
Pannon Egyetem

Műszaki Informatikai Kar

Matematika Tanszék

Gazdaságinformatikus BSc

SZAKDOLGOZAT

Linearizáló koordinátatranszformációk számítógépes támogatása

Horváth Márton

Témavezető: Dr. Szalkai István

2015



PANNON EGYETEM

MŰSZAKI INFORMATIKAI KAR

Gazdaságinformatikus szak

SZAKDOLGOZATTÉMA-KIÍRÁS

Horváth Márton

Gazdaságinformatikus BSc szakos hallgató részére

Linearizáló koordinátatranszformációk számítógépes támogatása

Témavezető: Dr. Szalkai István

A feladat leírása:

Mind statisztikai adatok feldolgozásakor, mind előzetes számítások alkalmával gyakran találkozunk a Descartes-féle koordinátarendszertől eltérő ábrázolási rendszerekkel. *A dolgozat célja* egyrészt a különféle ábrázolási módszerek, azok elméleti alapjai és gyakorlati felhasználásuk rövid bemutatása, másrészt olyan teljes számítógépes program kifejlesztése, amelynek segítségével egyetemi diákok és gyakorlati szakemberek alaposabban megismerhetik és felhasználhatják ezeket a módszereket, sőt kevésbé ismert transzformációkat is alkalmazhatnak.

A program a következő funkciókkal rendelkezen:

- tetszőleges egyváltozós függvény képletének beírása,
- a függvény értelmezési tartományának figyelembe vétele (hibakezelés),
- a koordinátatengelyek egymástól független transzformálása, legalább 10-10 féle módon, a transzformációk paramétereinek állításának lehetősége,
- a függvény ábrázolása a transzformált rendszerben,
- egyazon függvény ábráinak összehasonlítása többféle rendszerben,
- a munkamenet, adatok elmentésének és beolvasásának lehetősége.

A hallgató a legújabb, gyors fejlesztést lehetővé tevő, ingyenes szoftvereket és technológiákat alkalmazza.

A Szakdolgozó feladata:

1. A létező programok, alkalmazások rövid ismertetése, eltérések a jelen programtól.
2. A szükséges szoftverek, technológiák kiválasztása.
3. Tervezés, dokumentálás (rendszerterv, felhasználói kézikönyv, telepítési útmutató) és implementálás.
4. Tesztelés.

dr. Szalkai István
egyetemi adjunktus
témavezető

Dr. Hartung Ferenc
egyetemi docens
szakvezető

Veszprém, 2015. március 16.

NYILATKOZAT

Alulírott Horváth Márton hallgató, kijelentem, hogy a dolgozatot a Pannon Egyetem Matematika Tanszékén készítettem a gazdaságinformatikus végzettség megszerzése érdekében.

Kijelentem, hogy a dolgozatban lévő érdemi rész saját munkám eredménye, az érdemi részen kívül csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy a dolgozatban foglalt eredményeket a Pannon Egyetem, valamint a feladatot kiíró szervezeti egység saját céljaira szabadon felhasználhatja.

Veszprém, 2015. május 11.

Horváth Márton

Alulírott Dr. Szalkai István témavezető kijelentem, hogy a dolgozatot Horváth Márton a Pannon Egyetem Matematika Tanszékén készítette gazdaságinformatikus végzettség megszerzése érdekében.

Kijelentem, hogy a dolgozat védelemre bocsátását engedélyezem.

Veszprém, 2015. május 11.

Dr. Szalkai István

KÖSZÖNETNYILVÁNÍTÁS

Legnagyobb köszönettel témavezetőmnek, Dr. Szalkai Istvánnak tartozom, aki hatalmas segítséget nyújtott dolgozatom elkészítésében, rendkívül hasznos tanácsokkal látva el hétről hétre.

Köszönöm családomnak a tanulmányaim során nyújtott szerető támogatást és türelmet.

Végül, de nem utolsó sorban köszönet illet mindenkit, aki szakdolgozatom elkészítésének ideje alatt személyes tanácsaival segítette munkámat.

TARTALMI ÖSSZEFOGLALÓ

Szakdolgozatom témája koordináták és függvények ábrázolása nem ekvidisztáns koordináta-rendszerekben illetve a különböző rendszerek közti transzformáció szoftveres támogatása.

Dolgozatomban röviden áttekintem a koordináta-rendszerek alapfogalmait, különböző megjelenési formáit, ezek közül kiemelten foglalkozom az ekvidisztánstól eltérő rendszerekkel. Bemutatom a dolgozathoz készített szoftver fejlesztésének fázisait, használatát illetve kitérek a gyakorlati alkalmazásának fontosságára.

A szoftver használata során lehetőség nyílik tetszőleges ponthalmazok és függvények transzformálására (Descartes-féle) ekvidisztáns és nem ekvidisztáns koordinátarendszerek között. A szoftver tervezése teljes egészében sajátkezű, önálló munkám eredménye, az implementálásához felhasználtam néhány előre elkészített komponenst.

Kulcsszavak: koordináta, koordináta-rendszer, ekvidisztáns, függvény, függvénytranszformálás, függvénygrafikon, origó, síknegyed, értelmezési tartomány, értékkészlet, lineáris-, logaritmus-, reciprok-, hatvány-, gyök-, exponens-, normál eloszlás-, lognormál eloszlás-, tangens-, arcus-tangens-, tangens-hiperbolicus-, area-tangens-hiperbolicus skála.

ABSTRACT

Topic of my thesis is representation of coordinates and functions in non-equidistant coordinate systems and giving software-support for transformation of function graphs between different coordinate systems.

In my thesis I give a brief review of basic concepts and different manifestations of coordinate systems paying special attention to non-equidistant forms. I introduce the development stages of the software made for this thesis and highlight the importance of practical application.

Using the software an arbitrary coordinate or function can be transformed between (Cartesian) equidistant and non-equidistant coordinate systems. Software design is the result of my own, however during the implementation I have added some ready-to-use component in.

Keywords: coordinate, coordinate system, equidistant, function, function graph, transformation of function graphs, domain and codomain of a function, linear-, logarithm-, reciprocal-, power-, root-, exponent-, normal distribution-, lognormal distribution-, tangent-, arcus-tangent-, tangent-hyperbolicus-, area-tangent-hyperbolicus scale.

TARTALOMJEGYZÉK

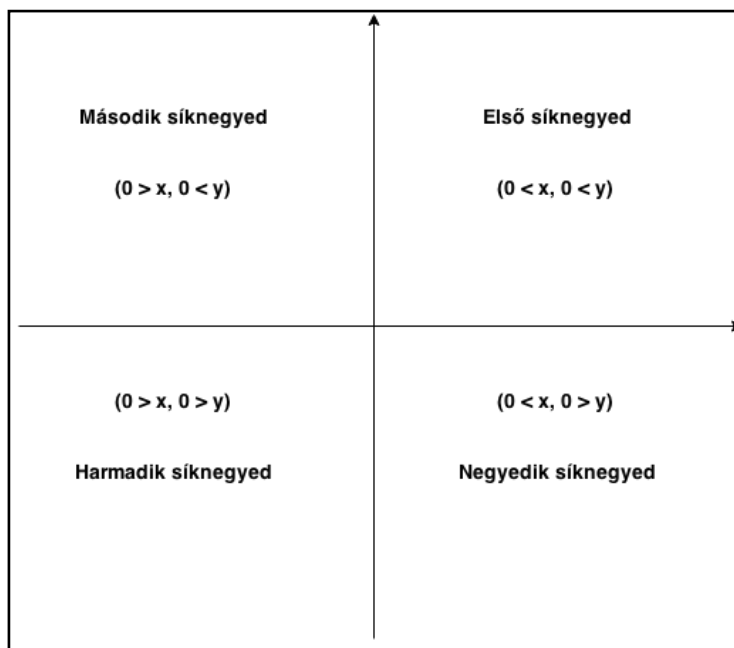
1. Bevezetés	9
1.1. Ekvidisztánstól eltérő ábrázolások	12
1.2. Történeti áttekintés	16
1.3. Modern változat	18
1.4. Céljai	20
2. Általános matematikai definíció	22
3. Speciális esetek	24
3.1. Szemilogaritmikus transzformáció	24
3.2. Logaritmikus transzformáció	26
3.3. Reciprok transzformáció	27
3.4. Normális eloszlás transzformáció	28
4. Az elkészített szoftver	32
4.1. Bevezetés	32
4.2. A problémakör azonosítása	32
4.2.1. A probléma leírása	32
4.2.2. Üzleti lehetőségek	32
4.3. A termék és környezete	33
4.3.1. Szereplők azonosítása	33
4.3.2. Magas szintű célok meghatározása	33
4.3.3. Felhasználói szintű célok	33
4.3.4. Egyéb követelmények és kényszerek	34
4.4. A megvalósítás folyamata	34
4.4.1. Technológiák	34
4.4.2. Fejlesztői környezet	35
4.4.3. Fejlesztés iterációkban	35
4.5. Használati esetek	36
4.5.1. Használati eset-modell	36
4.5.2. Használati eset: Függvény ábrázolása	37
4.5.3. Használati eset: Ponthalmaz ábrázolása	39
4.5.4. Használati eset: Dupla függvényábrázolás	40
4.5.5. Képernyőképek	43

4.6. Kiegészítő specifikáció	49
4.6.1. A felhasználás szabályai	49
4.6.2. Használhatóság	49
4.6.3. Megbízhatóság	49
4.6.4. Teljesítmény	50
4.6.5. Támogathatóság	50
4.6.6. Implementációs kényszerek	50
4.6.7. Komponensek	50
4.6.8. Interfészek	50
4.7. Felhasználói útmutató	51
4.7.1. Függvényábrázolás mód (function-plot)	51
4.7.2. Ponthalmaz ábrázolás mód (point-plot)	52
4.7.3. Dupla-ábrázolás mód (double-plot)	53
4.7.4. Beviteli szabályok	54
4.7.5. Tengelyek korlátai	55
4.8. Továbbfejlesztési lehetőségek	56
5. Irodalomjegyzék	57
6. Mellékletek	59
6.1. Eljárások	59
6.2. A mellékelt CD könyvtárszerkezete	60

1. Bevezetés

A sík pontjainak helyét megadhatjuk bizonyos báziselemekhez viszonyított számokkal. Ezek a báziselemek és a távolság méréséhez használt lépték meghatározzák a viszonyítási rendszert, amit **koordináta-rendszernek**, a **bázishoz** viszonyított számokat pedig **koordinátáknak** nevezünk.

A leggyakrabban alkalmazott koordináta-rendszer a **Descartes-féle** rendszer, melyet két irányított, egymásra merőleges számegyenes és az egység hossz definiál. A két számegyenes metszéspontja a nulla értéknél van, melyet **origónak** nevezünk. Ebben az esetben egy síkbeli P pont X és Y tengelyekhez tartozó koordinátáit Descartes-féle koordinátáknak nevezzük. A két koordináta rendezett számpárt alkot, ahol az első szám az X tengelyen, a második szám az Y tengelyen mért elmozdulást jelöli.



1.1. ábra: síknegyedek

A koordináta-rendszer a síkot négy részre osztja. A négy síknegyedben az *1.1 ábrán* látható módon változik a koordináták előjele.

A koordináta-rendszerben **függvénynek** nevezzük azt az egyértelmű hozzárendelést, ahol az **értelmezési tartomány** (az alaphalmaz azon részhalmaza,

melyen melyen a hozzárendelést értelmezzük) minden eleméhez, pontosan egy elemet rendelünk az **értékkészlet** (azon elemek halmaza melyeket az értelmezési tartomány elemeihez rendelünk) elemei közül. Függvény megadásához tehát szükséges az értelmezési tartomány, az értékkészlet és a hozzárendelés szabálya.

Az értelmezési tartomány és az értékkészlet összetartozó elempárjai egy-egy pontot határoznak meg a koordináta-rendszerben. Ezen pontok halmazát a függvény **grafikonjának** nevezzük.

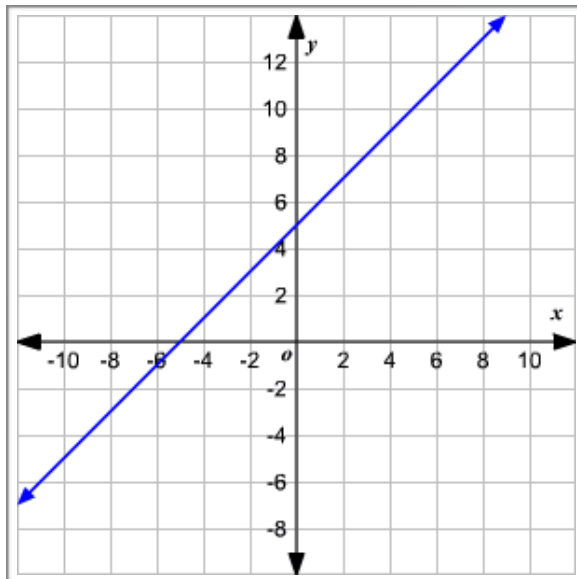
Tekintsünk néhány egyszerű függvénytípust és ezek grafikonját valamint a hozzárendelés szabályát:

- lineáris függvény

Grafikonja mindig egyenes, általános képlete

$$y = ax + b \tag{1}$$

ahol a és b tetszőleges racionális számok.



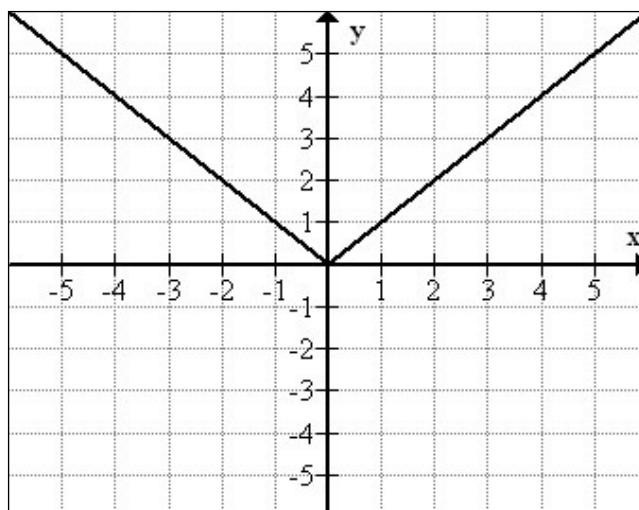
1.2. ábra: lineáris függvény

- abszolútérték függvény

Grafikonja törött vonal, képlete

$$y = |x|$$

(2)



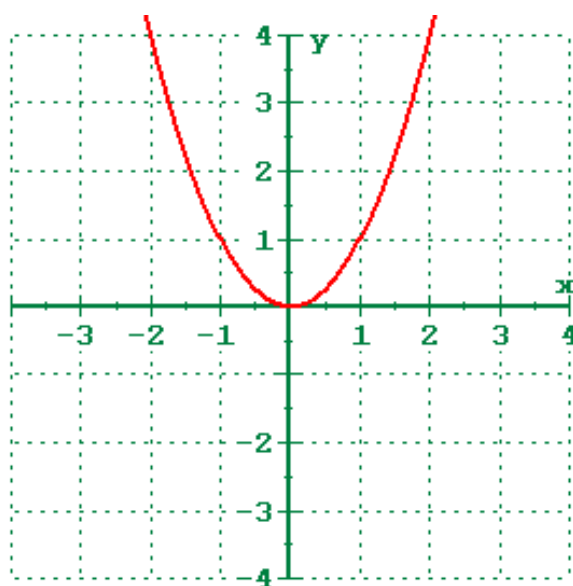
1.3. ábra: abszolútérték függvény

- másodfokú függvény

Grafikonja parabola, képlete

$$y = x^2$$

(3)

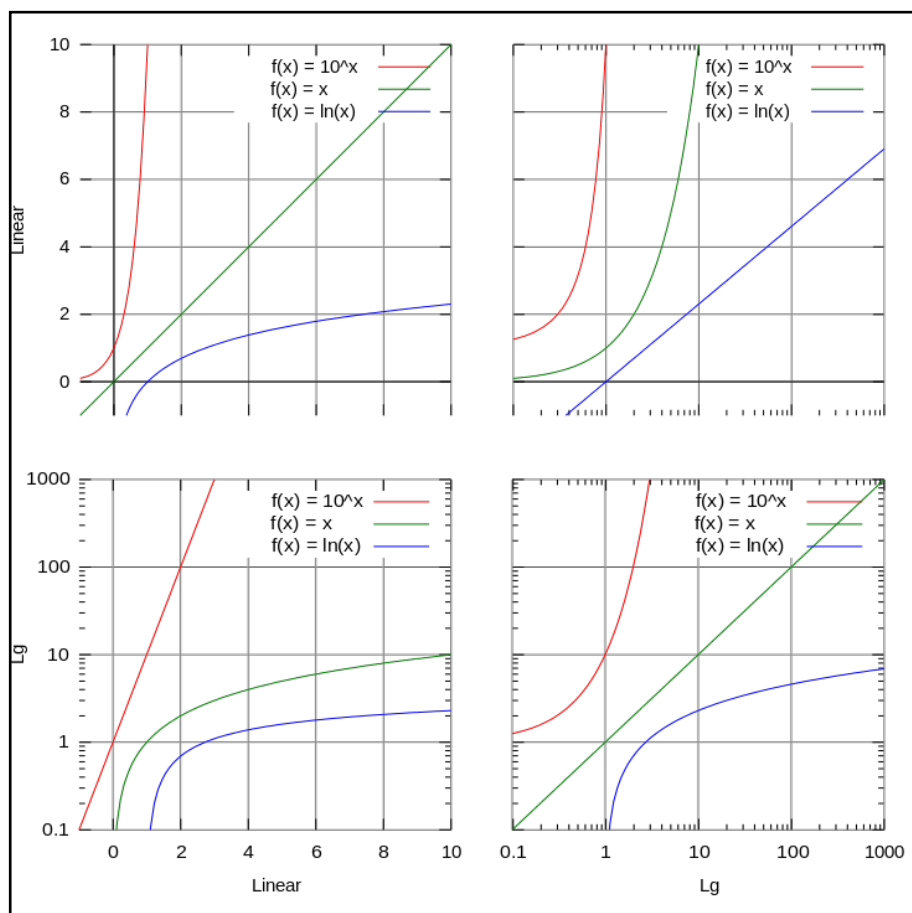


1.4. ábra: másodfokú függvény

A fenti függvénygrafikonok ábrázolásához használt koordináta-rendszerek számegyenesen ún. **ekvidisztáns** (egymástól egyenlő távolságra lévő) lépésközt alkalmazunk. Ez azt jelenti, hogy a számegyenesen egységnyi távolság mindig ugyanolyan mértékű elmozdulást jelent (pl.: a 0 és 1 közti távolság ugyanúgy 1 mértékű elmozdulást jelent, mint az 1 és 2 közti távolság). Más esetekben viszont szükséges az ekvidisztánstól eltérő ábrázolásmódok használata...

1.1. Ekvidisztánstól eltérő ábrázolások

Az ekvidisztánstól eltérő ábrázolásmódok alkalmazásaival gyakrabban találkozhatunk, mint gondolnánk. A legtöbb esetben akkor, ha egyszerre nagyon nagy és nagyon kicsi értékeket szeretnénk egy rendszerben hatékonyan ábrázolni,



1.1.1. ábra: lineáris és logaritmikus skálák [1]

vagy - ahogy az *1.1.1 ábrán* (Lásd még: *1.2.3, 1.2.4 ábra*) - különböző **koordináta-transzformációk** elvégzéséhez, **összefüggések** elemzéséhez használatosak. A fenti ábrán a **lineáris** és **logaritmikus skálák** valamint a szemi-logaritmikus és logaritmikus transzformációk, hatásai láthatók.

A dolgozathoz készített program többféle skálával is foglalkozik, most szemléltetésképpen tekintsük a logaritmus skálát, a logaritmikus és szemi-logaritmikus koordinátarendszereket és ezek alkalmazását a gyakorlatban.

Ha a lineáris skálán alkalmazzuk a logaritmus függvény inverz függvényét, logaritmus skálát kapunk. Hasonlóan, más beosztású (pl.: reciprok) skálák esetén is az inverzfüggvény adja meg a skála léptékét. Az *1.1.1 ábrán* lineáris és logaritmus koordinátatengelyek láthatók többféle kombinációban, illetve néhány egyszerű függvény képe az adott koordinátarendszerben.

Látható, hogy a logaritmus függvény képe abban az esetben válik lineárisrá, ha az X tengely logaritmus beosztású az Y tengely pedig lineáris, az inverzfüggvény képe pedig abban az esetben, ha az Y tengely logaritmus beosztású, az X tengely pedig lineáris. Ez a szabály más beosztású skálák esetén is érvényes, tehát a tengelyek transzformálásával a transzformáló függvény és annak inverzfüggvénye linearizálható, a függvény képe "kiegyenesíthető".

Az ilyen és ehhez hasonló tengelyeket a matematika, fizika, biológia, geológia és egyéb tudományágak számos területén alkalmazzák, tekintsünk ezek közül néhányat.

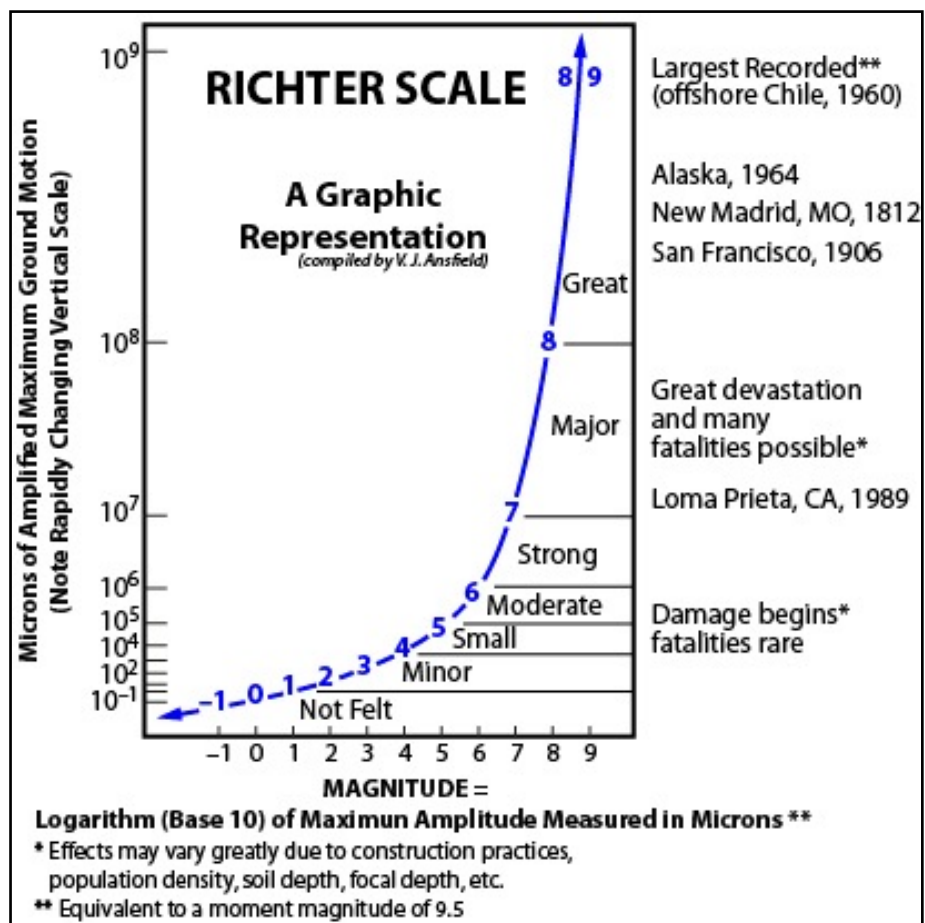
A **Richter-magnitúdó skála** a földrengések erősségének mérésére szolgáló legelterjedtebb eszköz. A mérőszám a földrengés epicentrumában felszabaduló energia logaritmusával arányos.

A **decibel skála** széles körben használatos a hangerő mérésére, de gyakran fordul elő akkusztikai és elektronikai méréseknél egyaránt. 1 B (bel) két érték tízszeres arányát jelöli. Képlettel:

$$A = 10 \cdot \log(P_2/P_1) \quad (4)$$

ahol A a logaritmus érték, P_2 a teljesítmény, amit P_1 -hez viszonyítunk. [8] A szélsőségek ábrázolásának lehetőségén kívül további nagy előnye, hogy skálája nagyon hasonló az emberi szem fényérzékeléséhez és az emberi fül rezgésérzékeléséhez, hiszen ezek is logaritmikus érzékenységek.

A **pH-érték skála** különböző anyagok kémhatásának mérésére szolgál. Ebben az esetben egy 0-tól 14-ig terjedő skálát használunk, mely középső értéke (7 = semleges), a tiszta víz kémhatásához igazodott. A pH egyenlő az oxóniumion-koncentráció tizes alapú logaritmusának ellentettjével. [9]



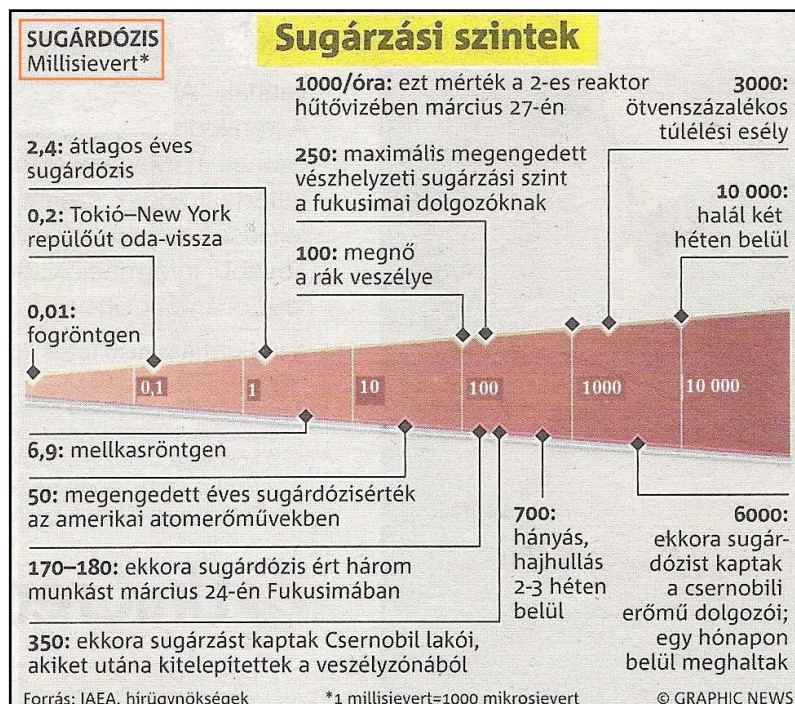
1.1.2. ábra: Richter-skála [2]

Az alábbi táblázat a decibel skála logaritmikus természetét néhány példával szemlélteti. [10]

DECIBEL	HANG	PÉLDA
10	szinte hallható	lehulló falevél

DECIBEL	HANG	PÉLDA
30	nagyon csendes	suttogás
50	korlátozott hang	működő hűtőszekrény
60	hallható	emberi hang
70	idegesítő	porszívó
90	rendkívül kellemetlen	közeli teherautó, ordítás
110	hangosabb	közeli helikopter, nagy dob
130		mennydörgés
150	maradandó károsodást okozhat	tűzijáték
170		lövés pisztollyal
194		Saturn rakéta

A fentieken kívül a logaritmus és más nem ekvidisztáns skálák további alkalmazásaival is gyakran találkozhatunk, elég csak földtörténettel foglalkozó dokumentumfilmek idővonalára a sugárdózis mérésére vagy a fényképezőgépünk rekeszértékeire gondolnunk.



1.1.3. ábra: Sugárzási szintek

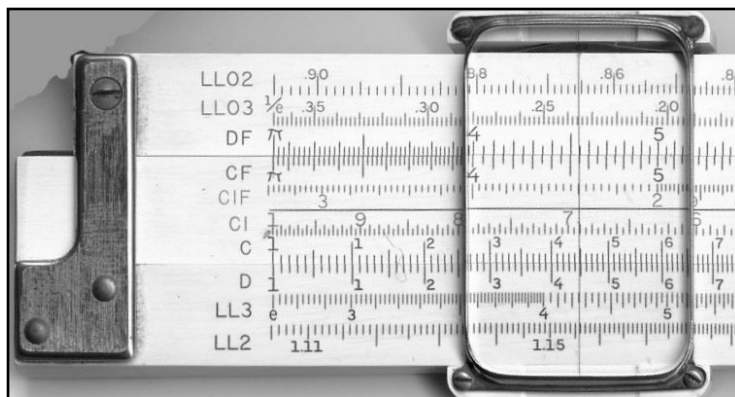
1.2. Történeti áttekintés

René Descartes (1596-1650) gyakran alkalmazott algebrai módszereket geometriai problémák megoldásához. Az 1637-ben írt *Discourse of Method* című művének *La Géométrie* függeléke nagy mértékben előre mozdította az analitikus geometria fejlődését. Descartes csak egyetlen tengellyel dolgozott és nem vette figyelembe a negatív számokat, műveiben még nem olvashatunk koordinátarendszerekről.

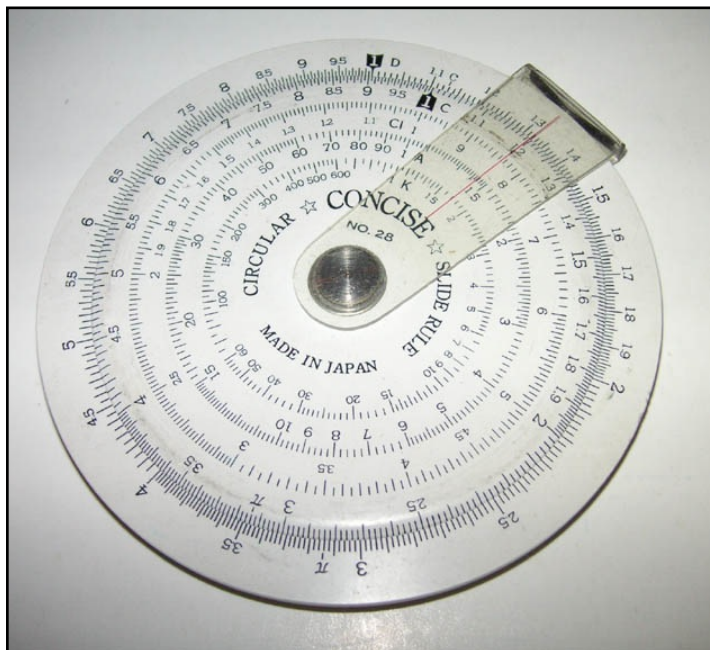
Descartes-al majdnem egyidőben *Pierre de Fermat* (1601-1665), francia matematikus is geometriai megoldások után kutatott. 1636-ban írt *Isagoge* című tanulmánya már analitikus geometriai tárgyúnak mondható.

Egy jól alkalmazható geometriai koordinátarendszer kifejlesztéséhez azonban el kellett ismernünk a nullát mint számot és be kellett vezetnünk a negatív számokat. 1748-ban *Leonhard Euler* (1707-1783) könyvében találkozhatunk már két tengelyű koordinátarendszerrel, mely a negatív számokat is használta. [3, 4]

A nem ekvidisztáns skálák használata a elsőként logarléc feltalálásával (1620-1630) terjedt el széles körben. Miután *John Napier* publikálta logaritmusról szóló művét, az oxfordi egyetemen *Edmund Gunter* feltalált egy logaritmikus skálából és mérőeszközből álló szerkezetet, mellyel szorozni és osztani lehetett. Később *William Oughtred* készített egy körlogarléctet és 1632-ben egyesítette találmányát Gunter eszközével, ezzel létrejött a mai értelemben vett logarléc.

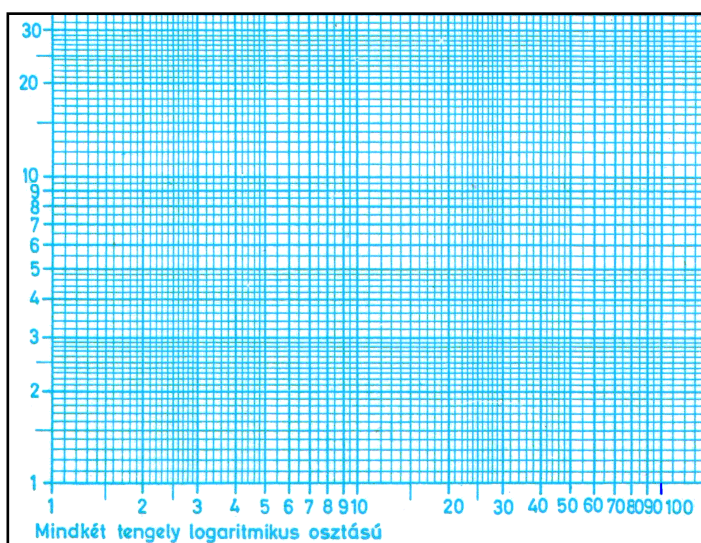


1.2.1. ábra: logarléc

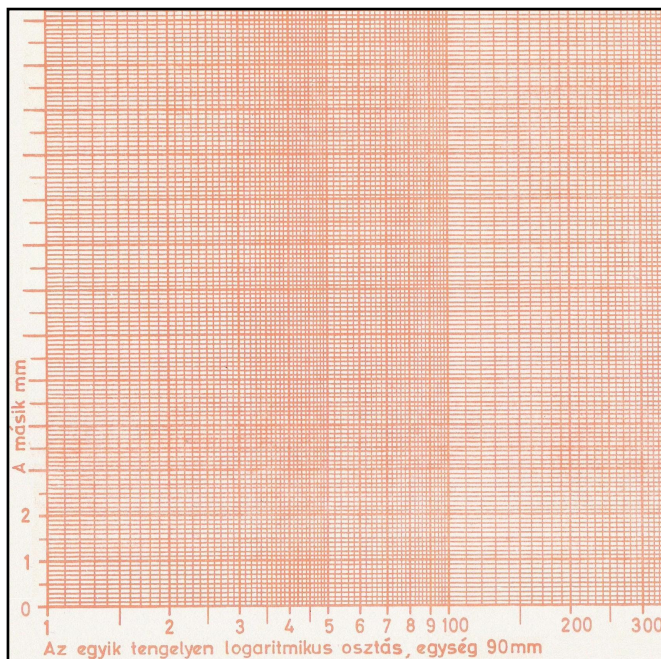


1.2.1. ábra: körlogarléc

Az alapvető logaritmikus skálákon kívül a gyakorlatban használatos logarlécek más (nem ekvidisztáns) skálákat is alkalmaztak, pl.: szinusz, koszinusz, tangens skálák. Később *Warner* bevezette a négyzet- és köbskálákat (1722), *Everard* az inverz skálát (1755), majd *Peter Roger* feltalálta a log-log skálát (1815). Az 1970-es, 1980-as években a logarléceket végül felváltották a tudományos kalkulátorok, számítógépek. [5]



1.2.3. ábra: log-log papír



1.2.4. ábra: semi-log papír

Egészen a közelmúltig mérnökök, diákok, tudósok gyakori kelléke volt a *logaritmusos papír*. Két elterjedt formája a *semi-log papír* és a *log-log papír*, melyeket - a később tárgyalt koordináta-transzformációkon túl (Lásd még: 1.1.1 ábra) - különböző intenzíven és széles skálán változó mennyiségek megjelenítésére használtak. Napjainkra őket is szinte teljesen felváltották a nagy mennyiségben elérhető számítógépes függvényrajzoló szoftverek.

1.3.Modern változat

Ahogy azt az előző alfejezetben tárgyaltuk, napjainkra a modern kor elvárásainak eleget téve a matematikusok, fizikusok, mérnökök eszköztárából eltűntek a mechanikus eszközök és nyomtatott anyagok, helyüket számítógépes szoftverek egész generációja vette át.

Az idő múlásával egyre több függvényrajzoló program jelent meg a kereskedelemben forgalmazott szoftverek piacán és a szabad felhasználású alkalmazások közt egyaránt.

Az ingyenesen elérhető - többnyire online - szoftvermegoldások előnye, hogy gyorsan, bérmentve és a világ bármely pontján használhatjuk őket valamint

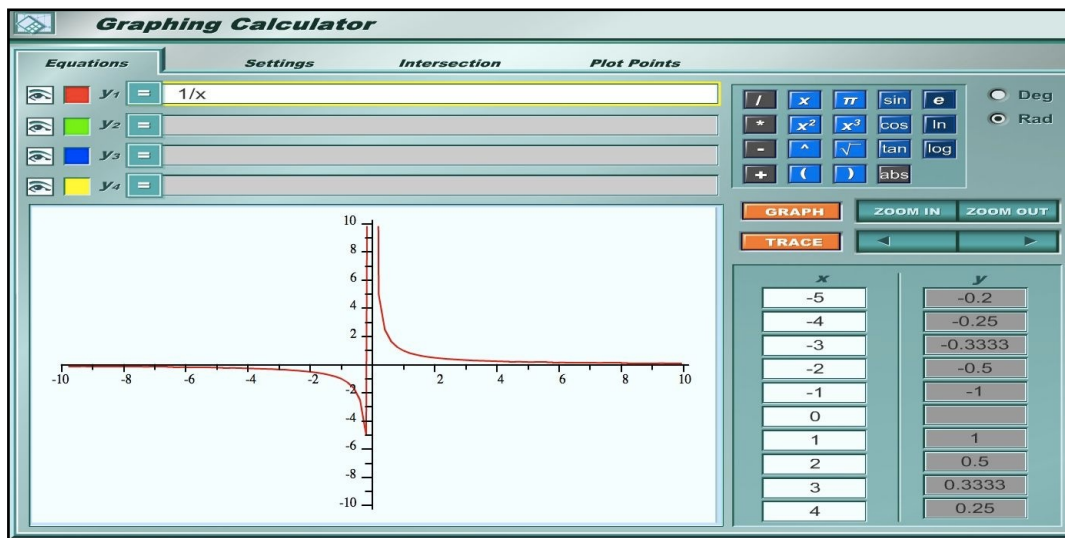
egyszerűségüknek köszönhetően nem szükséges a telepítésükkel, konfigurálásukkal időt töltenünk. Ezen programok nagy hátránya azonban, hogy a legtöbbjük nem ellenőrzött forrásból származik, ezért adott esetben hibás eredményt kaphatunk. Ilyen esetek jellemzően az informatikai eszközökkel nehezen kezelhető helyzetekben fordulnak elő, például a függvénykép szakadási pontjainak megállapításakor, vagy a végtelenhez tartó koordinátatengelyek ábrázolásakor.

Az 1.3.1 ábrán egy online függvényábrázoló megoldás látható, mely az

$$y = 1/x \quad (5)$$

függvény 0 szakadási pontjában függőleges vonalat húz. [6]

További hátrányként említhető, hogy bizonyos esetekben csak nagyon nehezen vagy egyáltalán nem található olyan alkalmazás, mely pontosan a felmerült problémára kínál megoldást.



1.3.1. ábra: hibás ábrázolás

Szakedolgozatom készítésekor a legfőbb célom olyan ingyenes online függvény- és pontábrázoló szoftver készítése volt mely a nem ekvidisztáns koordinátatengelyek használatával és linearizáló koordinátatranszformációkkal kapcsolatos speciális problémákra nyújt megoldást. Versenytársaival szemben

kimagasló előnye, hogy 12 féle skálát használhatunk mindkét koordinátatengelyen, egyszerre 4 függvény képét ábrázolhatjuk ugyanabban a rendszerben, egyszerre 2 koordinátarendszert is megjeleníthetünk, egyaránt ábrázolhatunk függvényeket és ponthalmazokat illetve a munkamenet elmenthető és megnyitható.

Hasonló ingyenes terméket hosszas keresés után sem találtam. Az alábbi listában - a teljesség igénye nélkül - felsorolom néhány gyakran használt kapcsolódó témájú online program internetcímét. Ezekben az esetekben ha van is lehetőségünk a tengelyek transzformálására, csak lineáris és logaritmus tengelyek közül választhatunk. Ez a legtöbb kereskedelemben kapható alkalmazás esetében is így van. (pl.: Microsoft Excel, LibreOffice, Apple Grapher stb.)

- <https://www.desmos.com/calculator>
- <http://fooplot.com/>
- <http://www.meta-calculator.com/>
- http://my.hrw.com/math06_07/nsmedia/tools/Graph_Calculator/graphCalc.html
- <https://graphsketch.com>
- <http://rechneronline.de/function-graphs/>
- <http://graph-plotter.cours-de-math.eu>
- <http://itools.subhashbose.com/grapher/>
- <http://www.bcalc.net>
- <https://mathway.com/graph>
- <http://www.shodor.org/interactivate/activities/GraphSketcher/>
- <http://www.intmath.com/functions-and-graphs/graphs-using-svg.php>

1.4.Céljai

A hagyományos, Descartes-féle koordinátarendszerben ábrázolt függvények esetén egyértelműen látszik a függvények különbözősége, de egy grafikon vagy felrajzolt ponthalmaz esetében nem azonosítható pontosan a függvény-összefüggés típusa és paraméterei.

Egyedül a lineáris összefüggés

$$y = mx + b \quad (6)$$

igazolható azzal a módszerrel, hogy egy **egyenes** illesztünk a pontokra (*régen: vonalzót*), mely megmutatja, hogy az összefüggés valóban lineáris-e illetve meghatározhatók az m és b paraméterek.

Az elkészített szoftver alapötlete, hogy a koordinátarendszer tengelyskáláinak módosításával az ábrázolt függvény vagy ponthalmaz alakja is változik, esetleg az összefüggés lineárisává válik, ezáltal nem lineáris esetben is használható az egyenes-illesztés módszere. Lásd: *1.1.1 ábra*, ahol a logaritmikus X tengellyel ellátott koordinátarendszerben a logaritmus függvény képe, a logaritmikus Y tengellyel ellátott koordinátarendszerben az exponenciális - amely a logaritmus inverze - függvény képe válik lineárisává vagy ha például az alábbi függvény képét egyenesítjük ki,

$$y = a \cdot \exp(bx) \quad (7)$$

akkor a kapott egyenes

$$y' = \lg(y) = bx + \lg(a) \quad (8)$$

grafikonjából a paraméterek meghatározhatók vonalzóval és szögmérővel.

A dolgozat témája ezen eljárás bemutatása, általánosítása, számítógépes támogatása és általánosan használható alkalmazás kifejlesztése.

2. Általános matematikai definíció

Ahogy a bevezetőben is tárgyaltuk, ekvidisztáns koordinátarendszerben ábrázolt pont koordinátái a pont báziselemekhez vagyis a koordinátatengelyekhez viszonyított távolságát adják meg. Az alábbi koordináta-transzformációkban azonban a sík egy P pontjának nem a fenti távolságok lesznek közvetlenül a koordinátái.

Jelöljük x_t -vel a P pont Y tengelytől való (X tengely mentén vett) **távolságát**, y_t -vel pedig a P pont X tengelytől való (Y tengely mentén vett) **távolságát**. Ezekben a távolságokban az x_f és y_f **feliratokat** írjuk és ezeket hívjuk a P pont **koordinátáinak**.

A módosítások célja, hogy bizonyos függvénykapcsolatok grafikonjai, mint *geometriai ponthalmazok* **lineárisak (egyenesek)** legyenek. Ez a gyakorlatban nem csak azt jelenti, hogy egy képlettel adott függvénykapcsolatot kell ábrázolnunk, hanem például egy mérésorozat adatait ábrázolva a megfelelő koordinátarendszerben a ponthalmaz rajzából kell eldöntenünk, hogy a mérési adatokra valóban a felírt összefüggés teljesül-e, és ha igen, melyek az összefüggés paraméterei.

Ezután rögzítsünk két tetszőleges, szigorúan monoton, lehetőleg folytonos függvényt:

$$\varphi, \psi : \mathbb{R} \rightarrow \mathbb{R} \tag{1}$$

Általában ezek értékészlete nem egyezik meg a valós számok halmazával, ezért az ilyen esetekben a sík nem minden pontjának lesznek koordinátái vagy nem minden koordináta jelenik meg geometriai pontként a papíron. ^{1 2}

¹ Szigorúan monoton függvények mindig invertálhatóak és inverzük is ugyanolyan típusú szigorúan monoton (nő vagy csökken)

² Folytonos függvények inverzei csak abban az esetben lesznek biztosan folytonosak ha véges (korlátos) és zárt intervallumon invertáljuk őket.

A sík P pontjainak (φ, ψ) -koordinátáinak az (x_f, y_f) számpárt definiáljuk, ahol

$$x_f = \varphi^{-1}(x_t) \text{ és } y_f = \psi^{-1}(y_t). \quad (2)$$

A fenti egyenlőség így is írható,

$$\varphi(x_f) = x_t \text{ és } \psi(y_f) = y_t \quad (3)$$

Definíció: Az X tengelyen az x_f feliratot $x_t = \varphi(x_f)$ távolságra írjuk fel és az Y tengelyen az y_f feliratot $y_t = \psi(y_f)$ távolságra írjuk fel és az x_f, y_f valós számokat hívjuk a $P = (x_t, y_t)$ síkbeli (geometriai) pont **$[\varphi, \psi]$ koordinátarendszerre vonatkozó koordinátáinak**.

Ponthalmaz vagy függvények ábrázolásakor csak az x_f, y_f feliratokat olvashatjuk a tengelyek skáláin vagyis az $y_f = h(x_f)$ függvény grafikonját kell felrajzolnunk, mely akkor lesz geometriailag egyenes, ha a pontokhoz tartozó x_f, y_f értékek közti kapcsolat lineáris, tehát:

$$y_t = mx_t + \beta. \quad (4)$$

3. Speciális esetek

A legegyszerűbb eset, ha csak egyetlen $y = F(x)$ függvényt szeretnénk linearizálni (“kiegyenesíteni”). Ekkor az X tengelyen maradjon a hagyományos ekvidisztáns skála, az Y tengelyen pedig az s számot (“feliratot”) a $t = F^{-1}(s)$ távolságra írjuk, vagyis az (1) képletben

$$\psi = F. \quad (1)$$

Ennek speciális esete a szemilogaritmikus- és Gauss- rendszer: $F(x) = \log(x)$ illetve $F(x) = \Phi(x)$ választással.

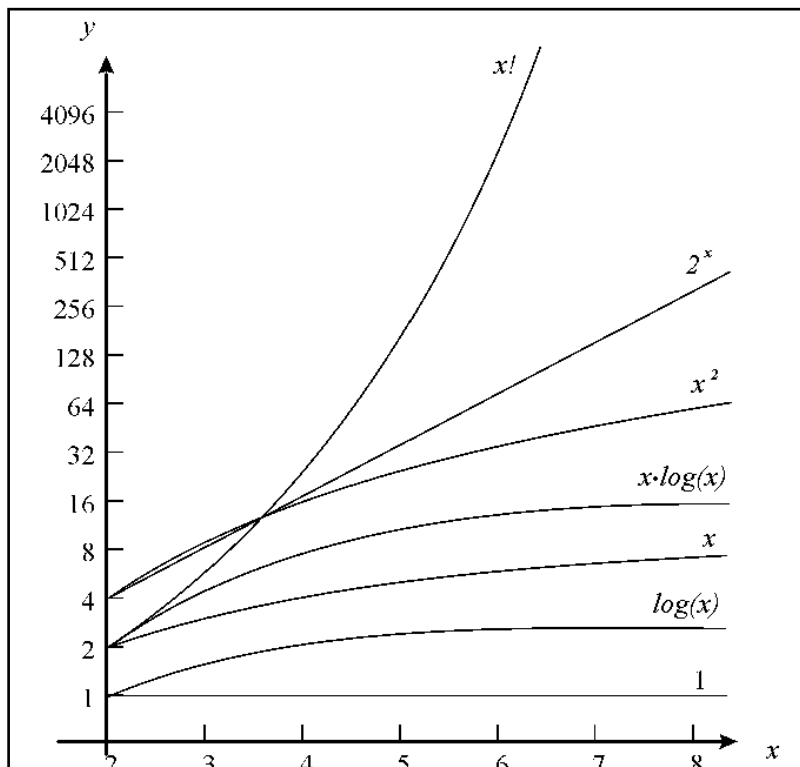
A hagyományos Descartes-féle koordinátarendszerben $\varphi = \psi = id$, vagyis $x_f = x_t$ és $y_f = y_t$. Tetszőleges rögzített (nemnulla) u, v, u_0, v_0 valós számok esetén az

$$x_t = u \cdot x_f + u_0 \text{ és } y_t = v \cdot y_f + v_0 \quad (2)$$

koordináta-transzformáció lényegében megfelel a “klasszikus” **lineáris függvénytranszformációknak** (vízszintes és függőleges nyújtás, zsugorítás, eltolás...).

3.1.Szemilogaritmikus transzformáció

A **szemilogaritmikus** rendszerben az egyik tengelyen egyenletes (id), a másik tengelyen logaritmikus beosztást készítünk. A 3.1.1 ábrán $\varphi = id$ és $\psi = \log_2$, vagyis $x_t = x_f$ de $y_t = \log_2(y_f)$ vagyis $y_f = 2^{y_t}$:



3.1.1 ábra: szemilogaritmikus rendszer [15]

Ebben az esetben egyedül a $h(x) = 2^x$ függvény képe egyenes. A kereskedelemben régebben kapható szemilogaritmikus papír egy darabja az 1.2.4 ábrán látható (narancssárga papír).

Állítás: Általában is igaz a következő összefüggés: Tetszőleges $b \in \mathbb{R}$ valós és $a, c, d \in \mathbb{R}^+$ ($a \neq 1$) pozitív valós számok esetén a

$$x_t = \varphi(x_f) = x_f \text{ és } y_t = \psi(y_f) = \log_a(y_f) \quad (3)$$

koordináta-transzformáció után minden

$$h(x) = d \cdot c^{bx} \quad (4)$$

alakú (exponenciális) függvény grafikonja geometriailag egyenes.

Bizonyítás: A (3) képlet az ábrázoláskor azt jelenti, hogy

$$y_f = h(x_f) = d \cdot c^{bx_f} \quad (5)$$

mindkét oldal \log_a értékét véve kapjuk:

$$\log_a(y_f) = \log_a(d \cdot c^{bx_f}) = b \cdot x_f \cdot \log_a(c) + \log_a(d) \quad (6)$$

vagyis (2) alapján valóban

$$y_t = mx_t + \beta \quad (7)$$

A (2) képletben szereplő koordináta-transzformációt **szemilogaritmikus** transzformációnak nevezzük és megkülönböztetésül (φ_s, ψ_s) jellel jelöljük.

3.2. Logaritmikus transzformáció

A **logaritmikus** rendszerben (másnéven **log-log** rendszer) *mindkét tengelyen* logaritmikus beosztást készítünk (nem feltétlenül azonos alappal). A kereskedelemben régebben kapható logaritmikus papír egy darabja az 1.2.3 ábrán látható (kék papír).

Állítás: Tetszőleges $a, b \in \mathbb{R}^+$, $a \neq 1$, $b \neq 1$ pozitív és $m \in \mathbb{R}$ valós számok esetén a

$$x_t = \varphi(x_f) = \log_a(x_f) \text{ és } y_t = \psi(y_f) = \log_b(y_f) \quad (8)$$

koordináta-transzformáció után minden

$$h(x) = d \cdot x^m \quad (9)$$

alakú (hatvány) függvény grafikonja geometriailag egyenes.

Bizonyítás: A (9) egyenlőség mindkét oldalának \log_a -át vesszük:

$$\log_a(y_f) = m \cdot \log_a(x_f) + \log_a(m) \quad (10)$$

és a $\log_b(y_f) = \frac{\log_a(y_f)}{\log_a(a)}$ összefüggés alapján

$$c \cdot y_t = m \cdot x_t + \beta, \quad (11)$$

vagyis

$$y_t = m' \cdot x_t + \beta'. \quad (12)$$

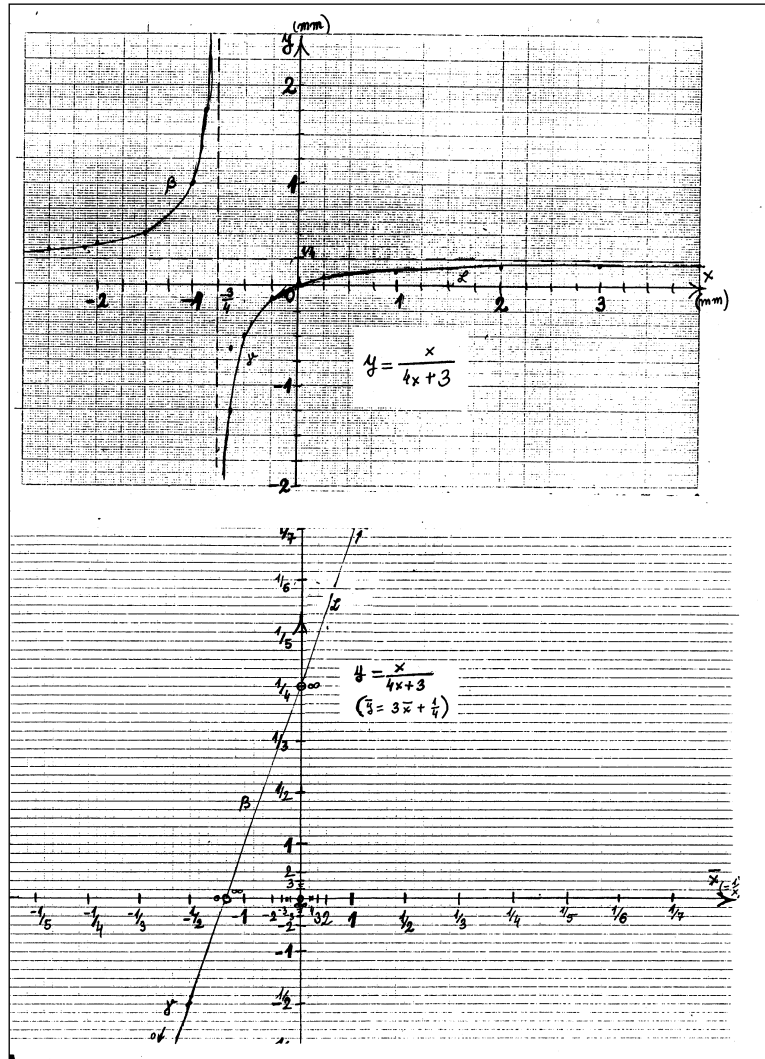
A fenti transzformációt **logaritmikus** transzformációnak nevezzük és megkülönböztetésül (φ_L, ψ_L) jellel jelöljük.

3.3. Reciprok transzformáció

A hiperbolikus rendszerben a szakirodalomban kevésbé ismert, ám annál hasznosabb **reciprok** skálát használjuk:

$$x_t = \varphi(x_f) = \frac{u}{x_f} \text{ és } y_t = \psi(y_f) = \frac{u}{y_f} \quad (13)$$

ekkor a $h(x) = \frac{ax+b}{cx+d}$ hiperbolák grafikonjai lesznek egyenesek-darabok. A síknegyedek és előjelek változatlanok a szokásos rendszerhez képest, de a tengelyeken lévő pontok nem valós számok, hiszen itt valamelyik koordináta nem értelmezhető (null-osztás). A síkot lényegében kifordítjuk, a végtelen távoli pontok jönnek a tengelyekre, az origóba pedig a $(+\infty, -\infty)$ pont kerül.



3.3.1 ábra: lineáris és reciprok rendszer

3.4. Normális eloszlás transzformáció

Ha egy mennyiség (jelenség, mérés) *normális eloszlást* "követ", akkor eloszlásfüggvénye (a mérések kumulatív összegzése) a következő képlettel írható le: $F_{m,\sigma}(z) = \Phi\left(\frac{z-m}{\sigma}\right)$ ahol m a mérés átlaga, és σ a szórása, és

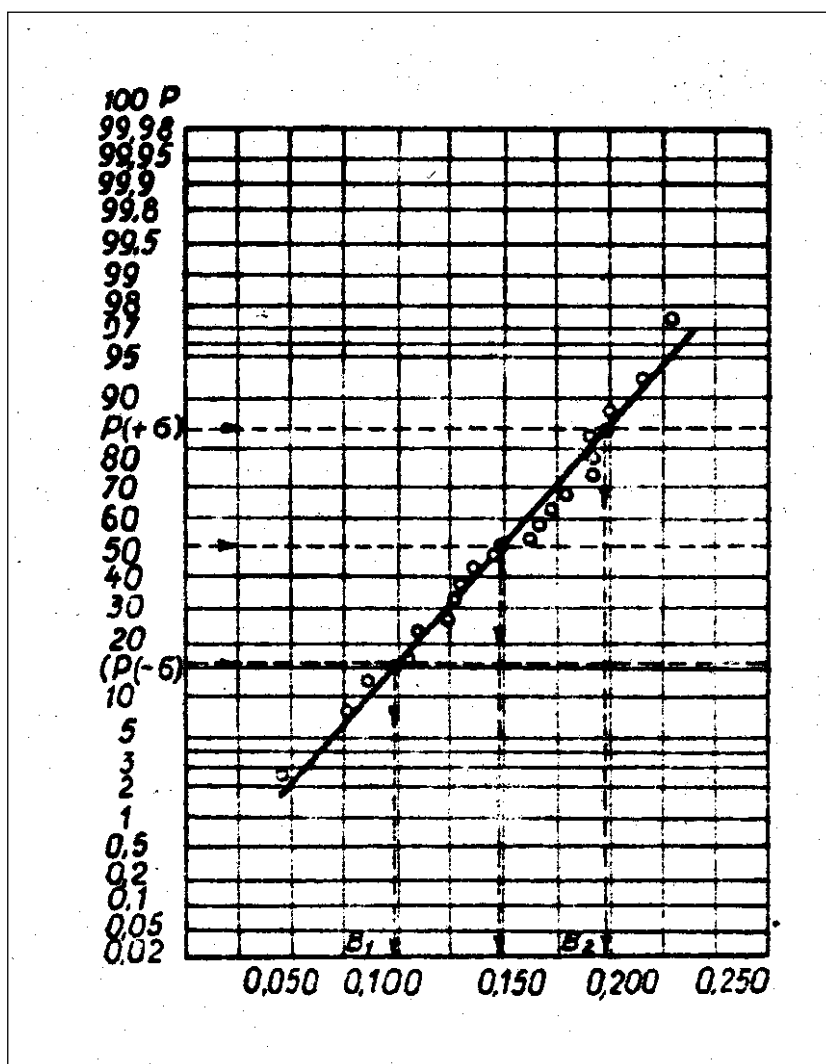
$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp\left(-\frac{t^2}{2}\right) dt. \quad (14)$$

Azonban **Liouville** tétele szerint $\Phi(x)$ -re nincs képlet, a szokásos függvényekkel, emiatt táblázatból szoktuk kikeresni. [12]

Tétel: $Dom(\Phi)=\mathbb{R}$, $0 < \Phi(x) < 1$ szigorúan növény folytonos, $\lim_{x \rightarrow -\infty} \Phi(x)=0$ és $\lim_{x \rightarrow +\infty} \Phi(x)=1$. Ezen kívül Φ specialitásai, hogy $\Phi(0)=1/2$ és Φ grafikonja középpontosan szimmetrikus a $(0, 1/2)$ pontra vagyis

$$\Phi(-x) = 1 - \Phi(x) . \tag{15}$$

A *linearizálás* módszerével lényegében a $\Phi(x)$ függvényt kell kiegyenesíteni, ahogy a 3.4.1 és a 3.4.2 ábrán látható.



3.4.1 ábra: Gauss-féle rendszer [13]

A Gauss-féle rendszerhez a $\Phi(x)$ függvény inverzét kell kiszámolnunk. Az X tengely beosztása a szokásos, de az Y tengelyen az s számot (“feliratot”) $t = \Phi^{-1}(x)$ távolságra kell írunk, ahol Φ^{-1} a Φ függvény inverze.

Például az $s = 0.999$ felirat körülbelül $t = \Phi^{-1}(0.999) \approx 3.9$ egységre lesz távol az X tengelytől.

Az X és Y tengely metszéspontjában (régén “origó”) az $s = 0.5$ felirat lesz, mert $t = \Phi^{-1}(0.5) = 0$ egység.

A (15) összefüggés miatt az Y tengely skálája az X tengely “alatt” hasonlít az X tengely “fölötti” skálázáshoz, például $s = 0.001$ felirat körülbelül $t = \Phi^{-1}(0.001) \approx -3.9$ egységre lesz távol az X tengelytől.

Liouville tétele miatt nincs képlet Φ inverzére sem, tehát az egyik módszer, hogy normális eloszlás táblázat (megtalálható szinte minden valószínűségszámítás könyvben, vagy pl.: <http://math.uni-pannon.hu/~szalkai/NormTabl.pdf>) alapján visszakeressük a megfelelő értékeket.

A másik módszer kevésbé ismert, pedig nagyon hasznos és modern. A [14] cikkben a következőt olvashatjuk:

$$y = \Phi(x) \approx 1/2 + 1/2 \cdot \tanh(0.8x) \quad (16)$$

ahol $\tanh(z)$ a “tangens hiperbolikus” függvény, képlete:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (y \in R) \quad (17)$$

Többek között:

$$-1 < \tanh(z) < +1 \quad (z \in R) \quad (18)$$

és

$$\tanh^{-1}(v) = \operatorname{Arth}(v) = \frac{1}{2} \ln \left(\frac{1+v}{1-v} \right) \quad (-1 < v < +1) \quad (19)$$

hiszen \tanh^{-1} (\tanh inverze) szokásos neve “*area tangens hiperbolikus*”. [11]

A (16) és (19) egyenlőségből

$$x = \Phi^{-1}(y) \approx \frac{10}{8} \cdot \tanh^{-1}(2y - 1) \quad (20)$$

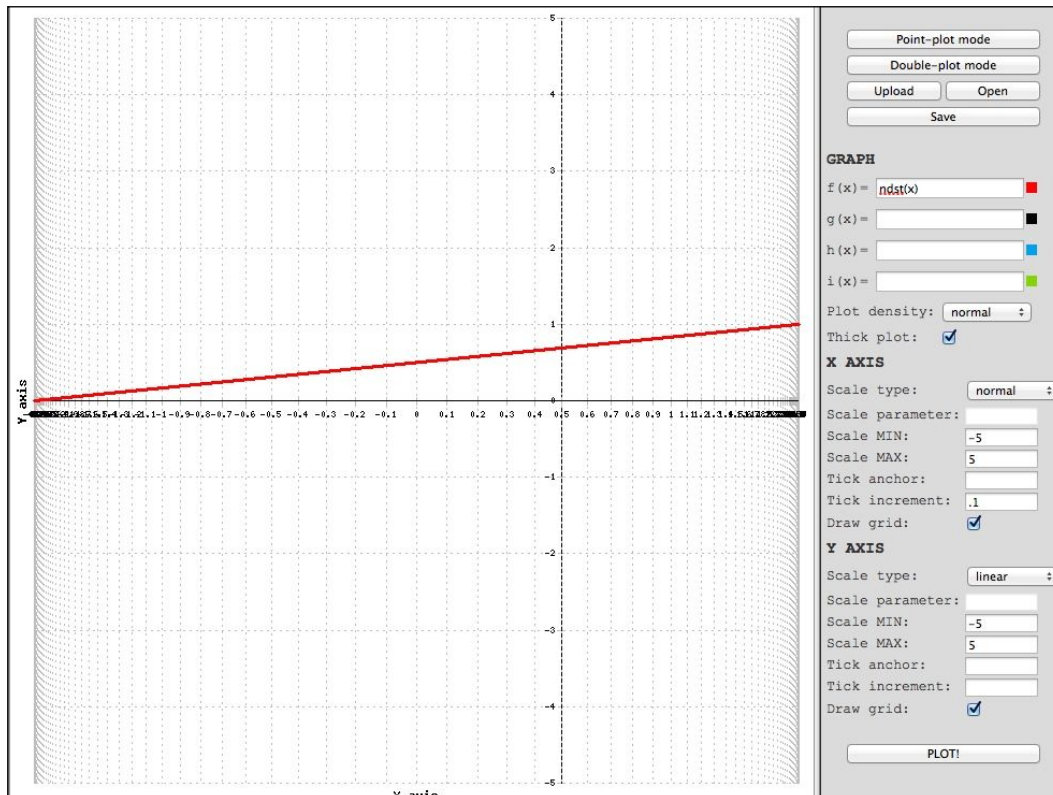
$$= \frac{5}{4} \cdot \frac{1}{2} \cdot \ln\left(\frac{1+(2y-1)}{1-(2y-1)}\right) \quad (21)$$

$$= \frac{5}{8} \ln\left(\frac{y}{1-y}\right) \quad (-1 < y < +1) \quad (22)$$

vagyis a Gauss- azaz Normális koordinátarendszerben az Y tengelyre az s feliratot

$$t = \Phi^{-1}(s) \approx \frac{5}{8} \cdot \ln\left(\frac{s}{1-s}\right) \quad (-1 < s < +1) \quad (23)$$

távolságra kell írunk.



3.4.2 ábra: $\Phi(x)$ függvény kiegyenesítése a kifejlesztett programmal

4. Az elkészített szoftver

4.1. Bevezetés

A kidolgozandó informatikai rendszer feladata lehetővé tenni a jól ismert koordináta-transzformációk mellett további kevésbé ismert és akár eddig még soha nem alkalmazott transzformációk használatát.

A szoftvert felhasználói ingyenesen, a világ bármely pontjáról elérhetik.

A rendszer korszerű, egyszerű, kényelmes, felhasználóbarát megoldásokkal támogatja a hatékony felhasználást.

4.2. A problémakör azonosítása

4.2.1. A probléma leírása

Hagyományos koordinátarendszerben függvények vagy pontthalmazok ábrázolásakor sokszor nem azonosítható pontosan a függvény-összefüggés típusa és paraméterei (csak lineáris összefüggés esetén).

A dolgozat során olyan alkalmazást szükséges kifejleszteni, mely a fenti problémára kínál megoldást oly módon, hogy a koordinátarendszer tengelyskáláinak módosításával megváltoztatja (transzformálja) az ábrázolt függvényt vagy pontthalmazt.

A transzformáció(k) elvégzése után az összefüggés esetleg lineárisává válik, ezáltal ismét azonosítható, leírható.

4.2.2. Üzleti lehetőségek

A program kezdeti verziója elsősorban a bevezetést, felhasználók toborzását szolgálja valamint a Pannon Egyetem professzorainak, mérnökeinek munkáját segíti. Hosszú távon elképzelhető az üzleti hasznosítás, pl. a weboldal magas látogatottsága esetén számottevő reklámbevétel lehetséges.

4.3.A termék és környezete

A rendszer egy központi számítógép (webszerver) segítségével működik. A megfelelő funkcionalitás megvalósításához szükséges, hogy a webszerver képes legyen PHP forráskód futtatására. A felhasználók saját számítógépükről (kliens) webböngésző segítségével használhatják a szoftvert. A rendszer nem igényel egyéb szolgáltatást.

4.3.1.Szereplők azonosítása

A rendszerben résztvevő szereplők a következők:

- Felhasználó
- Rendszeradminisztrátor

4.3.2.Magas szintű célok meghatározása

Magas szintű cél	Prioritás	Problémák	Jelen megoldások
Magas felhasználói létszám elérése vonzó környezet biztosításával	Magas	Reklámozás, felhasználói létszám növelése csak ingyenes eszközökkel elképzelhető	Reklámok, linkek elhelyezése pl.: az egyetem weboldalán
Kényelmes, hatékony felhasználói tér megvalósítása	Magas	Egyszerű, könnyen használható felület kell, kerülendők a felesleges műveletek	Átlátható felület, megfelelő felhasználói segédlet

4.3.3.Felhasználói szintű célok

- Felhasználó: kényelmes, kellemes környezet. Egyszerű kezelhetőség
- Rendszeradminisztrátor: kevés beavatkozás legyen szükséges, nagyon kis időráfordítással.

4.3.4. Egyéb követelmények és kényszerek

- A rendszernek folyamatos rendelkezésre állást kell biztosítania.
- A rendszernek hibatűrő módon kell működnie.
- A rendszernek lehetőséget kell biztosítania a továbbfejlesztésre.

4.4. A megvalósítás folyamata

4.4.1. Technológiák

A rendszer kifejlesztése során az alábbi programnyelveket és technológiákat használtam fel:

- HTML (HyperText Markup Language): Weboldalak készítéséhez használt leíró nyelv. A weboldal alapjait készítettem el vele.
- CSS (Cascading Style Sheets): A HTML oldalak megjelenését befolyásoló egyszerű nyelv. A weboldal objektumainak stílusának leírására használtam.
- JavaScript: Interaktív weboldalak létrehozásához használt parancsnyelv. A weboldal interaktív elemeit készítettem el vele. (PopUp ablakok, fileok fel- és letöltése, stb.)
- PHP (Personal Home Page): Dinamikus weboldalak készítéséhez használt általános szerveroldali szkriptnyelv. Koordinátarendszerek, függvények, pontthalmazok ábrázolása illetve egyéb funkciók kifejlesztéséhez használtam.
- PHPlot: Szabadon felhasználható és módosítható PHP könyvtár dinamikus üzleti, tudományos diagramok és táblázatok készítéséhez. Kiindulópontként szolgált a programban használt speciális koordinátatengelyek kifejlesztéséhez. (Az interneten elérhető verzióban lineáris és logaritmus tengelyek használhatók az ábrázoláshoz. <http://www.phplot.com>)

4.4.2. Fejlesztői környezet

A rendszer fejlesztésekor nem használtam kimondottan erre célra készített fejlesztői környezetet. A programkódot Sublime Text 2 szövegszerkesztőben írtam, a tesztelést pedig Apple Safari és Google Chrome böngészőben végeztem.

4.4.3. Fejlesztés iterációkban

A fejlesztést 1-4 hetes iterációkban végeztem (iteráció: miniprojekt, eredménye futtatható de még nem teljes értékű rendszer).

- **0. iteráció:** A megelőző iterációban körbejártam a téma matematikai oldalát, sorra vettem a már létező alkalmazásokat, elemeztem a szoftverrel szemben támasztott követelményeket majd ezek alapján megválasztottam a szükséges technológiákat, előzetes rendszertervet készítettem.
- **1. iteráció:** Az első iteráció során közelebbről megismerkedtem a szükséges technológiákkal, kiemelten a PHP programnyelvvvel és a PHPlot keretrendszerrel. Általános módszert dolgoztam ki a koordinátatengelyek transzformálására a PHPlot rendszerben. Elkészítettem a szoftver látványtervét majd ez alapján egy működő verziót melyben a funkciókat még nem implementáltam, az *id* függvény ábrázolására volt képes *lineáris* rendszerben.
- **2. iteráció:** A második iteráció során a kezdeti rendszert fejlesztettem tovább. Implementáltam az *exponenciális*, *hatvány* és *gyök* koordinátatengelyeket, a weblapon elhelyeztem egy beviteli mezőt a függvény képletének számára, ehhez készítettem egy egyszerű értelmező algoritmust. Itt már tetszőleges függvényeket ábrázolhatunk négy féle koordinátatengellyel.
- **3. iteráció:** A harmadik iterációban további beviteli mezőket és kezelőszerveket helyeztem el a weblapon, melyekkel a szükséges beállítások elvégezhetők. Implementáltam a *reciprok*, *normális eloszlás* és *lognormális eloszlás* skálákat. Elkészítettem a munkamenet mentésével és megnyitásával

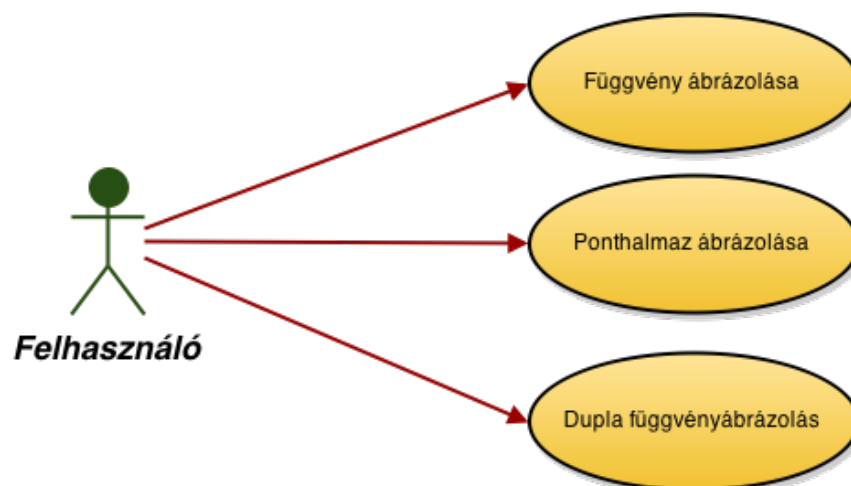
kapcsolatos funkciókat. (file mentése, file feltöltése a szerverre, file megnyitása, file tartalmának ellenőrzése és értelmezése)

- **4. iteráció:** A negyedik iteráció során implementáltam a *tangens*, *arkusz tangens*, *tangens hiperbolikus*, *area tangens hiperbolikus* skálákat. A *hatvány* és *gyök* skálákhoz elhelyeztem egy újabb beviteli mezőt, mellyel paramétert adhatunk meg a skála kialakításához. (pl.: a hatvány skála, $h(x^a)$, ahol a megadható paraméter) Elkészítettem a ponthalmazok ábrázolásához használható különálló funkciót, ebben a meglévő skálák szintén használhatók. Elkészítettem a dupla ábrázolásához használható funkciót, mellyel a képernyőn két koordinátarendszert (és egyszerre két függvény képét vagy akár ugyanazon függvény képét más-más rendszerben) ábrázolhatunk.

- **5. iteráció:** Az ötödik iterációban lehetővé tettem, hogy egy koordinátarendszerben akár négy függvény képét is egyszerre ábrázolhassuk. Kidolgoztam az adatbevitelre vonatkozó hibakezelés módszerét. (pl.: hogy viselkedjen a rendszer hibás függvényképlet beírása esetén, vagy az egyes skálák határainak rossz megadása esetén, stb) Az utolsó iteráció alatt négy fős csoportot kértem fel, a rendszer tesztelésére, ehhez már egy interneten keresztül elérhető webszervert használtam. A rendszer a későbbiekben a Pannon Egyetem szerverén lesz elérhető.

4.5.Használati esetek

4.5.1.Használati eset-modell



4.5.2. Használati eset: Függvény ábrázolása

Elsődleges szereplő:

- Felhasználó

Szereplők érdekei:

- Felhasználó: A kívánt függvény(ek) megfelelő ábrázolása a lehető legrövidebb idő alatt, és legkevesebb munka befektetésével.

Előfeltételek:

- A felhasználó böngészőjében megnyitotta a szoftvert. (beírta a webcímét)

Utófeltételek:

- A kívánt függvény(ek) a rendszer megfelelően ábrázolta.

Sikeres forgatókönyv:

- (1) A felhasználó beírja az ábrázolni kívánt függvény(ek) képlet(ei)t a *GRAPH / f(x), (g(x), h(x), i(x))* mező(k)be.
- (2) Kiválasztja az ábrázolás részletességét (*GRAPH / Plot density*), alapértelmezett esetben “normal” érték.
- (3) Esetleg vastagított ábrázolást választ (*GRAPH / Thick plot*)
- (4) Kiválasztja az *X* tengely skálatípusát az *X AXIS / Scale Type* mezőben, alapértelmezett esetben “linear” érték.
- (5) Hatvány (*power*) és gyök (*root*) skálatípus esetén kiválasztja az *X* tengely skálaparaméterét az *X AXIS / Scale Parameter* mezőben.
- (6) Kiválasztja az *X* tengely skálamínimumát az *X AXIS / Scale MIN* mezőben.
- (7) Kiválasztja az *X* tengely skálamaximumát az *X AXIS / Scale MAX* mezőben.
- (8) Kiválasztja az *X* tengelyen elhelyezni kívánt kitüntetett jelzést az *X AXIS / Tick anchor* mezőben. (ezután a többi jelzés a kitüntetett jelzéshez igazodik)

- (9) Kiválasztja az X tengelyen a jelzések gyakoriságát az X *AXIS / Tick increment* mezőben.
- (10) Ha nem kíván rácsot rajzolni a koordinátarendszerbe, akkor a X *AXIS / Draw grid* mezőt inaktív állapotba állítja.
- (11) A 4. - 10. pontokat végrehajtja az Y tengely esetében is.
- (12) A kirajzoláshoz megnyomja a *PLOT!* gombot.
- (13) Esetleg a felhasználó a *Save* gomb megnyomásával elmenti (letölti) a munkamenet adatait.

Alternatív forgatókönyvek:

- Előzőleg elmentett munkamenet folytatása:
 - (1) A felhasználó az *Upload* gomb megnyomásával feltölt egy munkamenet fület a szerverre.
 - (2) Az *Open* gomb megnyomásával betölti az adatokat a megfelelő mezőkbe
 - (3) A kirajzoláshoz megnyomja a *PLOT!* gombot.
- A felhasználó az adatbevitel során hibás adatot ad meg:
 - (1) A rendszer jelzi, hogy melyik adat hibás és útmutatást ad az adott mező kitöltésére.
 - (2) Folytatódik az adatbevitel a sikeres forgatókönyv megfelelő pontjától.

Technológiai követelmények:

- Kis sávszélességű hálózati kapcsolat esetén is gyorsan működjön a rendszer.

Felhasználási gyakoriság:

- Csúcsidőben szinte folyamatosan.

Képernyőképek:

- 4.5.3.1 képernyőkép
- 4.5.3.2 képernyőkép (adatbeviteli hiba)

4.5.3. Használati eset: Ponthalmaz ábrázolása

Elsődleges szereplő:

- Felhasználó

Szereplők érdekei:

- Felhasználó: A kívánt pontthalmaz megfelelő ábrázolása a lehető legrövidebb idő alatt, és legkevesebb munka befektetésével.

Előfeltételek:

- A felhasználó böngészőjében megnyitotta a szoftvert. (beírta a webcímét)
- A *Point-plot mode* gomb megnyomásával átváltott pontthalmaz ábrázolás módba.

Utófeltételek:

- A kívánt pontthalmazt a rendszer megfelelően ábrázolta.

Sikeres forgatókönyv:

- (1) A felhasználó beírja az ábrázolni kívánt pontok halmazát a *POINT SET* mezőbe.
- (2) Kiválasztja az *X* tengely skálatípusát az *X AXIS / Scale Type* mezőben, alapértelmezett esetben “*linear*” érték.
- (3) Hatvány (*power*) és gyök (*root*) skálatípus esetén kiválasztja az *X* tengely skálaparaméterét az *X AXIS / Scale Parameter* mezőben.
- (4) Kiválasztja az *X* tengely skálaminimumát az *X AXIS / Scale MIN* mezőben.
- (5) Kiválasztja az *X* tengely skálamaximumát az *X AXIS / Scale MAX* mezőben.
- (6) Kiválasztja az *X* tengelyen elhelyezni kívánt kitüntetett jelzést az *X AXIS / Tick anchor* mezőben. (ezután a többi jelzés a kitüntetett jelzéshez igazodik)
- (7) Kiválasztja az *X* tengelyen a jelzések gyakoriságát az *X AXIS / Tick increment* mezőben.

- (8) Ha rácsot kíván rajzolni a koordináta-rendszerbe, akkor a *X AXIS / Draw grid* mezőt aktív állapotba állítja.
- (9) A 2. - 8. pontokat végrehajtja az *Y* tengely esetében is.
- (10) A kirajzoláshoz megnyomja a *PLOT!* gombot.

Alternatív forgatókönyvek:

- A felhasználó az adatbevitel során hibás adatot ad meg:
 - (1) A rendszer jelzi, hogy melyik adat hibás és útmutatást ad az adott mező kitöltésére.
 - (2) Folytatódik az adatbevitel a sikeres forgatókönyv megfelelő pontjától.

Technológiai követelmények:

- Kis sávszélességű hálózati kapcsolat esetén is gyorsan működjön a rendszer.

Felhasználási gyakoriság:

- Csúcsidőben szinte folyamatosan.

Képernyőképek:

- 4.5.3.3 képernyőkép
- 4.5.3.4 képernyőkép (adatbeviteli hiba)

4.5.4. Használati eset: Dupla függvényábrázolás

Elsődleges szereplő:

- Felhasználó

Szereplők érdekei:

- Felhasználó: A kívánt függvények megfelelő ábrázolása a lehető legrövidebb idő alatt, és legkevesebb munka befektetésével.

Előfeltételek:

- A felhasználó böngészőjében megnyitotta a szoftvert. (beírta a webcímét)
- A *Double-plot mode* gomb megnyomásával átváltott dupla ábrázolás módba.

Utófeltételek:

- A kívánt függvényeket a rendszer megfelelően ábrázolta.

Sikeres forgatókönyv:

- (1) A felhasználó beírja az ábrázolni kívánt függvények képleteit a *GRAPH / f(x)*, mezőkbe. (mindkét oldalon)
- (2) Kiválasztja az ábrázolás részletességét (*GRAPH / Plot density*), alapértelmezett esetben “normal” érték. (mindkét oldalon)
- (3) Esetleg vastagított ábrázolást választ (*GRAPH / Thick plot*) (mindkét oldalon)
- (4) Kiválasztja az *X* tengely skálatípusát az *X AXIS / Scale Type* mezőben, alapértelmezett esetben “linear” érték. (mindkét oldalon)
- (5) Hatvány (*power*) és gyök (*root*) skálatípus esetén kiválasztja az *X* tengely skálaparaméterét az *X AXIS / Scale Parameter* mezőben. (mindkét oldalon)
- (6) Kiválasztja az *X* tengely skálaminimumát az *X AXIS / Scale MIN* mezőben. (mindkét oldalon)
- (7) Kiválasztja az *X* tengely skálamaximumát az *X AXIS / Scale MAX* mezőben. (mindkét oldalon)
- (8) Kiválasztja az *X* tengelyen elhelyezni kívánt kitüntetett jelzést az *X AXIS / Tick anchor* mezőben. (ezután a többi jelzés a kitüntetett jelzéshez igazodik) (mindkét oldalon)
- (9) Kiválasztja az *X* tengelyen a jelzések gyakoriságát az *X AXIS / Tick increment* mezőben. (mindkét oldalon)
- (10) Ha nem kíván rácsot rajzolni a koordinátarendszerbe, akkor a *X AXIS / Draw grid* mezőt inaktív állapotba állítja. (mindkét oldalon)
- (11) A 4. - 10. pontokat végrehajtja az *Y* tengely esetében is. (mindkét oldalon)
- (12) A kirajzoláshoz megnyomja a *PLOT!* gombot. (mindkét oldalon)

Alternatív forgatókönyvek:

- A felhasználó az adatbevitel során hibás adatot ad meg:

- (1) A rendszer a megfelelő oldalon jelzi, hogy melyik adat hibás és útmutatást ad az adott mező kitöltésére.
- (2) Folytatódik az adatbevitel a sikeres forgatókönyv megfelelő pontjától.

Technológiai követelmények:

- Kis sávszélességű hálózati kapcsolat esetén is gyorsan működjön a rendszer.

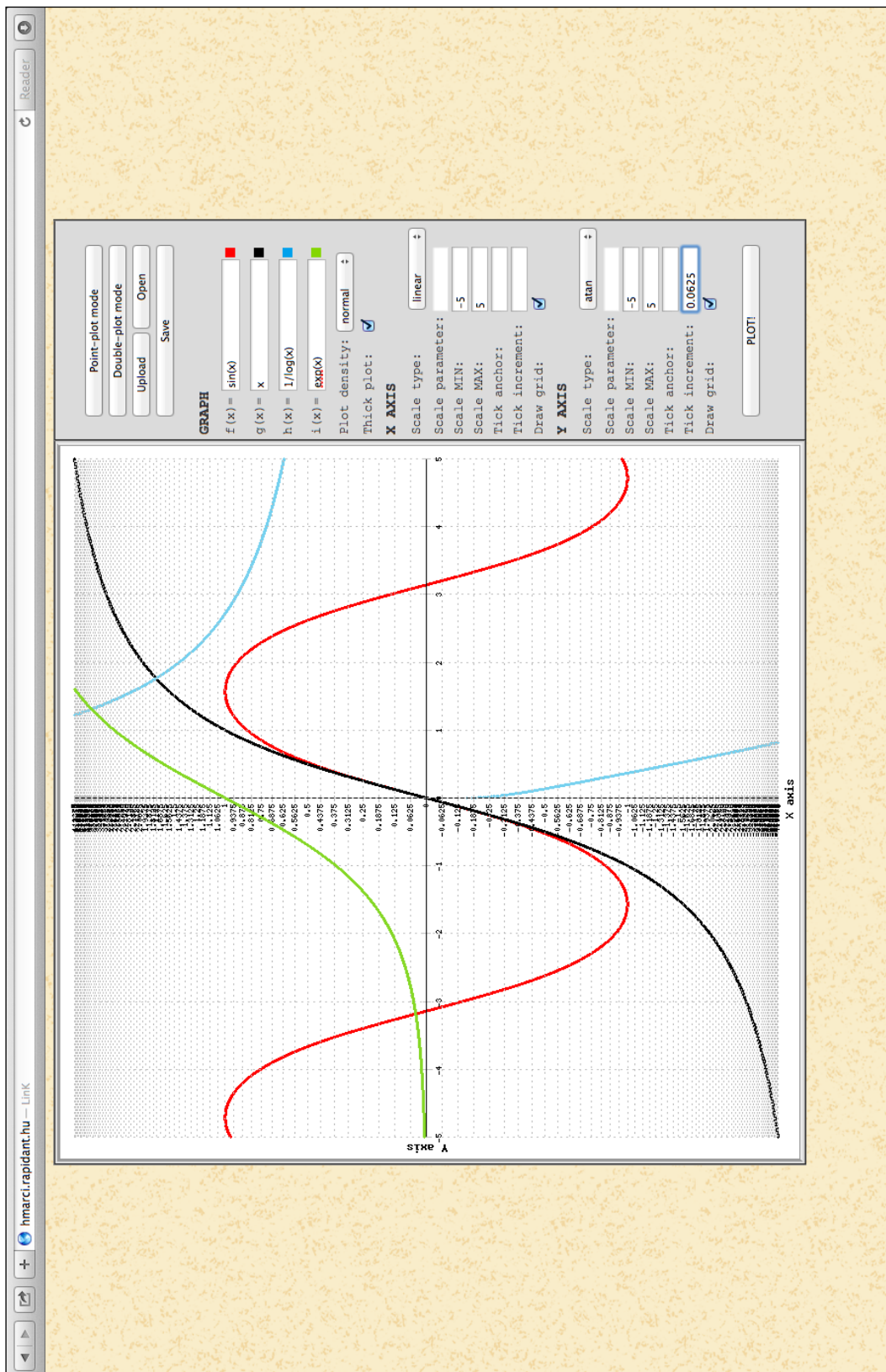
Felhasználási gyakoriság:

- Alkalmanként.

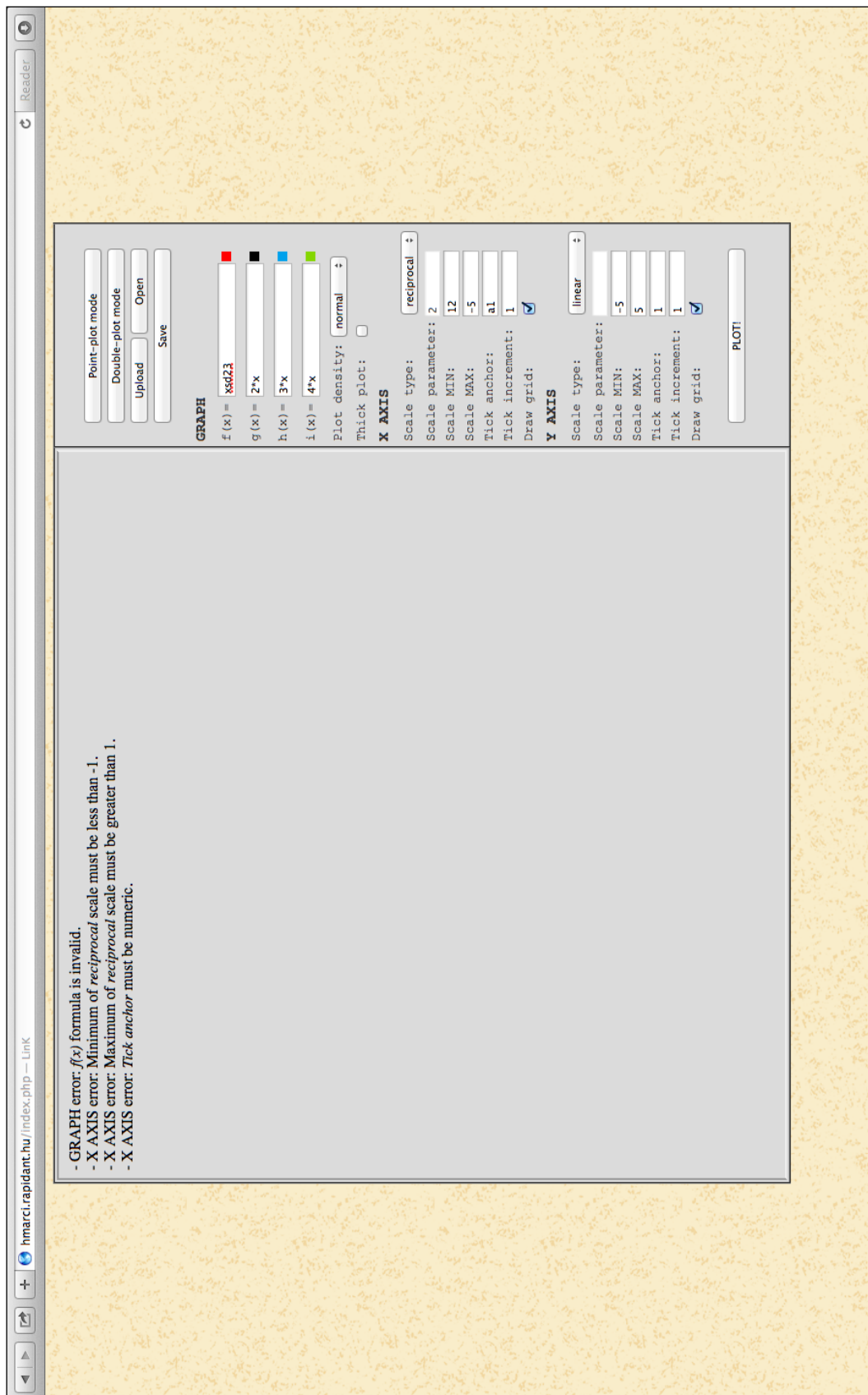
Képernyőképek:

- *4.5.3.5 képernyőkép*
- *4.5.3.6 képernyőkép (adatbeviteli hiba)*

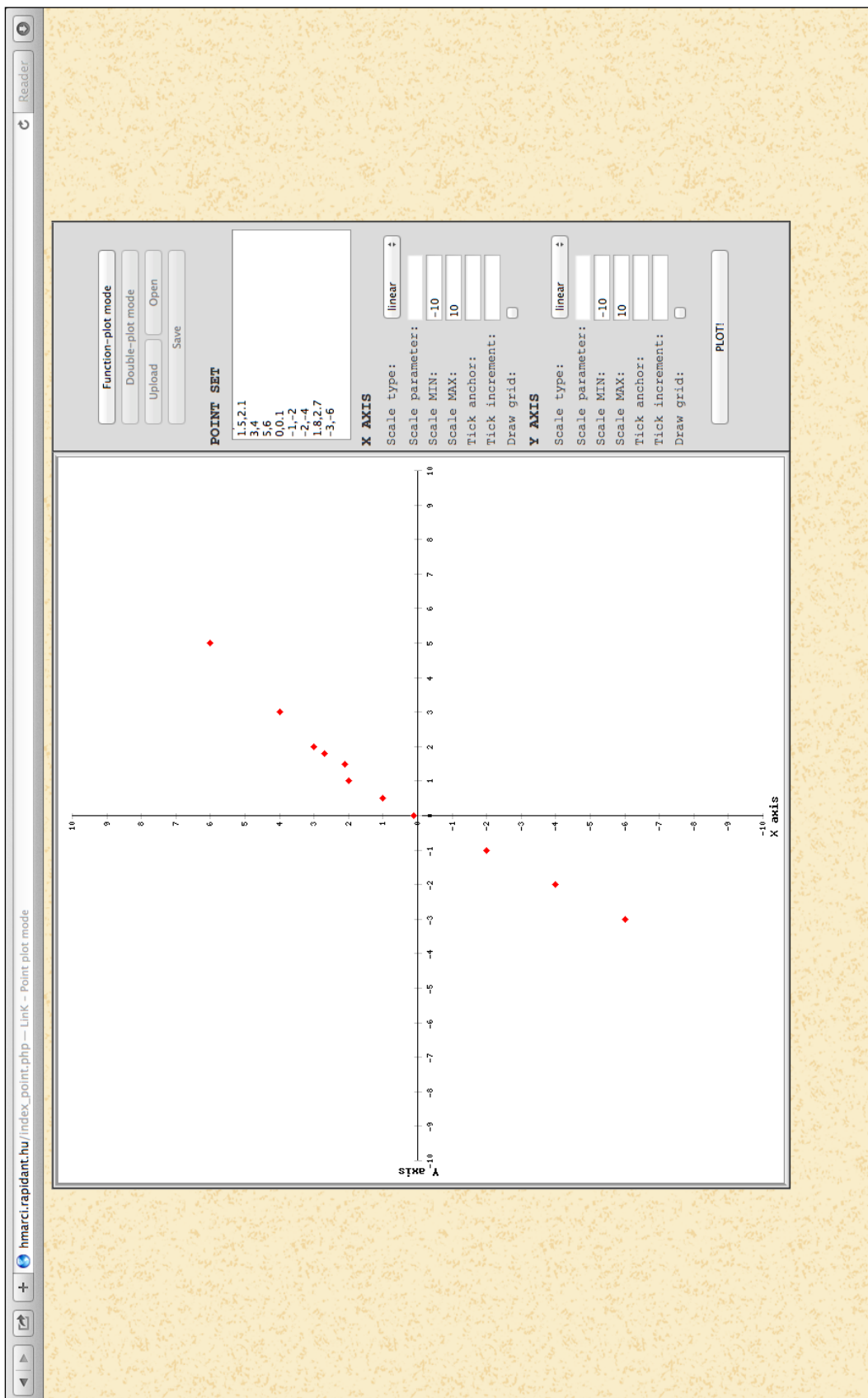
4.5.5. Képernyőképek



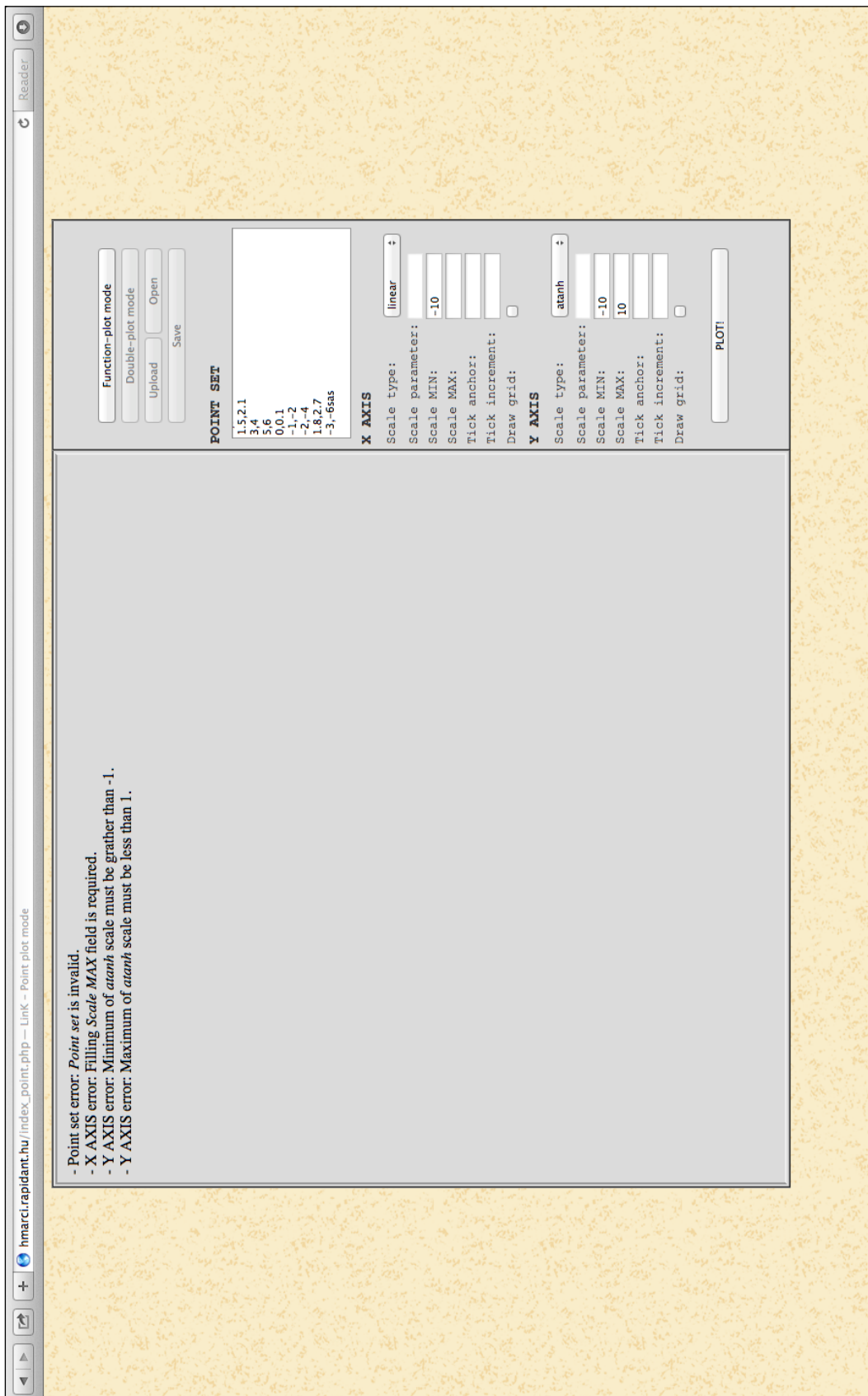
4.5.3.1 képernyőkép



4.5.3.2 képernyőkép



4.5.3.3 képernyőkép



4.5.3.4 képernyőkép



4.5.3.5 képernyőkép

hmarci.rapidant.hu/index_double.php - Link - Double plot mode

Reader

GRAPH
 $f(x) = \text{pow}(x, 3)$
 Plot density: normal
 Thick plot:

X AXIS
 Scale type: logarithm
 Scale parameter:
 Scale MIN: 1
 Scale MAX: 10
 Tick anchor:
 Tick increment:
 Draw grid:

Y AXIS
 Scale type: logarithm
 Scale parameter:
 Scale MIN: 0
 Scale MAX: 10
 Tick anchor:
 Tick increment:
 Draw grid:

- Y AXIS error: Minimum of logarithm scale must be greater than 0.

GRAPH
 $f(x) = \text{pow}(x, 3)$
 Plot density: normal
 Thick plot:

X AXIS
 Scale type: linear
 Scale parameter:
 Scale MIN: 1
 Scale MAX: 10
 Tick anchor:
 Tick increment:
 Draw grid:

Y AXIS
 Scale type: linear
 Scale parameter:
 Scale MIN: 1
 Scale MAX: 10
 Tick anchor:
 Tick increment:
 Draw grid:

PLOT

4.5.3.6 képernyőkép

4.6.Kiegészítő specifikáció

A kiegészítő specifikáció tartalmazza a rendszer azon követelményeit, melyek a használati esetekben nem jelennek meg.

4.6.1.A felhasználás szabályai

A dolgozathoz elkészített szoftvert az interneten bárki (magánszemély vagy cég) elérheti, szabadon használhatja (nem kereskedelmi célokra) és módosíthatja. A rendszer működése matematikai szabályokon alapszik. Használatához bizonyos matematikai alapismeret (függvényábrázolás, koordinátarendszerek stb.) illetve alapvető informatikai ismeretek szükségesek valamint az előző fejezetekben tárgyalt definíciók, bizonyítások tanulmányozása is ajánlott.

4.6.2.Használhatóság

A rendszer iránti alapvető követelmények:

- Egyszerű használhatóság: A felhasználói útmutató hiányában is az alapvető funkcionalitásnak egyértelműnek kell lennie.
- Sebesség: A rendszer válaszidejének (a felhasználói interakciókra adott válaszok a felhasználóhoz való visszajutásának ideje) ésszerű keretek közt kell maradnia. (pl.: néhány másodperc)
- A felhasználói felület legyen áttekinthető és izléses.

4.6.3.Megbízhatóság

Esetleges rendszerhiba vagy kommunikációs hiba miatti leállások után a felhasználók egyértelmű jelzést kapjanak arról, hogy sikeres vagy sikertelen volt-e a művelet. Továbbfejlesztési lehetőség az esetleges rendszerhibák naplózása.

4.6.4.Teljesítmény

A rendszernek akkor is teljesítenie kell a vele szemben támasztott követelményeket (Lásd *4.5 Használati esetek*, *4.6.2 Használhatóság*) ha a felhasználók száma nagy mértékben megnövekszik.

4.6.5.Támogathatóság

A rendszerkövetelmények idővel változhatnak (pl.: továbbfejlesztés) ezért a rendszernek ezen szabályokat rugalmasan kell kezelnie, ezeknek egyszerűen konfigurálhatónak kell lennie.

4.6.6.Implementációs kényszerek

A követelmények szempontjából nincsenek konkrét technológiai elvárások, de a rendszernek online elérhetőnek kell lennie ezért valamilyen webes technológia alkalmazása feltétlenül szükséges.

4.6.7.Komponensek

Vásárolt komponensek nincsenek, ingyenes, nyílt forráskódú komponensként a PHPlot (Lásd még *4.4.1*) könyvtárat használtam fel.

4.6.8.Interfészek

A rendszernek webszerveren (vagy szimulált webszerver) kell futnia, a felhasználók számára egy átlagos kiépítésű számítógép használatával elérhetőnek kell lennie.

4.7.Felhasználói útmutató

A program használatával tetszőleges függvényeket és ponthalmazokat ábrázolhatunk lineáris és különböző nem ekvidisztáns tengelyű koordinátarendszerekben.

A tengelyek skálázásának 12 féle módja érhető el, ezáltal a használható koordinátarendszerek száma 144.

Tengelyskálák:

- **Lineáris** (linear)
- **Logaritmus** (logarithm)
- **Exponenciális** (exponent)
- **Hatvány** (power)
- **Gyök** (root)
- **Reciprok** (reciprocal)
- **Normál eloszlás** (normal)
- **Lognormál eloszlás** (lognormal)
- **Tangens** (tan)
- **Arkusz tangens** (atan)
- **Tangens hiperbolikus** (tanh)
- **Area tangens hiperbolikus** (atanh)

4.7.1.Függvényábrázolás mód (function-plot)

Függvényábrázolás módban egyszerre négy függvény képét jeleníthetjük meg tetszőleges beállítások mellett (Lásd 4.5 Használati esetek).

A felhasználói felület két fő részre osztható. Jobb oldalon található a kezelőszervek, bal oldalon pedig a rajzoló ablak, melyben megjeleníthetjük adatainkat.

A kezelőszervek az alábbi funkciókat látják el, fentről lefelé haladva:

- Point-plot mode: Átváltás ponthalmaz ábrázolás módra;
- Double-plot mode: Átváltás dupla-ábrázolás módra;

- Upload: A megnyíló ablakban kiválaszthatunk egy előzőleg lementett fület, mely megfelelő formátumban tartalmazza a megjelenített függvényeket és a hozzájuk tartozó beállításokat.
- Open: Ezzel a gombal betölthetjük az utoljára feltöltött file tartalmát a beviteli mezőkbe.
- Save: A megjelenített függvények és beállítások mentése fileba.
- GRAPH / f(x), g(x), h(x), i(x): Adatbeviteli mező a megjeleníteni kívánt függvények számára.
- GRAPH / Plot density: Itt kiválasztható a rajzolás részletessége. normál értéknél 5000, high értéknél 10000, very_high értéknél pedig 20000 pontot ábrázolhatunk az X tengely szélességében.
- GRAPH / Thick plot: Aktív állapotban az ábrázolt pontokat vastagon kiemelve jeleníthetjük meg.
- X AXIS (Y AXIS) / Scale Type: tengely skála típusának kiválasztása.
- X AXIS (Y AXIS) / Scale parameter: Hatvány és gyök skála kiválasztása esetén paraméterezhetjük a tengely skáláját, a paraméter előbbi esetben a hatvány, utóbbi esetben a gyök kitevője.
- X AXIS (Y AXIS) / Scale MIN: Skála minimum.
- X AXIS (Y AXIS) / Scale MAX: Skála maximum
- X AXIS (Y AXIS) / Tick anchor: Megadhatunk egy kitüntetett értéket a tengelyen, mely mindenképpen megjelenik.
- X AXIS (Y AXIS) / Tick increment: Megadhatjuk a tengely beosztásának sűrűségét.
- X AXIS (Y AXIS) / Draw grid: Aktív állapotban a rajzolásakor segéd rácsot jeleníthetünk meg.
- PLOT!: Kirajzolás

4.7.2. Ponthalmaz ábrázolás mód (point-plot)

Ponthalmaz ábrázolás módban tetszőleges elemszámú ponthalmazt jeleníthetünk meg tetszőleges beállítások mellett (Lásd 4.5 Használati esetek).

A felhasználói felület két fő részre osztható. Jobb oldalon található a kezelőszervek, bal oldalon pedig a rajzoló ablak, melyben megjeleníthetjük adatainkat.

A kezelőszervek az alábbi funkciókat látják el, fentről lefelé haladva:

- Function-plot mode: Átváltás függvényábrázolás módra;
- POINT SET: Ebben a mezőben adhatjuk meg az ábrázolni kívánt pontokat. A pont koordinátáit egymástól vesszővel, a pontokat egymástól ENTER-rel választjuk el!
- X AXIS (Y AXIS) / Scale Type: lásd 4.7.1
- X AXIS (Y AXIS) / Scale parameter: lásd 4.7.1
- X AXIS (Y AXIS) / Scale MIN: lásd 4.7.1
- X AXIS (Y AXIS) / Scale MAX: lásd 4.7.1
- X AXIS (Y AXIS) / Tick anchor: lásd 4.7.1
- X AXIS (Y AXIS) / Tick increment: lásd 4.7.1
- X AXIS (Y AXIS) / Draw grid: lásd 4.7.1
- PLOT!: lásd 4.7.1

4.7.3. Dupla-ábrázolás mód (double-plot)

Dupla ábrázolás módban egyszerre két koordinátarendszert használhatunk a megjelenítéshez. Ideális ugyanazon függvények más-más rendszerben való ábrázolásához (Lásd 4.5 Használati esetek).

A felhasználói felület négy fő részre osztható. Jobb, és bal oldalon található a kezelőszervek, középen pedig a két rajzoló ablak, melyben megjeleníthetjük adatainkat.

A kezelőszervek az alábbi funkciókat látják el, fentről lefelé haladva:

- GRAPH / f(x): lásd 4.7.1
- GRAPH / Plot density: lásd 4.7.1
- GRAPH / Thick plot: lásd 4.7.1
- X AXIS (Y AXIS) / Scale Type: lásd 4.7.1
- X AXIS (Y AXIS) / Scale parameter: lásd 4.7.1

- X AXIS (Y AXIS) / Scale MIN: lásd 4.7.1
- X AXIS (Y AXIS) / Scale MAX: lásd 4.7.1
- X AXIS (Y AXIS) / Tick anchor: lásd 4.7.1
- X AXIS (Y AXIS) / Tick increment: lásd 4.7.1
- X AXIS (Y AXIS) / Draw grid: lásd 4.7.1
- PLOT!: lásd 4.7.1

4.7.4. Beviteli szabályok

Függvényképlet beírásakor az alábbi szabályokat alkalmazzuk:

- A független változót x formában adhatjuk meg.
- Tizedesjegyek elválasztására a $.$ használható.
- Függvényábrázolás és dupla-ábrázolás módban a függvényképlet

beviteléhez az alábbi egy-operandusú műveletek használhatók:

- természetes alapú logaritmus: $\log(\text{operandus})$
- exponens: $\exp(\text{operandus})$
- négyzetgyök: $\sqrt{\text{operandus}}$
- színusz: $\sin(\text{operandus})$
- koszinusz: $\cos(\text{operandus})$
- tangens: $\tan(\text{operandus})$
- kotangens: $\cot(\text{operandus})$
- arkusz színusz: $\arcsin(\text{operandus})$
- arkusz koszinusz: $\arccos(\text{operandus})$
- arkusz tangens: $\arctan(\text{operandus})$
- arkusz kotangens: $\text{arccot}(\text{operandus})$
- színusz hiperbolikus: $\sinh(\text{operandus})$
- koszinusz hiperbolikus: $\cosh(\text{operandus})$
- tangens hiperbolikus: $\tanh(\text{operandus})$
- kotangens hiperbolikus: $\text{coth}(\text{operandus})$
- area színusz hiperbolikus: $\text{asinh}(\text{operandus})$

- area koszinusz hiperbolikus: $\text{acosh}(\text{operandus})$
- area tangens hiperbolikus: $\text{atanh}(\text{operandus})$
- area kotangens hiperbolikus: $\text{acoth}(\text{operandus})$
- normál eloszlás: $\text{ndst}(\text{operandus})$
- lognormál eloszlás: $\text{logndst}(\text{operandus})$
- egész rész: $\text{integer}(\text{operandus})$
- A bevitelhez az alábbi két-operandusú műveletek használhatók:
 - logaritmus: $\text{log}(\text{operandus}_1, \text{operandus}_2)$, ahol operandus_1 az a szám, melynek logaritmusát kiszámítjuk, operandus_2 a logaritmus alapja
 - hatvány: $\text{pow}(\text{operandus}_1, \text{operandus}_2)$, ahol operandus_1 a hatvány alapja, operandus_2 a hatvány kitevője
- A bevitelhez az alábbi állandók használhatók:
 - π (=3.14159265...)
 - e (= 2.71828)

4.7.5. Tengelyek korlátai

A transzformált tengelyeknek létezhetnek korlátai attól függően, hogy milyen transzformációt alkalmazunk rajta. Egy transzformált tengely kiterjedése az alkalmazott transzformáló függvény érték készlete. Pl.: A logaritmus tengelyhez a logaritmus függvény inverz függvényét (exponenciális függvény) kell alkalmaznunk a tengelyen ezért a transzformált tengely a $]0, +\infty[$ tartományban jeleníthető meg.

Ábrázolhatósági megfontolásból a reciprok tengely minimuma -1-nél kisebb, maximuma pedig 1-nél nagyobb kell, hogy legyen.

4.8. Továbbfejlesztési lehetőségek

A rendszerbe az alábbi funkciók nem kerültek integrálásra, ezek iránymutatást adhatnak a későbbi továbbfejlesztéshez:

- A dupla-ábrázolás mód helyett esetleg több koordinátarendszert lehetne megjeleníteni egy képernyőn. (4 vagy több) Ez a képernyőméret, képernyőfelbontás függvényében esetleg kiválasztható.
- A függvényösszefüggések megállapításához régebben használt “vonalzós” módszert modellezni lehetne a képernyőn például a függvényre vagy ponthalmazra legjobban illeszkedő egyenes megjelenítésével.
- Az előző pontot bővíteni lehetne úgy, hogy a megjelenített egyenes paraméterei és az összefüggés típusa leolvashatóak legyenek a képernyőről. (ugyanis transzformált rendszerben az egyenesre illeszkedés nem lineáris összefüggés jelent)
- További tengelyek implementálása...

5. Irodalomjegyzék

(1) http://en.wikipedia.org/wiki/Logarithmic_scale HORVÁTH M. (letöltés dátuma 2015.05.05)

(2) <http://imgbuddy.com/richter-scale-graph.asp> HORVÁTH M. (letöltés dátuma 2015.05.05)

(3) http://www.bethlen.hu/matek/mathist/forras/Koordinata_rendszer.htm
HORVÁTH M. (letöltés dátuma 2015.05.05)

(4) http://4dlab.info/article_short_history_of_the_cartesian_coordinates.htm
HORVÁTH M. (letöltés dátuma 2015.05.05)

(5) <http://hu.wikipedia.org/wiki/Logarléc#T.C3.B6rt.C3.A9nete> HORVÁTH M. (letöltés dátuma 2015.05.05)

(6) http://my.hrw.com/math06_07/nsmedia/tools/Graph_Calculator/graphCalc.html HORVÁTH M. (letöltés dátuma 2015.05.06)

(7) <http://math.uni-pannon.hu/~szalkai/> HORVÁTH M. (letöltés dátuma 2015.05.06)

(8) http://wiki.ham.hu/index.php/Logaritmus_egysegek#A_decibel
HORVÁTH M. (letöltés dátuma 2015.05.08)

(9) <http://www.vilaglex.hu/Lexikon/Html/Kemhatas.htm> HORVÁTH M. (letöltés dátuma 2015.05.08)

(10) <http://www.fuldugo.hu/hallas-wiki/hangero-decibelben> HORVÁTH M.
(letöltés dátuma 2015.05.08)

(11) <http://math.uni-pannon.hu/~szalkai/Hypfvmind-.doc> Szalkai István:
Hiperbolikus függvények alaptulajdonságai, HORVÁTH M. (letöltés dátuma
2015.05.10)

(12) <http://math.uni-pannon.hu/~szalkai/Anal-Tk1B-c.pdf> Szalkai István,
Mikó Teréz: Közgazdaságtan matematikai alapjai, HORVÁTH M. (letöltés
dátuma 2015.05.10)

(13) LÁSZLÓ, Z. (1975) László Zoltán: Matematika VII -
Valószínűségszámítás, Egyetemi jegyzet, Veszprém, 1975

(14) ROBIN, A.C.: A quick approximation to the normal integral, The
Mathematical Gazette, 81 (1997), 95-96. old.,

(15) SZALKAI, I. (2001). Szalkai István: Diszkrét matematika és
algoritmuselmélet alapjai, Pannon Egyetemi Kiadó, 2001.

16. Mellékletek

16.1. Eljárások

A **PHPlot** könyvtár változóit és eljárásait a *phplot* osztály foglalja egységbe. A szoftver kifejlesztéséhez a fenti osztályt kibővítettem (*my_phplot* származtatott osztály) az alábbi eljárásokkal:

```
function set_plot_detail($detail)
```

(beállítja a rajzolás részletességét)

```
function print_debug($file)
```

(naplózást készít az elvégzett műveletről a megadott fileba)

```
function print_plot_data($file)
```

(munkamenet fileba mentése)

```
function position_in_plot($y)
```

(megadja, hogy az \$y koordináta az aktuális koordinátarendszerhez képest, hol helyezkedik el, /alatta, fölötte, benne/)

```
function CalculateFunctionData($new_function)
```

(elvégzi az ábrázoláshoz szükséges számításokat a felhasználó által beírt függvényképlet és egyéb adatok alapján)

```
function GetXScaleType()
```

(visszatér az X tengely skálatípusával)

```
function GetYScaleType()
```

(visszatér az Y tengely skálatípusával)

```
function set_x_axis_pos($value)
```

(beállítja az X tengely pozícióját az Y tengelyen)

```
function set_y_axis_pos($value)
```

(beállítja az Y tengely pozícióját az X tengelyen)

Módosított eljárások, melyek felülírják a *phplot* osztály megegyező nevű eljárásait:

```
function SetXScaleType($which_xst)
```

```
function SetYScaleType($which_yst)
```

```
protected function CalcAxisPositions()  
protected function CalcTranslation()  
function xtr($x_world)  
function ytr($y_world)
```

16.2.A mellékelt CD könyvtárszerkezete

- **[szakdolgozat]**: ez a mappa tartalmazza a szakdolgozatot szerkeszthető (Apple Pages) és PDF formátumban
- **[linkek]**: ez a mappa tartalmazza az irdalomjegyzékben szereplő internetes hivatkozásokat letöltve
- **[program]**: ez a mappa tartalmazza az elkészített szoftver forráskódját és a szükséges egyéb fileokat