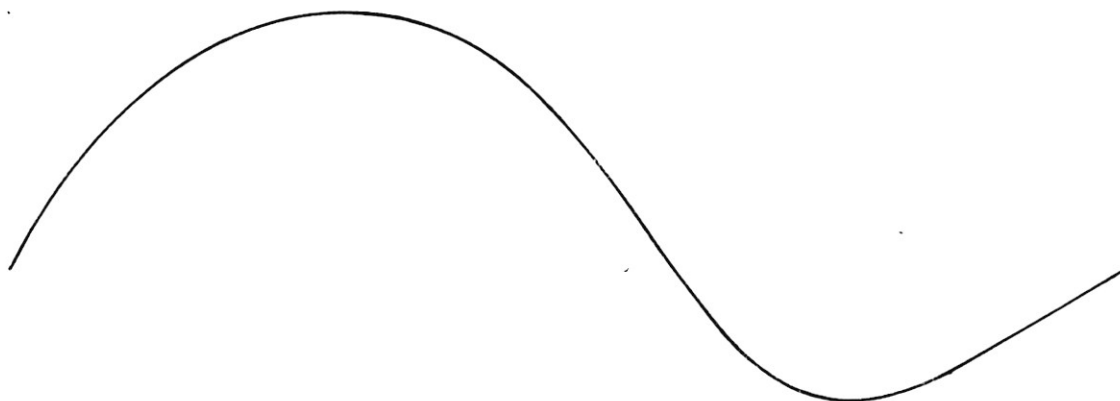


Tehát olyan függvények grafikonjáról lesz szó, amelyek egy számhoz (időponthoz) egy vektort rendelnek. A vektort koordinátaival írtuk fel.

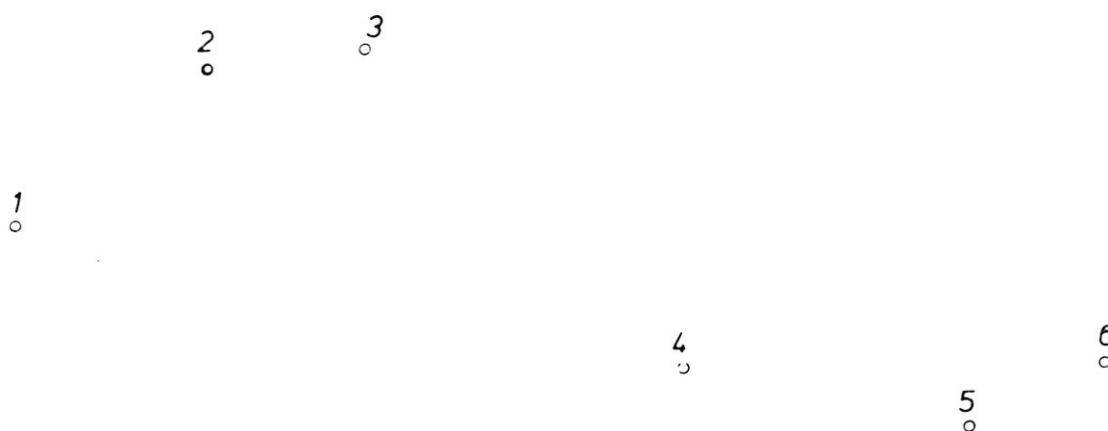
2.16.2 NEMLINEÁRIS INTERPOLÁCIÓ

Tételezzük fel, hogy egy olyan görbét akarunk leírni, amelyről lényegében alig tudunk valamit. Mindössze néhány pontja ismert, ill. e pontok azon sorrendje, amely sorrendben a görbe a pontokon átmegy.

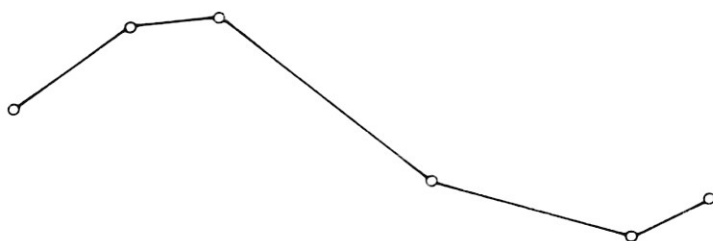
Tekintsünk egy példát a probléma megértésére! A következő görbét akarjuk leírni:



A görbéből azonban csupán néhány pont ismert:



A legegyszerűbb megoldás a görbe pontok közötti alakjának meghatározására az lenne, ha a pontokat egyenes szakaszokkal kötnénk össze.



Ez a megoldás sok esetben meg is felel a céloknak. Miképp kell azonban eljárni akkor, ha jobb közelítést akarunk elérni? Ez azért is érdekes, mert a természeti jelenségeket leíró görbék nem „cickakkos”, hanem általában „sima” lefutásúak. Ezért tehát a következő lépésben a görbét „lekerekítjük”, pl. oly módon, hogy 3 pontra egy körivet — vagy valami más alkalmas görbét — fektetünk, és ezáltal további pontokat határozunk meg:



Az eredményül kapott görbe már sokkal „kerebbebbnek” néz ki:



Ezután az a kérdés, hogy melyik függvény alkalmas arra, hogy a pontokat legjobban közelíthessük? A leggyakrabban a *Lagrange*-féle interpolációs eljárást alkalmazzák a függvény meghatározására.

Ha az ismert pontok száma n , akkor felírunk egy olyan $(n-1)$ -ad fokú polinomot, amely az összes pontot tartalmazza. Pl. a következő pontokat ismerjük:

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix}; \quad k = 1, 2, 3.$$

Ekkor egy másodfokú polinomot írhatunk fel:

$$ax^2 + bx + c = y$$

Az x_k és az y_k behelyettesítésével a következő egyenleteket kapjuk:

$$ax_1^2 + bx_1 + c = y_1;$$

$$ax_2^2 + bx_2 + c = y_2;$$

$$ax_3^2 + bx_3 + c = y_3.$$

Az egyenletrendszer mátrix alakban:

$$\mathbf{Y} = \mathbf{M} \cdot \mathbf{A},$$

ahol

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}; \quad \mathbf{M} = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \quad \text{és} \quad \mathbf{A} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}.$$

Ha $x_1 \neq x_2 \neq x_3$, akkor az M mátrix sor-, ill. oszlopvektorai lineárisan függetlenek, és így a mátrix rangja $r=3$. Ebből az következik, hogy az A vektor az

$$A = M^{-1} \cdot Y$$

összefüggéssel egyértelműen meghatározható. Számos okunk van arra, hogy az eljárást ebben az általános formában ne alkalmazzuk. Csak a két legfontosabbat említve: könnyen belátható, hogy amennyiben az ismert pontok száma több mint 4, az M^{-1} inverz mátrix meghatározása jelentős problémát okoz. A másik ok az, hogy a magasabb fokszámú polinomok gyakran meredek hullámot vetnek, pedig a görbénket éppenhogy simítani akarjuk. Milyen követelményeket kell tehát a megfelelő függvényekkel szemben támasztani? A függvényeknek

- 1) csak 1 maximumuk és csak 1 minimumuk legyen;
- 2) legyen inflexiós pontjuk.

Ezek a tulajdonságok jellemzik pl. a harmadfokú polinomokat.

Nézzük meg először a harmadfokú polinomok egy speciális fajtáját. A binominális tétel alapján:

$$\begin{aligned} 1 &= ((1-u) + u)^3 = \sum_{r=0}^3 \binom{3}{r} (1-u)^{3-r} u^r = \\ &= (1-u)^3 + 3(1-u)^2 u + 3(1-u) u^2 + u^3 \end{aligned}$$

Az összegzendő

$$B_r(u) = \binom{3}{r} (1-u)^{3-r} u^r$$

polinomokat nevezzük *Bernstein*-féle polinomoknak, amelyek könnyen ellenőrizhető tulajdonságai a következők:

- 1) $B_r(u)$ -nak az $u = 0$ r -szeres gyöke;
- 2) $B_r(u)$ -nak az $u = 1$ $(n-r)$ -szeres gyöke;
- 3) $B_r(u)$ -nak csak egy maximuma van az $I = [0; 1]$ intervallumban, mégpedig $u = r/3$ -nál.

Ezért a *Bernstein*-féle polinommal rendszerint az $I = [0; 1]$ intervallumban dolgozunk.

Az 1) és 2) tulajdonságok miatt a $B_r(u)$ polinomok lineárisan függetlenek és ezért bázist képeznek a harmadfokú polinomok terében. Egy tetszőleges, legfeljebb harmadfokú poli(u) polinom tehát előállítható a $B_r(u)$ polinomok lineáris kombinációjaként.

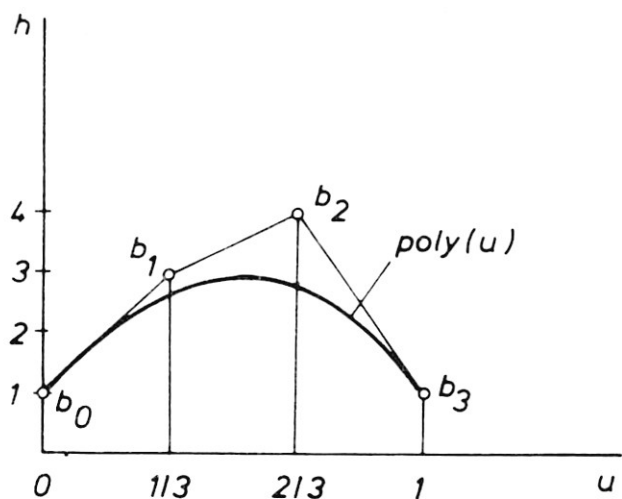
$$\text{poli}(u) = b_0 B_0(u) + b_1 B_1(u) + b_2 B_2(u) + b_3 B_3(u)$$

A b_k szorzótényezőket interpolációs pontoknak, magát a polinomot interpolációs polinomnak és az $(i/n; b_k)$ töréspontokkal jellemzett poligont interpolációs poligonnak nevezzük.

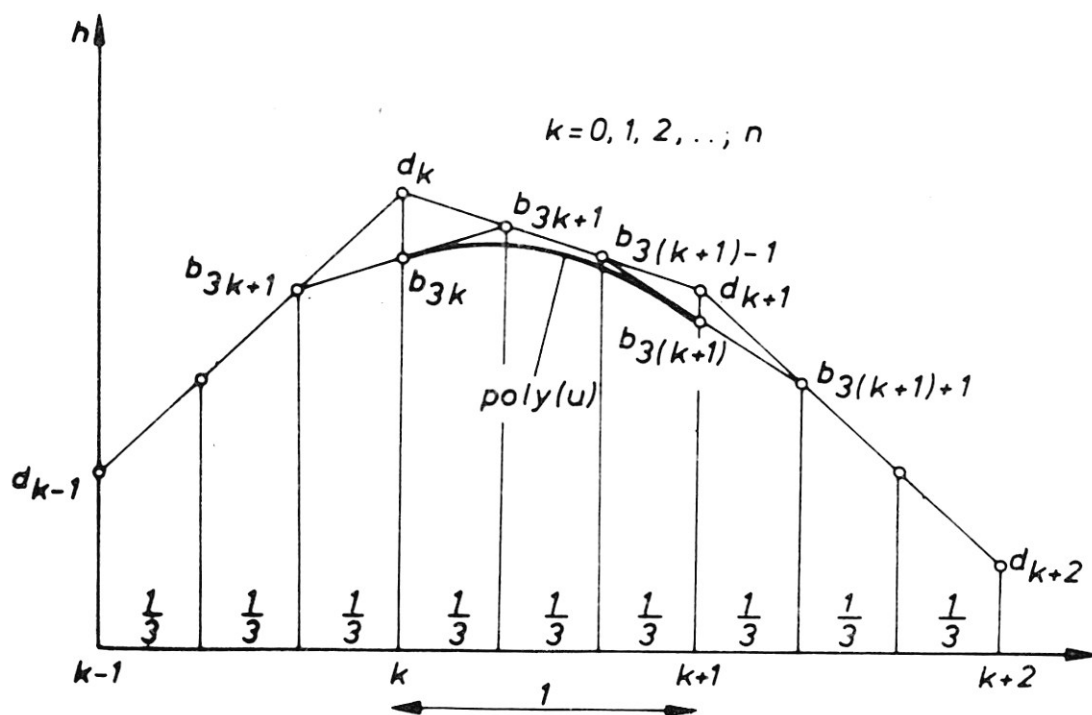
Példaként nézzük a következő harmadfokú polinomot:

$$\text{poli}(u) = 1(1-u)^3 + 3 \cdot 3(1-u)^2 u + 4 \cdot 3(1-u) u^2 + 1 u^3.$$

Ekkor az interpolációs pontok ($b_0 = 1, b_1 = 3, b_2 = 4, b_3 = 1$) és az interpolációs poligon a következő módon ábrázolható:



Nézzük meg más szemmel is az ábrát! A $\text{poly}(u)$ függvény simítja az interpolációs poligont. Amint ez könnyen kimutatható, a b_0 és a b_3 pontban a $\text{poly}(u)$ polinom megegyezik az interpolációs poligonnal, egészen a második deriváltig. Úgy mondjuk, hogy a $\text{poly}(u)$ polinom az interpolációs poligon *spline*-interpolációját alkotja. Egy poligonnak a $P_k = (k; d_k)$ pontokkal való interpolációjával, az interpolációs poligon segítségével egy sokszögvonallal simítható.



Nem kívánunk mélyebben belemerülni a matematikai elméletbe, ezért fogadjuk el posztulátumként a következőket.

A d_k és a d_{k+1} polinomok között akarunk interpolálni. Mivel egy *spline*-hoz 4 interpolációs együttható szükséges, szerkesszük meg a $b_{3k}, b_{3k+1}, b_{3(k+1)}$ pontokat, amelyek megfelelnek a b_0, b_1, b_2 és b_3 interpolációs pontoknak. (Létezésük összefügg a közelítendő függvény b_{3k} és $b_{3(k+1)}$ pontokban való kétszeres, folytonos differenciálhatóságával.)

A következő egyenletek írhatók fel:

$$b_{3k} = \frac{1}{6}(d_{k-1} + 4d_k + d_{k+1});$$

$$b_{3k+1} = \frac{1}{3}(2d_k + d_{k+1});$$

$$b_{3(k+1)-1} = \frac{1}{3}(d_k + 2d_{k+1});$$

$$b_{3(k+1)} = \frac{1}{6}(d_k + 4d_{k+1} + d_{k+2}).$$

A közelítő interpolációs függvény tehát:

$$\begin{aligned} \text{poli}(u) &= b_{3k}(1-u)^3 + b_{3k+1} \cdot 3(1-u)^2u + b_{3(k+1)-1} \cdot \\ &\cdot 3(1-u)u^2 + b_{3(k+1)}u^3 = \dots = \frac{1}{6}(1-u)^3d_{k-1} + \\ &+ \left(\frac{1}{2}u^3 - u^2 + \frac{4}{6}\right)d_k + \left(-\frac{1}{2}u^3 + \frac{1}{2}u^2 + \frac{1}{2}u + \frac{1}{6}\right)d_{k+1} + \frac{1}{6}u^3d_{k+2}. \end{aligned}$$

Vezessük be a következő rövidítéseket:

$$D_1(u) = \frac{1}{6}(1-u)^3$$

$$D_2(u) = \frac{1}{2}u^3 - u^2 + \frac{4}{6}$$

$$D_3(u) = -\frac{1}{2}u^3 + \frac{1}{2}u^2 + \frac{1}{2}u + \frac{1}{6}$$

$$D_4(u) = \frac{1}{6}u^3$$

Mátrix alakban felírva (nemsokára megmutatjuk, hogy miért éppen ezt az alakot választottuk):

$$h = \text{poli}(u) = (d_{k-1}d_kd_{k+1}d_{k+2}) \cdot \begin{pmatrix} D_1(u) \\ D_2(u) \\ D_3(u) \\ D_4(u) \end{pmatrix}; \quad u \in [0; 1]$$

Térjünk most vissza interpolációs feladatunk kiindulópontjára!

Célunk az volt, hogy az $(x_k; y_k)$, $k = 1, 2, \dots, n$ pontokból simítsuk a poligont. Könnyen belátható, hogy ezt a következő képlet felírásával érhetjük el (a paraméteres alakban az u paraméter veszi át t szerepét):

$$\begin{pmatrix} x(u) \\ y(u) \end{pmatrix} = \begin{bmatrix} x_{k-1} & x_k & x_{k+1} & x_{k+2} \\ y_{k-1} & y_k & y_{k+1} & y_{k+2} \end{bmatrix} \cdot \begin{pmatrix} D_1(u) \\ D_2(u) \\ D_3(u) \\ D_4(u) \end{pmatrix}; \quad u \in [0; 1]$$

Ez a sík (R^2) interpolációs görbéje. Hasonlóan megadható összefüggés az R^n n -dimenziós térre:

$$\begin{pmatrix} x_1(u) \\ x_2(u) \\ \vdots \\ x_n(u) \end{pmatrix} = \begin{bmatrix} x_{1(k-1)} & x_{1(k)} & x_{1(k+1)} & x_{1(k+2)} \\ x_{2(k-1)} & x_{2(k)} & x_{2(k+1)} & x_{2(k+2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n(k-1)} & x_{n(k)} & x_{n(k+1)} & x_{n(k+2)} \end{bmatrix} \cdot \begin{pmatrix} D_1(u) \\ D_2(u) \\ D_3(u) \\ D_4(u) \end{pmatrix}$$

Ez a képlet a d_0 és a d_1 pontok közötti interpolációra nem érvényes, ugyanígy a d_{n-1} és a d_n pontok között sem. Itt a következő peremfeltétel érvényes:

$$b_0 = d_0 \quad \text{és} \quad b_{3m} = d_m.$$

A

$$D_{11}(u) = \frac{1}{6}u^3 - u + 1$$

$$D_{12}(u) = \frac{1}{3}u^3 + u$$

$$D_{13}(u) = \frac{1}{6}u^3$$

Jelölésekkel a d_0 és a d_1 pontok közötti interpolációra kapjuk:

$$h = (d_0 d_1 d_2) \cdot \begin{pmatrix} D_{11}(u) \\ D_{12}(u) \\ D_{13}(u) \end{pmatrix}; \quad u \in [0; 1]$$

A d_{n-1} és a d_n pontok közötti interpolációra pedig:

$$h = (d_n d_{n-1} d_{n-2}) \cdot \begin{pmatrix} D_{11}(u) \\ D_{12}(u) \\ D_{13}(u) \end{pmatrix}.$$

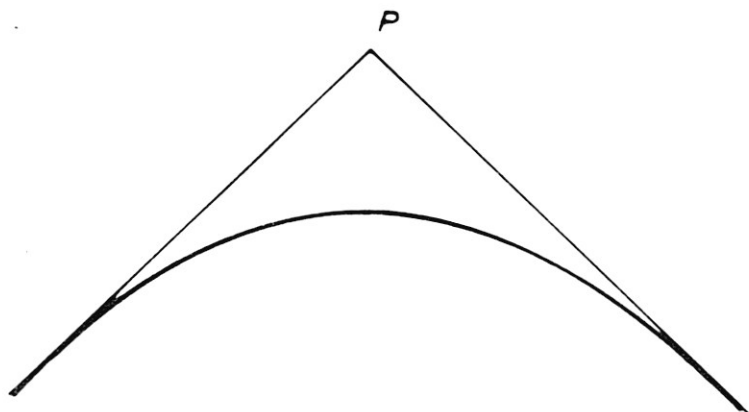
A közelítő függvény a b_{3k} és a $b_{3(k+1)}$ pontokban egészen a második deriváltig megegyezik az interpolációs poligonnal. Ezért lehet két egymást követő d_k pont között közelíteni, majd ezeket a görbedarabokat egymáshoz fűzni anélkül, hogy az így előállított görbe töréspontokat tartalmazna, ill. hogy a görbének folytonos, de nem differenciálható helyei lennének.

A következő program alkalmazza ezt az eljárást egy sokszögvonalnak a $[-10; -10] - [10; 10]$ tartományban való simításra. Ennél az eljárásnál a közelítő görbe nem megy át az adott pontokon, ezért nagyon jól alkalmazható a mérésel nyert, tehát természetüknél fogva hibával terhelt mennyiségek közelítésére. Ha a közelítő görbének a megadott pontokon át kell mennie, akkor a d_k értékek helyett a b_{3k} értékeket kell megadni és a d_k pontokat kell megszerkeszteni.

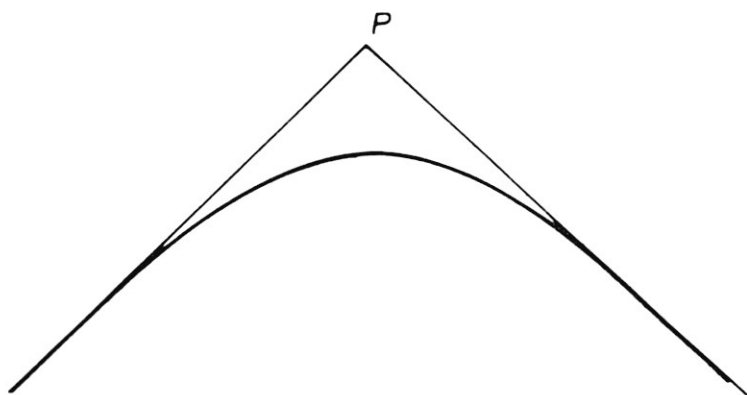
Útmutatás

A pontok számát és a pontok koordinátáit DATA-sorokban adtuk meg a program végén. A megadott értékeket és azok számát az Olvasó tetszőlegesen megváltoztathatja. A program mindig a poligon síkjának egy négyzet alakú kivágatát mutatja. A négyzet élhosszúsága 1,

középpontja az $(m_x; m_y)$ pont. Ha a P_1 pont egybeesik P_n -nel, azaz zárt poligonról van szó, akkor csekély mértékben megváltozik az eljárás. Ez esetben ui. nincs a poligonnak kezdete és vége, tehát az egész poligont az általános képlettel interpolálhatjuk. A kezdő- és végpontra vonatkozó speciális esetet itt tehát elfelejthetjük. Egyszerűen azt tesszük, hogy n -et 2-vel megnöveljük $n+2$ -re és a P_2, P_3 pontokat mint P_{n+1} és P_{n+2} a poligon végéhez fűzzük. A simítás szokásos esetben így néz ki:

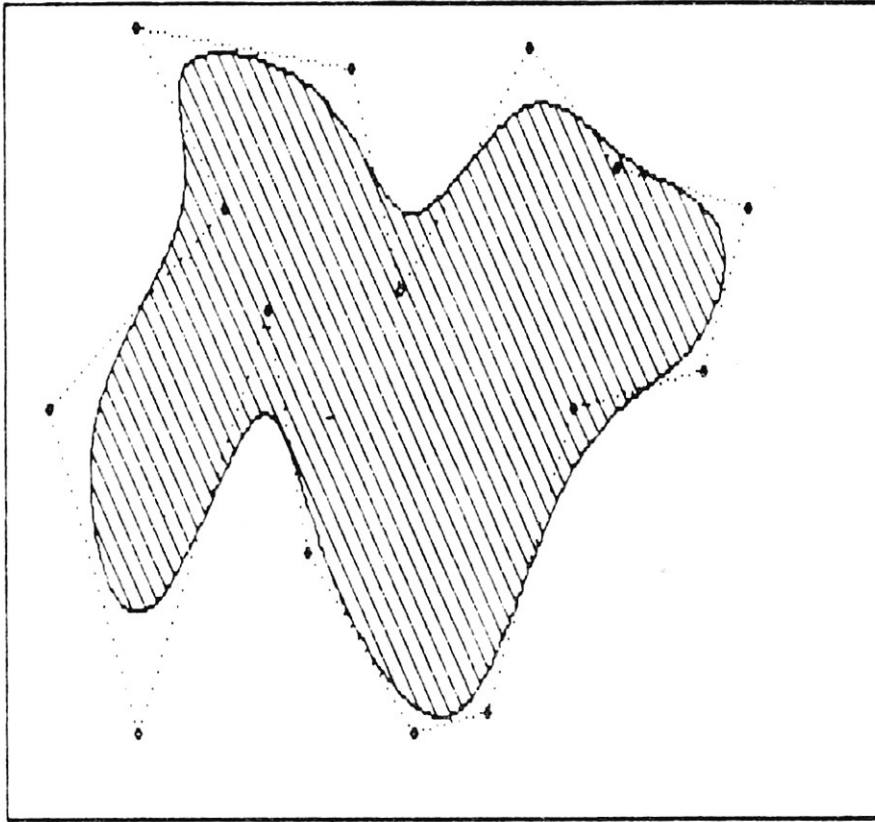


Ha a P pontot súlyozni akarjuk, akkor az eljárás kétszeri alkalmazásával már ilyen közelítő görbét kapunk:



Programleírás

Sor	Magyarázat
120—140	A kivágat meghatározása
150—180	A képernyő-paraméterek beállítása
190—250	A közelítő függvény definiálása
260—290	A koordináták olvasása és a poligon megrajzolása
300—350	Az első <i>spline</i> megrajzolása
360—430	A középső <i>spline</i> -ok megrajzolása
440—490	Az utolsó <i>spline</i> megrajzolása
500—520	DATA-sorok
530—620	A másik kivágat



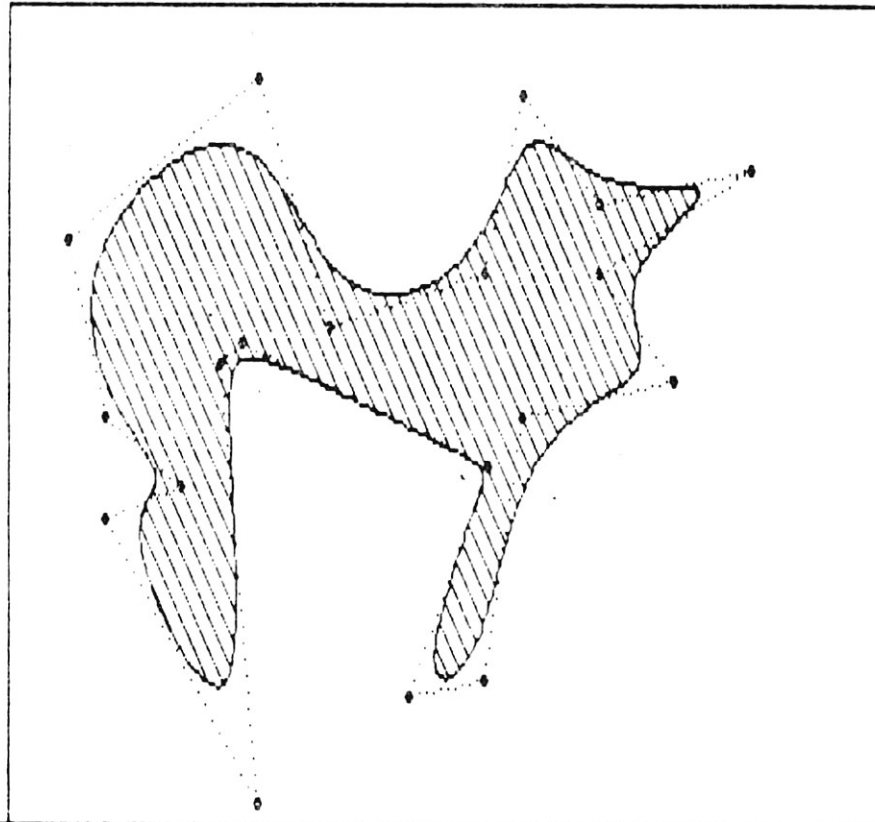
Poligon interpolációja

Élhosszúság

? 20

Középpont

? 0,0



Poligon interpolációja

Élhosszúság

? 23

Középpont

? 0,0

FIGYELEM: a program nem tartalmaz magyarázatokat, ezért a gyakorlati rész bevezetőjét
FELTÉTLENÜL el kell olvasni!

```
50 REM ... P18 ...
100 REM ..... Poligon simitasa
110 CLS:PRINT "Poligon interpolacioja"
120 INPUT "Negyzet elhosszusaga";L
130 INPUT "Kozeppont (MX,MY)";MX,MY
140 A=MX-L/2:B=MY-L/2:C=MX+L/2:D=MY+L/2
150 READ N:DIM X(N+3),Y(N+3)
160 CLS:SCREEN 2:KEY OFF:MC (150,1)-(638,198),,1
170 PRINT "-----s";:PRINT "poligon":
    PRINT "interpolacioja"
180 WINDOW (A,B)-(C,D)
190 DEF FNB1(U)=((1-U)^3)/6
200 DEF FNB2(U)=(U^3)/2-U^2+4/6
210 DEF FNB3(U)=(-U^3)/2+(U^2)/2+U/2+1/6
220 DEF FNB4(U)=(U^3)/6
230 DEF FNB11(U)=(U^3)/6-U+1
240 DEF FNB12(U)=-(U^3)/3+U
250 DEF FNB13(U)=(U^3)/6
260 FOR I=1 TO N:READ X(I),Y(I)
270 IF I>1 THEN LINE -(X(I),Y(I)),, ,&H8888
280 CIRCLE (X(I),Y(I)),.1
290 NEXT:IF X(1)=X(N) AND Y(1)=Y(N) THEN GP=1:
    GOSUB 620:GOTO 360
300 I=2:PSET (X(1),Y(1))
310 FOR U=0 TO 1 STEP .03
320 X=X(I-1)*FNB11(U)+X(I)*FNB12(U)+X(I+1)*FNB13(U)
330 Y=Y(I-1)*FNB11(U)+Y(I)*FNB12(U)+Y(I+1)*FNB13(U)
340 LINE -(X,Y)
350 NEXT
360 IF GP=1 THEN N=N+2
370 FOR I=2 TO N-2
380 FOR U=0 TO 1 STEP .03
390 X=X(I-1)*FNB1(U)+X(I)*FNB2(U)+X(I+1)*FNB3(U)+
    X(I+2)*FNB4(U)
400 Y=Y(I-1)*FNB1(U)+Y(I)*FNB2(U)+Y(I+1)*FNB3(U)+
    Y(I+2)*FNB4(U)
410 IF GP=1 AND I=2 AND U=0 THEN PSET (X,Y):GOTO 430
420 LINE -(X,Y)
430 NEXT:NEXT:IF GP=1 THEN 530
440 I=N-1
450 FOR U=0 TO 1 STEP .03
460 X=X(I+1)*FNB11(1-U)+X(I)*FNB12(1-U)+
    X(I-1)*FNB13(1-U)
470 Y=Y(I+1)*FNB11(1-U)+Y(I)*FNB12(1-U)+
    Y(I-1)*FNB13(1-U)
480 LINE -(X,Y)
```

```

490 NEXT
500 DATA 16:REM ..... Az adatok szama
510 DATA -9,0,-7,-8,-4,2.5,-3.1,-3.5,-.7,-8,1,-7.5,3,
    0,6,1,7,5,4,6
520 DATA 2,9,-1,3,-2.1,8.5,-7,9.5,-5,5,-9,0
530 PRINT :PRINT "Elhosszusag"
540 PRINT " " :LOCATE 6,1:INPUT L
550 PRINT "Kozeppont"
560 PRINT " " :LOCATE 8,1:INPUT MX,MY
570 A=MX-L/2:B=MY-L/2:C=MX+L/2:D=MY+L/2
580 LINE (A,B)-(C,D),,B
590 INPUT "OK";Q$
600 IF Q$="N" THEN LOCATE 4,1:GOTO 530
610 RESTORE:MC :GOTO 150
620 X(N+1)=X(2):X(N+2)=X(3):Y(N+1)=Y(2):Y(N+2)=Y(3):
    RETURN

```

Jegyzetek

Ára: 380,-Ft

Markus Weber könyve a számítógépes grafika, elsősorban a 3 dimenziós ábrázolás lehetőségeivel ismerteti meg azokat az érdeklődő Olvasókat, akik élni kívánnak IBM PC, illetve ezzel kompatibilis számítógépük grafikus lehetőségeivel.

- Hogyan rajzolhatjuk meg egy ház perspektivikus képét?
- Mit kell tennünk ahhoz, hogy megjelenítsük a statisztikai adatokból nyerhető információkat?
- Mi az üzleti grafika?
- Milyen képet mutat számítógépünk képernyőjén a St. Helen hegy a vulkánkitörés előtt és után?

Az Olvasó a könyv tanulmányozása során elsajátíthatja a grafika programozásának tudnivalóit. Ehhez nyújt segítséget a matematikai alapokat összefoglaló rész, illetve a grafikus ábrázolás gyakorlati alkalmazásaira bemutatott több BASIC-nyelvű programlista.