

12. Számítástechnika és Analízis

Simonovits Miklós:

Függelék a Laczkovich-T. Sós Analízis tankönyvhöz,
Bővített változat, a CD-re

Részletesebb változat, CD-re

Ebben a változatban figyelmen kívül hagyhatjuk azokat a kompromisszumokat, amelyeket a tankönyvbeli változatban részben az arányok betartása, részben a terjedelmi korlátozások miatt vállalnunk kellett.

- (a) Több feladatot fogalmazunk meg.
- (b) Kicsit részletesebb magyarázatokat mellékelünk.
- (c) Nagyobb ábrákat használunk,
és azok jobban hasonlítanak az eredetire.
- (d) Néhány további kulcsszó elmagyarázására is kitérünk.
- (e) Ha lehet, nem törjük el a programjainkat lapváltással.

12.1. Bevezetés

A függelék célja, hogy rámutassunk arra, hogyan használható a számítógép

- (a) az analízisbeli fogalmak *szemléltetésére*,
- (b) analízisbeli feladatok *megoldásának megsejtésére*.

Az (a)-ban tehát elsősorban arról van szó, hogy egy definícióra vagy tételre gyártunk könnyen illusztrációkat, míg (b)-ben arról írunk, hogy van egy feladat, amelynek nem tudjuk a megoldását, de hogy megsejtsük, kísérleteket végzünk.

A gép jól használható számelméletben, analízisben, fizikában: kis energiával hatékony programokat írhatunk. Más területeken, pl. kombinatorikában a programok megírása több gondot jelent és kevésbé látványos az eredmény.

Itt az **analízis illusztrálására szorítkozunk**. Ezen belül elsősorban a

1. határérték megsejtésével,
2. függvények ábrázolásával, elemzésével,
3. polinom-approximációval foglalkozunk.

Számítástechnika oldaláról az alábbiakat vizsgáljuk meg:

- Kész programokat hogyan és mire használhatunk?

- Mikor kell komolyabb programcsomagokat használni?
- Mire használhatunk magunk által írt egyszerű programokat?
- Mely fogalmaknál hasznos, ha számítógépet használunk?
- Hogyan csap be a számítógép, ha nem vigyázunk?

Mindezt elsősorban a matematikai oldalról, másodsorban a számítástechnikában a MAPLE matematikai programcsomag és a QBASIC programnyelv szempontjából vizsgáljuk meg.¹

*Nem célunk itt a számítógépes ismeretek taglalása, amit a számítástechnikáról leírnak, az inkább csak emlékeztető. Célunk, hogy megszeretessük a számítógép használatát, és megválaszoljuk a fenti kérdéssort. Filozófiánk, hogy nagyon kevés számítástechnikai ismerettel is nagyon sok dolgot kiszámolhatunk a tudományokban. Itteni programjaink többnyire egy rövid INPUT résszel, egy rövid output résszel és egy 3-6 soros maggal rendelkeznek: valóban nagyon egyszerűek. Ugyanilyen szemléletű, de sokkal részletesebb anyag található Fried Katalin és Simonovits Miklós 2005-ben a TypoTeX-nél megjelent, *A problémamegoldás számítógépes iskolája c.* könyvében.*

12.2. Milyen profi programokat használjunk?

A címben feltett kérdésre sokféle válasz adható, és remélhetően mindegyik válasz gyorsan avul el. Az alábbiakat érdemes figyelembe venni:

1. Ha *egy* ilyen programot megtanulunk, utána bármelyik másikat már sokkal könnyebben tanuljuk meg: ilyen értelemben beszélhetünk nemelavuló tudásról.
2. Mindig olyan programot érdemes használni, amelyiket a környezetünkben mások is használnak, és amelyikről megkérdezhetünk valakit, ha elakadunk. Ez utóbbi kérdés sokkal fontosabb és bonyolultabb, mint gondolnánk.
 - (a) Vannak akik szakkönyvek alapján szeretik elsajátítani a komputer-programok használatát. Ilyen olvasóból is kétfajta van: az egyik mindig csak a minimumot olvassa el abból ami szükséges. Ha elakad, két perc alatt megtalálja, amire szüksége van, azt elolvassa, majd azonnal visszarakja a könyvet a helyére. A másik

¹Választhattuk volna a MATHEMATICA programcsomagot, és a PASCAL programnyelvet is, de – ami a PASCAL-t illeti, – a számunkra szükséges nagyon egyszerű programoknál a QBASIC használata a legegyszerűbb. Emellett a programok átírása PASCAL-ra majdnem automatikus. A könyvhöz tartozó CD-n találunk néhány PASCAL programot. A MATHEMATICA vagy a DERIVE programokkal *is* elsősorban a helyhiány miatt nem foglalkozunk.

típus előbb alaposan átolvass egy szakkönyvet és csak utána kezd dolgozni a programmal.

- (b) Sokan leülnek mások mellé, őket megfigyelve elkezdnek próbálkozni, és ha elakadnak, megkérdezznek valakit, majd annak segítségével tovább folytatják a munkájukat.

Ahhoz, hogy eldöntsük, melyik programot használjuk, *ismernünk kell magunkat, a környezetünket*, és az elérhető könyveket. Ennek alapján kell eldöntenünk, mit választunk.

3. A következő programcsomagokat ajánljuk az Olvasó figyelmébe:

- (a) MAPLE: A Waterloo-i egyetemen (Kanada) fejlesztették ki.² Mindent tud, amire egy matematikusnak szüksége lehet.
- (b) MATHEMATICA: Wolfram Research által kifejlesztett programcsomag, kb. ugyanazt tudja, mint a MAPLE.
- (c) DERIVE: Egy nagyon kicsi, de nagyon hatékony program, korai, ún. DOS-os változatai egy floppyra ráférnek. DOS-os verziója kicsit elavult, de mindent elvégez, amit szeretnénk, pontosan, gyorsan, szépen. Nagyon könnyen elsajátítható a használata. (Ha több helyünk volna, nagyon megérdemelne itt egy függelék.)

A felépítés. A fejezet első részében csak a MAPLE és hasonló programok szempontjából vizsgáljuk a számítástechnika felhasználását. A váltás a 12.5 alfejezetben kezdődik. Innen egy ideig a QBASIC programok alkalmazásáról írunk, majd a befejező részben visszatérünk a MAPLE használatára.

Kiegészítő anyag található még Simonovits Miklós hon-lapján:

<http://www.renyi.hu/~miki/tankonyv.htm> -en.

12.2.1. Kipróbáljuk a MAPLE-t



Analízisben gyakran segíthet egy jó ábra egy fogalom megértésében, vagy bizonyos esetekben az igazság megsejtésében. Ehhez gyakran segít a megfelelő program. Az itt következő első részt kedvcsinálónak szánjuk, bepillantás a MAPLE világába.

Minden alaposabb magyarázat nélkül elmondjuk, hogy ha az f függvényt akarjuk ábrázolni az $[a, b]$ intervallumon, akkor a $>$ „prompt” után

plot(f,x=a..b);

²Elnevezése is erre utal: MAPLE a kanadai tölgy-juhar, melynek levele Kanada szimbóluma.

-t kell beírni, pl. $f(x) = x \sin x$, $a = 0$, $b = 4\pi$ esetén

```
plot(x*sin(x),x=0..12.56);
```

vagy

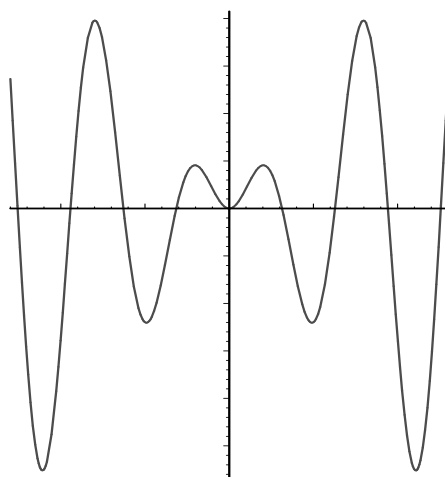
```
plot(x*sin(x),x=0..4*Pi);
```

stb. alakban.

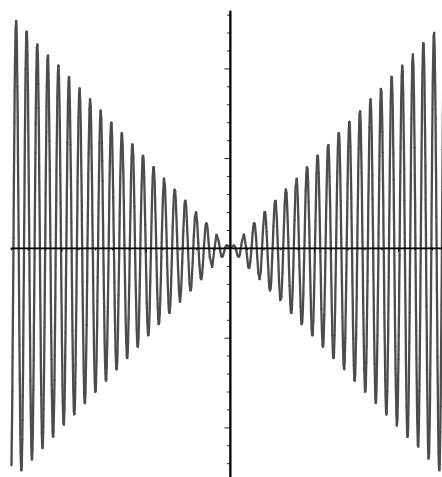
Intuición és képi megjelenítés. Függvényekkel való ismerkedésünkhöz a MAPLE egy nagyerejű függvénykirajzoló program is. Az alábbi 3 kép pl. a MAPLE program segítségével készült, és az $x \sin x$ függvényt ábrázolja, a $[-13, 13]$, $[-130, 130]$, ill. a $[-65, 65]$ intervallumokon. A 1/a ábrát pl. a

```
> plot(x*sin(x),x=-13..13);
```

adja.³ Legelőször túl rövid intervallumot választottunk, nem láttuk, amit látni akartunk volna. Ezért megtízszereztük az intervallumot, amitől túl sok hullámot kaptunk, és hogy egy kicsit jobban lássuk a részleteket, visszacsökkentettük az intervallum hosszát.

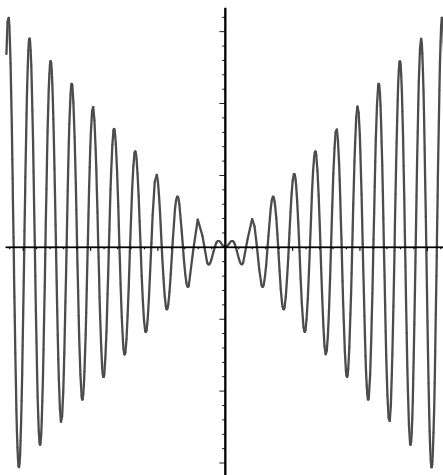


$x \in [-13, 13]$
1/a. ábra: $x \sin x$



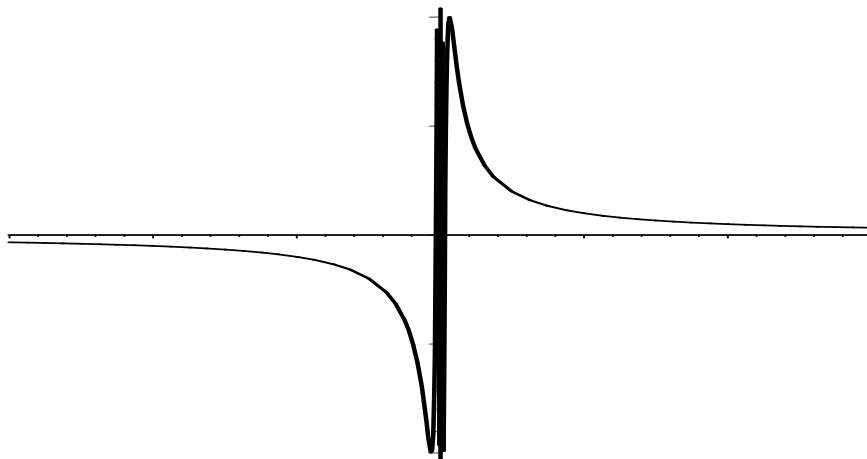
$x \in [-130, 130]$
1/b. ábra: $x \sin x$

³A „>” promptot nem mi írjuk oda, azt a MAPLE adja.



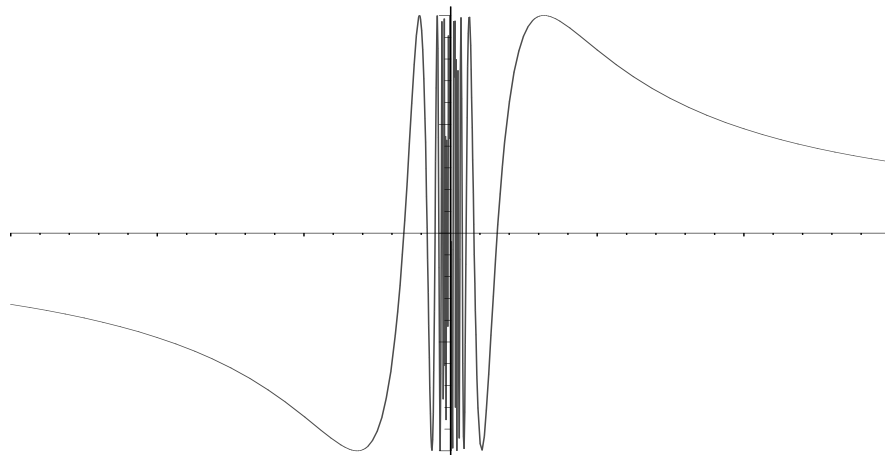
1/c. ábra: $x \sin x$
 $x \in [-65, 65]$

Itt meg a $\sin \frac{1}{x}$ függvényt ábrázoltuk.



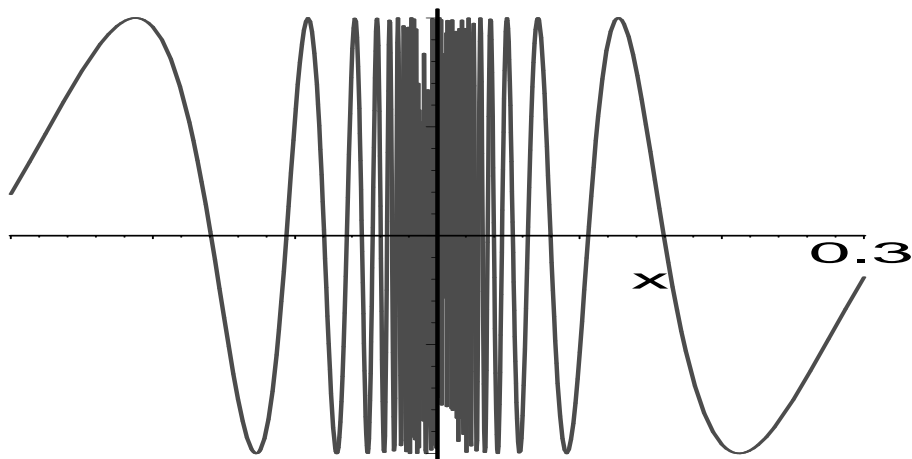
2. ábra: $\sin \frac{1}{x}$, $x \in [-30, 30]$

Az ábrán nem igazán látható, mi történik a 0 körül. Ezért „rá-zoomolunk”: tizedakkora intervallumot veszünk.

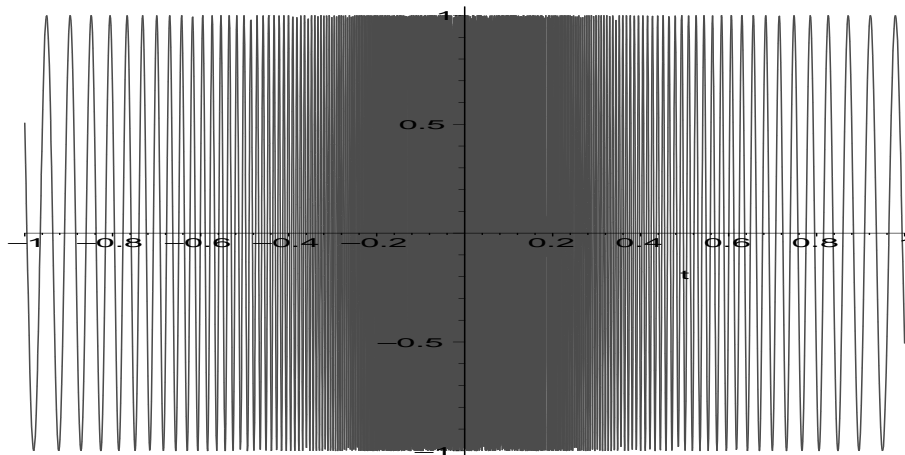


3. ábra: $\sin \frac{1}{x}$, $x \in [-3, 3]$

Mivel a hullámok a 0 körül nagyon besűrűsödtek, az alábbiakban kicsit átskáláztuk: x helyébe $x/100$ -at írtunk:



4. ábra: $\sin \frac{1}{x}$, $x \in [-0.3, 0.3]$



5. ábra: $\sin \frac{100}{x}$, $x \in [-1, 1]$

Itt tehát ugyanazt a függvényt vizsgáltuk meg „mikroszkópon” keresztül, csak a (vízszintes) nagyítás változott. (A 5. ábra, főleg nyomtatásban eredetileg teljesen befeketedett. Kérdés, hogy ez siker-e. Kicsit szürkébbre vettük, hogy a részletek kevésbé vesszenek el.)

Hogyan készítettük a fenti ábrákat? Egyebek között ezt szándékozunk bemutatni a következőkben.

12.3. A MAPLE: Első lépések



Itt a MAPLE 8-at ismertetjük. Az ilyen programoknak különböző verziói vannak, részben a megcélzott közönség (diákok/oktatók vs. profik), részben a kibocsájtás időpontja szerint. Jelenleg bármelyik verzió a MAPLE 5-től felfele megfelel számunkra, bár vannak közöttük apróbb eltérések. (A CD anyaga LINUX-ban, Maple8-cal készült.)

12.3.1. HELP használata

Ha egy ilyen programmal akarunk dolgozni, legelőször azt kell megtudnunk, hogyan kell elindítanunk és hogyan kell kiszállnunk belőle. A MAPLE esetén, LINUX-ban az **xmacle** parancs indítja el, WINDOWS-ban a megfelelő ikon.

Bármilyen, kicsit is bonyolultabb programot használunk, nagyon fontos a HELP (magyarul SÚGÓ) használatának elsajátítása. Erre főleg akkor van szükségünk, ha nincs a programhoz megfelelő leírásunk. Ezért érdemes a program használata előtt a HELP-jével is megismerkednünk. A MAPLE HELP-je nagyon jó. Van egy „New User’s Tour” (a MAPLE bemutatása az

újoncoknak), illetve Topic Search⁴. Persze ezek használatához kell egy kicsit angolul is tudnunk, és pl. a Lagrange interpolációt nem Lagrange-nál, hanem az Interpolation-nál találjuk: kell egy kis találékonyság is.

12.3.2. Példák a Help használatára

Pozitív példa: Szeretnénk sorozat vagy függvény határértéket számolni. Mivel a határérték angolul „limit”, beírjuk a Help Topic Search-be hogy limit. Két limit-et kapunk vissza, az egyik Limit, nem értékeli ki a kifejezést, a másik kiértékeli.

A Help-ben látjuk, mit kell beírni, hogy a MAPLE határértéket számoljon. Beírjuk:

```
> limit((n^2+n+1)^(1/2)-n, n=infinity);
```

Kiírja:

$$\frac{1}{2}$$

Beadjuk:

```
> limit((1-(1/n))^n, n=infinity);
```

Kiírja:

$$e^{(-1)}$$

Függvények határértékét ugyanúgy számoljuk, mint sorozatok határértékét. Beírjuk:

```
> limit(sin(x)/sin(3*x),x=0);
```

 Kiírja:

$$\frac{1}{3}$$

Negatív példa. Szeretnénk a Stirling formulát használni. Rákattintunk a Help-re, beírjuk, hogy Stirling, erre az elsőfajú és másodfajú Stirling számokhoz jutunk, amire legtöbb embernek nincs szüksége. (A Stirling formulát sem használja mindenki, de jóval többen, mint a Stirling számokat.)

1. feladat. Keressük meg a Help-ben, hogyan kell polinomot faktorizálni. Alkalmazzuk ezt $x^{128} - 1$ -re.

⁴Téma szerinti kereső: Topic = téma, search = keres.

12.3.3. Hogyan használjuk a MAPLE-t?

Rövid leírás

A MAPLE „munkalapokkal” dolgozik, ún. „Maple Worksheet”-ekkel, ezek save-elhetők, .mws kiterjesztéssel. Munkánkat később újra behívhatjuk, ha pl. határértéket számoltunk és limesz.mws címmel elmentettük, azt a szokásos módon újra behívhatjuk, pl. LINUXban **xmapple limesz.mws**-sel.

Az eredmény kimenthető (exportálható) LATEX-ben is. Az ilyenkor készített ábrákat POSTSCRIPT formátumban, .eps kiterjesztéssel is kimentí. ⁵

Amikor elindítjuk a MAPLE-t, kapunk egy ilyen munkalapot, arra írjuk be a parancssorokat. Ennek „untitled” a neve, azaz, címnélküli, amikor ki akarunk lépni, a program megkérdezi, el akarjuk-e menteni, és ha igen, milyen néven az eddigi munkánkat.

(a) Minden beírt sort `;`-vel vagy `:`-tal fejezünk be. A `;` esetében kiírja az output-ot: vagy szebb formában, amit beírtunk, vagy ha **plot**-ot írtunk be, kirajzolja, amit kérünk, és feldolgozza azt. A `:`-s sorzárás esetén nem ír ki semmit, de feldolgozza.

(b) A MAPLE csaknem minden olyan függvényt ismer, amellyel a könyvben találkozhatunk. Néha nem pont ugyanúgy jelöli, pl. $\sin x$ helyett $\sin(x)$ -et kell írunk, stb. Ezekből definiálhatunk további függvényeket.

(c) Definiáláshoz az `:=`-t használjuk, pl. beírjuk:

```
> g:=abs(x)^(3/2)/(1+x^2);
```

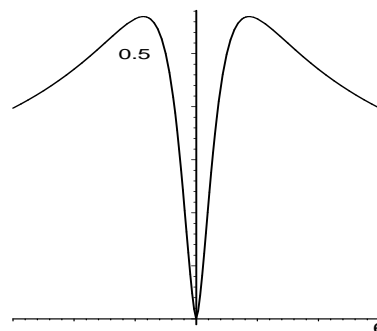
Az `ENTER` leütése után a program szépen is kiírja, amit beírtunk. Most pl.

$$g(x) := \frac{|x|^{3/2}}{1+x^2}$$

Ha ábrázolni szeretnénk, beírjuk:

```
> plot(g,x=-6..6);
```

A 6. ábrát kapjuk. (Korábbi MAPLE verziókban előzőleg a **with(plots)**; segítségével a rajzoló programkönyvtárat is be kell hívnunk, (l. 12. old.).



6. ábra: $g(x) = \frac{|x|^{3/2}}{1+x^2}$

Figyeljük meg, hogy itt a MAPLE választotta meg a skálázást.

2. feladat. Állítsuk elő az 1–6. függvényábrákat MAPLE-lel.

⁵A kozelit.mws munkalap LATEX kimentésekor keletkezik egy (kicsit használhatatlan) kozelit.tex file, és az ábrákat kozelit01.eps, kozelit02.eps, ... alakban kapjuk. A POSTSCRIPT ábrák készítése nagyon hasznos, ha valamit TEX/LATEX-ben akarunk írni!

3. feladat. Állítsuk elő az $y = x(\pi - x) \sin(8x)$ függvény ábráját MAPLE-lel, a $[-2\pi, 2\pi]$ intervallumon, majd a $[0, \pi]$ -n. (MAPLE-ben a π beírható Pi-ként!)

12.3.4. Szimbolikus számolások

A MAPLE (MATHEMATICA, DERIVE) két dolgot tud nagyon jól: (a) szimbolikusan számolni, azaz formulákat manipulálni, és (b) matematikai objektumokat ábrázolni.

Tanulmányaink során gyakran kell bonyolultabb formulákat egyszerűbb alakra hoznunk. Ezt bármely szimbolikus programcsomag könnyedén megteszi.

Szeretnénk pl. egyszerűsíteni az $(a^6 - b^6)/(a^4 - b^4)$ kifejezést. Beütjük:

```
> f := a^6-b^6;
```

Erre azt kapjuk: $f := a^6 - b^6$. Kíváncsiságból faktorizáljuk:

```
> factor(f);
```

Az eredmény:

$$(a - b)(a + b)(a^2 + ab + b^2)(a^2 - ab + b^2)$$

Beadjuk: $> g:=a^4-b^4;$ ⁶

$$g := a^4 - b^4$$

A hányadosra alkalmazzuk a **simplify (f/g)** ; parancsot, mire ezt kapjuk:

$$\frac{a^4 + a^2 b^2 + b^4}{a^2 + b^2}$$

4. feladat. Faktorizáljuk $x^{16} - y^8$ -t.

Kipróbáljuk a MAPLE „trigonometrikus” erejét is. Beadjuk: **expand(sin(3*x))**;
mire azonnal kifejti $\sin x$ és $\cos x$ hatványai szerint: $4 \sin(x) \cos(x)^2 - \sin(x)$
Ezt még mi is ki tudnánk számolni, de a gép az **expand(sin(13*x))**;
hatására a $\sin(13x)$ -et is egy pillanat alatt kifejti. Egyszerre csak kitégülnak
a lehetőségeink.⁷

⁶Menet közben kezdünk tömöríteni: nem írunk mindent új sorba, nem mindig írjuk ki a gép választát. Kiemelt képletnél a sorvégi pontot gyakran elhagyjuk.

⁷Ehhez kapcsolódik a 9.23. Tétel után található 5. és 6. feladat.

A szimbolikus számolás azt jelenti, hogy míg a gépek korábban csak számokat, vektorokat, stb. tudtak összeszorozni, az általunk leírt programok-csomagok algebrai kifejezésekkel, absztrakt szimbólumokkal is számolnak, legalább olyan jól, mint egy matematikus. Ha azt kérjük, hogy differenciálják az

$$f := b^{x^2} \cdot \sin(x)$$

függvényt, akkor kiadják az eredményt, (néha nem abban a formában, melyben szeretnénk):

> **diff(f,x);**

$$2b^{(x^2)}x\ln(b)\sin(x) + b^{(x^2)}\cos(x)$$

Ha pl. azt írjuk be, hogy

> **solve(x^2-x-1);** azaz a program oldja meg az $x^2 - x - 1 = 0$ egyenletet, akkor az

$$\frac{1}{2} + \frac{\sqrt{5}}{2}, \frac{1}{2} - \frac{\sqrt{5}}{2}$$

sort kapjuk vissza: nem tizedestört közelítést.

12.3.5. A kötelező óvatosság

Szeretnénk figyelmeztetni az olvasót, hogy a gép gyakran becsap. Máskor pedig olyan választ ad, amivel nem tudunk semmit kezdeni.

(a) Például, ha egy függvényt szeretnénk számítógéppel megsejteni, (és konkrét értékeit ki tudjuk számolni), ez nehéz olyankor, amikor az adott függvényhez nagyon közel van egy másik, sokkal egyszerűbb függvény. Ezért számítógéppel juthatunk helyes és helytelen eredményre egyaránt, és kell ahhoz némi gyakorlat, hogy tudjuk, mikor bízhatunk meg egy géppel kapott eredményben.

(b) Amikor használni kezdjük a gépet, tudnunk kell, milyen formában keressük a választ. Ha pl. az ikerprímek iránt érdeklődünk, nem kérdezhetjük meg, hogy van-e végtelen sok ikerprím, de könnyen írhatunk olyan programot, amelyik az 1000000 alattiakat kiírja.

12.3.6. Rajzolás, függvényábrázolás MAPLE-lel

Mondjuk, szeretnénk ábrázolni az $f := x \ln(x)$ -et. Többnyire azzal kezdjük, hogy beírjuk:

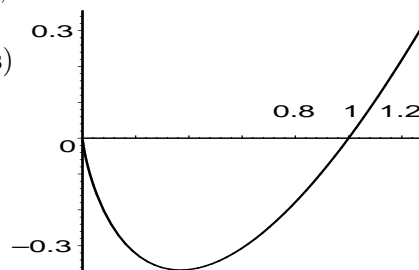
> **restart;** (memóriatörlés)

Beírjuk:

> **f := x*ln(x);**

A gép kiírja:

$$f := x \ln(x)$$



7. ábra: $x \log x$

Beírjuk:

> **plot(f,x=0..1.3);**

A függvényt az 7. ábrán látjuk.

... Innentől kezdve a „kiírja” részt többnyire elhagyjuk! Mindenesetre, ha a `;` helyébe `:`-ot írunk, akkor a MAPLE is elhagyja a „visszajelzést”.

Ha egyszerű rajzokat készítünk, a MAPLE 8-on nem kell, de a régebbi verziókban, ill. a bonyolultabb rajzoláshoz be kell hívunk a **plots** programkönyvtárat:

Beütjük:

> **with(plots);** Az alábbi üzenet arról értesít, hogy a korábbiakhoz képest valami változott. (Ez számunkra nem lényeges.)

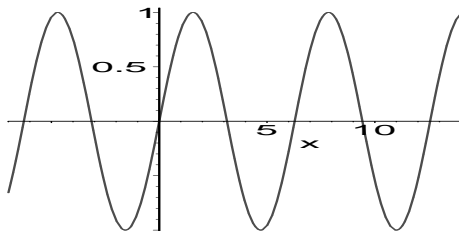
Warning, the name changecoords has been redefined

Ezután a MAPLE kiírja, milyen parancsok váltak elérhetővé a **with(plots)** hatására. Nagyon sok parancsot ír ki, mi itt három alkalmazásra szorítkozunk: **plot**, **plot3d**, **implicitplot**.

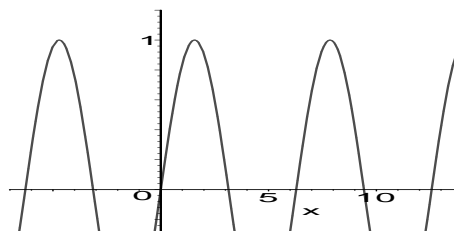
(A) Az értelmezési tartomány és értékészlet megszorítása. Függvényábrázoláskor be kell írunk, mettől meddig akarjuk a függvényt ábrázolni.

A

> **plot(sin(x),x=-7..14);** kirajzolja a \sin függvényt -7 -től 14 -ig. Az eredmény a 8/a ábrán látható:



8/a. ábra: $\sin x, x \in [-7, 14]$



8/b. ábra: $\sin x, x \in [-7, 14]$

Néha csak egy adott téglalapon belül szeretnénk a függvényt ábrázolni.

```
> plot(sin(x),x=-7..15,y=-0.3..1.2);
```

Az eredmény a 8/b ábrán látható. Erre lehet pl. szükségünk, ha meg szeretnénk pontosabban állapítani az 5 és 7 közötti gyök helyét. (Persze, itt, most tudjuk, hogy az 2π . Tegyük azonban úgy, mintha nem tudnánk.)
Beadjuk⁸

```
> plot(sin(x),x=5..7,y=-0.2..0.2);
```

Az eredmény a 9. ábrán látható.

(B) Több görbe egy koordináta-rendszerben

Ha két vagy három (vagy akárhány) görbét akarunk egyazon koordináta-rendszerben kirajzolni, akkor szögletes zárójelek közé tesszük őket:

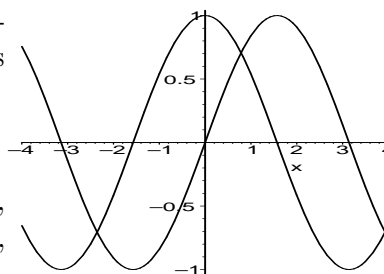
```
> plot([f,g],x=a..b);
```

 alakban. Pl. a

```
> plot([sin(x),cos(x)],x=-7..7);
```

a 9. ábrán látható. Ha azt szeretnénk, hogy mindkét görbét feketén rajzolja ki, használjuk a

```
> plot([f,g],x=a..b,color=black);
```

 alakot.

9. ábra: két fv. egyszerre

Ha azt szeretnénk, hogy vastagon rajzolon, használjuk a

```
> plot([f,g],x=a..b,thickness=3);
```

 alakot.

Mi történik a következő parancs beírására?

```
> plot([x^2*sin(x),x^2],x=-1..10,thickness=[3,1],color=[blue,black]);
```

Hogyan magyarázható, amit megfigyelünk?

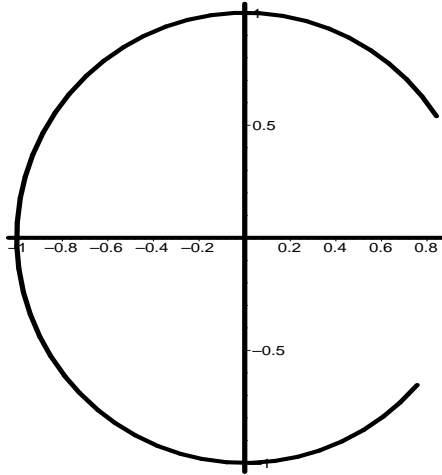
(C) paraméteres görbe kirajzolása.

Próbáljuk ki, mi történik, ha a] zárójelet „rossz” helyre tesszük:

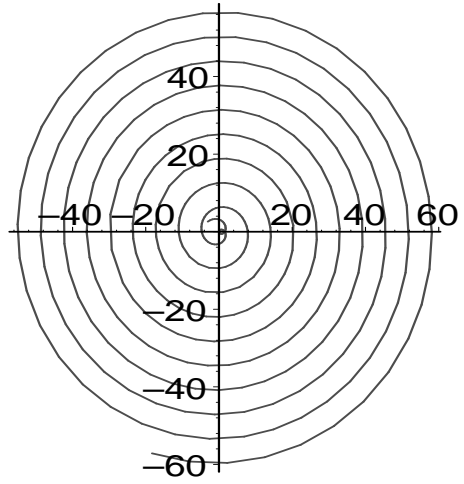
```
> plot([f,g,t=a..b]);
```

Ilyenkor az $x = f(t)$, $y = g(t)$ „pályát” rajzolja ki a MAPLE. Ezzel is találkozunk még: ez matematikában is és fizikában is nagyon fontos. Így kapjuk az alábbi ábrákat, a MAPLE által skálázva.

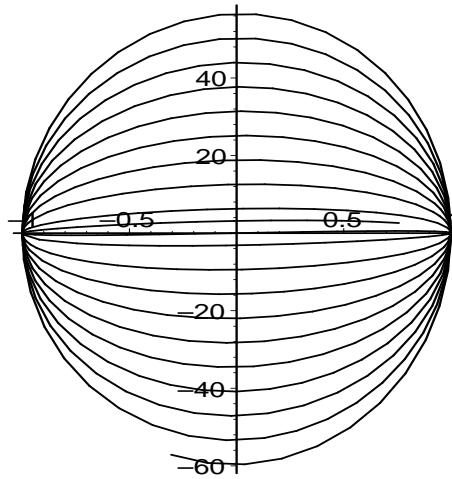
⁸Ebben a függelékben az ábrákat a kinyomtathatóság végett kicsit átalakítjuk: megvastagítjuk a vonalakat, felnöveljük a beírásokat és ahol zavaróvá válnak, ott elhagyjuk őket.



$(\sin(t), \cos(t)), t \in [-4, 1]$
10/a. ábra: Körív:



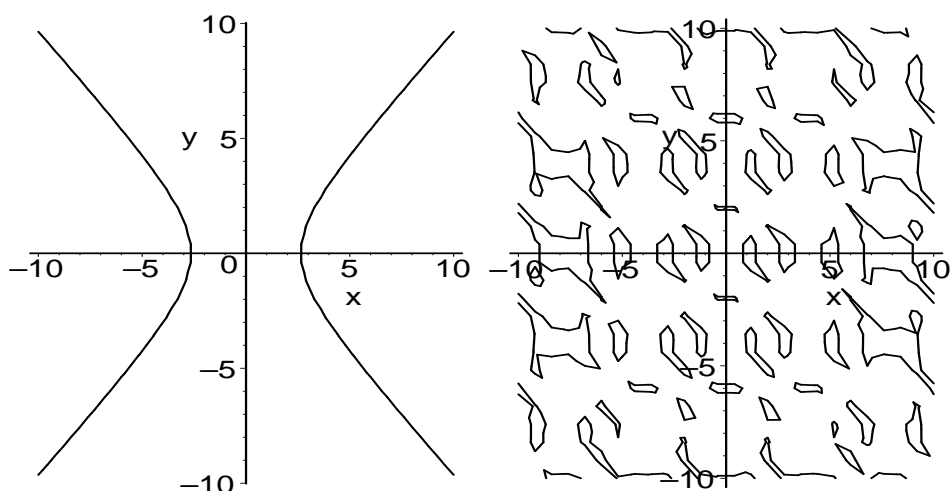
$(t \sin(t), t \cos(t)), t \in [-4, 60]$
10/b. ábra: Spirál



10/c. ábra: Lemaradt egy $t!$
 $(\sin(t), t \cos(t)), t \in [-4, 60]$ -t rajzoltuk ki

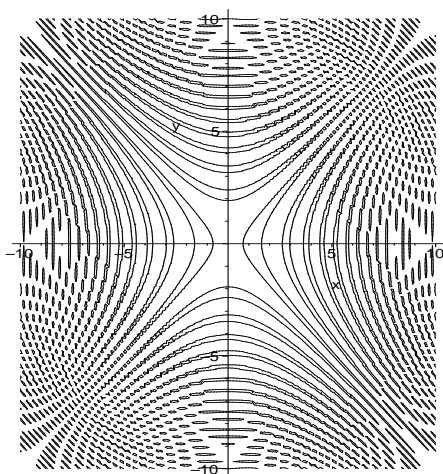
(D) **Implicitplot.** Az **implicitplot** azt jelenti, hogy beírunk egy $f(x, y) = c$ alakú egyenletet és a program kirajzolja a megfelelő pontok mértani helyét.

> **implicitplot(x²-y²=7,x=-10..10,y=-10..10,color=black);**



hiperbola
11/a. ábra:

hiperbolaszereg
11/b. ábra: durva rajz



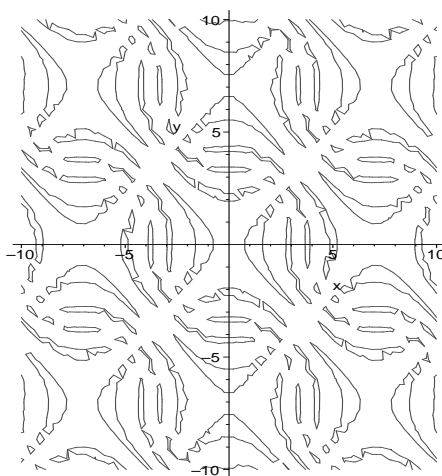
11/c. ábra: finom rajz
hiperbolaszereg

Itt tehát a 11/a ábra az $x^2 - y^2 = 7$ kirajzolása. A „color=black” a kirajzolás színét állította feketére. Szerettünk volna görbesereget generálni,

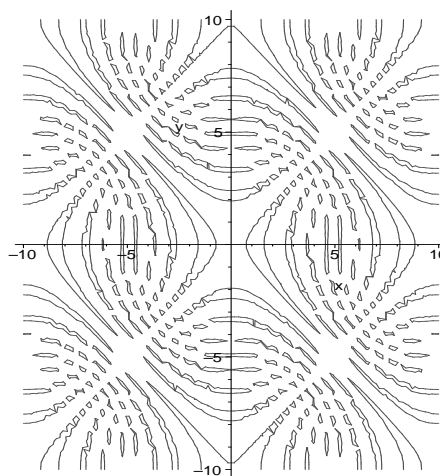
így beírtuk a programnak, hogy **implicitplot**-tal ábrázolja a $\sin(x^2 - y^2) = 0.5$ „mértani helyet”. Ez adta a 11/b ábrát. Nagyon érdekes, de világos, hogy valami nem jó rajta. Túl kevés pontot használt a rajzoláshoz. Beírtuk:

> **implicitplot(sin(x^2-y^2)=0.5,x=-10..10,y=-10..10,numpoints=2000);**
 azaz használjon sokkal több pontot. Észrevehetően lelassult a gép, de javult az ábra. Végül **numpoints=16000**-rel, rövid várakozás után a 11/c ábrát kaptuk: a keresett (két) hiperbolásereget.

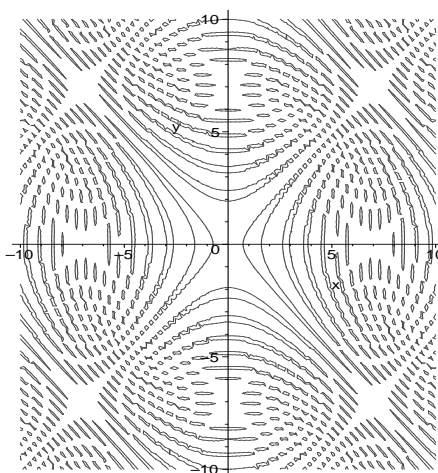
Érdekes megfigyelni a jó ábrához vezető utat:



12/a. ábra: 2000 pont



12/b. ábra: 4000 pont



12/c. ábra: 8000 pont

5. feladat. Próbáljuk ki ugyanezeket ellipszisszeregre, azaz,

> **implicitplot(sin(x²+3*y²)=0.5,x=-10..10,y=-10..10,numpoints=16000);**
-rel, illetve a hozzávezető durvább rajzokkal.

12.3.7. Racionális törtfüggvények



Megint a MAPLE-lel kezdjük. Megvizsgáljuk az

$$y = \frac{x^3 + x^2 - 7x}{x^2 - 3x + 1}$$

racionális törtfüggvényt. Érdemes a nevező gyökeinek megkeresésével kezdeni. Beütjük $f := x^3 + x^2 - 7x$ -et és $g := x^2 - 3x + 1$ -et, majd a **solve(g=0);**-val folytatjuk (solve=megoldani). Az eredmény:

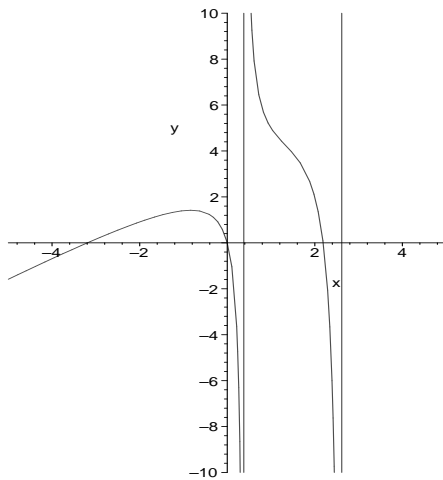
$$\frac{3}{2} + \frac{\sqrt{5}}{2}, \frac{3}{2} - \frac{\sqrt{5}}{2}$$

Minket a tizedestört kifejtés is érdekel. Beütjük: **evalf(solve(g=0));** ahol az **evalf** az evaluate-float-ot rövidíti, (evaluate=kiértékelni). Az eredmény:

$$2.618033988, 0.381966012$$

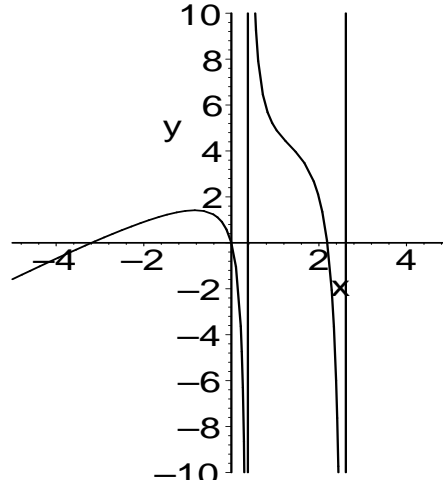
Most behívjuk a grafika csomagot, hogy ábrázoljuk f/g -t, bár már elmondtuk, hogy erre az egyszerűbb grafikus műveletekhez, a legújabb MAPLE verziókban nincs szükség.

> **with(plots);**



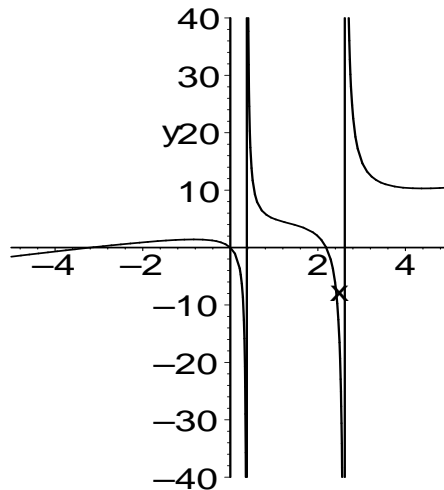
`plot([f,g],x=-6..6,y=-18..18);`

13/a. ábra:



`plot(f/g,x=-5..5,y=-10..10);`

13/b. ábra:



13/c. ábra:

`plot(f/g,x=-5..5,y=-40..40);`

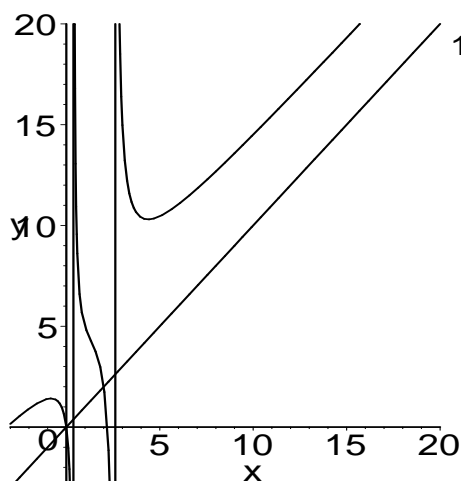
A 13/b ábrán azért kellett az $y \in [-10, 10]$ -zel levágnunk az ábrát, hogy bizonyos részleteket ne nyomjon agyon az y irányú skálázás. Így a nevező 2. gyöke fölötti részletek tűntek el. Ha $y \in [-40, 40]$ -nel vágunk, a 13/c ábrát kapjuk.

Az itt megadott három ábránk szép, de több hibája is van. Először is, a 13/a ábrán látható, hogy túl vékonyak a vonalak. Így, mint már említettük,

ezeket kiigazítjuk és a számokat is gyakran kinagyítjuk, mint a 13/b-c ábrán, ill. néhol elhagyjuk őket.

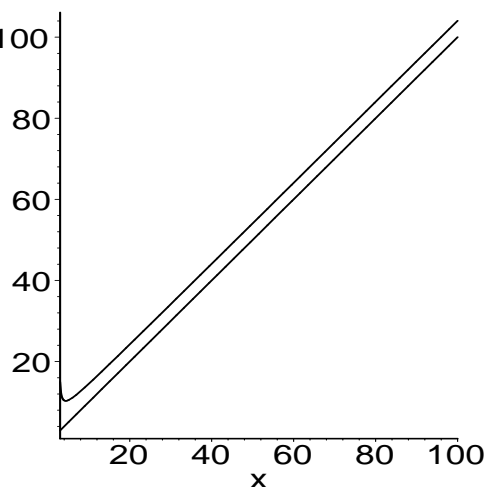
Kifogásolható az is, hogy nem azonos a skálázásunk az x és az y irányban. Ezen könnyen segíthetünk, ha mindkét irányban ugyanolyan hosszú intervallumot adunk meg. Ilyenkor azonban fontos részletek tűnhetnek el.

Szeretnénk megtudni, hogyan viselkedik függvényünk a második gyök ($x = 2.618$) körül. Ezért kirajzoljuk $[-3, 20]$ -ban is. Most y -t is ugyanígy vágjuk.



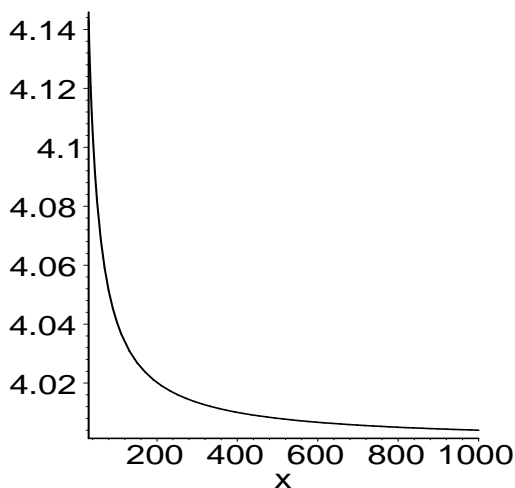
`plot(f/g,x=-3..20,y=-3..20);`

14/a. ábra:



`plot([f/g,x],x=3..100);`

14/b. ábra:



14/c. ábra:

plot((f/g)-x,x=30..1000);

Mi van nagy értékekre? Elkészítettük a 14/b ábrát, amiből arra következtethetünk, hogy $y = x$ aszimptotája a görbének. Hogy ezt jobban megvizsgáljuk, a 14/c ábrán a különbségüket vettük. Ebből arra következettünk, hogy a különbség 4-hez tart. Hogy ezt jobban lássuk, (utólag) megszorítottuk y -t [4, 4.2]-re.

Könnyű megmutatni, hogy ez a sejtés valóban helyes, azaz

$$\lim_{x \rightarrow \infty} \left(\frac{x^3 + x^2 - 7x}{x^2 - 3x + 1} - x \right) = 4.$$

(Lásd a 9.2. Tételt.)

6. feladat. Elemezzük az

$$\frac{x^2 + \sqrt{x}}{x - \sqrt{x}}$$

függvényt a fenti módszerekkel. Elemezzük a könnyű példáit ugyanígy (is).

12.4. MAPLE mint programozható kalkulátor

Ez a rész logikailag előbb jönne, de mivel – bár fontos, – kevésbé figyelemfelkeltő, így ide helyeztük el.

Sorozatok határértéke: A MAPLE egyebek között egy nagyerejű programozható kalkulátor. Így egy sorozat n -edik tagját beprogramozhatjuk, azt kiszámolhatjuk, és ezzel a határértékét megsejthetjük. Szeretnénk pl. az $(1 - \frac{1}{n})^n$ -ről megtudni, konvergens-e, és ha igen, mihez tart?

	Ezt írtuk be	Maple:	Jelentése
1a.	<code>> a := (1-(1/n))^n;</code>		A sorozat n -ik tagja
1b.		$(1 - \frac{1}{n})^n$	
2a.	<code>> evalf(subs(n=5,a));</code>		helyettesítsen
2a.			a -ba $n = 5$ -öt
2b.		0.32768	
3a.	<code>> evalf(subs(n=5,1/a));</code>		Találgatunk
3b.		2.718309	Talán $1/e$ -hez tart?
4a.	<code>> limit((1-(1/n))^n, n=infinity);</code>		
4b.		$e^{(-1)}$	Igen, $1/e$ -hez tart

A 2a. sorban két dolgot vontunk össze: behelyettesítettünk egy $a(n)$ „függvénybe” $n = 5$ -öt, majd (erre tört-alakot adott volna vissza) majd

evalf(...); paranccsal átszámítottuk az eredményt tizedestörtre. Ha visszamegyünk a 2a. sorra és átírjuk 5-öt 15-re, akkor 0.355264-et kapunk. Ha $n = 5000$ -et írunk be, akkor 0.3678 az eredmény, $n = 50000$ -re 0.36787576. Érezhetjük, hogy a sorozat konvergens, de kevés az esélyünk, hogy megsejtsük, mihez tart. Ha véletlenül kipróbáljuk a reciprokát, észrevehetjük, hogy az $1/e$ -hez tart.

A veszélyes MAPLE: „Sajnos” a MAPLE túl sokat tud: nem csak segíthet, de le is szoktathat a gondolkodásról. Fentebb, a 4a. sorban beírtuk, hogy számítsa ki a

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n$$

-et és lám, azonnal megmondta, hogy $1/e$ -hez tart.

7. feladat. Alkalmazzuk a fenti „módszert” más sorozatokra is: a 3. fejezet 3.1 példájában számos jelöltet találunk erre. Alkalmazzuk az ott található esetekre, pl. a könyvben szereplő

$$\lim_{n \rightarrow \infty} (\sqrt{n+1} - \sqrt{n}) = 0. \quad (214)$$

elemzésére, majd további könyvbeli nem-összeg feladatokra.

Ebben a függelékben megpróbálunk egy picit továbblépni a könyv anyagán, hiszen ha teljesen arra szorítkoznánk, amit a könyv matematikailag is kielemez, akkor nem érezhetnénk át, hogy a gép hatékony segítőeszköz. Ebben a szellemben elemezhetőek a 60. oldal (e), (f), (j) feladatai.⁹ A

> **sum(f,k=a..b)**

a matematikában használt $\sum_{k=a}^b f(k)$ MAPLE-beli alakja: összegezi az $f(k)$ függvényt $k = a$ -tól $k = b$ -ig. Ennek segítségével megsejthetjük a 60. oldal összeg alakú határértékeit. Az alábbiakban egy ilyen esetet bemutatunk. A 60. oldalon szerepel \sum jelölés nélkül a

$$\lim_{n \rightarrow \infty} \sum \frac{1}{k(k+1)}$$

meghatározása. A MAPLE ezt az összeget zárt alakban állítja elő (ahogyan mi is, ha meg akarjuk oldani az ezzel kapcsolatos feladatokat).

⁹Az itt található oldalszámhivatkozások technikai okokból elcsúszhatnak.

	Ezt írtuk be	Ezt kapjuk	Jelentése
4a.	<code>> b := 1/(k(k+1))</code>		Az összeg k -ik tagja
4b.		$b := \frac{1}{k(k+1)}$	
5a.	<code>>evalf(sum(b,k=1..10000));</code>		összegezzünk 10000-ig.
5b.		0.9999000100	sejthető: $\rightarrow 1$
6a.	<code>>f:=sum(b,k=1..n);</code>		Próbálkozunk
6b.		$f := -\frac{1}{n+1} + 1$	Meglepetés: kiszámolta!

Valóban, úgy látszik, a MAPLE sokkal okosabb, mint gondoltuk.

12.4.1. Még mit érdemes tudnunk a Maple-ról?

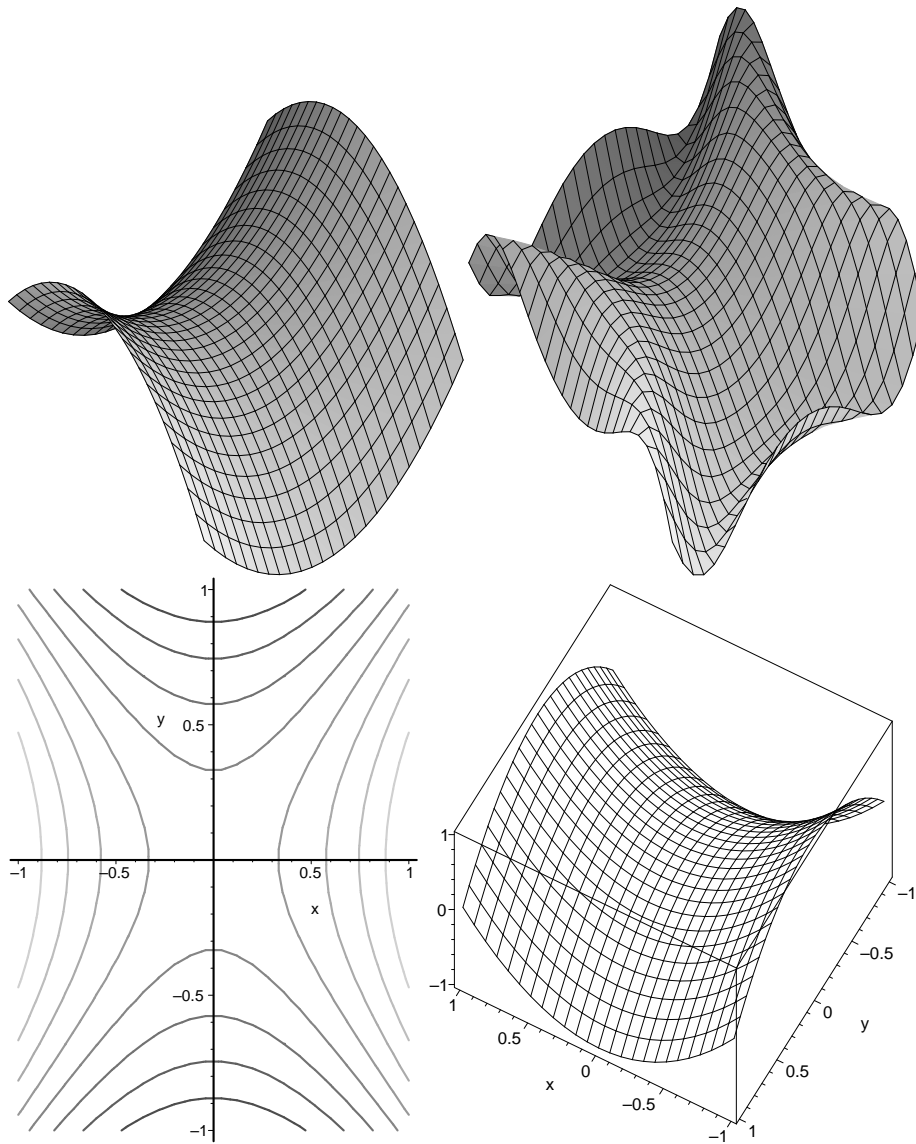
Sok mindent. Az alábbi táblázat részben ismétlés is, és néhány újabb parancsot is beírtunk, kipróbálásra.

Az Olvasó persze ne higgye, hogy itt a legfontosabbakat gyűjtöttük össze: a leghelyesebb próbálkozni, megpróbálni ráérezni ezen „játék” ízére, majd a Help, vagy egy megfelelő könyv segítségével továbblépni.

Csodákat azonban ne reméljünk: a MAPLE segíthet a gondolkodásban de nem helyettesíti azt!

Kulcsszó/Példa	Megjegyzés	értelmezés
<code>expand(...)</code>		kifejt
<code>plot(f,x=a..b)</code>	<code>with(plots)</code>	ábrázolja $f(x)$ -et $x \in [a, b]$ -re
<code>z=plot3d(f,x=a..b,y=c..d)</code>	<code>with(plots)</code>	ábrázolja $f(x, y)$ -t $x \in [a, b]$, $y \in [c, d]$ -re
<code>simplify(...)</code>		egyszerűsít
<code>solve(f=0,x)</code>		Megold egy $f = 0$ egyenletet x -ben
<code>subs(x=t^2,g)</code>	<code>substitute</code>	függvénybe behelyettesítünk $x = t^2$ -et
<code>evalf</code>	<code>evaluate</code> <code>float</code>	tizedestört alakba írunk... (kiértékelünk)

Az alábbiakkal kicsit előre ugrunk, a többváltozós függvényekre.



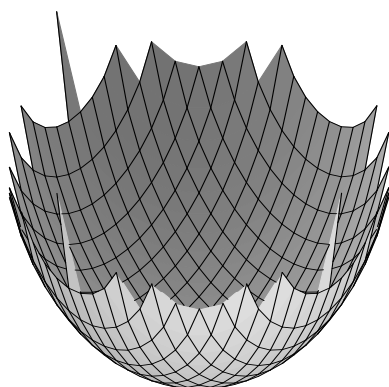
Itt a $z = (x^2 - y^2)$, illetve a $z = \sin(x^2 - y^2)$ 3-dimenziós felületet ábrázoltuk $x \in [-2,2]$, $y \in [-2,2]$ négyzet felett (majd az egerrel elforgattuk) (`plot3d(x^2-y^2,x=-1..1,y=-1..1);`)

Itt az $(x^2 - y^2)$ -et szintvonalasan, illetve kockába zárva rajzoltuk ki

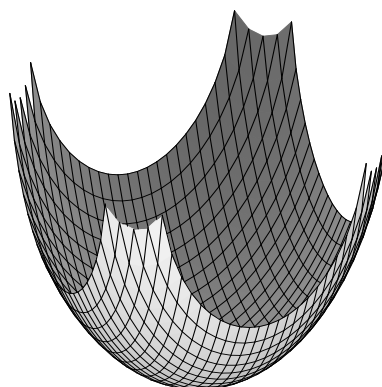
Az 15/a ábrán pl. egy félgömböt rajzoltunk volna ki, impozáns, de látjuk, a széleivel még nem minden tökéletes. Ezt használtuk:

`plot3d(-sqrt(1-x^2-y^2),x=-1..1,y=-1..1);`

A bajt az okozta, hogy az $x^2 + y^2 = 1$ körvonal felett a felület érintősíkja függőlegessé válik. Ez jól látható, ha csak $x \in [-0.8, 0.85], [-0.8, 0.85]$ felett ábrázoljuk a felületet, majd rákattintva az egérrel az ábrára, azt elforgatjuk. A finomabb (30×30) rács is segíthet. Az eredmény a 15/b ábrán látható.



Alapértelmezés
15/a. ábra: $-\sqrt{1-x^2-y^2}$



$x \in [-0.8, 0.8], y \in [-0.8, 0.8], \text{grid}=[30,30]$
15/b. ábra: $-\sqrt{1-x^2-y^2}$

12.4.2. MAPLE és az egyenlőtlenségek

Képesek vagyunk egy ábrára több függvényt is kirajzolni egyszerre. Így egyenlőtlenségeket is vizsgálhatunk. Láttuk, ha pl. az f , g és h függvényeket szeretnénk kirajzolni egyszerre, akkor a `plot([f,g,h],x=a..b);` alakot használjuk.

A matematikai bizonyításokban az egyenlőtlenségek fontos szerepet játszanak. Ezt láttuk már pl. a Bernoulli egyenlőtlenség esetében, a Jensen egyenlőtlenségnél, és még sok más esetben. Segíthet-e a MAPLE (vagy a MATHEMATICA, DERIVE) egyenlőtlenségek kezelésében? Igen, pl. kirajzolhatunk függvényeket egyazon koordináta-rendszerben, és rögtön látjuk, hogy egyik nagyobb-e mint a másik. Az alábbiakban ezt mutatjuk be, kicsit leegyszerűsítve.

A Bernoulli egyenlőtlenség. (1. az 1.5. Tétel.)

Beírjuk a MAPLE-be:

> `f:=(1+x)^n;`

> `g:=1+n*x;`

$(f = (1+x)^n)$

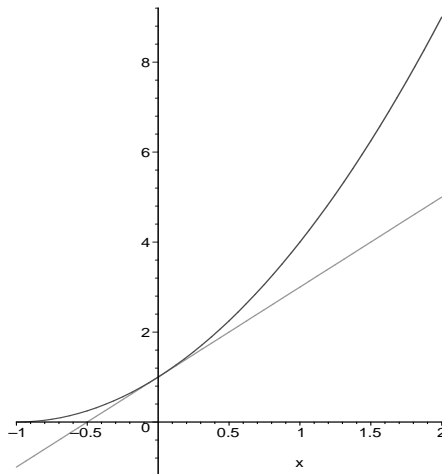
(a lineáris közelítés)

```

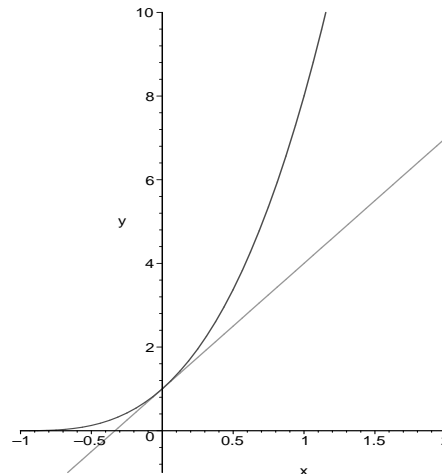
> fn:=subs(n=3,f);           (subs=substitute = behelyettesít)
> gn:=subs(n=3,g);
> with(plots);
> plot([fn,gn],x=-2..3,y=-1..10,color=black);           (kirajzol)

```

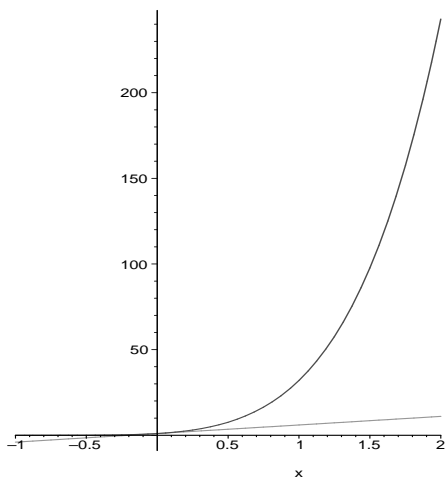
Fent tehát megadtuk a Bernoulli egyenlőtlenség mindkét oldalát, (f, g) -t, majd behelyettesítettünk $n = 3$ -at, kaptuk a 15/b ábrát. Ha $n = 2$ -et helyettesítünk, a 15/a ábrát kapjuk, $n = 5$ -re a 15/c ábrát. A fenti utolsó MAPLE-sorban az $y = -1..10$ elhagyható, de nélküle a MAPLE saját maga dönti el a függőleges nyújtást. Így mi adtuk meg: csak a -1 és 10 közötti értékeket kértük. Az utolsó sorban a **color=black** megparancsolja, hogy a gép a görbékét feketében rajzolja ki. Enélkül az elsőt pirosan, és a továbbiakat újabb és újabb színnel kapnánk meg.



$1 + 2x$
16/a. ábra: $(1 + x)^2$



$1 + 3x$
16/b. ábra: $(1 + x)^3$



16/c. ábra: $(1+x)^5$
 $1+5x$

12.4.3. Polinom közelítés (lokális)

Az itt következő téma, ahogyan ezt a ?? fejezetben is láttuk, még folytatása az egyenlőtlenségek tárgyalásának, ugyanakkor átvisz a polinomközelítésre: Függvényeket fogunk közre hozzájuk közeli polinomokkal.

Fontos számunkra, hogy

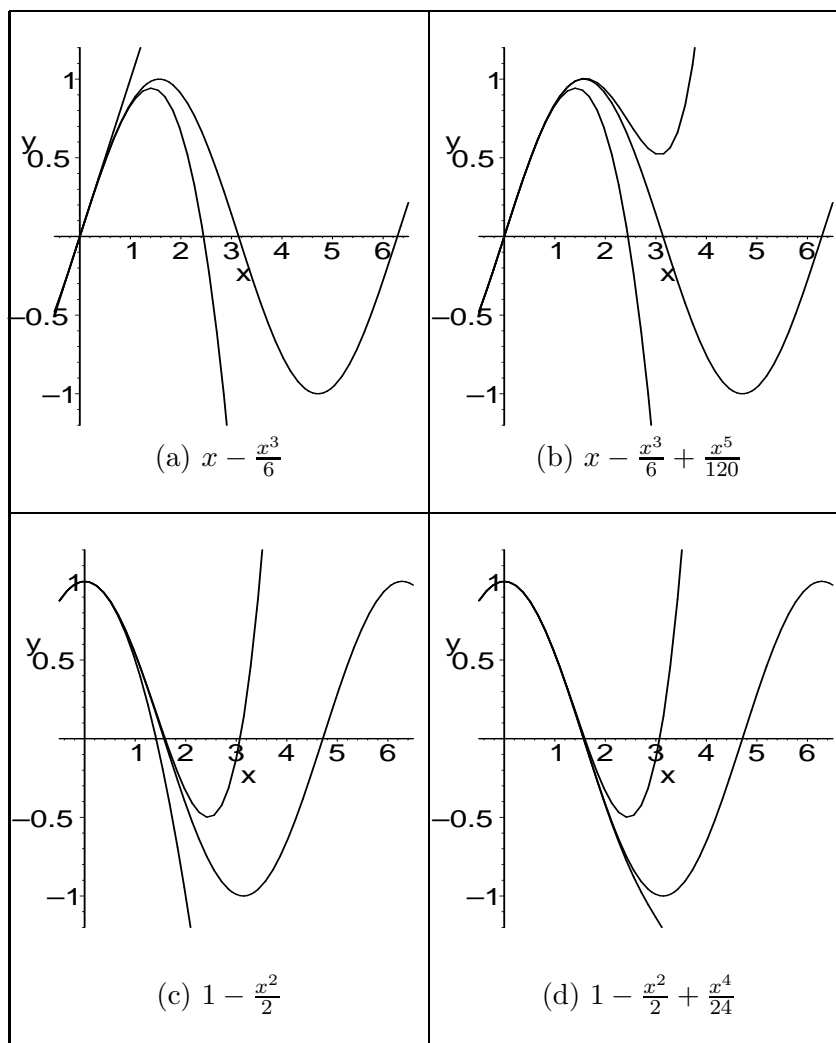
$$x - \frac{1}{6}x^3 < \sin x < x \quad \text{ha } x > 0.$$

8. feladat. Mutassuk be ezt a MAPLE segítségével.

A fenti feladat továbbvitele:

9. feladat. Mutassuk be a MAPLE segítségével a következőt:

$$x - \frac{1}{6}x^3 < \sin x < x - \frac{x^3}{6} + \frac{x^5}{120} \quad \text{ha } x > 0.$$



17. ábra: $\sin x$, ill. $\cos x$ közrefogása Taylor polinomjaikkal

A 17/a-b ábrákon azt látjuk, ahogyan a $\sin x$ -et a Taylor polinomjai közrefogják és egyre jobban megközelítik, a 17/c-d ábrákon pedig ugyanezt a $\cos x$ -re. A 18. ábrán a $\sin x$ közelítése látható a 11-edfokú Taylor polinomjával.

10. feladat. Mutassuk be MAPLE segítségével, ¹⁰ hogy

$$1 - \frac{1}{2}x^2 < \cos x < 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 \quad \text{ha } x > 0.$$

¹⁰Ez BASIC programmal még egyszerűbben megtehető.

Az ábrázolás néha segíthet bizonyításainkban, de azokat nem pótolja. Az idevágó bizonyítások megtalálhatók a DIFFERENCIÁLHATÓ FV.-ek VIZSGÁLATA c. fejezetben, a feladat-sorozatban.

A jelen könyv folytatásában szerepelni fog a Taylor sorfejtés, eddig még csak a végesített formájával találkoztunk. (L. 11.6. Definíció, 11.7. Tétel.) Hogy kiismerjük magunkat a MAPLE-ben, helyesebb mégis sorfejtésről beszél-nünk, ami ($a = 0$ körüli sorfejtésnél) azt jelenti, hogy

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n f^{(k)}(0) \frac{x^k}{k!} = f(x) \quad \text{ha } n \rightarrow \infty.$$

Az általunk tekintett esetekben persze Taylor polinommal való közelítésről van szó és a megfelelő fejezetben megfogalmazott egyenlőtlenségekről.¹¹

Ha pl. a $\log(1+x)$ -et akarjuk 4 tagig sorbafejteni, 0 körül, beírjuk:

> **f := ln(1+x);**

$$f := \ln(1+x)$$

> **g:=taylor(f,x=0,4);** hatására a MAPLE kiadja:

$$g := x - \frac{1}{2}x^2 + \frac{1}{3}x^3 + O(x^4)$$

Az így kapott formulával kicsit óvatosabban kell bánnunk, mintha az $O(\cdot)$ nem volna ott. Ezért az ilyen „alakzatokkal” a MAPLE sem bánik úgy, mint közönséges polinomokkal.

11. feladat. Próbáljuk átvinni az e^{-x} -re $x > 0$ azt, amit a $\sin x$ -szel csináltunk ebben a részben. (Számunkra az a különbség az e^x és az e^{-x} között, hogy az utóbbi Taylor polinomjainak együtthatói alternáló előjelűek.)

12.4.4. Határozatlan integrál, primitív függvény

A primitív függvényt is kiszámolja a Maple. Itt a $\log x$ és az $x \sin(x)$ primitív függvényének példáján mutatjuk be ezt. A primitív függvény kereséséről a 11. fejezetben volt szó. Az f x -től függő függvény határozatlan integrálját **int(f,x)** adja meg:

¹¹Ha a MAPLE Help-jébe beírjuk a Taylor szót, a "Taylor Expansion" = „Taylor sorfejtés”-re ugrik.

	Ezt írtuk be	Ezt adta vissza	Jelentése
1a.	<code>> g := ln(x);</code>		Ezt írtuk be.
1b.		$\ln(x)$	Ezt írta ki
2a.	<code>> int(g,x);</code>		keressen primitív függvényt
2b.		$x \ln(x) - x$	$\ln x$ primitív fv.-e
3a.	<code>> int(x*sin(x),x);</code>		keressen primitív fv-t.
3b.		$\sin(x) - x \cos(x)$	$x \sin x$ primitív fv.-e

12. feladat. Keressük meg az alábbi függvények primitív függvényét a MAPLE segítségével, majd differenciálással ellenőrizzük az eredményt.

$x \log x$	$\tan(x)$	$\frac{1}{x^2-x-1}$	$x^2 e^x \cos^3(x)$	$\tan(x)^3$	$e^x \sin(3x) x^2 e^x$	e^{x^2}
------------	-----------	---------------------	---------------------	-------------	------------------------	-----------

A legutolsó integrált nem tudjuk „zárt alakban” integrálni. Az eredmény fontos a valószínűségi számításban, így bevezetünk erre egy új függvényt: az **erf**-et (Neve az error-fv-re utal.) Próbáld ki integrálni ezt a MAPLE-lel.

12.4.5. Mit tud még a Maple?

Mindent tud, amire egy átlagos matematikusnak szüksége lehet.

Tud integrálni, mátrixokat szorozni, azok inverzét megkeresi, sajátértékeit kiszámolja, karakterisztikus polinomjait megadja, . . . egyenleteket, egyenlet-rendszereket old meg, és sok olyasmit tud, ami a magasabb matematikához tartozik.¹²

12.5. Mikor használjunk saját programokat?

A saját programon olyan programot értünk, amelyiket mi írtunk.

Az első kérdés:

Lehet-e Maple-ben programokat írni?

Igen, lehet. Minden, ami BASIC-ben megcsinálható, Maple-ben is megcsinálható. Amit mi *itt* BASIC-ben megcsinálunk, azt Maple-ben is könnyű megcsinálni.¹³

¹²Ha valamire szükségünk van, ami itt nincs felsorolva, érdemes a HELP-ben „rákeresni”.

¹³Ezzel szemben a BASIC nem ismeri a szimbolikus műveleteket: nem tud differenciálni, integrálni, gyököt keresni, . . . Ezen felül, sok mindent, ami a MAPLE-ben adott, a BASIC-ben magunknak kell kitalálnunk, megcsinálnunk.

Persze, első kérdés, hogy **mi program és mi nem az**. Itt a programot azzal definiálom, hogy van benne ciklus és/vagy elágazás.

Ha a MAPLE nyelv helyett mégis a BASIC nyelvet választjuk, annak lehet az oka, hogy a BASIC-ben is nagyon egyszerű a programok megírása, a BASIC jobban elérhető (ami változhat) és a BASIC programokban jobban beláthatunk a dolgok algoritmikus lelkébe.

Egy adott feladat megoldását segítő saját magunk által írt program egyaránt lehet hasznos és káros. Felhívhatja a figyelmünket arra, amit egyébként nem, vagy csak nehezebben vennénk észre. Másrészt elterelheti a figyelmünket valami fontosabbról, leblokkolhatja a gondolatainkat, nagyon sok időnkot elviheti. Próbáljunk megmaradni az arany középben.

12.5.1. Melyik nyelvet használjuk?

Majdnem mindegy: használjuk azt, amelyik a legszimpatikusabb. Ilyen egyszerű feladatra a BASIC is teljesen megfelel. Mi ennek egy „nyelvjárását”, QBASIC-et fogunk használni.

Az alábbiakban (tömören) a következő feladatokat vizsgáljuk:

1. Megsejtjük sorozatok határértékét BASIC programokkal.
2. Ezen belül rekurziók határértéket is vizsgáljuk.
3. Az „archimedeszi” rekurzió egy változatával kiszámoljuk π -t.
4. Összegek határértékét is megvizsgáljuk. (Ezen belül a görbe alatti terület közelítését.)
5. Diszkutálunk függvényeket egyszerű BASIC programokkal.

Feltesszük, hogy az olvasó gépén van QBASIC.

Először indítsuk el a programot.

Erre megjelenik a QBASIC integrált környezet, ami azt jelenti, hogy egy olyan editor jelenik meg, amelyikben programokat írhatunk és azokat kiírhatjuk lemezre, továbbfejleszthetjük, futtathatjuk. ¹⁴

¹⁴Ezzel szemben van olyan programnyelv-használat is, ahol egy editorban írjuk a programot és egy másik helyen ”kompiláljuk”, majd a ”kompilált programot” futtathatjuk.

12.5.2. Rövid összefoglaló a BASIC-ről

QB Ez nem egy számítástechnika tankönyv. Csak nagyon kevés BASIC kulcsszót használunk, és a számítástechnikai csiszolásokat majdnem teljesen elhagyjuk. Itt mégis összefoglaljuk a BASIC kulcsszavak jelentését, csak annyira, amennyire szükségünk lesz rájuk.

CLS	Clear Screen: letörli a képernyőt.
SCREEN 12	Grafikus képernyő, rajzoláshoz, finom felbontás
INPUT x	Bekéri az x változó értékét.
PRINT x	Kinyomtatja az x változó értékét.
FOR i=1 TO n NEXT i	A hozzátartozó NEXT-ig végrehajtja a közöttük levő parancsokat
IF <felt> THEN <para>	Az IF utáni feltételt ellenőrzi és teljesülése esetén a THEN utáni parancsokat végrehajtja
REM	Feljegyzésekre szolgál, ami utána van, a program futását nem befolyásolja
END/STOP	leállítja a programot
a ^b	Hatványozás: a^b
SQRT(x)	\sqrt{x}

12.5.3. Sorozatok szemléltetése, határértéke

A legrégebbi BASIC verziókban minden sornak volt egy sorszáma, ez a QBASIC-ben is működik, de felesleges. Egy régi program mondjuk így nézett ki:

Tegyük fel, hogy $\lim_{n \rightarrow \infty} \sqrt[n]{n}$ -re vagyunk kíváncsiak: létezik-e, és ha igen, mennyi? Írjuk meg az alábbi programot:

1. Program (Sorozat határértéke).

```
10 PRINT "n="
15 INPUT n
20 a=n^(1/n)
30 PRINT a
```

Magyarázat: Ez a program a 2. sorában bekéri n értékét, az 1. sor emlékeztet, hogy n értékét kéri. a 3-ikban n -edik gyököt von belőle, a 4-ikben kiírja az eredményt. Ha beadjuk $n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ -et, rendre a következőket kapjuk

$\sqrt[n]{n}$:	1	2	3	4	5	6	100	1000	10000
	1	1.412	1.422	1.412	1.379	1.348	1.047	1.00693	1.00092

Ebből két dolgot is észrevehetünk:

- (a) A sorozat 1-hez tart.
- (b) Megsejthetjük, hogy a hiba: $\Delta_n := a_n - \lim a_n \sim \frac{c \log n}{n}$. (Hogy hogyan, arra a 12.6.1. alfejezetben térünk vissza.)

12.5.4. $a_n = (1 + \frac{1}{n})^n$ határértéke.

Nem várhatjuk el programjainktól, hogy egy mélyen elméleti tétel bizonyításában segítsenek, de pl. a 87. oldalon található 5.4 tétel és környéke kiválóan illusztrálható ilyen programokkal. Most az a feladatunk, hogy az ott megadott sorozatokat vizsgáljuk meg programjainkkal. Előbb azonban kicsit csiszoljuk az előző programot.

Kérdés: Hogyan tehetjük a fenti programot kényelmesebbé?

Például úgy, hogy ciklusba szervezzük:

2. Program (Sorozat határértéke II.).

```
10 REM ciklus
20 CLS
30 FOR n=1 TO 100
40 a=(1+(1/n))^ n
50 PRINT a,
60 NEXT n
```

Magyarázat: Programjaink **keret-programok**: van egy soruk, ahova tetszőleges képletet írhatunk, és akkor a program az ennek megfelelő sorozatot számolja. A változatosság kedvéért itt az $(1 + \frac{1}{n})^n$ -et írtuk be. Ennek hatására a gép kiírja az első 100 tagot. Ez itt megengedi, hogy megsejtsük a határértéket, de lassan konvergáló sorozatoknál esetleg teljesen félrevezethet. Az eredmény (kis változtatások és válogatás után!)

$(1 + \frac{1}{n})^n$:	1	2	3	4	5	7	100	1000	10000
	2	2.25	2.37	2.44	2.49	2.57	2.705	2.7169	2.71815

Itt egy újabb számítástechnikai fogalmat, a **for-next** ciklust vezettük be. A 30-50 sor-pár azt intézi el, hogy a közöttük levő sorokat a program végrehajtja $n = 1, \dots, 100$ -ra. A 2. sor egy Clear-Screen, azaz képernyőtörlés, kellemesebbé teszi a program futtatását, de nem funkcionális. Az 1. sor egy "remark" = „megjegyzés”, a program futását nem befolyásolja, de segít a program áttekintésében.

A hosszabb, vagy sok ember által használt programoknál fontos az áttekinthetőség: a program, az input és az output áttekinthetősége.

Az OUTPUT esetében érdemes a PRINT-ről azt tudni, hogy `PRINT x;` kinyomtatja x -et, majd a „helyzetjelző”, hogy hova írunk, legfeljebb 1 karaktert megy tovább, `PRINT x,` után egy tabulátornyit ugrik, és ha a `PRINT x` után semmit nem teszünk, akkor a következő sorra ugrik. A `PRINT x;` típusú nyomtatás arra jó, hogy sok adatot írassunk ki egy sorba, a `PRINT x,` ugyanezt rendezettebben teszi meg. Ha szöveget akarunk kiírni, azt idézőjel közé tesszük. Az `INPUT "sebesség"; a` a `PRINT "szoveg"; : INPUT a` rövidítése. Mindkettő **feliratozott INPUT**: bekér egy adatot, de előbb megmondja, mit.

13. feladat. (a) A 2. programba írjuk be a 55. oldalon található alábbi (10)-(12) sorozatot:

$$\left(1 + \frac{1}{n}\right)^n, \quad \left(1 + \frac{1}{n}\right)^{n^2}, \quad \left(1 + \frac{1}{n^2}\right)^n,$$

és sejtjük meg határértékeiket a géppel.

(b) Tegyük ugyanezt a többi ottani sorozattal is.

Egy másik variáns, a $h_n = n(\sqrt[n]{n} - 1)$ sorozatra:

3. Program (Sorozat határértéke III.).

```
REM n-edik gyök n, ciklus
CLS
FOR i=1 TO 1000
INPUT "n="; n
IF n < .0001 THEN END
a=n*(n^(1/n)-1)
PRINT a
NEXT i
```

Magyarázat: Ez a program tipikusan olyan, amilyent az ember csak sajátmágának készít, hogy valamit gyorsan kiszámítson, nem érdemes túlcicomázni.

Arra való, hogy egymás után az adott sorozat akárhány tagját kiszámítsuk. Az ilyesmit valójában nem „for-next” ciklussal, hanem vagy „repeat-until” vagy „while” ciklussal szoktuk megoldani, amit itt nem akartunk elmagyarázni. Az egyszerűség kedvéért megint az $\sqrt[n]{n}$ sorozatot vizsgáltuk.

Mit kapunk ezzel a programmal? (Helyhiány miatt csak ritkítva és ke-rekítve írjuk ki az eredményeket. Ezt az előző program is megadta volna.)

$n(\sqrt[n]{n}-1)$:	$n =$	2	3	4	5	6	100	1000	10^4	10^6
	$h_n =$	0.83	1.33	1.66	1.9	2.09	4.71	6.93	9.21	13.8

Ebből megsejthetjük, hogy az $\sqrt[n]{n} \rightarrow 1$ -ben a hiba nagyobb nagyságrendű, mint $1/n$. Kipróbálhatjuk, hogy kisebb, mint $1/\sqrt{n}$. Ha viszont $\frac{\log n}{n}$ hibával, azaz $h_n = \frac{n}{\log n}(\sqrt[n]{n} - 1)$ -nel próbálkozunk, a következőt kapjuk:

$\frac{n}{\log n}(\sqrt[n]{n}-1)$:	3	4	5	6	100	1000	10^5	10^6
	1.21	1.195	1.18	1.17	1.023	1.0035	1.00046	1.000035

Ebből azt sejtjük, hogy $\sqrt[n]{n} - 1 \sim \frac{\log n}{n}$. (Ez a példa annyiból csak „iskolapélda”, hogy itt nem túl nehezen be is bizonyíthatjuk sejtésünket.)

14. feladat. Szeretnénk tudni az $a_n = (1 + \frac{1}{n})^n$ sorozat konvergenciasebességét. Módosítsd az előző programok valamelyikét úgy, hogy segítsen ezt megsejteni.

Mi az, ami ebben a programban „tákolmány”?

1. A program feltételezi, hogy 1000-nél kevesebb értéket kérdezőnk meg.
2. A leállítása egy tetszőleges $x \leq 0$ -val történik, de a program pozitív valós számokra is működik. Viszont, ha leállító feltételnek $x \leq 0$ -t írtunk volna, akkor nem lehetnénk biztosak, hogy a gép hogyan viselkedik $x = 10^{-100}$ -nál.

Ha egy BASIC programot hamarabb akarunk leállítani, mint ahogyan magától leállna, használjuk a CTRLBREAK-et.

15. feladat. Nézzük meg programmal, mi a határéréke és a konvergencia-sebessége $a_n = \sqrt[n]{2}$ -nek.

16. feladat. Nézzük meg programmal, mi a határéréke és a konvergencia-sebessége $a_n = \sqrt{n+1} - \sqrt{n-1}$ -nek.

12.5.5. Sorösszegzéssel kapcsolatos programok

Reciproknégyszetösszeg. Nagyon fontos és nem triviális matematikai tétel, hogy

$$\sum_{i=1}^n \frac{1}{i^2} \rightarrow \frac{\pi^2}{6} \quad \text{ha } n \rightarrow \infty.$$

Most ezt a tényt „vizsgáljuk meg” számítógéppel.

4. Program.

```
PRINT " Reciproknegyzetosszeg"  
10 INPUT "n="; n  
FOR i=1 TO n  
20 s=s+(1/i)^2  
NEXT i  
30 PRINT s  
40 PRINT SQR(6*s);
```

Magyarázat: Mivel sem az első sor, sem az utolsó nem szükséges igazán a programhoz, a limesz (sorösszeg) keresésére tulajdonképpen egy 5-soros programot használunk. 10 bekéri, hogy hány tagot adunk össze, 20 a ciklusban az összeget képi, az s változóba rakva be a mindenkori összeg-értéket. A 30-as sor kiírja az összeget, amiről láthatjuk, hogy konvergál, az utolsó sorban kiírjuk $\sqrt{6s}$ -t, arról észrevehető, hogy π -hez tart. Ebből láthatjuk, hogy $s \rightarrow \frac{\pi^2}{6}$.

17. feladat. A $\sum_{k=1}^n \frac{1}{k} - \log n$ sorozat konvergens: egy ún. Euler konstanshoz tart. Írjunk programot ennek illusztrálására: írjuk kicsit át a fenti programot.

18. feladat. A $\sum_{k=1}^n \frac{-1^k}{k} - \log n$ sorozat konvergens: $\frac{\pi}{4}$ -hez tart. Írjunk programot ennek illusztrálására: módosítsuk egy kicsit a fenti programot.

A görbe alatti terület. A ???. és a ????. oldalon található (egyik) feladat egy parabola alatti területrész kiszámításáról szól. Ez tulajdonképpen az integrálszámítás alapfeladata, amellyel az integrálszámítás keretében alaposabban is megismerkedünk. A következő program ezt illusztrálja.

5. Program.

```
REM terület-szamitas  
10 INPUT " a="; a: INPUT " b="; b: INPUT " n="; n  
20 PRINT "a vizsgalt intervallum: ["; a; ","; b; "]"  
30 lep=(b-a)/n: s=0  
PRINT "felosztas-szam="; n, "lepeshossz=", lep  
40 FOR i=1 TO n  
50 x=a+i * lep  
60 y=x * x  
s=s+y  
70 NEXT i  
PRINT "A terület kozelito erteke: ", s*(b-a)/n
```

Magyarázat: 10-ben bekérjük az intervallum végpontjait, továbbá, hogy hány felosztó pontot használunk. 20 pusztán ellenőrzésképen kiírja ezeket. 30 kiszámítja a felosztással kapott kis intervallumok hosszát. A 40–70 ciklus végzi az igazi munkát, az s -ben összeadogatja a kis téglalapok területeit: pontosabban a magasságukat, amelyet az utolsó sorban megszoroz a téglalapok közös alapjával.

Ez a program is **keret-program**: akármilyen függvényre alkalmazható. A programban csak egyetlen sor tükrözi, hogy az $y = x^2$ függvénnyel van dolgunk, a 60-as sor. Ha ezt a sort más függvényre cseréljük, annak a grafikonja alatti (előjeles) területet kapjuk meg.

Érdeemes megjegyezni, hogy az ún. trapézösszegek lényegében ugyanennyi munkával sokkal jobban közelítenek.

12.5.6. Rekurziók

QB A rekurzió fogalmával legelőször az 55. oldalon találkoztunk. A rekurziók a kezdők számára kissé nehezen tekinthetők át. Ilyenkor egy egyszerű programocska valóban sokat segíthet. Nézzük pl. az 55. oldalon található (15) sorozatot:

$$a_1 = 0, \quad a_{n+1} = \sqrt{2 + a_n}.$$

Konvergens-e? Mihez tart?

6. Program (Rekurziós határérték).

```

REM gyokos rekurzio
INPUT "Meddig irjam ki? n="; n
(i) a=0
FOR i=1 TO n (ciklus eleje)
(r) a=SQR(2+a) (a rekurzió)
PRINT i, a
NEXT i (ciklus vége)

```

Ha a programot $n = 14$ -vel futtatjuk, az alábbiit kapjuk:

1	2	3	4	5	6	7	11	12	13	14
1.414	1.848	1.96	1.990	1.997	1.9994	1.99985	1.999999	2	2	2

Ebből azt látjuk, hogy a sorozat valószínűleg 2-höz tart és hogy $n = 12$ körül a gép által használt kerekítés már pontosan 2-t ad. Azt ne higyjük, hogy onnantól kezdve $a_n = 2$. Az sem igaz, hogy mindegyik gép vagy program ugyanúgy kerekít.

Ha ezt a programot akarjuk más rekurzióra alkalmazni, akkor két helyen kell változtatnunk: (r)-ben a rekurzió és (i)-ben az induló értékeken.

A következő program az ún. Newton féle gyökkereső algoritmust használja annak talán legegyszerűbb esetében, a gyökvonáshoz. (Lásd a ???. oldal.) Legegyszerűbb szemléletes magyarázata az, hogy ha adott a -hoz b jól közelíti \sqrt{a} -t, de alulról, akkor a/b is jól fogja közelíteni, de felülről, a számtani közepük pedig még sokkal közelebb lesz \sqrt{a} -hoz.

7. Program (Newton gyökvonás).

```

      REM newton
(d)  DEFDBL A-B
      INPUT "a="; a
      INPUT "Hany iteraciot:"; n
      b=1
      FOR i=1 TO n
(r)  b=(b+(a/b))/2
      PRINT i, b
      NEXT i

```

Magyarázat: A rekurzió az (r) sorban van elrejtve. A (d) sor dupla pontosságúvá teszi a -t és b -t. Enélkül nem igazán figyelhetnénk meg a konvergencia hibáját, mert néhány lépés után eltűnne a kerekítésben. Ez kiderül, ha a programot alkalmazzuk valamilyen négyzetszámra, mondjuk, 4-re, (d) sorbeli **DEFDBL** kihagyásával.

Mit figyelhetünk meg? Ha pl. a programot 10000-re alkalmazzuk, akkor eleinte a hiba feleződik, majd amikor a hiba már, pl. $1/2$ alá ment, akkor lényegében minden hiba az előző hiba négyzetével becsülhető:

Ha $a \geq 1$ és $a_n = \sqrt{a} + h_n$, akkor

$$\begin{aligned}
 a_{n+1} - \sqrt{a} &= \frac{1}{2} \left(a_n + \frac{a}{a_n} \right) - \sqrt{a} = \frac{1}{2} \left(\sqrt{a} + h_n + \frac{a}{\sqrt{a} + h_n} \right) - \sqrt{a} \\
 &= \frac{1}{2} \left(\sqrt{a} + h_n + \frac{a - h_n^2}{\sqrt{a} + h_n} + \frac{h_n^2}{\sqrt{a} + h_n} \right) - \sqrt{a} = \frac{1}{2} \frac{h_n^2}{\sqrt{a} + h_n} < \frac{h_n^2}{2}.
 \end{aligned}$$

Természetesen ez pl. $h_n = 2$ esetében nem sokat ér, de ha egyszer $h_n < 1$, onnan az a_n sorozat gyorsan fog konvergálni \sqrt{a} -hoz.

12.5.7. Rekurzió π közelítésére

A π „kiszámolása” izgalmas kérdés. Nem mintha valami számolgotás önmagában izgalmas lehetne, de a matematika egyik legfontosabb konstansáról van szó,

amelyik a matematikusok számára a legritkábban jön elő, mint a kör félkerülete, sokkal inkább, mint egy fontos konstans a valószínűségszámításban, a komplex függvénytanban, stb. Itt az $n!$ Stirling-közelítésében bukkan fel. (L. a ?? oldal.)

Az alábbi középiskolai „feladat” megoldására van szükségünk.

19. feladat. Bizonyítsuk be, hogy ha a_n az egységkörbe írható szabályos n -szög oldalhossza, és $m_n = a_n^2/4$, akkor

$$m_{2n} = \frac{1}{2}(1 - \sqrt{1 - m_n}). \quad (215)$$

Induljunk ki az egységkörbe írt szabályos 6-szögből. Erre $m_6 = \frac{1}{4}$. A (215) alapján $m_{12}, m_{24}, m_{48}, m_{96} \dots$ rekurzive könnyen számolható, és $\pi = \frac{1}{2} \lim n \cdot a_n$. Ezen alapul az alábbi program:

8. Program (Szabályos sokszögek a körben).

```

      REM pi kozelítése
(1)  DEFDBL S
      INPUT „Hanszor duplazunk”;z
(3)  n=6: m=.25
(4)  FOR i=1 TO z
(5)  m=(1-SQR(1-m))/2
(6)  n=2*n
(7)  s=n*SQR(m)
(8)  PRINT i,s
(9)  NEXT i

```

Magyarázat: (1) azt kéri, hogy a kerületet dupla pontossággal számolja a gép: DBL a double=dupla rövidítése. A (3) sor beállítja a rekurzió kezdőértékét: szabályos hatszögre az oldalhossz 1, így $m_6 = \frac{1}{4}$. (5)-ben számítjuk ki a kétszeres sokszög oldalhosszának megfelelő m_{2n} -et, (6)-ban duplázzuk az oldalszámot, (7)-ben számoljuk az új kerületet.

12	3.1058	96	3.14103	$6 \cdot 2^{20}$	3.1415926	$6 \cdot 2^{31}$	3
24	3.1326	192	3.14145	$6 \cdot 2^{29}$	3.1374	$6 \cdot 2^{33}$	0
48	3.1393	384	3.14156	$6 \cdot 2^{30}$	3.18	Baj van!	

Részletesebben:

1	12	3.10583
2	24	3.13263
3	48	3.13935
4	96	3.14103
5	192	3.14145
6	384	3.14156
7	768	3.141583936817931
8	1536	3.141590425508945
9	3072	3.141592133056985
10	6144	3.141592474566481
11	12288	3.141592645321215
12	24576	3.141592645321215
13	49152	3.141592645321215
14	98304	3.141592645321215
15	196608	3.141592645321215
16	393216	3.141592645321215
17	786432	3.141592645321215
18	1572864	3.141592645321215
19	3145728	3.141592645321215
20	6291456	3.141592645321215
21	1.258291E+07	3.141592986830656
22	2.516582E+07	3.141592303811738
23	5.033165E+07	3.14159776795893
24	1.006633E+08	3.141586839655041
25	2.013266E+08	3.141674265021758
26	4.026532E+08	3.141674265021758
27	8.053064E+08	3.14307274017004
28	1.610613E+09	3.1486604294525
29	3.221225E+09	3.137475099502783
30	6.442451E+09	3.181980515339464
31	1.28849E+10	3
32	2.57698E+10	4.242640687119285
33	5.153961E+10	0
34	1.030792E+11	0
35	2.061584E+11	0

Ennek a programnak az eredménye nagyon függ attól, milyen programmal számoljuk a rekurziót. A program $z = 20$ -szal jó eredményt ad, $z = 21$ -re túlmegy π , ami csak hibás lehet, $z = 30$ körül megbolondul a program, $z = 33$ -ra teljesen rossz eredményt ad!

A baj abból származik hogy a kerületet, mint az oldalszám-szor oldal-

hosszat, $\infty \times 0$ típusú határértékként közelítjük. Ez könnyen kijavítható, ha felhasználjuk, hogy

$$1 - \sqrt{1 - m} = \frac{m}{1 + \sqrt{1 - m}}. \quad (216)$$

A részleteket átugorjuk. (Itt a gyöktelenítés éppen a numerikusan instabil kis különbségektől való megszabadulást jelenti.)

12.5.8. Stirling formula

A Stirling formuláról volt szó (a ?? oldalon), részletesen majd később fog szerepelni. Vannak pontosabb és durvább formái. Mi számítógéppel a ?? oldalon kimondott formáját illusztráljuk, hogy

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n. \quad (217)$$

9. Program.

```

REM === Stirling Formula ===
(1)  DEFDBL F, K, P
      CLS : INPUT "n="; n
      REM === A pontos ertek ===
(2)  pontos=1
      FOR i=1 TO n: pontos=pontos*i
      NEXT i
      REM === A kozelites ===
(3)  e=2.71828182845#
      kozel=(n/e)^n:
      kozel=kozel*SQR(2*n*3.141592653#)
      PRINT n; " | "; pontos; " | "; kozel,
      PRINT " arany: "; pontos/kozel

```

Magyarázat: Itt olyan nagy számokkal is dolgozunk néha, hogy nagyon óvatosnak kell lennünk. Az egyik óvatossági lépésünk, hogy (1)-ben dupla pontosságot állítunk be. A (2) utáni ciklus kiszámolja $n!$ pontos értékét. Utána kiszámoljuk a közelítő értéket, (217) alapján, majd kinyomtatjuk a két értéket és a hányadost. A 3.14... és a 2.71... utáni #-ot a gép tette oda, jelezve, hogy a számot dupla pontosságúnak veszi.

Próbáljuk ki a fenti programot. Ha $\alpha(n)$ -nel jelöljük a hányadost, $\alpha(5) = 1.016$, $\alpha(7) = 1.0119, \dots$, $\alpha(20) = 1.00417, \dots$. A Stirling formula tehát már kis értékekre is jól közelít és később tovább javul. (Vigyázat, ez nem bizonyítás!)

20. feladat. A Stirling formulának két része van: a $\sqrt{2\pi}$ szorzó és a főrész:

$$\left(\frac{n}{e}\right)^{n+(1/2)}.$$

A formula élészíthető:

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \frac{c}{n} + O\left(\frac{1}{n^2}\right)\right)$$

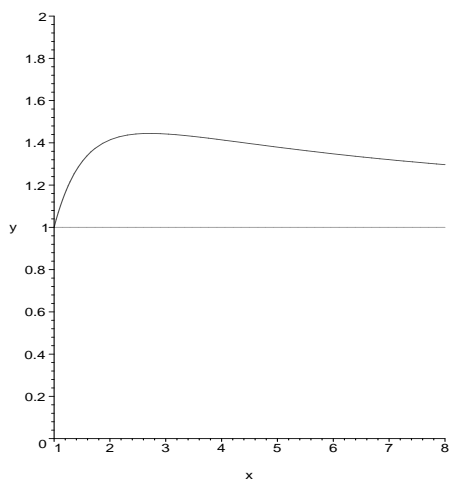
alakban. Próbáljuk megtalálni c értékét (egyszerűbb $1/c$ értékét keresni!)

21. feladat. Illusztráljuk a MAPLE, és a QBASIC segítségével, illetve bizonyítsuk be, hogy $\left(1 + \frac{1}{n}\right)^{n+\alpha}$ monoton csökkenő, ha $\alpha > 1/2$ és monoton növekvő, ha $\alpha < 1/2$. Mi a helyzet $\alpha = 1/2$ -re? Mi ennek a következménye a Stirling formula szempontjából?

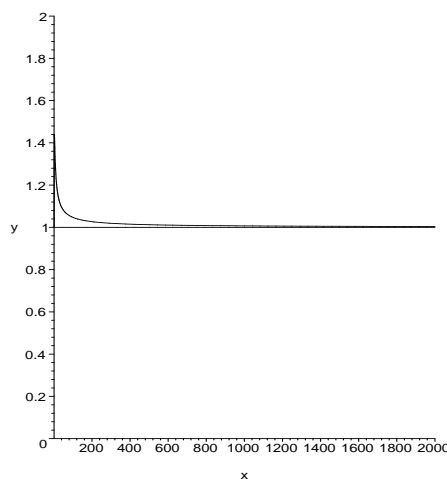
12.6. A matematika oldaláról

12.6.1. Konvergencia-sebesség

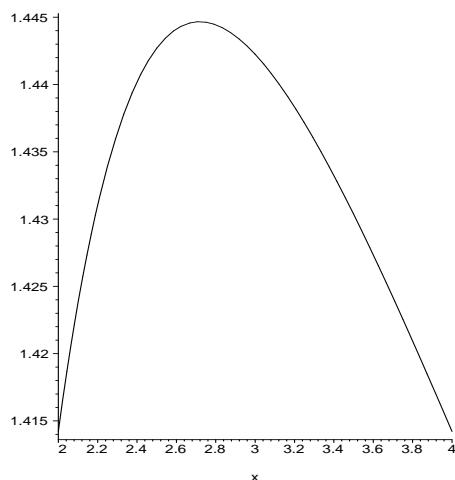
Most a MAPLE segítségével támadjuk meg iskola-feladatunkat: Arra vagyunk kíváncsiak, mihez tart $\sqrt[n]{n}$, és milyen gyorsan. Fogjuk fel a sorozatot úgy, mint az $f(x) = x^{1/x}$ függvény határértékét az $x = n$ sorozat mentén. Ha MAPLE-ben ábrázoljuk az $x^{1/x}$ -et, a következő adódik.



18/a. ábra: $x \in [1, 8]$



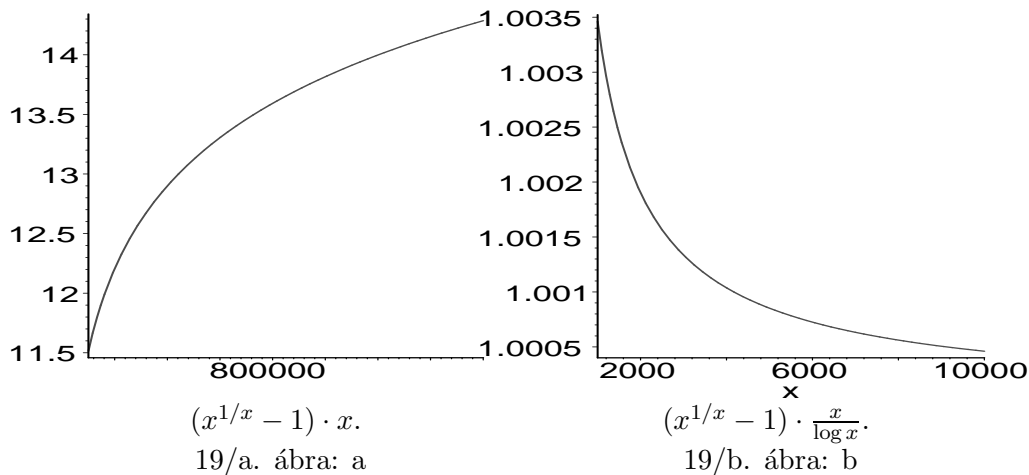
18/b. ábra: $x \in [1, 2000]$



18/c. ábra: $x \in [2, 4]$

Ebből valóban jól látható, hogy a függvény eleinte kicsit felmegy, majd lemegy, és nem túl gyorsan, de 1-hez tart. A 18/c ábra azt mutatja, hogy a függvény (valahol) az $x = e$ -ben éri el a maximumát.

(Amikor a függvénydiszkussziót tanuljuk, akkor ennek igazolása könnyű feladat: csak azt kell tudni, hogy $f(x)^{g(x)}$ helyett gyakran érdekesebb a logaritmusát vizsgálni.)



19/a. ábra: a

19/b. ábra: b

Mármost, a hiba nagyságrendjének megállapítása mehet „felezéses-próbál-
gatásos eljárással”, de nagyon óvatosnak kell lennünk. Ugyanazt tesszük,
mint a QBASIC programunknál. A két kapott függvény tükrözi ezt. A 19/a

függvénye végtelenhez tart (legalábbis úgy látszik), a míg a 19/a ábrán

$$(x^{1/x} - 1) \cdot \frac{x}{\log x} \rightarrow 1.$$

Az olvasó joggal lehet kíváncsi, hogy a konvergenciasebesség, amit a gépen próbáltunk megsejteni, matematikailag hogyan kezelhető. A válasz az, hogy ott, ahol ez felvetődött, ott még nem kezelhető, viszont amikor már tudjuk, hogy 0-ban $(e^x)' = 1$, akkor ez könnyűvé válik, lényegében azzal ekvivalens. Vázolunk egy megközelítést: a fentiből $\frac{e^u - 1}{u} \rightarrow 1$, ha $u \rightarrow 0$. Így $n = x = 1/u$ -t helyettesítve

$$\sqrt[x]{x} - 1 = x^{\frac{1}{x}} - 1 = \frac{1}{u^u} - 1 = e^{-u \log u} - 1 \sim -u \log u = \frac{\log x}{x}.$$

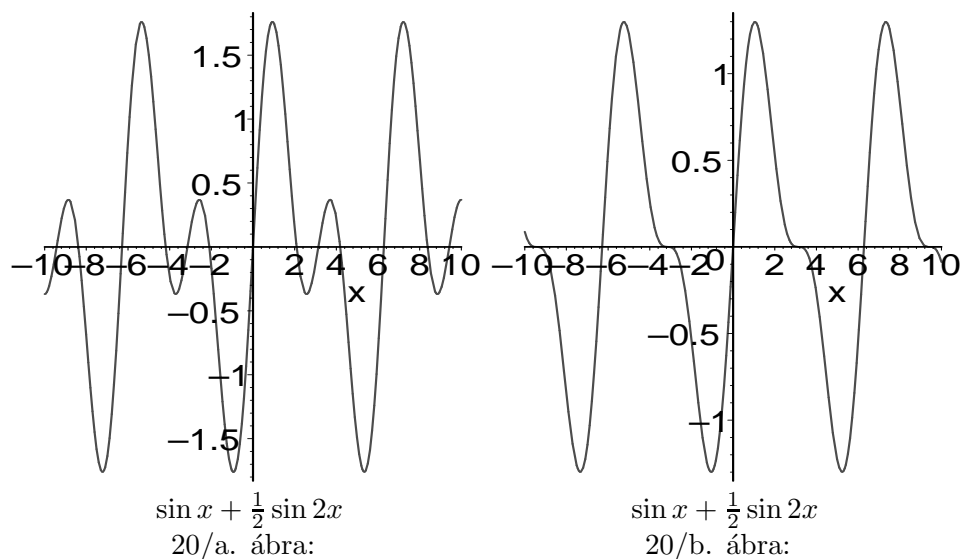
(Lásd a ???. oldal megfelelő feladatát.)

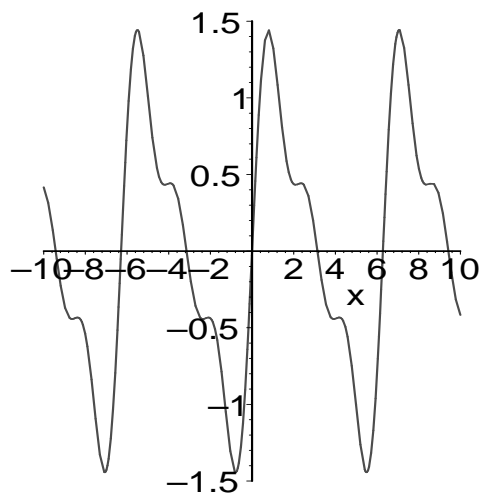
12.6.2. Trigonometrikus összegek

Megint csak abból indulunk ki, hogy mi az, amit MAPLE-ban elintézhettünk. A matematikában nagyon fontosak és érdekesek a trigonometrikus polinokok. Ezek (megfelelő normálással) a

$$\sum a_n \sin nx + b_n \cos nx$$

alakú függvények. Az alábbiakban néhány ilyen mutatunk be.





20/c. ábra:
 $\sin x + \frac{1}{2} \sin 2x + \frac{1}{3} \sin 3x$

A 20/a ábrám az $f(x) = \sin x + \sin 2x$, a 20/b-n $f(x) = \sin x + \frac{1}{2} \sin 2x$, a 20/c-n pedig az $f(x) = \sin x + \sin 2x + \frac{1}{3} \sin 3x$. Kérdezhető, hogy ezen függvények sorozata, pontosabban, a

$$f_n(x) = \sum_{k=1}^n \frac{1}{k} \sin kx \quad (218)$$

konvergál-e valamilyen értelemben valamilyen függvényhez.

22. feladat. Próbáljuk megsejteni, hogy mi a határértéke a (218) függvény-sorozatnak. (Kifejezhető a „törtrész” függvénnyel!)

Más szóval, az a kérdés, hogy mi mondható az (218)-ban megadott függvénytörzsről? Ha pl. f_{100} -at szeretnénk ábrázolni MAPLE-val, az is megtehető, de ehhez meg kellene tanulnunk a MAPLE programozását, legalábbis a ciklusírást. Ehelyett most egy BASIC programmal ábrázoljuk ugyanezt.

Az alábbiakban tehát egy olyan programot írunk, amelyik a (218)-ban megadott $f_n(x)$ -et rajzolja ki a képernyőre: célja, hogy kísérletezzünk, szemléljessük a közelítés sebességét.

A program meglehetősen „fapados”, a kényelmesebbé tételére szolgálnak az utána következő feladatok.

Azzal a megjegyzéssel kezdjük, hogy ma már BASIC-ben nincs mindegyik sornak sorszáma, csak annak amelyikre hivatkozunk. A mostani programban egyetlen sorszám sem kellene, ezeket a magyarázathoz írtuk be.

A következő program egyik újdonsága, hogy QBASIC-ben görbét rajzolunk.

10. Program (Fűrészfog).

```
10 CLS
   INPUT " n="; n: INPUT " height="; height:
   INPUT " b="; b
30 SCREEN 12
   FOR i=0 TO 1024
     oldy=y
     x=i*b/1024: REM x=x/10
     y=0
     FOR m=1 TO n
       y=y+height*SIN(m*x)/(m)
     NEXT m
60 PSET (i, 240) : PSET (i, y+240)
70 IF i > 0 THEN LINE (i-1, oldy+240)-(i, y+240)
   NEXT i
   PRINT "n="; n, "height="; height, "b="; b,
   REM Jo adatok: 5, 20, 30, vagy 50, 80, 50, ...
```

Magyarázat:

Itt három grafikus utasítást használunk: a grafikus képernyőre váltást: 30-ban, a pont-kirakást a 60 teszi meg, a simább görbe érdekében viszont (oldx, oldy)-ba elrakjuk (x,y) értékét, majd 70-ben az új pontokat összekötjük a régiekkel: a **LINE(i,y)-(i-1,yold)** a két pont közé húz egy egyenes szakaszt.

Az 10-es sor letörli a képernyőt, elhagyható lenne. A következő két sorban a program három paraméterét kérjük be: szabad egy sorba több utasítást írni, de -tal kell azokat elválasztani. Itt éppen azt kértük be, hogy hány tagú legyen a sin polinom, milyen magasra skálázzuk (y irányban) a görbét, hogy ne legyen túl lapos de ki se fusson a képernyőről, és hogy milyen $[0, b]$ intervallumon dolgozzunk. 30 átvisz a grafikus képernyőre, ahol rajzolni szoktunk. Ha a sorban IF-THEN-t használunk, akkor az IF feltétel teljesülése esetén (l. a 70-es sor), utána következő utasításokat mindet végrehajtja. 60 első fele kirajzolja az x-tengely i-edik pontját, a második fele pedig a felette levő pontot.

Ha a 70-es sort elhagyjuk, a program ugyanúgy fog működni, az előző sor második fele kirakja a pontokat. Amit kapunk, az néha kicsit szaggatott

lesz. Ez a sor egyenes szakaszokkal köti össze a konzekutív pontokat, így a kapott kép sokkal szebb lesz. Viszont a kezdőpontnál vigyáznunk kell.

Az utolsó előtti sor is felesleges: csak a szórakozottság ellensúlyozására van: elvben lefuttathatjuk a programot valamilyen adatokkal, megtetszik a kapott kép, de hirtelen nem emlékszünk, mik voltak az input adatok. Ez a sor kiírja azokat. Az utolsó sor egy feljegyzés magunk számára, hogy milyen változókkal sikerült számunkra kellemes képet „produkálnunk”.

Az alábbi feladatokban implicite feltesszük, hogy a függvényeink nem túl vadak: ismerve azon 200-2000 pontban az értékeiket, ahol megvizsgáljuk őket, a köztes intervallumokban jó közelítéssel lineárisoknak képzelhetjük őket.

23. feladat (Függvényábrázolás maximum-kereséssel). Változtassuk meg a 10. programot úgy, hogy

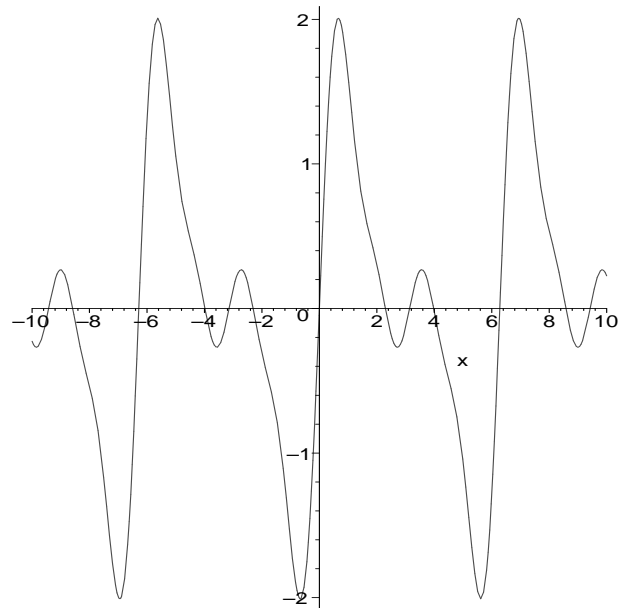
(a) előbb kiszámolja a függvény maximumát és minimumát, majd – egy második menetben, a maximum és minimum ismeretében – automatikusan olyan skálázást használjon, amelyikben a függvény görbéje elfér a képernyőn, de egyben azt kellemesen ki is tölti.

(b) Mindig tegye el az utolsó előtti és az azelőtti függvényértékeket, és ha az utolsó előtti nagyobb az előtte és utána következőnél, akkor ott húzzon egy függőleges vonalat (jelezve, hogy ott valószínűleg maximumhely van).

24. feladat (Függvényábrázolás diszkusszióval). Változtassuk meg az előző programot úgy, hogy

(c) Ahol a függvény monoton növekszik, ott pirosan rajzoljon, (**color(4)**) ahol a függvény monoton csökken, ott kéken (**color(1)**).

(d) A (c) helyett jelezzük ki a konvex és konkáv szakaszokat. (A konvexekre h lépésköz esetén $f(x) - f(x - h) > f(x - h) - f(x - 2h)$ jellemző.)



Ide berakunk egy érdekes alakú periódikus függényt, amely egy képlet-elírás nyomán született. Az (218)-beli $f_4(x)$ -ben $\sin 2x$ egyötthatója lemaradt.

12.6.3. Görbeseregek ábrázolása

25. feladat. Az e^x függényt az tűnteti ki az a^x alakú függvények között, hogy a 0-ban 1 a meredeksége, $f'(0) = 1$. Ennek vizsgálatára írjunk egy programot, amelyik $f(x, a)$ alakú, a -val paraméterezett görbeseregeket rajzol ki. (Itt $f(x, a) = a^x$.)

Megoldás:

11. Program.

- (1) SCREEN 12
- (2) ax=100: ay=100: lep=.05: e=2.71821828#
- (3) x=0: FOR y=-1 TO 2 STEP .03: GOSUB rajzol: NEXT y
- (4) y=0: FOR x=-2 TO 4 STEP .03: GOSUB rajzol: NEXT x
- (5) LINE (200-200, 200+200)-(200+200, 200-200)
- (6) FOR a=e-2 TO 2+2 STEP .25
- (7) FOR x=-2 TO 4 STEP lep
- (8) IF ABS(a-e) < .1 THEN szin=4 ELSE szin=3

```

(9) yold=y: oldx=x-lep:
(10) y=a ^x:
(11) GOSUB rajzol
(12) NEXT x
(13) INPUT w$
(14) NEXT a
(15) END
(16) rajzol:
(17) IF x < -1.9 THEN RETURN
(18) LINE (200+ax*oldx, 300-ay*yold)-(200+ax*x, 300-ay*y), szin
(19) RETURN

```

Magyarázat: Ebben a programban a kirajzolást egy „rajzol” nevű szubrutin-nal oldottuk meg. Három különböző helyről is meghívjuk, a szubrutin a LINE paranccsal egy kis vonalkát húz az előző pont és az új pont között. Ha csak egy pontot tennénk ki (a PSET(X,Y) paranccsal) a kapott görbe pontokból állna, kicsit szagatott lenne. A RETURN hatására a program mindig az azutáni parancsra tér vissza ahonnan ideugrott.

Rajzoláskor gondot szokott okozni, hogy az ábrák kifutnak a képernyőről, vagy pedig nagyon kicsire sikerülnek. Ezzel többnyire el kell játszanunk. A szubrutinos formának itt elsősorban az az előnye, hogy a beszkálázás így könnyebb.

Az (1)-ben térünk át a grafikus képernyőre. A SCREEN 12-es paramétere biztosítja, hogy színesen rajzolhassunk. (2)-ben állítjuk be a paramétereket: „ax”, „ay” az x ill. y irányú nagyítás, „lep” a lépésköz, e a természetes logaritmus alapja. „szin” adja a színt. Itt csak annyi kell nekünk, hogy amikor az a^x -ben az a e -hez közel jut, akkor más színnel rajzoljunk ki.

(3) és (4) a koordináta-tengelyeket rajzolja ki.

A program lelke a (9)-(12) rész: ha más függvényre vagyunk kíváncsiak, csak a (10)-et kell megváltoztatnunk és a skálázást. (9)-ben tesszük el az előző értéket, hogy a görbéinket szakaszokból állíthassuk össze, pontok helyett.

A görbesereg paraméterét, a -t a (6) és a (14) sor szervezi ciklusba. A (13) sor minden görbe kirajzolása után megállítja a programot, azzal, hogy bekéri w \$ értékét. Így szöveget kér (w \$ jelzi ezt), elég csak az ENTER-t lenyomnunk. (Ha „input a”-t írnánk számot kérne, ezt kellene beírni.)

A (15)-beli END azért kell, hogy a szubrutinba csak a GOSUB-on keresztül érkezhessünk be. Enélkül az utolsó a után is bemegy a szubrutinba,

majd hibát jelez.

(8) intézi el, hogy e^x más színnel jelenjen meg, mint a többi hatványgörbe.

12.6.4. Globális polinom-közelítés

Mikor használjunk komolyabb programcsomagokat? Ha fogalmakat akarunk számítógéppel megközelíteni, gyakran egyszerű programokat írhatunk a probléma-mára, olyankor a gép segít. Alább egy olyan problémát mutatunk be, ahol egyértelmű, hogy BASIC-ben és MAPLE-ben egyaránt megoldhatjuk, de a BASIC program megírása sok időt vehet igénybe.

Interpoláció. Azt a problémát nevezik interpolációnak, amikor egy f függvénynek adott az értéke n pontban és olyan $n - 1$ -edfokú polinomot keresünk, amelynek az adott pontokban ugyanez az értéke. A 281. oldalon olvashatunk róla. A MAPLE segítségével viszont kiismerhetjük. A tárgyalás alább vázlatos, de a gép használata mellett követhető.

A $[0, 1]$ intervallumon vett egyenletes felosztást tekintjük:

Adott egy f folytonos függvény a $[0, 1]$ -en, azt az $x_n = i/n$ pontokkal n egyenlő részre osztjuk, majd megkeressük azt a $P_n(x)$ polinomot, amelyre $P_n(x_i) = f(x_i)$ ($i = 0, \dots, n$). Azt reméljük (naivan) $P_n(x) \rightarrow f(x)$ minden $x \in [0, 1]$.



Az $|x|$ -hez keresünk egy 7 ponton interpoláló (6-odfokú) majd egy 13 ponton interpoláló (12-edfokú) polinomot, majd kirajzoljuk ezeket. Behívjuk a MAPLE-t. Ezután a leghelyesebb a Help-ben megkeresni a „CurveFitting”-et és kicsit tanulmányozni. Mi ezt átugorjuk itt. A

> **with(CurveFitting);**

hívja be a megfelelő programkönyvtárat,

> **with(plots)** kell majd a kirajzoláshoz (legalábbis a korábbi verziókban).

> **f := abs(x);** (megadjuk a függvényt)

$$f := |x|$$

Beírjuk a **PolynomialInterpolation**-ba a 7 „egész” ponton a függvényértékeket.

> **g:=PolynomialInterpolation([-3,-2,-1,0,1,2,3],[3,2,1,0,1,2,3],x);**

$$g := \frac{1}{60} x^6 - \frac{1}{4} x^4 + \frac{37}{30} x^2 \quad (\text{a polinom})$$

> `plot([f,g],x=-3..3);` (kirajzoltatjuk)

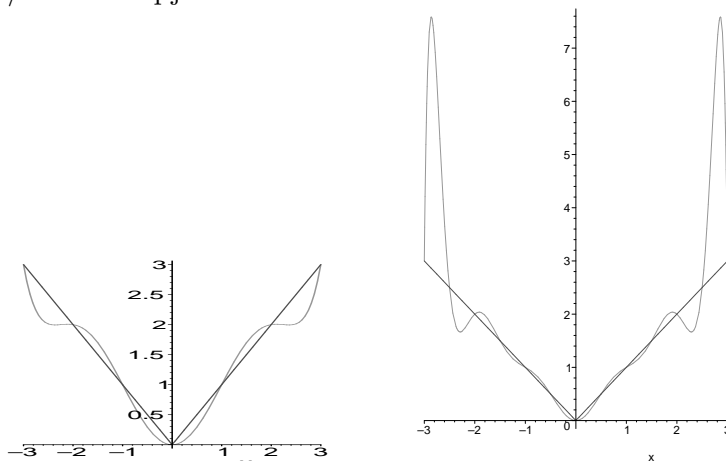
A 21/a ábra mutatja az eredményt. Most a „fél-egész” pontokat is beadjuk:

> `h:=PolynomialInterpolation([-3,-2.5,-2,-1.5,-1,-0.5,0,0.5,1,1.5,2,2.5,3], [3,2.5,2,1.5,1,0.5,0,0.5,1,1.5,2,2.5,3],x);`

(A két sor a gépben egyetlen sor.) A gép ezen a ponton kiírt egy hosszú polinomot, csúnya együtthatókkal, amit most kihagyok. Beadjuk, hogy

> `plot([f,h],x=-3..3);`

mire a 21/b ábrát kapjuk:



Az automatikus skálázás torzít

21/a. ábra: `plot([f,g],x=-3..3);` 21/b. ábra: `plot([f,g],x=-3..3);`

A számítógép tehát azt sugallja, hogy ez az interpoláció egyáltalán nem közelít.

A Bernstein polinomok. A POLINOM APPROXIMÁCIÓ fejezetben találkoztunk velük, ahol Weierstrass approximációs tételével kapcsolatban. A következő képlet adja meg az f -et a $[0,1]$ -en közelítő n -edfokú Bernstein polinomot:

$$g(x) = B_n(x; f) = \sum_{k=0}^n \binom{n}{k} f\left(\frac{k}{n}\right) x^k (1-x)^{n-k} \quad (219)$$

Két függvény 8-adfokú közelítését rajzoljuk ki, de az alábbi „munkalapban” a függvényt átírva, bármelyik másik függvény tetszőleges fokú polinomközelítését is kiszámolhatjuk.¹⁵ A 22/a ábrán $\arctg(9x - 3)$, és a Bernstein polinomja. A 22/b-n pedig $g = |x - 0.4|$ és a Bernstein polinomja látható. (Készakarva nem $\frac{1}{2}$ -re szimmetrikus függvényt választottunk, mert az nem mutatná, ha x -et és $1 - x$ -et felcseréltük volna..)

¹⁵Persze, ha nagyon sokat kívánunk a géptől, a program elszállhat.

Az alábbi MAPLE Work Sheet -et (bernstein.mws) írtuk erre a célra:

```

> restart: (memóriatörlés)

> f:=abs(t-.4): (függvény megadása)

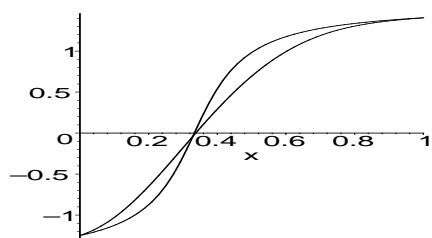
> g:=sum(binomial(n,k)*x^k*(1-x)^(n-k)*subs(t=k/n,f),k=0..n):

> h:=subs(n=8,g): (n = 8-adfokúval approx.)

> plot([h,abs(x-.4)],x=0..1,color=black); (kirajzoljuk
g, h-t.)

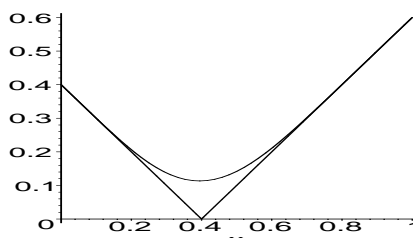
```

Az következő ábrákat kaptuk.



$\arctan(9x - 3), x \in [0, 1]$

22/a. ábra:



$|x - 0.4|$

22/b. ábra:

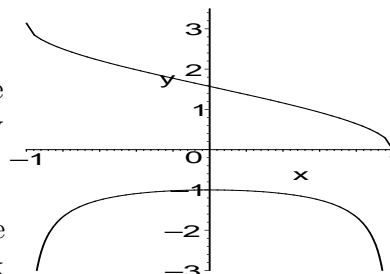
Magyarázat

- Újraindításkor célszerű mindig a **restart**-ra ráállni a kurzorral és nyomogatni az **ENTER**-t: ez azért kell, hogy először a memória törlődjék: tiszta memóriával induljunk. Ez a MAPLE használatánál nagyon fontos: ha erről elfelejtkezünk, helytelen eredményt kapunk. (Van egy Restart gomb is!)
- A **subs(x=...,f(x,y))** behelyettesítés, az összeget a **sum** és az $\binom{n}{k}$ -t a **binomial(n,k)** parancsokkal állíthatjuk elő.
- A 2. sorban adjuk meg a függvényt.
- A 3., kivastagított sor állítja elő a Bernstein polinomot, a (219) alapján. Ebben a **sum(u(k),k=0..n)**; a $\sum_{k=0}^n u(k)$ -val ekvivalens. A **subs(t=k/n,f)** pedig azt jelenti, hogy f -be helyettesítsünk $t = \frac{k}{n}$ -et.
- A következő sorban döntjük el, hányadfokú polinommal approximálunk.
- Az utolsó sor kirajzolja az eredeti függvényt és a közelítését.

A fenti „program” más függvényekre is működik: a 2. és az utolsó sorában kell **abs(x-.4)**-et lecserélni.

26. feladat. Miért nem működik a

`> plot([h,f],x=0..1,color=black);`
sor a fenti programban? Hogyan lehetne elérni mégis, hogy a függvényt csak egy helyen kelljen beírni?



27. feladat. Írjuk be a MAPLE-be $\arccos x$ -et, deriváljuk, majd rajzoljuk ki egy koordináta-rendszerbe őt és a deriváltját. (lásd az 23. ábrát.)

23. ábra: $\arccos x$ és deriváltja

Hivatkozások

- [1] Fried Kati és Simonovits Miklós: A számítástechnikai gondolkodás iskolája, 2005, Typotex.
- [2] Szász Pál I. 689. old. 404. old 685. old