

Trigonometrikus görbék szintonikus genezise teknőcgeometriával

Farkas Károly

farkas.karoly@nik.bmf.hu

BMF NIK

Absztrakt. A tananyag megfelelően kicsiny lépésekre bontása a metodika egyik alaptétele. A teknőcgeometria a szintonikus hatásával hatékonyan segíti a megértést, a rögzítést. Programozást tanulni elsősorban azért hasznos, mert ez a gondolkodási képességeket intenzíven fejleszti. Bonyolultabb matematikai görbék (függvények) megismerését segíti, ha azokat egyszerűbb görbék szuperpozíciójával állítjuk elő. A trigonometrikus függvények megértése a matematikai tanulmányok során rendszerint egy gondolkodási szint meghaladását igényli. Ezt segíthetjük az itt bemutatásra kerülő módszertani példánkkal. Prezentációnk a Logo programnyelv, a logo-pedagógia középiskolai használatára, felhasználására mutat példát. A trigonometrikus görbék generálását egyrészt korábbi publikációink, továbbá originális újszerű részletek egységes mikrovilágával, rendszerével mutatjuk be.

1. Pedagógiai megfontolások

A programozás tanítással, a Logo alkalmazásával a jelen tanulmányban is kifejtett egyik fő céllunkat, már az első a módszertani konferencián is megfogalmaztuk. Ez változatlan. „A matfóbia kifejezést, most Papert [1] után, mint valamilyen nehéznek tűnő ismeretrendszerrel való iszonyódat alkalmazzuk. Bármely tantárgynál - a kimondottan örömtantárgyként indult informatikánál is - kialakulhat ez az ellenézés. A tanulmányok kezdetén ez igen károsan befolyásolja a további fejlődést, felsőoktatásban pedig már gyakran gyógyíthatatlan. A megelőzés módja az egyes tantárgyak megszerettetése. Ezt, a megkedveltetést fontosabbnak tartjuk, mint magát a tanítást.” Tehát annál, hogy mit tanítunk, véleményünk szerint fontosabb a hogyan tanítunk!

Hipotézisünk szerint mindegyik görbe előállítható mozgások szuperpozíciójával. Kövessük Pólya György tanácsát: „Oszd a problémát részekre!” Szókratész, Skinner, és sokan megmutatták, hogy a tanítás lényeges mozzanata a megismerendő, a megtanulandó ismeret megfelelően kicsiny részekre bontása. Bonyolultabb (matematikai) görbék megalkotására, megismerésére is tehát célszerűnek látszik, azok egyszerűbb elemekből állítjuk össze.

A matematikai tanulmányok során sokan a trigonometrikus függvényeknél (görbéknel) veszítik el a fonalat. A nehézség abból is adódhat, hogy csak ritkán látunk a valóságban trigonometrikus görbét, így a tanulónak nincs gondolati fogódzója, nincsen képből. Eddig a matematika órán sokaknak segített, ha elegendő számú trigonometrikus görbét szerkesztettek. Ez a munka azonban időigényes, és csak kevesek számára szórakoztató. Az elegendő szám könnyebben érhető el, ha számítógéppel ábrázolunk. A figyelem fenntartását, a megértést az is segíti ilyenkor, hogy a látvány dinamikusabb.

Ezen előnyöket sokféle matematikai programmal kiaknázhajjuk. Amennyiben a Logo technó-geometriáját alkalmazzuk, akkor ehhez járul (vagy talán még jelentősebb is) az a kognitív haszon, hogy a trigonometrikus görbék előállítása testszintonikus, a görbék megrajzolását (tényleges, vagy képzelt) mozgásélményekhez kapcsolhatjuk.

Az egyenesekkel, körökkel még csak megbirkózzunk, az alternáló mozgás is emészthető. Építjük fel tehát, ezen elemekből a trigonometrikus mikrovilágot!

A technógeometria szintonikus (átélhető, mozgásémlékeinkhez köthető) értéke talán, a Logo programnyelv kellő elterjedtségének köszönhetően ismert. Technóöket tanítsunk mozogni, és az összetevő mozgásokat a bemutató technóöünk összegezze, szuperponálja!

A továbbiakban az első módszertani konferencián ismertetett példáinkat tovább fejlesztve a bemutatásra kerülő görbék előállítása ismét nem csak esztétikai élményt jelenthet, de a matematikával-számítógéppel való gondolkodásfejlesztés talán figyelemre méltó példája, a számítógép célszerű felhasználásának egyik lehetséges módja a középiskolai matematika – fizika oktatásában, de a programozástanulásnak is egy lehetséges útja.

2. Technógeometriai példák

A bemutatásra kerülő példák folyamatos munka eredményei. Az első módszertani konferencia utáni bővítések egy részét Csink Lászlóval publikáltuk [2]. Ezeket és további példákat most MicroWorlds Ex Logo változattal készítettük. A program Demo változata szabadon használható. [3]

2.1. Alpalgoritmus

Görbéink előállításához három főszereplőre van szükség, két összetevőmozgás eredőjeként állítjuk elő a kívánt harmadikat. Tehát teremtsünk három technóöt: Ádámot, Évát és Káint. (Ádám legyen kék, Éva piros, Káin fekete, mindenki szorítsa tollát a rajzlaphoz, hogy haladásakor vonalat húzzon maga után, mutassák magukat). A MicroWorlds Ex programnyelven ezt imperatív programozási technikával (parancsokkal) így valósíthatjuk meg:

```
newturtle "Ádám setc 116 st pd
newturtle "Éva setc 16 st pd
newturtle "Káin st pd
```

Ádámnak a módsöusa: kattintásra az „a” jelű utasítás (vagy utasításlista) végrehajtása,

Éva feladata „b” végrehajtása,

Káin pedig úgy az „a”-t, mint a „b”-t hajtsa végre!

```
Set "Ádám "onclick [forever [run :a]]
Set "Éva "onclick [forever [run :b]]
Set "Káin "onclick [forever [run :a run :b]]
```

Az „a” és a „b” utasítások legyenek infinitizimálisan kicsiny elmozdulások. Amint majd látni fogjuk a kellően kicsiny mozdulatok a szinusznál egy teknőclépés, egy foknyi fordulás, a tangens görbénél század teknőc-lépések, századfok fordulások.

Ha ezek után kiadjuk a „mindenkire kattintottam!” (`everyone [clickon]`) parancsot, attól függően, hogy mi az „a” és mi a „b” utasítás tartalma, Káin különféle görbéket jár be, Ádám és Éva mozgásainak szuperpozícióját, az „a” és a „b” mozgásoknak eredőjét hajtja végre.

Ha a két összetevő mozgás egyenes vonalú, egyenletes mozgás, az eredő szintén az.

```
make "a [setx xcor + 1]
make "b [sety ycor + 1]
```

Az egyenes vonalú mozgások szuperpozíciójára írható eljárás ennek megfelelően:

```
to egyenes
  make "a [setx xcor + 1]
  make "b [sety ycor + 1]
  everyone [clickon]
end
```

Leállítás után a teknőcöket alaphelyzetbe hozhatjuk egy eljárással:

```
to origoba
  everyone [setpos [0 0]]
end
```

Amennyiben kört akarunk Káinnal rajzoltatni, Ádám haladjon előre, Éva forduljon kicsit:

```
to kör
  make "a [fd 1]
  make "b [rt 1]
```

```

everyone [clickon]
end

```

Korábban publikáltuk, hogyan generálhatunk szereplőinkkel, spirálokat, cikloisokat, szinuszgörbét, Lissajous görbéket. Most ezek közül felidézzük a szinusz geneziséét.

Rávezető gyakorlat a teknőcöknek, az ordináta értékek egy intervallumban tartása:

```

to fűrészfog
  make "a [setx xcor + 0,5]
  make "b [fd 1 if or ycor > 80 ycor < -80 [rt 180]]
  ask [Éva Káin] [seth 0]
  ask [Ádám Éva Káin] [clickon]
end

```

2.2. Szinusz

A szinusz előállítás: Ádám egyenesen halad, erre a mozgásra Éva merőleges irányú harmonikus rezgőmozgását illesztjük. A harmonikus rezgőmozgás egy körmozgás vetületeként kapható.

A feladatot kellően kis lépésekre bontva, Évának legyen segítségül Lucifer körmozgása, amely vezérlőgörbének Éva mindenkor y koordinátáját veszi át.

Teremtjük meg Lucifert, akinek metódusa körözés.

```

newturtle "Lucifer setc 45 st pd

```

Éva pedig kérje el mindenkor Lucifertől az y koordináta értékét, s ezt vegye fel.

```

make "b [sety ask "Lucifer [ycor]]

```

Ádám egyenesen halad x irányban.

Ezek felhasználásával a szinusz eljárás így is megírható:

```

to szinusz
  ;a színpadon már ott a három főszereplő: Ádám, Év, Káin,
  valamint Lucifer

```

```
Lucifer, st pd setx -57 seth 0
set "Lucifer "onclick [forever [fd 1 rt 1]]
make "a [setx xcor + 1]
make "b [sety ask "Lucifer [ycor]]
everyone [clickon]
end
```

Az Ádám-Éva-Káin rendszerben a szinusz genezisének kulcsa tehát az, hogy Éva átveszi a köröző Lucifer y koordináta értékét.

Az amplitúdót Lucifer pályakörének átmérője, a periódus nagyságát Ádám haladási sebessége határozza meg. (A szinuszos mozgásról köztudott, hogy egyik összetevője lehet a forgómozgás y vetülete. A forgattyús mechanizmus elemzésekor az egyes alkatrészek mozgásait a szinusz függvény segítségével írtuk le. A szinuszgörbe előbbieken ismertetett genezisést én más úton, a cikloisok nyújtásával találtam meg. Újszerű meghatározásom szerint így a szinuszgörbe a nyújtott ciklois limesze.)

2.3. Koszinusz görbe

A koszinusz a szinusz x tengelyen történő eltolásával kapható. Az eltolás $\pi/2$. Mikrovilágunkban ezt úgy hozhatjuk létre, hogy Lucifert máshonnan, más nézési iránnyal indítom. A szinusz-generálásakor nézési iránya induláskor észak volt, most kelet felé állítom be.

```
to koszinusz2
Lucifer, st pd sety 57 seth 90
set "Lucifer "onclick [forever [fd 1 rt 1]]
make "a [setx xcor + 1]
make "b [sety ask "Lucifer [ycor]]
everyone [clickon]
end
```

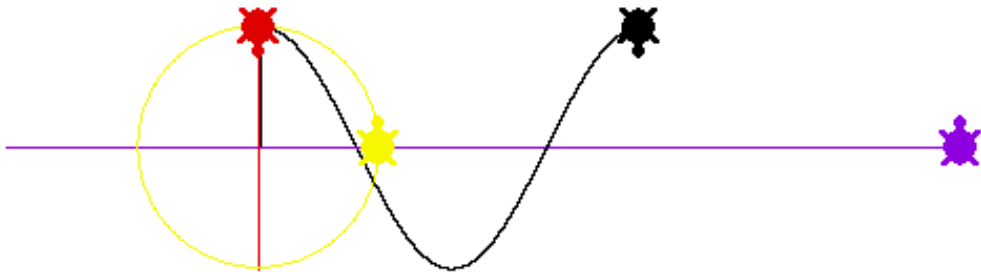
A koszinusz görbe másik, szintonikusabb genezise a koszinusz, mint az x tengelyen található vetület szerinti:

```
to koszinusz
Ádám, setx -120
Lucifer, st pd setx 57 seth 0
set "Lucifer "onclick [forever [fd 1 lt 1]]
```

```

make "a [setx xcor + 1]
make "b [sety ask "Lucifer [xcor]]
everyone [clickon]
end

```



1. ábra. Koszinusz, mint két mozgás eredője

Az origót most is Ádám és Éva nyomvonalának metszéspontja jelöli. Lucifer „teknőcegységkört” rajzol. (Amelynek kerülete 360 lépés, így átmérője $360/\pi$, azaz 114 teknőclépés.) Most Lucifer köröző mozgását pályája jobb szélső pontjában kezdi (induló helyzete setx 57), és rt (jobb) fordulatok helyett lt (bal) fordulatokat tesz, mivel a teknőcgeometriában a forgásszög nulla értéke a térképészeti keleti irány, a pozitív forgásirány pedig a matematikában megszokottal ellentétes. Éva pedig Lucifer x koordinátáit veszi át.

2.4. Tangens szuperpozícióval

Jelen tanulmány originális része a tangens görbe genezise. Ehhez szükségünk lesz, a köt eljárásra, amit több görbe generálására (pl. epicikloisok) korábban már használtunk.

A köt parancs kiadása után a felszólított teknőc, mindenkor a parancs paraméterével megnevezett teknőc felé fordul, s tartja a tőle levő távolságát.

```

to köt :a
;kötöm az aktuális teknőcöt az „a” teknőchöz
towards :a
make "d distance :a
make "alfa ask :a [heading]

```

```

make "x ask :a [xcor]
make "y ask :a [ycor]
setpos list :x + :d * sin :alfa :y + :d * cos :alfa
end

```

Hogyan származtassuk a tangens görbét? A tangens koordinátageometriai definíciója segített a megoldásban. A tangens a forgószög mozgó szárával az egységsugarú kör egy-null pontjában húzott érintőből kimetszett szelet hossza. Nosza, ezt utánozzák a teknőcök!

Ádám ismét töretlenül haladjon egyenes útján. A forgószög szára pedig legyen egy diszkó lámpa fénysugara, egy lézersugár, amely pásztáz. Kell tehát egy forgó (a tangens fő periódusának megrajzolásához egy lengő) teknőc. Ez legyen megint Lucifer, aki irányítja Évát. Lucifer mintegy kivont lézerekarddal (a szög mozgó szára) forgolódjon (kardot emelgessen), s ez a kardlámpa sugár világítson rá a Lucifertől jobbra egységnyi távolságban elhelyezett képernyőre. A forgás tengelye tehát most párhuzamos az ernyővel. A forgástengely irányban szemlélve a lézersugár keltette pont fel-fel szalad egy egyenesen. Ennek a pontnak megfelelően mozogjon egy újabb teknőc, mondjuk „Kígyó”. (Lucifer a „fényhordozó” produktuma). A pont függőleges távolsága az x tengelytől, a tangens érték, ezt kell tehát Évának mindenkor felvenni.

A fénypontra ültetet Kígyó, két tulajdonsággal kell, hogy rendelkezzen, mindenkor Lucifert nézi, ahhoz kötve mozog, és mindenkor rajta marad az érintőn, tehát egy egyenesen is mozog. (A gépészetben ez a kulisszás hajtómű, a forgattyús mechanizmusnak bizonyos inverze).

A köt eljárás segítségével, tudja például Ádám maga körül forgatni Évát. Ha „a” értéke folyamatosan jobbra fordulás, „b” értéke a forgó teknőchöz kötés, az eredő körözés:

```

to forgatás
make "a [rt 1]
make "b [köt „Ádám]
everyone [clickon]
end

```

Visszatérve a tangenshez, az létrehozható, ha a Kígyó Luciferhez van kötve, és követi azt a fénypontot, amely az érintőn szaladgál. Teremtsük meg a Kígyót!

```

newturtle "Kígyó setc 65 st pd

```

A tangens_főperiodusok eljárás a tangens fő periódusait rajzolgatja egymás után.

```

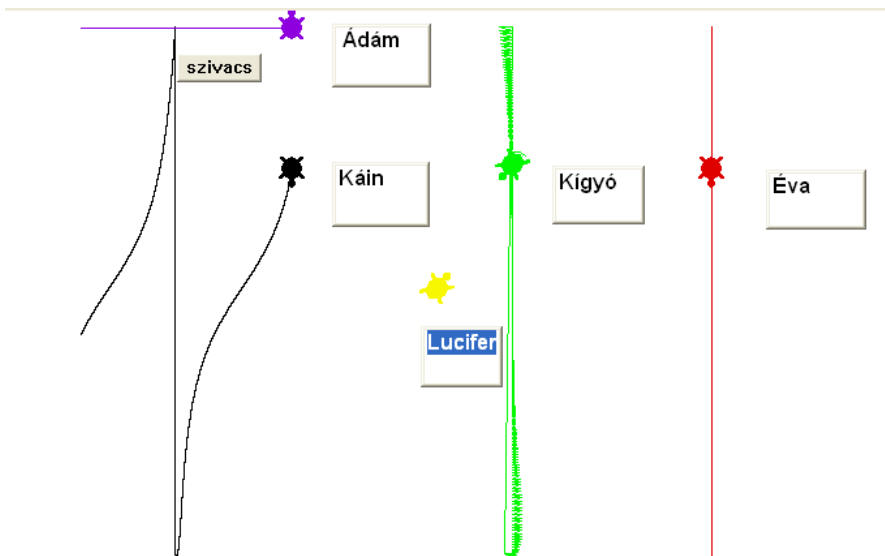
to tangens_főperiodusok
Ádám, sety 200 setx -300

```

```

Éva, setx 210
Káin, st setx -300 pd
Lucifer, setc 55 seth 160 st
Kígyó, st
set "Ádám" onclick [forever [setx xcor + 0.025]]
set "Éva" onclick [forever [sety ask "Kígyó [ycor]]]
set "Káin" onclick [forever [setx ask "Ádám [xcor] sety ask
"Éva [ycor]]]
set "Lucifer" onclick [forever [ifelse heading > 13 [lt
0,05] [seth 166]]]
set "Kígyó" onclick [forever [köt "Lucifer setx 57]]
everyone [clickon]
end

```



2. ábra. Tangens főperiódusok generálása

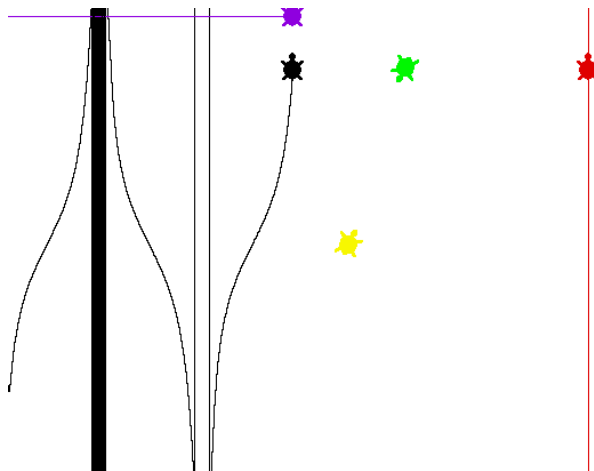
2.5. Kotangens

Amennyiben Lucifer nem csak emelgeti magát, hanem körbe forog (a forever utáni szögletes zárójelbe pl. lt 0,02 kerül), Káin váltogatva egy tangens, egy kotangens periódust rajzol.

```
to tg_ctg
```



```
Ádám, sety 200 setx -300
Éva, setx 210
Káin, st setx -300 pd
Lucifer, setc 45 seth 160 st
Kígyó, st pu
set "Ádám "onclick [forever [setx xcor + 0.025]]
set "Éva "onclick [forever [sety ask "Kígyó [ycor]]]
set "Káin "onclick [forever [setx ask "Ádám [xcor] sety ask
"Éva [ycor]]]
set "Lucifer "onclick [forever [lt ,05]]
set "Kígyó "onclick [forever [köt "Lucifer setx 57]]
everyone [clickon]
end
```



3. ábra. Tg-ctg mozgás

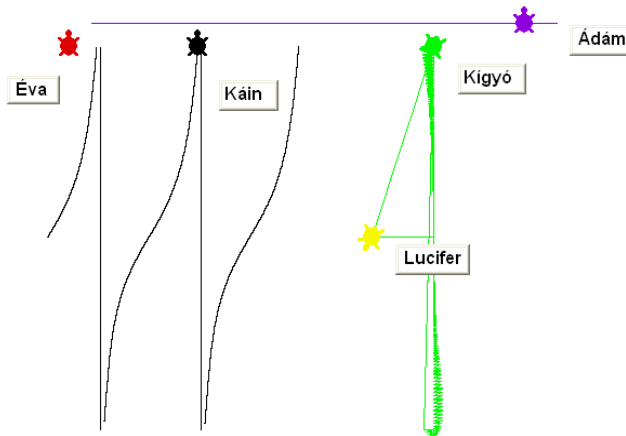
2.5. Tangens függvénygörbe

A teljes tangens függvény generálásához most már csak Lucifernek északról dél felé tartó forgása során a lézertámpával maga mögé kell világítania, így tudja a sugárral mindenkor a jobb oldali érintőt metszeni. Ehhez írtuk a köt2 eljárást, amelyben Kígyó úgy van Luciferhez kötve, hogy mindig mögötte jár.

A képernyő szélén a teknőcök átugranak a képernyő túloldalára, a képernyőt mintegy önmagában záródó sikként kezelik, hogy ne legyenek ezek a felesleges átugrás nyomvonalak az aszimptoták közelében, ezért ott fel kell mentenünk a Kígyót munkája alól, és ezzel kihagyjuk a

rajzlapról lefutó teknőcök zűrzavarát. (Az aszimptoták változó elhelyezkedését eredményezné, hogy a teknőc drámajátékban a szereplők csak kvázi parallel dolgoznak. Az egyprocesszoros számítógép a teknőcökkel csak egymás után, sorban, külön-külön tud foglalkozni. Ennek kompenzálására írtunk be az eljárásba a modell lényegéhez nem tartozó utasításokat.)

A tangens rajzoló eljárásunk (és az eddig közölt többi) az MWLogo honlapon megtalálható [4].



4. ábra. Tangens görbe genezise

3. Összegzés

Három teknőc főszereplésével, és további mellékszereplőkkel mutattuk be az alapvető trigonometrikus függvénygörbék generálását. Ezzel folytattuk a különféle görbék teknőc-geometriával való genezisével foglalkozók [5], [6], [7] munkásságát.

Az intrinsic görbék ilyen módon való megalkotása a konstruktív, kísérletező tanulásnak is jó példája. Ennek során a matematika és az informatika tantárgy között erős koncentrációt valósítottunk meg. A módszert alkalmazzuk Budapesten a Hetényi Géza Humán Szakközép iskola informatika óráin.

Az első táblázat összefoglalja néhány görbe teknőcgeometriai geneziséét:

Az alakzat/mozgás, :a és :b eredője	Ádám metódusa :a	Éva metódusa :b	Megjegyzés
Rezultáns egyenes	setx xcor + 1	sety ycor + 1	
Kör	fd 1	rt 1	
Ciklois	setx xcor + 1	fd 1 rt 1	
Forgás-körözés	rt 1	köt "Ádám	
Spirál	rt 1	köt "Ádám fd 1	
Fűrészfog	setx xcor + 1	fd 1 if or ycor>80 ycor <-80[rt 180]	
Színusz	setx xcor + 1	sety ask "Lucifer [ycor]	Lucifer jobbra köröz
Koszínusz	setx xcor + 1	sety ask "Lucifer [xcor]	Lucifer balra köröz
Tangens	setx xcor + 1	sety ask "Kígyó [ycor]	Kígyó: forgó Lucifer- sugár „dőféspontja”
Lissajous	setx ask "Lili [xcor]	sety ask "Lucifer [ycor]	Lili és Lucifer köröz- nek

1. táblázat. Teknőcgörbék genezise

Irodalom

1. Papert, S.: *Mindstorms*, Basic Books, New York (1981).
2. L. Csink, K. Farkas: *Turtle's Curves*. In: R. T. Mittermeyer M. M. Sysło (ed): *Proceedings Informatics Education Contributing Across the Curriculum*, Torun, Poland, (2008) 76-86.
3. <http://www.microworlds.com>
4. <http://mwlogo.fw.hu>
5. Armon, U.: *An Algorithm that Translates Intrinsic Equations of Curves into Intrinsic Procedures of These Curves*. In: M. Turcsányi-Szabó (ed:) *Proceedings Sixth European Logo Conference*, Budapest (1997) 378-389.
6. Alexandrov, K. Soprunov, S.: *One Approach to Mechanic Simulations in Logo*. In: M. Turcsányi-Szabó (ed:) *Proceedings Sixth European Logo Conference*, Budapest (1997) 114-121.
7. Foltinowicz I.: *Cicloids and limacons in the turtle graphics*. In: I. Kalaš (ed:) *Proceedings of the 11th European Logo Conference*, EuroLogo 2007, Bratislava, Slovakia, (2007) 19-24.