

Számítógép alkalmazása a matematika tanításában

Kovács Zoltán



Szegedi Tudományegyetem

2011

Tartalomjegyzék

1. Bevezetés	3
2. Előszó	3
3. A számítógépes matematikaóra tartalmi és formai keretei	4
3.1. Mikor használjunk számítógépet?	4
3.2. Milyen módon használjuk a számítógépet?	5
4. A matematikai szoftverek osztályozási lehetőségei	6
4.1. Technikai osztályozás	6
4.2. Téma szerinti csoportosítás	7
5. Beszerzés	9
5.1. Telepítés Windows alá	9
5.2. Telepítés Linux alá	10
6. A szoftverek	12
6.1. wxMaxima	12
6.2. Sage	12
6.3. matek.hu kalkulátor	13
6.4. GeoGebra	13
6.5. Mathematica	14
6.6. Euklides	14
6.7. Euler 3D	14
6.8. JSXGraph	14
6.9. GNU XaoS	15
6.10. LyX	15
6.11. \TeX macs	15
6.12. Moodle	15

Előszó az elektronikus változathoz

Ezen dokumentum eredetije egy elektronikus wiki-szócikk. Jelenleg olvasható formája a „Jó szomszédok a közös jövőért” (IPA HU-SRB/0901/221/088, 2010-2011) projekt támogatásával valósult meg. Az interneten lapozható, HTML alapú formátum a matek.hu tudástárban érhető el.

1. Bevezetés

Ez az oldal a **Számítógép alkalmazása a matematika tanításában** egyetemi kurzushoz készített elektronikus jegyzet. A kurzus a Szegedi Tudományegyetem matematika BSc alapszakának matematika szakirányán kötelező tárgy.

A tárgy célja leendő matematika tanárok számára olyan ismeretek átadása, melyek a számítógéppel segített matematika oktatásában elsődlegesek. A kurzus során elméleti, gyakorlati és technikai tudnivalók ismertetésére is sor kerül.

A jegyzet ebben a formájában inkább kibővített vázlatnak tekinthető, mintsem teljes és részletes leírásnak. Több, a matek.hu tudástárban olvasható szócikkben további tudnivalók olvashatók. Sok külső hivatkozás is szerepel, amelyeknek célszerű utánajárni, mivel a leírtakat bővebben megvilágítják. Habár a külső hivatkozások egy része angol nyelvű, az Olvasót bátorítjuk ezen külső cikkek megismerésére is.

A jegyzet fejleszthető és kiegészíthető, akár az Olvasó számára is (ehhez regisztráció szükséges). Reményeink szerint az oldal fejlődése az aktuális szoftverek megjelenésével folyamatosan fog történni, az új szócikkekre új hivatkozások jelennek majd meg, az elavult információkat pedig törölni fogjuk. Amennyiben az Olvasó pontosítani, bővíteni, kiegészíteni vagy javítani tudja ezt az anyagot, köszönettel vesszük segítségét.

2. Előszó

A személyi számítógépek megjelenésével az oktatásban is egyre inkább megjelentek és elterjedtek azok a módszerek, amelyek a számítógépet feladatmegoldóként, segítőként, illusztrációs eszközként használják. Míg kezdetben egy iskolában legfeljebb egy vagy néhány számítógép volt, ma már szinte minden diáknak van saját számítógépe: a mobiltelefonja (hogy az otthoni számítógépet ne is említsük). Így a matematika számítógépes oktatása egy dinamikusan fejlődő pályát futott be, s ez a fejlődés még ma is tart.

Példák és kapcsolódó gondolatok — 1–1 mondatban. Az OLPC projekt 2011 novemberéig több, mint 2 millió gyermeklaptopot biztosított a fejlődő országok iskoláinak. A GeoGebra projektben kiemelt figyelmet kap a GeoGebraMobile fejlesztése, melynek célja, hogy a szoftver okostelefonokon is futathassák a tanulók. Az okostelefonok iskolai használatának engedélyezésére ma még nincs kiforrott gyakorlat, Magyarországon inkább tiltják vagy nehezményezik, mintsem bátorítják használatukat; mindehhez az informatikáról való szemléletünk megújulása is szükséges.

Ez a fejlődés a tanári hozzáállást is átalakította. Amellett, hogy ma a tanárképzésben a számítógépes eszközök oktatása kötelező elem, az évtizedes gyakorlattal rendelkező tanárok is maguk érzik szükségét, hogy továbbképzéseken, tanfolyamokon "érjék utol" tanítványaikat, s kihasználják a technológiában rejlő lehetőségeket.

Példák és kapcsolódó gondolatok — 1–1 mondatban. Ismeretes, hogy Magyarország nem egy vezető középiskolájában a mai napig nem elfogadott, hogy a matematika oktatásába szoftveres segítséget vigyünk: a matematika és az informatika munkaközösség olykor nem dolgozik össze, a meglévő szoftvereket sem használják, az ajándékba kapott "dobozos" szoftvert ki sem bontják. Többek között a szegedi GeoGebra tanfolyamok, így többek között a 2010-es, valamint a 2010 novemberére óta a Varga Tamás Napokon kezdeményezett GeoGebra konferenciák is a változás pozitív előszelei.

A belső indíttatás mellett a külső kényszernek is szerepe van: számos iskolában előírás, hogy a nemrégiben beszerzett digitális táblákat szakértő tanárok használják, vagyis kihasználják az azok nyújtotta lehetőségeket. A központilag előírt tanártovábbképzések egy része a digitális írástudást és annak fejlesztését tűzi ki célul.

3. A számítógépes matematikaóra tartalmi és formai keretei

3.1. Mikor használjuk számítógépet?

A számítógép használatának jogosultsága a matematika oktatásában és kutatásában régóta vita tárgya. A négyosztályos 1976-ban publikált számítógépes bizonyítása volt az első áttörés, ami a számítógépes programokat a matematikusok egy komoly hányada számára úgymond "partneri szintre" emelte. Azóta számos számítógépes program látott napvilágot, amelyek hasznosságát a matematikus társadalom nagyrészt elfogadta. Fontos megjegyezni, hogy a számítógép az emberi kreativitást csak annyiban tudja leutánozni, amennyire azt számára megtanították; hogy a mesterséges értelem végül is utolérheti és helyettesítheti-e valaha is az emberi zsenialitást, az továbbra is kérdés, és egyéb filozófiai-vallási kérdéseket is felvet.

Oktatási szempontból a legfontosabb kérdések egyike, hogy milyen célból kívánjuk használni a számítógépet; itt most számítógépen akár számológépet is érthetünk: minden olyan eszközt, amely elektronikusan működik, és gépi algoritmusokat használ számítások elvégzésére.

Példák és kapcsolódó gondolatok — 1–1 mondatban. A papíron számított négyzetgyökök hosszú évtizedekig kötelező tananyag volt iskoláinkban, de immáron nem része a követelményeknek — feltétlenül baj-e ez? Egy időben közelebbi példa: az irracionális függvények függvénytábla segítségével történő kiszámítása még része a tantervnek — vajon meddig, és feltétlenül baj-e ez?

A matematikai ismeretszerzés, de általában bármilyen ismeretszerzés a fokozatosságon alapul, és bizonyos tényeket bizonyítás nélkül kell elfogadnunk. Nem véletlen, hogy a szorzótáblát mind a mai napig a "bemagoltatás" módszerével tanítják, de hasonló példa, hogy minden további bizonyítás nélkül elfogadjuk, hogy az $y = x$ függvény grafikonja folytonos.) A matematikaoktatás végső célja olyan emberek nevelése, akik alkotó módon tudnak a világ problémáival megküzdeni, és abban eredményesek; ez nem csupán az elméleti, de a gyakorlati életben is fontos. Nem mindegy, hogy ki a célközönségünk: egy matematikus számára érdekes és fontos lehet egy bizonyítás minden részletének aprólékos kiírása, míg egy vegyész hallgató számára az a fontos, hogy biztonságos tudjon deriválni (hiszen később integrálni is tudnia kell).

Példák és kapcsolódó gondolatok — 1–1 mondatban. Bruno Buchberger "black box-white box" cikke azt tanácsolja, hogy bizonyos esetekben (és ezek az esetek akár többségben is lehetnek) kifejezetten káros, ha túlságosan részletekbe megyünk, egy ilyen példa a matematikai háttér túlságos magyarázata a nem értő hallgatóságnak — azonban ha valaki valóban kíváncsi a részletekre, miért nem bonthatná ki a "fekete dobozként" ismertetett eljárásokat, gondolatokat? A matek.hu egyik tervezett továbbfejlesztési iránya, hogy a végeredmények részletes levezetését is meg lehessen kapni, atomi szinten.)

Számos esetet lehet felsorolni, amikor a számítógép többet árt, mint használ:

- Ha a diák absztrakciós lépéseket hagy ki azáltal, ha a gép gondolkozik helyette. (Pl. megcsinálja a házi feladatát, ahelyett, hogy csak ellenőrizné.)
- Ha a vizualizáció eltereli a figyelmet, és nem kiegészíti. ("Szép, de nem értem." Pozitív példa: derivált szemléletes jelentése.)
- Ha a szoftver hibás működése miatt rossz ismeret alakul ki. (Néhány példa: Még egyetemi szakdolgozatba is kerülhet ilyen szarvashiba, pl. a $\sin x$ ábrázolása félkörökkel. Az interneten keresve a Google által felajánlott "javasolt" függvényábrázoló is lehet pontatlan, pl. az egészrész vagy törtrész függvény ábrázolásánál könnyen félrevezető ábrát kapunk, "on-line function grapher", próbáljuk meg a **floor(x)** függvényt ábrázolni. A Se veled, se nélküled előadás első példái arra világítanak rá, hogy mindig fel kell készülnünk rá: a számítógép sugalmazhat hibás, pontatlan eredményeket is.)

Hasznos a számítógép, ha:

- Már rutinná kimunkált műveletsort felgyorsít, automatizál. Így a feladat lényegi, izgalmas részére koncentrálnunk.
- Sejtés megfogalmazásában segít.
- Már ismert fogalmat elmélyít, más szemszögből is bemutat.
- Bizonyos nehezen megfogalmazható fogalmat szemlélet útján vezet be (pl. derivált fogalma a függvényen mozgó érintővel).

Feladatok.

1. Keressünk további hibás függvényábrázoló programokat! Tipp: próbálkozzunk az $\frac{1}{\tan x+x}$ ábrázolásával. Ha a szoftver képes a függvény értelmezési tartományát is megállapítani, miért lesz nehéz dolga ezzel a függvénnyel?
2. Vizsgáljuk meg a matek.hu segítségével a tangensfüggvény deriváltjait: 1., 2., 3.! Folytatván a számolást, fogalmazzunk meg sejtéseket a deriváltakban megjelenő tagkifejezésekről!

3.2. Milyen módon használjuk a számítógépet?

Tanórán.

- Koncentrált géptermi óra. Kis létszámmal célszerű, 10 fő fölött segítővel. Irányított, alaposan megtervezett óra, frontális ("csináljuk együtt") és/vagy önálló munka ("csináld magad", munkalappal). Alkalmi lehetőség (előre lefoglalt gépterm); rendszeres géptermi órák esetén is csak időszakosan szabad a gépeket használni. (Tehát az idő nagyobbik része ne számítógépes legyen. Kivétel: ha kifejezetten számítógépes programokat bemutató tanfolyamról, szakkurzusról van szó.)
- Projektoros bemutató. Csak tisztán ezt használva többnyire "álmosító". Figyeljünk az interaktivitásra! Hasznos lehet fehér táblás teremben tartani az órát (hogy írni is kelljen az előadónak is, ezáltal a hallgatónak is), esetleg digitális táblát alkalmazni (a még több interakcióhoz).
- A számítógép mindenkinél ott lévő "kicsi eszköz" (akár egy számológép). Mindez anyagi, szervezési problémákkal járhat. (Magyarországon néhol, külföldön pedig számos helyen használnak Texas Instruments gépeket, valamint okostelefonokat is.)

Technikai tippek:

1. Lehetőleg próbáljunk ki előre a helyszínen mindent. (Ez sokszor részben vagy teljes egészében lehetetlen!) Gondoljunk arra, hogy a hallgatóságnak esetleg más belépési kódja van, mint nekünk, és azzal esetleg az általunk bemutatott szoftver nem érhető el.
2. Legyen a tarsolyunkban egy B (esetleg C) terv technikai problémák vagy áramszünet esetére.
3. A laptopok és a projektorok összekapcsolása nem mindig automatikus, néha speciális billentyűkombináció (Fn helyett Ctrl-Alt) segít, néha a távirányító valamelyik elrejtett funkciója, néha újraindítás. (A projektorok nem mindig indulnak azonnal újra, csak néhány perc hűtés után!)
4. Legyen a közelben technikus/rendszergazda kolléga.
5. Tartsunk fegyelmet!
6. Amennyire lehetséges, törekedjünk arra, hogy a hallgatóság értelmi és technikai szinten homogén legyen. Ha egyedül tartjuk a frontális órát, időről időre (6-8 percenként) fussunk végig a termen, hogy mindenki tud-e haladni a tervezett tempóban.
7. A lézeres mutató hasznos, s célszerű csereelemről is gondoskodni.

Házi feladathoz.

- Webről letölthető statikus anyagok (dolgozathoz mintapéldák).
- Webről letölthető vagy on-line dinamikus anyagok. (Példa: <http://www.geogebra.org/en/upload/files/magyar/hogabo/Fuggvenyek/linearis.html>.)
- Statikus vagy dinamikus feladatlapok. (Pl. Google Docs űrlapok, WeBWorK, STACK.)
- Teljes oktatási modul. (Pl. Moodle tananyag.)
- Keresési feladat. (Keresőgépeken, pl.: Google; szakértői rendszerekben: pl. WolframAlpha; tudástárakban: pl. Planetmath.)

Órai felkészüléshez (tanáré).

- Dolgozatfeladatlap (matematikai szövegszerkesztő célszerű).
- Feladatmegoldás kutatóként (komputeralgebrai vagy dinamikus geometria rendszer célszerű).

Feladatok.

1. Az általunk látogatott egyetemi kurzusokon figyeljük meg, hogy mely oktatók használnak számítógépet. Milyen módon használják a gépet? Lekötik a figyelmet vagy inkább ellankasztják? Milyen arányban van az óra számítógépes és nem számítógépes része? Milyen trükköket és tipikus hibákat figyelhetünk meg, vagyis mi az, amit mi is követendőnek tartunk, s mi az, amit nem?

4. A matematikai szoftverek osztályozási lehetőségei

A tanár számára másodlagos kérdés, hogy az általa használt szoftver milyen típusú számítógépen működik, milyen licenccel futtatható. A mai világban egyre inkább elfogadott és elvárt, hogy egy számítógépes program futtatása semmilyen technikai nehézségekbe ne ütközzön, "csak működjön és kész". Ahogyan a matematikai szoftver is tömegcikké válik, úgy lesz egyre kényelmesebb ezeknek a programoknak mind a telepítése, mind a használata. Ha az iskolában a rendszergazda "a kezünk alá dolgozik", valóban minden automatikusan működhet. Nem minden iskolában van azonban rendszergazda, és bizonyos technikai ismeretekkel lehet, hogy ő sem rendelkezik. Lássuk ezeket összefoglalva.

4.1. Technikai osztályozás

Architektúra, operációs rendszer. A legelterjedtebb hardver ma az IBM PC kompatibilis személyi számítógép, melyek többsége az Intel által kifejlesztett processzorokon alapul. Adott számítógép típuson rendszerint többféle operációs rendszer is futtatható, ilyen szoftver pl. a Microsoft Windows, a Mac OS X vagy a Linux. Az operációs rendszereken belül is többféle variáns létezik, pl. a Windowson belül is létezik Windows 7, Windows Vista vagy Windows XP, amelyek részben kompatibilisek lehetnek egymással; azonban egy adott szoftver lehet, hogy csak egy előre adott variánsra fog működni.

Szakértelmet kíván tehát, hogy eldöntsük, hogy egy általunk használni kívánt szoftver abban a gépteremben vagy azokon a számítógépeken, melyeken dolgozni kívánunk, működni fog-e. A gyártók is felismerték a bőség zavarának ezt a problémáját, és ahelyett, hogy minden egyes géptípusra, operációs rendszerre és azon belüli variánsra kifejlesszék a szoftverük speciális változatát, olyan technológiákat kezdenek el alkalmazni, amelyek lehetőleg "minden" rendszeren automatikusan működnek. Ez az elv a *platformfüggetlenség* elve, amely a géptípusok különböző architektúrája ill. az operációs rendszerek különbözősége esetén is sikeres lehet.

A kompatibilitás (azaz a hordozhatóság) megvalósítása a platformfüggetlen szoftverek esetében többnyire egy szoftveres kompatibilitási réteg segítségével történik. Létezik architektúra szintű (ilyen szoftver pl. a QEMU) és operációs rendszer szintű emuláció vagy virtualizáció (ilyen pl. a Wine, amellyel Windows programokat futtathatunk más operációs rendszereken) is (az emuláció és virtualizáció kifejezéseket itt szinonimaként használjuk). Manapság azonban az egyre inkább elterjedt Java ill. Javascript programozási nyelvek kerülnek abba a szerepkörbe, hogy az ezeken a nyelveken készült alkalmazások minden számítógépen futtathatók lesznek, mivel mindkét nyelven hatékony virtuális gépeket ill. futtató környezeteket készítettek az elmúlt évek során. Mindez a végfelhasználó számára azt eredményezi, hogy egy népszerű alkalmazást (legyen az az APEH által készített Abev szoftver vagy a GeoGebra dinamikus geometria szoftver) szinte bárhol futtatni tudunk, függetlenül architektúrától és operációs rendszertől. Másrészt az egyre inkább elterjedő, web alapú on-line szoftverek (ilyen pl. a matek.hu) a korszerű webböngészőkben hatékonyan és lényegében ugyanolyan módon futnak a mögöttes Javascript motor segítségével.

A jövő mindenképpen a kompatibilis alkalmazásoké, de ma még számos olyan szoftver van, amelyet az általunk használt platformon külön telepítőkészlettel kell installálnunk. Ilyen példa a nagy komputeralgebra rendszerek feltelepítése, így pl. a Maple esetén külön DVD-t kell beszerezni, ha Windowsra vagy Linuxra akarjuk telepíteni. Sőt, számos szoftvernek nem is létezik speciális változata a ritkábban használt platformokra, így pl. az Euler 3D szoftver csak Windowson fut, más operációs rendszeren nem (vagy csak emulációval).

Licencelés. A PC bumm idején, azaz a '80-as évek elején Magyarországra csak nem hivatalosan ill. félhivatalosan juthattak be számítógépek, hiszen a politikai berendezkedés miatt hazánk sokáig a Nyugat elektronikai termékeinek tiltólistáján szerepelt. A szoftverek beáramlását azonban nem tudták lassítani a politikai határok: a szoftvermásolást sokszor hobbi programozók, önszerveződő számítógépes klubok karolták fel, miután az országba nehezen ellenőrizhető úton bejutott az eredeti szoftverek "tört", "crackelt" változata.

Így Magyarországon egészen a '90-es évek elejéig-közepéig nem is volt hivatalosan tilos a szoftvermásolás, és még manapság is a társadalom komoly hányada szerint erkölcsileg elfogadhatónak számít a kereskedelmi szoftverek lemásolása, még akkor is, ha a törvény amúgy néhány éve akár szabadságvesztéssel is sújthatja¹ az ilyen jellegű illegális tevékenységet. Tanárként külön figyelemmel kell legyünk arra, hogy a megszokott, ám tisztességesnek semmiképpen sem mondható jogtalan szoftverhasználatot ne erősítsük tanítványainkban.

Már csak annál is inkább, mert számos olyan alternatíva létezik, amely a kereskedelemben kapható (rendszerint borsos árú) szoftvereket ingyenes megfelelőekkel helyettesíti. Az ilyen számítógépes programokat szabad szoftvereknek nevezzük, mert készítőik a szoftver használatát és továbbadását szabadon biztosítják. (A "szabad"-ság nem feltétlenül ugyanaz, mint az ingyenesség. Tehát teljesen ésszerű lehet, ha egy matematika tanár a magántanítványától valami jelképes díjat kér azért, hogy CD-re felmásolta neki a szoftvert, és készít hozzá egy 1-2 oldalas ismertetőt.)

Egyes matematikusok és szoftverfejlesztők még ennél is tovább mennek. A zárt forráskódú programok, melyek működését a szerző cégek üzletpolitikai megfontolásokból nem árulják el, tudományos szempontból is aggályosak lehetnek. Ugyanis ha nem ismerhetjük meg egy bonyolult rendszer (pl. egy komputeralgebrai szoftver) pontos működését, akkor nem is bízhatunk abban teljes egészében, hogy jól számol, vagyis a számítások eredménye biztosan helyes. Így a tudományos szférában és az oktatásban sokszor jobban elfogadottak a szabad szoftverek, mint kereskedelmi társaik.

Feladatok.

1. Az általunk leggyakrabban használt számítógépnek gyűjtsük össze a jellemző technikai paramétereit: Milyen architektúrájú a számítógép? 32 vagy 64 bites? Milyen erős a CPU? Hány magos? Mekkora a fizikai RAM? Milyen típusú a videokártya? Mekkora a merevlemez? Van-e számítógépben hálózati kártya, wifi, bluetooth? Hány USB csatoló van? Van-e optikai meghajtó; ha igen, tud-e a számítógép DVD lemezt írni, vagy csak olvas, továbbá képes-e kétrétegű DVD-k kezelésére? Ha laptopunk van, van-e beépített mikrofon és webkamera? Milyen további extrák jellemzik a számítógépünket?
2. Keressük meg az interneten a GPL 3-as verziójának magyar fordítását. Próbáljuk az Előszót néhány mondatban összefoglalni.

4.2. Téma szerinti csoportosítás

A technikai részleteknél a tanár számára fontosabb, hogy **milyen** szoftverek is léteznek, milyen típusú szoftverekből válogathat.

Komputeralgebrai (CAS) rendszerek. A középiskolás oktatási keretekbe nehezebben férnek bele, de mélységük, kidolgozottságuk és a mögöttük álló szerzői apparátus miatt ezek a legismertebb, klasszikus matematikai szoftverek. Az egyik első ilyen ismert, nyilvánosan is elterjedt rendszer az 1960-as években kifejlesztett Macsyma volt (ennek van egy ma is hozzáférhető és széles körben használt változata, a Maxima). Nagy karriert és nyilvánosságot futott be a Derive nevű szoftver, melynek fejlesztése nemrégiben teljesen lezárult.

Ma a legismertebb kereskedelmi CAS rendszer a Mathematica és a Maple. Szabad szoftver alternatíváik a Maxima/wxMaxima, a SAGE és a Mathpiper.

A komputeralgebrai rendszerek közösek abban, hogy a számításokat szimbolikusan végzik el (nem numerikusan, bár kérésre ezt is meg tudják tenni). Egy CAS rendszertől elvárjuk, hogy szimbolikusan tudjon egyenleteket, egyenletrendszereket megoldani, sorozatokat és függvényeket vizsgálni határérték és folytonosság tekintetében, s képes legyen differenciálni és integrálni is. A CAS rendszereket többnyire a saját programozási nyelvükön kell programozni, amely a Maple esetében a Pascalhoz hasonlít, a Sage-nél pedig a Python programozási nyelv kibővítése. (A Mathematica egyedi nyelvezetét a Mathpiper is átvette.)

¹<http://www.mindentudas.hu/techbazis/szamitogep/20020416tizev.html>

Dinamikus geometriai (DGS) rendszerek. Történetük a '80-as évek elejére nyúlik vissza. Az elsők egyike, a Cabri és a Geometer's Sketchpad igen nagy népszerűségnek örvend ma is.

Dinamikus geometriai rendszeren olyan szoftvert értünk, amely geometriai objektumokat jelenít meg (rendszerint 2 dimenzióban). Az objektumok ún. bázispontokon (vagy bázisobjektumokon) alapulnak, amelyeket ha megváltoztatunk, a teljes objektumrendszer (vagyis az egész ábra) velük együtt változik. Ilyen pl. egy háromszög nevezetes vonalainak rajza: ha a háromszög csúcsait (ezek a bázispontok) az egérrel arrébb húzzuk ("vonszoljuk"), a nevezetes vonalak is folyamatosan az új, megfelelő helyükre kerülnek.

Adattároló alkalmazások. Ebbe a csoportba olyan szoftverek tartoznak, amelyek nagyobb mennyiségű adatot képesek főként táblázatos formában tárolni, majd bizonyos szempontok alapján visszakeresni, ábrázolni. Tipikusan ilyenek a statisztikai szoftverek (pl. az SPSS vagy az R), de a Microsoft Excelt vagy az OpenOffice Calcot is ide sorolhatjuk.

Grafikus, illusztrációs szoftverek. Bár a fenti csoportokban szinte minden szoftver képes arra, hogy grafikusán is megjelenítse a számértékeket, a grafikus alkalmazások kifejezetten erre a célra lettek kifejlesztve. Többnyire egy-egy speciális szakterülettel kapcsolatos ábrázolások, animációk sorolhatók ide, pl. a Gnuplot függvényábrázoló program vagy a XaoS fraktálrajzoló.

Célszoftverek. A matematika nagyon tág tudomány, és sokszor olyan szoftverre lehet szükségünk, amely egy konkrét matematikai probléma vizsgálatára van tervezve. Ilyen pl. az a C nyelven írt szoftver, amely a π jegyeit számítja ki akár sok millió tizedesjegy pontossággal². Az ilyen feladatokhoz a CAS rendszerek többnyire túl lassúak, mivel a feladatot a gép rendszerszintjéhez közeli programozási nyelven kell ahhoz megfogalmazni, hogy a gép hatékonyan és gyorsan legyen képes a választ megadni.

Matematikai szövegszerkesztés és ábrakészítés. A matematikai szakszövegek nyomdai minőségű előállítására külön szoftvereket készítettek lelkes matematikusok és programozók. Ilyen szoftver pl. a \TeX , az Eukleides vagy az Asymptote: az előbbivel elegáns matematikai cikkeket, az utóbbiakkal pedig professzionális ábrákat hozhatunk létre.

Ebbe a kategóriába tartoznak azok a szoftverek is, amelyekkel internetes matematikai publikációk létrehozására van lehetőség. Gyakran azok a szoftverek, melyek papír alapú szövegszerkesztésre lettek tervezve, lehetőséget adnak az anyag webes exportálására is. Ilyen pl. LyX vagy a \TeX macs program. Más szoftvereket kifejezetten webes megjelenítésre terveztek, mint pl. a JSXGraph rendszert, mellyel dinamikus geometriai ábrákat tehetünk weboldalra, vagy a MathJaxot, amellyel a weboldalra professzionális minőségű matematikai szövegeket írhatunk. Látható, hogy a fenti kategóriák között vannak átmenetek, tehát nem egy program a fentiek közül akár több kategóriába is beletartozhat.

Oktatásszervezés, számonkérés. Az oktatói munka fontos eleme a tananyag rendszerezése, a közzététel technikája, és maga a számonkérés is. Léteznek általános célú szoftverek, amelyek ebben nem csupán a matematika tanár számára nyújtanak segítséget, hanem a teljes tananyag vagy akár egy-egy dolgozat feladatainak elkészítését tantárgytól függetlenül támogatják. Ilyenek például a tesztkészítő programok, mint pl. — az időközben karbantartó nélkül maradt — keduca vagy a Google-féle űrlapkészítő.

Kifejezetten matematikai tesztek létrehozására készült a WeBWorK és a Stack, valamint a matek.hu 1. verziójának tematikus modulok része, továbbá a MaW. Ezeken felül számos egyedi célszoftver létezik, amellyel egy vagy néhány konkrét feladattípust gyakorolhatunk, pl. ilyenek a Szegedi Tudományegyetem bizonyos kurzusaihoz létrehozott feladatsorok is.

Feladatok.

1. Mi a π szám tizedestört alakjában az 1234567. számjegy?
2. Keressünk az interneten képernyőképeket a fenti szoftverekről! Illesszük be egy szöveges dokumentumba a képernyőképeket, s küldjük el email-ben a kurzus oktatójának! (Legalább 10 szoftverhez keressünk képernyőképet!)
3. Készítsük el kedvenc matematikai képletünket \TeX formátumban! (Segítségünkre lehet pl. a \TeX on-line.)

²<http://wmi.math.u-szeged.hu/~kovzol/e4-2009.php>

5. Beszerzés

A számítógépes program beszerzése előtt fontos, hogy legyen min futtatnunk a szoftvert, vagyis szükségünk van számítógépre. Iskolai használat esetén a gépterem vagy gépterem már előre adottak, tehát inkább az a kérdés, hogy az általunk használni kíván szoftver működni fog-e az iskola hardverén. Ha magunknak szeretnénk számítógépet venni, vagy diákjaink érdeklődnek, milyen hardverre van szükségük a matematikai szoftverek futtatásához, akkor viszont célszerű körülnéznünk a számítástechnikai üzletekben, átnéznünk néhány friss reklámújságot, s kikérni informatikához értő szakember véleményét.

Ebben az írásban nyilván csak támpontokat adhatunk, hogy milyen számítógépet érdemes vásárolni. A matematikai szoftverek többsége manapság nem igényel nagy erőforrásokat. Néhány komolyabb szoftver, mint pl. a Mathematica vagy a \TeX macs természetesen igényelheti, hogy korszerű számítógépen futtassuk, és a program betöltődése is eltarthat az 5 másodperctől egészen az 1-2 perces indulásig (pl. a GeoGebra mögött futó Java virtuális gép gyors betöltődéséhez hasznos, ha erős a hardverünk). Általában azonban elmondható, hogy a szoftverek fejlesztői (különösen a szabad szoftverek) számára kihívás, hogy a lehető legnagyobb felhasználói réteget célozzák meg a programukkal, és ezért a szoftvert folyamatosan tökéletesítik, hogy minél kisebb erőforrással minél nagyobb hatékonyságot és gyorsaságot érjenek el.

5.1. Telepítés Windows alá

A legelterjedtebb operációs rendszerhez többnyire bináris formában, azaz futtatható .exe formájában érkezik a szoftver; néha a telepítő .zip fájlba van csomagolva, amelyet előbb ki kell tömöríteni. Olykor .msi fájlt kell telepíteni.

Sok programot felkészítettek arra, hogy rendszergazdai jog nélkül is fel lehessen telepíteni a számítógépre, bizonyos programoknál azonban erre nincs mód: legtöbbször írásvédelmi hibával leáll a telepítő. Sok esetben azonban ezeket a hibaizeneteket figyelmen kívül hagyhatjuk, a program működése ettől még elfogadható lesz, legfeljebb néhány funkciót nem használhatunk tökéletesen.

Fontos, hogy csak megbízható forrásból telepítsünk szoftvert. Az .exe és az .msi fájlok vírust, rosszindulatú programot is tartalmazhatnak.

Ha tehetjük, kérjük meg a rendszergazdát, hogy ő telepítse a szoftvert, így a számítógép házirendjének betartása egy kézben marad.

Fontos, hogy a szoftver működését az éles használat (tanóra) előtt magunk is próbáljuk ki. Előfordulhat, hogy a szoftver más Windows változatokhoz készült, és apró futási problémák jelentkeznek (az ékezetes betűk közül néhány hibásan jelenik meg, időzítési probléma lép fel, a könyvtárstruktúra speciális karaktereire nincs felkészülve az alkalmazás stb.).

Ha már nincs többé szükség az alkalmazásra, a telepítő szoftver által a gépre felmásolt törlő (általában UNINSTALL nevű) alkalmazást kell indítani. Rendszergazdai módban telepített szoftvereknél rendszerint a program regisztrálja magát a Windows adatbázisában is, és törlése a *Programok telepítése, eltávolítása* Windows-funkciónál is megtörténhet.

Telepítés forrásból. A szabad szoftverekhez rendszerint a forráskódot is mellékelnie kell a fejlesztőnek. Ha a program működését szeretnénk módosítani, megváltoztatni (pl. magyar nyelvi támogatás készítésekor vagy új funkció leprogramozásakor), akkor merülhet fel, hogy a szoftvert újrafordítsuk. Ehhez olyan fordító környezetre van szükség, melyet külön fel kell telepíteni. Attól függően, hogy a programot milyen nyelven írták, a fejlesztői környezet (IDE) különféle lehet: a C/C++ nyelvű programokat a Microsoft Visual Studio, a C++Builder, a DevC++ vagy az Anjuta, a Java szoftvereket az Eclipse IDE-ben lehet célszerű szerkeszteni és újrafordítani.

Bizonyos szoftvereket kifejezetten Linux/Unix alá írtak, ez esetben a Windows alá átfordítás nagyobb szakértelmet igényel. Sok esetben a Cygwin emulációs réteg segítségével képesek lehetünk Windows alatt is futtatni Unix alapú programokat.

Feladatok.

1. Telepítsük a Google Chrome-ot, a GeoGebrát és a XaoS programot bináris csomagból.

5.2. Telepítés Linux alá

A korszerű Linux rendszerekben a szoftverek telepítése internetes tárolókon keresztül történik. A Linux változat (disztribúció) kiadója többnyire az alapértelmezetten beállított tárolókban már a felhasználó rendelkezésére bocsátja azokat a szoftvereket, amelyekre várhatóan szüksége lesz. Mivel ezek a szoftverek szabad szoftverek, a beszerzés sem a kiadó, sem a végfelhasználó számára nem ütközik másolásjogi problémába, így a letöltésük és telepítésük néhány egérgattintással megtörténhet.

Többféle Linux rendszer létezik, és ezek szinte mindegyike ad grafikus felületet a szoftverek telepítéséhez és törléséhez. Pl. Ubuntu Linuxon az Alkalmazások → Ubuntu szoftverközpont, Fedorában a Rendszer → Adminisztráció → Szoftver telepítése/eltávolítása menük és menüpontok segítenek. Ezek a rendszerek parancssori telepítést is lehetővé tesznek; bizonyos esetekben célszerű lehet ezek használata is (pl. távoli eléréskor). Példaként a $\text{T}_{\text{E}}\text{X}$ macs szoftver telepítését mutatjuk be többféle Linux rendszeren is. A $\text{T}_{\text{E}}\text{X}$ macs, akárcsak a legtöbb más szoftver, ún. telepítő csomagban érhető el. Minden ilyen csomagnak van egy egyedi neve, ez az Ubuntuiban **tex-macs**, Fedorában és SuSE Linuxon **$\text{T}_{\text{E}}\text{X}$ macs** (figyeljünk a kis- és nagybetűk közötti különbségre!). A telepítés parancssorból tehát a következőképpen történik Ubuntuiban:

```
sudo apt-get install texmacs
```

Fedorában:

```
sudo yum install TeXmacs
```

SuSE Linuxban:

```
sudo yast -i TeXmacs
```

További részletek olvashatók itt.

Amennyiben a szoftver nincs az alapértelmezett tárolóban, de ismert, hogy az interneten milyen tároló URL-en keresztül tölthetjük le a kívánt alkalmazást, úgy a tárolót kell előbb felvenni a rendszer adatbázisába, és utána lehet a telepítést indítani. Példa: 1., 2.

Telepítés forrásból. Linux/Unix ill. Mac OS X rendszereken a szoftverek forrásból telepítése általában könnyebb, mint Windowson, de némi szakértelmet mégis kíván. Fontos, hogy az alaptelepítésen túl a megfelelő fejlesztői környezetet ki kell alakítani, Ubuntu (és Debian³) Linux rendszereken ez rendszerint a **build-essential** csomag feltelepítésével történik. Ha olyan szoftvert kívánunk forráscsomagból telepíteni, amely grafikus elemeket is tartalmaz (a legtöbb program ilyen), úgy rendszerint további csomagok telepítése szükséges: Ubuntu rendszeren a **libx11-dev** csomagra van még szükség.

A forráscsomagok rendszerint .tar.gz, .tgz, .tar.bz2 vagy .zip kiterjesztéssel érkeznek; ezeket a szoftvereket a fejlesztő weboldaláról, vagy egy kifejezetten szabad szoftvereknek otthont adó közösségi oldalról (ilyen például a SourceForge) lehet letölteni. A forráscsomagok kitömörítéséhez rendelkezésre áll grafikus alkalmazás, de kiterjesztéstől függően különféle parancssorokkal is dolgozhatunk. Kitömörítés előtt nyissunk meg egy terminálablakot, s lépünk be abba a könyvtárba, ahová a forráscsomagot letöltöttük: ez rendszerint a ~/Letöltések könyvtár:

```
cd ~/Letöltések
```

Ha a forráscsomag .tar.gz és .tgz kiterjesztésű, akkor pl. a xaos-3.5.tar.gz fájl kitömörítése így történik:

```
tar xzf xaos-3.5.tar.gz
```

Ugyanez .tar.bz2 kiterjesztés esetén (minta):

```
tar xjf csomagnév-verzió.tar.bz2
```

Ugyanez .zip kiterjesztés esetén (minta):

```
unzip csomagnév-verzió.zip
```

A forrásprogram kitömörítése után 3 vagy 4 további lépésre van szükség, hogy a programot telepítsük. Feltéve, hogy a terminálablak továbbra is nyitva van, lépünk be parancssorból abba a könyvtárba, ami kitömörítéskor létrejött (ezt az **dir** vagy az **ls** paranccsal tudjuk megvizsgálni), a XaoS forráscsomag esetén ez a következőképpen történik:

```
cd xaos-3.5
```

³Debian rendszerekre többnyire érvényes a telepítő csomagokkal kapcsolatos minden leírt tudnivaló

Konfigurálás. Először meg kell vizsgálnunk, hogy a program lefordításához minden segédeszköz rendelkezésre áll-e. Ilyen segédeszközök lehetnek azok az ún. külső függvénykönyvtárak, amelyek nem tartoznak szorosan a programhoz, de használatuk az alkalmazás számára elengedhetetlen. Ilyen pl. a XaoS program esetén a komplex számok közötti gyors műveletvégzéshez szükséges GSL vagy NASM rutinkönyvtár.

A programhoz mellékelni szokás egy **configure** nevű programot, amely leellenőrzi, hogy a kérdéses segédprogramok rendelkezésre állnak-e. Az ellenőrzés rendszerint így történik:

```
./configure
```

A konfigurálás során vagy konkrét hibaüzenetet kapunk, vagy csak egy figyelmeztetést, hogy bizonyos csomagokat célszerű lenne telepíteni. A XaoS esetében a legtöbb Linux rendszeren ez a 4 sor árulkodik arról, hogy hiányzanak bizonyos rutinkönyvtárak:

```
configure: WARNING: There is no driver available. XaoS will not start.
configure: WARNING: You need zlib to compile XaoS with image saving support.
configure: WARNING: You need pnglib to compile XaoS with image saving support.
configure: WARNING: You need GSL or NASM to comple XaoS with user formulas.
```

A probléma egy lehetséges feloldásához Ubuntu-n az **xserver-xorg-dev**, **zlib1g-dev**, a **libpng-dev** és a **libgsl0-dev** csomagokat kell feltelepíteni. (A **nasm** csomagot nem szükséges, mert a program anélkül is tud működni.) Ennek telepítése után a konfigurálást újra kell kezdeni.

Fontos kérdés, hogyan találhatjuk ki, milyen nevű csomagokra van szükség. A legkönnyebb, ha a Linux disztribúciónk grafikus csomagtelepítőjének keresőjét használjuk. Hasznos lehet továbbá a program telepítési útmutatóját is átolvasni, ezek többnyire angol nyelvű leírások.

Módosítás. Amennyiben új funkciót vagy javítást kívánunk a szoftveren eszközölni, úgy a forráskódba írjuk be a változtatásokat. Nyilvánvalóan ehhez programozói ismeretek és nagy gyakorlat szükséges. (Példa: a XaoS bővítése új fraktáltípussal.)

Fordítás. A fordítás rendszerint a

```
make
```

utasítás kiadásával történik.

Telepítés. Ha nem kívánjuk a Linux/Unix gépre bejelentkezni tudó minden felhasználónak elérhetővé tenni a programot, akkor nincs további dolgunk: a kész program elérhető a forráscsomag könyvtárfájában egy előre meghatározott helyen. (A XaoS esetében pl. a bin könyvtárban.)

Ha adminisztrátori jogkörrel rendelkezünk, akkor a

```
sudo make install
```

parancssor segítségével tudjuk előidézni, hogy a szoftver egyes komponensei a Linux rendszer megfelelő könyvtáraiba bekerüljenek, és ezáltal minden felhasználó teljes funkcionalitással tudja használni a programot.

Futtatás élő (live) CD/DVD-ről vagy USB-ről. A Linux disztribútorok néhány éve egy olyan technikát terjesztettek el és fejlesztettek magas színvonalra, amely a számítógép merevlemezének módosítása nélkül is lehetőséget ad egy teljes értékű Linux rendszer kipróbálására, sőt, akár a mindennapi munkára is.

A technológia lehetőséget ad arra, hogy szabad matematikai szoftvereket úgy próbáljunk ki a számítógépünkön, hogy az azon futó operációs rendszert és szoftvereket (többnyire Windowst) módosítanánk. Több ilyen kezdeményezés is van, melyek közül kifejezetten magyar célközönségnek készült az OSSMath.

Feladatok.

1. Telepítsük az RTZME alkalmazást Ubuntu vagy Xubuntu alá a <http://matek.hu/c10/> oldal leírásait követve!
2. Töltsünk le a SourceForge-ról egy matematikai szoftvert, s fordítsuk le forrásból!
3. Telepítsünk fel Linuxra az alapértelmezett tárolókból 5, általunk választott matematikai szoftvert! Melyiknek van magyar menürendszere és kezelőfelülete? Melyikhez van magyar leírás? Használhatónak tartjuk-e az oktatásban ezeket? Miért vagy miért nem?

6. A szoftverek

Ebben a részben a kurzushoz kapcsolódó konkrét szoftverek rövidebb-hosszabb leírása olvasható.

Fontos megjegyezni, hogy a felsorolt szoftvereken kívül számos (több ezer) nagyon hasznos, jó minőségű matematikai szoftver létezik, amelyek közül nagyon sok ingyenesen használható. Így tehát a következő lista egy többé-kevésbé szubjektív válogatás az elérhető szoftverek közül. Hasznos ha az Olvasó maga is keres további matematikai szoftvereket, kipróbálja azokat, és ha beválnak, beépíti oktatói-kutatói munkájába.

6.1. wxMaxima

A Maxima komputeralgebrai rendszerhez írt modern kezelőfelület, frontend. A Maxima a Macsyma rendszer továbbfejlesztése, melyet William Schelter tett közkinccsé.

Magyar nyelvű leírások.

- Maxima – Egy nyílt forráskódú CAS alternatíva (Blahota István, Linuxvilág, 2006. szeptember)
- SFD 2009 konferencia
- Rövid bevezetés a wxMaximába (munkalap WXM formátumban)
- A Maxima komputeralgebra rendszer (Horváth Anita, szakdolgozat, Szegedi Tudományegyetem, Bolyai Intézet, 2003)
- Live CD/DVD megoldások a számítógéppel segített matematikaoktatásban (Csenki Dániel, szakdolgozat, Szegedi Tudományegyetem, Bolyai Intézet, 2009)
- Student csomag készítése a Maxima komputeralgebrai rendszerhez (Kókai Zoltán, Kovács Gergely, Illés Szabolcs, szakdolgozatok, Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 2005)
- Kalkulus és Maxima (Blahota István, elektronikus jegyzet, ideiglenes változat, 2010)

Feladatok.

1. Tekintsük meg a matek.hu tudástárban lévő, készülő wxMaxima leírást! Ha tudjuk, bővítsük, tegyük jobbá az anyagot!
2. Oldjuk meg wxMaximával a $\sin x = 0$ egyenletet! Mit tapasztalunk? Miért?
3. Oldjuk meg wxMaximával az $y' = y$ differenciálegyenletet! Tipp: használjuk az **ode2** parancsot és a Google-t!
4. Számítsuk ki a $\lim_{n \rightarrow \infty} n \cdot \sin \frac{1}{n}$ határértéket a wxMaximával!
5. Próbáljuk ki, mi történik, ha az 1,4142 számnak megkeressük a lánc tört alakját. Fogalmazzunk meg sejtést (és bizonyítsuk is be)!
6. Számítsuk ki az $x^4 - 1$ és $x^6 - 1$ polinomok legnagyobb közös osztóját és legkisebb közös többszörösét!
7. Készítsük el az $e^{-(x^2+y^2)}$ kétváltozós függvény 3 dimenziós grafikonját! Forgassuk el úgy az ábrát, hogy jól látszódnak a felület lényeges vonásai!

Megoldások.

6.2. Sage

A Sage komputeralgebrai rendszer egy alapjaitól újraírt szoftver, melynek fő gondolata a modulokból való építkezés és a programozási környezet. A matek.hu tudástárban külön szócikk olvasható róla.

Feladatok.

1. Látogassunk el a <http://wiki.sagemath.org/interact/> weblapra, és az **Explanatory example** programlistáját próbáljuk bemásolni egy üres Sage munkalapba. Azáltal, hogy egyes sorokat próbaképpen elhagyunk majd visszaírunk, továbbá módosítunk, tegyünk kísérletet arra, hogy a program 12 sorának működését egyesével elmagyarázzuk.
2. Írjuk egy 10×10 -es mátrixba sorfolytonosan az 1, 2, 3, ..., 100 számokat. Mennyi lesz a mátrix determinánsa? Adjunk a jelenségre matematikai magyarázatot. Tipp: Merítsünk ötleteket a <http://sage.math.u-szeged.hu/home/pub/16/> és <http://sage.math.u-szeged.hu/home/pub/42/> munkalapokból.

6.3. matek.hu kalkulátor

A matek.hu kalkulátor egy on-line számológép. Célja a középiskolai és egyetemi oktatásban a diákok a szimbolikus számításokat könnyen elérhetővé tenni, mivel használatához csupán egy internet böngészőre van szükség. A matek.hu tudástárban külön szócikk olvasható róla.

Feladatok.

1. Oldjuk meg a $\sin x = 0$ egyenletet! Mit tapasztalunk? Vajon miért?
2. Számítsuk ki a $\lim_{n \rightarrow \infty} n \cdot \sin \frac{1}{n}$ határértéket!
3. Számítsuk ki az $x^4 - 1$ és $x^6 - 1$ polinomok legnagyobb közös osztóját és legkisebb közös többszörösét!
4. Készítsük el az $e^{-(x^2+y^2)}$ kétváltozós függvény 3 dimenziós grafikonját! Forgassuk el úgy az ábrát, hogy jól látszódnak a felület lényeges vonásai!

6.4. GeoGebra

A GeoGebra egy dinamikus geometriai rendszer, melynek fő célközönsége a középiskolás diákság. A szoftver emellett sikerrel használható az általános iskolában és a felsőoktatásban is. A GeoGebrának rendkívül nagy irrodalma van, kiterjedt közösségi háló tartozik hozzá, és a matek.hu tudástárban is olvasható róla külön szócikk.

Feladatok.

1. Készítsünk webes munkalapot, melyen az $a \cdot \sin(b \cdot x + c) + d$ függvényt lehet ábrázolni. Figyeljünk arra, hogy teljes értékű oktatási anyagot készítsünk, vagyis legyen az exportált HTML anyagnak címe, bevezetője, s legyen a végén valamilyen feladat is az olvasó számára!
2. A háromszög körülírható körén futtassunk végig egy pontot, s tekintsük a közben keletkező Simson-egyenesek nyomát (a GeoGebrában az egyenesre jobb gombbal kattintva a Nyomvonal opciót kell bejelölni). Milyen geometriai formát ad az egyenessereg? Exportáljuk ki a kapott képet PNG formátumba, és küldjük el a kurzus oktatójának email-ben!
3. Kísérletezzünk a táblázat ablakkal! (Példákat a GeoGebra szócikkben találunk.) Ha sikerül hasznos példát létrehozni, küldjük el a kész .ggf fájlt a kurzus oktatójának email-ben.
4. Látogassunk el a http://www.maa.org/joma/Volume7/Hohenwarter2/guidelines_dyn_worksheets.html weblapra, ahol angol nyelven találunk instrukciókat, hogyan is célszerű exportált HTML anyagot készíteni. Ragadjunk ki az angol szövegből 3 javaslatot, amelyeket fordítsunk le magyarra!
5. Hasonlítsuk össze a jelenlegi stabil (e sorok írásakor 3.2-es) és a hamarosan bevezetésre kerülő béta (e sorok írásakor 4.0-s) verziót. (A béta verzió elindítható a <http://www.geogebra.org/webstart/4.0/geogebra.html> címen.) Látunk-e különbséget a menürendszerben, az ikonokban? Ha igen, soroljunk fel néhány eltérést!

6.5. Mathematica

A Mathematica a Wolfram Research által fejlesztett professzionális komputeralgebrai rendszer, főként matematikusok számára. A középiskolai oktatásban csupán részben javasolt alkalmazni, mert angol kezelőfelülete, fejlett (olykor bonyolult) szintaxisa, nagy erőforrásigénye és (a konkurens termékekhez képest) magas ára hátrányt jelenthetnek a tanórán. Ha tehetjük (pl. az oktatási intézménynek van belőle legalább 1 példánya), mégis célszerű megismerkednünk a szoftverrel, hiszen tanítványaink később a felsőoktatásban vagy a munkahelyen elképzelhető, hogy használni fogják. Ezen kívül a 6-os verziószám felett könnyen készíthetünk vele oktatási segédanyagokat, melyeket az ingyenesen használható Mathematica Playerrel bárki megtekinthet. A Mathematica Player továbbá lehetőséget ad arra, hogy a <http://demonstrations.wolfram.com> weboldalon található több ezer (jelen sorok írásakor mintegy 7000) oktatási anyagból letöltsünk számunkra hasznosakat, és azokat ingyenesen felhasználjuk.

Feladatok.

1. A Manipulate parancs segítségével készítsünk interaktív munkalapot, melyen az $a \cdot \sin(b \cdot x + c) + d$ függvényt lehet ábrázolni. (Tipp: a **?Manipulate**-tel segítségével pontos szintaxist kaphatunk, és a megjelenő rövid leírás végén a »-re kattintva az *Examples* részben konkrét példákat is találunk.) Ellenőrizzük, hogy az általunk lementett .nb fájl valóban megnyitható az ingyenes Mathematica Playerrel (ez utóbbit töltsük le előbb a Wolfram weblapjáról)!
2. Keressünk a több ezer segédanyag között olyat, amivel az Archimédész-féle π -közelítést szemléltethetjük!

6.6. Euklides

László István és Simon Péter elsősorban középiskolás közönségnek írt dinamikus geometriai szoftvere, melynek jelenlegi tulajdonosa a Mozaik Kiadó. Zárt forráskódú. Kis erőforrásigényű, nagy tudású Windows-os alkalmazás. Ingyenesen letölthető tesztverzióval rendelkezik. Fejlesztése lényegében befejeződött, a matematika oktatásában betöltött szerepét emiatt lényegében a GeoGebra veszi át fokozatosan.

Feladatok.

1. Illusztráljuk Euklidesben Bonaparte Napóleon tételét! (Keressünk az interneten GeoGebrában készült illusztrációt is ugyanehhez a matematikai tételhez!)

6.7. Euler 3D

Petró Tamás szoftvere, mellyel 3 dimenziós geometriai alakzatokat készíthetünk és vizsgálhatunk. Jelenlegi tulajdonosa a Mozaik Kiadó. Zárt forráskódú. Viszonylag kis erőforrásigényű, sok funkciót tartalmazó Windows-os alkalmazás. Ingyenesen letölthető tesztverzióval rendelkezik. Fejlesztése lényegében befejeződött.

Feladatok.

1. Készítsünk egy szöveges dokumentumot, amelyben lépésről-lépésre elmagyarázzuk, hogyan kell szabályos oktaédert készíteni Euler 3D-vel! Először (elméletben) tervezzük meg, milyen koordinátákat fogunk beírni, majd a fontosabb lépéseket képernyőképekkel is illusztráljuk!

6.8. JSXGraph

A Bayreuth-i Egyetem Javascriptben írt dinamikus geometriai rendszere, mellyel elsősorban weblapokba ágyazhatunk geometriai szerkesztéseket, grafikonokat.

Feladatok.

1. Készítsük el Jessiescript segítségével a háromszög Euler-egyenesét! Kezdésként induljunk el a <http://wmi.math.u-szeged.hu/~kovzol/jessiescript> oldalon található HTML fájljából (a forráskódot a legtöbb böngészőben a CTRL-U billentyűkombinációval tudjuk megtekinteni), valamint a <http://jsxgraph.uni-bayreuth.de/wp/documentation/> oldalon a Jessiescript referenciakártya összefoglalójából!

6.9. GNU XaoS

Komplex számokon értelmezett, főként szökési időn alapuló fraktálokat bemutató, nagyrészt magyar nyelven is elérhető grafikus program, elsődleges szerzője Jan Hubička és Thomas Marsh. A matek.hu tudástárban is olvasható róla leírás.

6.10. LyX

Matematikai szövegszerkesztő, mely a szokásos WYSIWYG ("what you see is what you get": "azt kapod, amit látsz") megjelenítés helyett a WYSIWYM ("what you see is what you mean": "azt látod, amire gondolsz") megjelenítési filozófia szerint működik.

A szoftvert 1995-ben alkotta meg Matthias Ettrich, a későbbi KDE tervezője.

A LyX számos nyelven elérhető, köztük magyarul is. A beépített dokumentáció alapján a program használata alaposan megismerhető, s így a program lehetőségeinek kiaknázásával a LyX-szel akár könyveket, szakdolgozatokat is készíthetünk.

A LyX a \LaTeX -re épül. (Nagyrészt) magyar nyelvű weblapja a <http://www.lyx.hu> oldalon érhető el.

6.11. T_EXmacs

Joris van der Hoeven matematikai szövegszerkesztője. Működési elve sokban hasonlít a \LaTeX -hez, de a mögöttes tipográfiai motort a szerző teljes egészében újraírta. A T_EXmacs ezen kívül egy általános célú tudományos dokumentumszerkesztő: támogatja ábrák készítését és tudományos szoftverekkel (pl. a Maximával, a Sage-dzsel és a GeoGebrával) való munkamenetek importját és exportját is.

A T_EXmacs hivatalos weblapja a <http://www.texmacs.org>.

6.12. Moodle

Általános célú e-Learning rendszer, mellyel teljes kurzusok tananyaga kezelhető. Mind a kurzus jegyzete, feladatok, és azok automatikus kijavítása-pontozása lehetséges. A rendszer lehetőséget ad névtelen és autentikált bejelentkezésre is. Számos nyelven elérhető, így magyarul is. Telepítése rendszergazdai ismereteket kíván, de egy létező Moodle rendszerhez oktatóként csatlakozni és saját írásokat megosztani nem bonyolult.

Hivatkozások

[1] Kovács Zoltán: Számítógép alkalmazása a matematika tanításában – 2008/2009 (I. félév)

[2] Kovács Zoltán: Számítógép alkalmazása a matematika tanításában – 2006/2007 (II. félév)