

L^AT_EX: rajzolás és ábrázolás

TikZ és pgfplot

Vadon Viktória

2023/24/I. félév

Mi az a TikZ? I

- vektorgrafikus rajzoló \LaTeX -ben
- csomag betöltése
 - `\usepackage{tikz}`
 - könyvtárakkal bővíthető:
 - `\usetikzlibrary{konyvtar1,konyvtar2}`
 - dokumentáció (link) – pgf, TikZ, és könyvtárak is
- rajz
 - `\tikz{}` parancsban
 - vagy `\begin{tikzpicture}` környezetben
 - ezeken belül \LaTeX -étől különböző, TikZ szintaxis!
 - de fordítás szokásos \LaTeX compilerrel, nem kell segédprogram
 - beamer-ben együttműködik az időzítés/overlay parancsokkal, animálható rajzok

Mi az a TikZ? II

- színfalak mögött:
- pgf
 - portable graphics format
 - egy grafikus (rajzoló) nyelv
 - \TeX -hez hasonlóan egy alacsony szintű nyelv
 - ő a „motor” vagy „backend”
- TikZ
 - \LaTeX -hez hasonlóan egy magasabb szintű jelölő nyelv
 - „user interface” vagy „frontend” a pgf-hez

Vászon, koordináták I

- a TikZ egy vektorgrafikus rajzoló, koordináták alapján dolgozik
- szokásos Descartes-féle koordináta-rendszer
 - x tengely jobbra, y tengely fel
 - szögek az x tengelytől indulva, óramutatóval ellentétesen, fokban mérve
 - használhatók Descartes-féle vagy polár koordináták, *akár vegyesen is*
- a koordinátázás, origó elhelyezése tetszőleges
- a kész képet vágja körbe, a koordinátáktól függetlenül

Példa, szintaxis I

- TikZ rajz alapegysége: *path* (útvonal)

```
\tikz{ \draw (0,0) -- (1,0) (0,1) -- (1,1)
(2,0.5) circle [radius=0.5] ; }
```



- **pirossal**: path kezdetén egy „rajz mód” *parancs* (ld. path típusa és funkciói, 5 szakasz), végén ; (pontosvessző)
- **zölddel**: koordináták, amiket egymás után bejár – útvonal/path, erről kapja a nevét
- **kékkel**: köztük rajz utasítások
 - ahol hiányzik, ott csak ugrik / rajzolás nélkül mozog, példánkban (1,0) és (0,1) között
 - --: egyenes szakasz húzása
 - circle: adott *középpont* körül kör rajzolása

Példa, szintaxis II

- path-on belül szóközöket, sortöréseket figyelmen kívül hagy, tetszőlegesen tagolható a kód
- egy ábrában több path is lehet
 - kódolás sorrendjében rajzolja őket
 - akár egymásra is, amit később rajzolunk, kitakarhatja a korábban ott lévőt!
- formázási beállítások, mint szín, vonalvastagság, stb., egy-egy path-ra vonatkoznak

Koordináták I

- Descartes koordináta-rendszer: x tengely jobbra, y tengely felfelé
- origó tetszőleges
- alapértelmezett mértékegység: cm, de tetszőleges \LaTeX által ismert hossz használható
- tört és negatív koordináták is
- **Descartes** koordináták: (x, y)
- **polár** koordináták: $(\text{szög}:\text{sugár})$
 - szögeket x tengelytől indulva, óramutatóval ellentétes irányban, fokban mér
- **cycle** kulcsszó: path első koordinátájához visszatér
- példaül

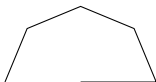
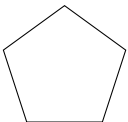
```
\tikz{ \draw (0,0) -- (60:1) -- (1,0) -- cycle ; }
```



Koordináták II

- relatív koordináták
 - `++(koord)`: ahová érkezik, az lesz az új vonatkoztatási pont
 - `+(koord)`: a vonatkoztatási pont marad, ami eddig volt
 - működik Descartes és polár koordinátákkal is
 - path elején kell egy abszolút koordináta
- például

```
\tikz{ \draw (0,0) -- ++(1,0) -- ++(72:1)
-- ++(144:1) -- ++(216:1) -- cycle ;
\draw (3,0) -- +(1,0) -- +(45:1)
-- +(90:1) -- +(135:1) -- +(180:1) ; }
```



Path kezdő parancs – path típusa, funkciója

- `\draw`: körvonal rajzolása
- `\filldraw`: körvonal rajzolása és az alakzat kitöltése
- `\fill`: csak kitöltés
- `\shade`: színátmenetes kitöltés (ld. színeknél, 1 szakaszban)
- `\clip`: körbevágás
 - az *utána* rajzolt path-okra érvényes
 - azaz: a továbbiakban az adott alakzaton kívül nem rajzol semmit
- stb. (de nem kell mindet tudni)
- általános: `\path[]` parancs
 - [] argumentumba a rajzolás módja(i): `draw`, `fill`, `shade`, stb. kulcsszavak kombinációja
 - egyéb opciók is ide kerülnek majd, mint kitöltés színe, vonalstílus, stb.

TikZ ábra és path transzformációi I

- rajz parancs vagy környezet opcionális argumentumával transzformálható a teljes ábra: `\tikz [opciók] {}`,
`\begin{tikzpicture} [opciók]`
 - nagyítás (`\scalebox`-hoz hasonló eredménnyel)
 - nagyítás: pl. 0.5-szörösre `scale=0.5`
 - nyújtás: `\xscale=0.5,yscale=1` x vízszintesen felére, függőlegesen változatlan
- path opcionális argumentumában is alkalmazhatók transzformációk `\path [opciók] ... ;`
 - nagyítás, nyújtás fenti módon `scale=`, `xscale=`, `yscale=`
 - forgatás pl. `rotate=30`: 30 fokkal pozitív szögirányban (óramutatóval ellentétesen)
 - eltolás pl. `xshift=1cm, yshift=2cm` jobbra 1cm, fel 2cm
 - nagyítást alkalmazza *utoljára*, így az eltolást is felnagyítja (!)

2 Vonalak

- Egyenes szakaszok
- Rácsozás
- Görbék
- Kör, ellipszis
- Körív, ellipszisív

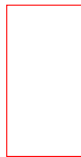
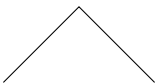
Egyenes szakaszok I

- a következő „parancsok” a két, összekötendő koordináta közé kerülnek
- egyenes szakasz: --, vagy to
 - töröttvonal: (koord) -- (koord) -- (koord), stb.
- gyorsparancs „L” betűhöz/„lépcsőhöz”: vízszintes vonal előbb -|, függőleges vonal előbb |-
 - előbb: a rajzolás/mozgás irányában nézve
- mindkét L betű együtt/téglalap (két átellenes sarok között): rectangle

Egyenes szakaszok II

● példa

```
\begin{tikzpicture}
\draw (0,0) -- (1,1) -- (2,0) ;
\draw (3,0) -| (4,2) ;
\draw (5,0) |- (6,2) ;
\draw (7,0) rectangle (8,2) ;
\end{tikzpicture}
```



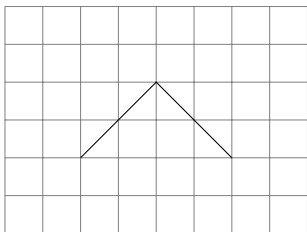
Rácsozás I

- téglalaphoz hasonlóan, két átellenes sarok koordinátája közé:
grid kulcsszó
- hasznos path opciók:
 - `help lines`: halványszürke szín
 - `ultra thin`: vékony vonalak (ld. vonalvastagság 4)
 - osztás, sűrűség: pl. `step=0.5` fél egységként –
alapértelmezés egységnyi (1cm)

Rácsozás II

- például

```
\tikz{ \draw[help lines, ultra thin, step=0.5]  
  (-1,-1) grid (3,2) ;  
  \draw (0,0) -- (1,1) -- (2,0) ; }
```



Görbék I

- egyszerű görbék: (koord) **to** [opciók] (koord)
 - kis hajlítás az egyenes szakaszhoz képest: to [bend right], to [bend left] – jobbra ill. balra, a rajzolás/mozgás irányából nézve
 - megadható a „kitérés szöge”, pl. to [bend right=45]
 - ki-és belépési szög külön megadásával: pl. to [out=30,in=150]
 - itt a szögek abszolút szögirányban értendők (0°: x tengely +, 90°: y tengely +, stb.)
 - kilépés: kezdőpont/korábbi koordináta, belépés: végpont/későbbi koordináta

Görbék II

- Bézier-görbe
 - másod- vagy harmadrendű polinom, aminek érintői a végpontok és kontrollpont(ok) által meghatározott egyenesek
 - másodrendű, egy kontroll ponttal: `(koord) .. controls (koord) .. (koord)`
 - harmadrendű, két kontroll ponttal: `(koord) .. controls (koord) and (koord) .. (koord)`

Kör, ellipszis I

- adott koordinátára mint középpontra illeszthetők, vonatkoztatási pont nem változik
- kör (koord) `circle (sugár)`
- ellipszis (koord) `circle (xsugár, ysugár)`
- ferde ellipszis: `rotate=` path opció
- vagy alternatív szintaxis: (koord) `circle [x radius=1, y radius=2, rotate=30]`, forgatás pozitív irányban és fokban megadva

Körív, ellipszisív I

- kivételes szintaxis
 - kezdőpont az aktuális koordináta
 - kezdő-és végszög, ill. sugár határozza meg
 - a végpontot ezekből ő számolja(!)
 - a végpont lesz az új vonatkoztatási pont
- körív: (koord) `arc (kezd:bef:sugár)`
 - alternatív: (koord) `arc [start angle=kezd, end angle=bef, radius=sugár]`
- ellipszisív: (koord) `arc (kezd:bef:xsugár and ysugár)`
 - alternatív: (koord) `arc [start angle=kezd, end angle=bef, x radius=xsugár, y radius=ysugár]`
- pozitív szögirányban húzza az ívet
 - ha a komplementer ív kell, cseréljük meg a kezdő-és befejezőszöget

- 3 Színek, formázás
 - Színek, kitöltés
 - Kitöltés színátmenettel
 - Kitöltés mintával
 - Vonaltvastagság
 - Nyilak
 - Vonaltstílus
 - Vonalak vége és csatlakozása

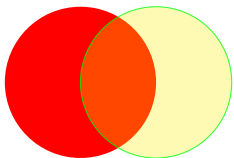
Színek, kitöltés I

- betölti az `xcolor` csomagot, így az alapszínek név szerint használhatók, és a `!`-es színkeverés is; `xcolor` csomag opcióival nagyobb választék
- vonal és/vagy kitöltés színe
 - vonal színe: `\draw[color=red], \draw[red], \path[draw=red]`
 - kitöltés színe: `\fill[color=yellow], \fill[yellow], \path[fill=yellow]`
 - kettő együtt: `\filldraw[draw=red,fill=yellow], \path[draw=red,fill=yellow]`
 - alapértelmezés mindig fekete
- átlátszatlanság: `opacity=0.5` path opció, 0 és 1 közti szám
 - 0: teljesen átlátszó, 1: teljesen átlátszatlan (alapért.)
 - kitöltésre külön átlátszatlansági tényező adható meg: `fill opacity=0.8`

Színek, kitöltés II

- például

```
\tikz{ \fill[red] (0,0) circle (1);  
\filldraw[draw=green,fill=yellow,opacity=0.8,  
fill opacity=0.3] (1,0) circle (1); }
```



Kitöltés színátmenettel I

- kitöltés színátmenettel: `shade (\shade, \shadedraw, vagy \path[shade, ...])`

- alapértelmezés: felül szürke, alul fehér

```
\tikz{ \shade (0,0) rectangle (1,1);}
```



- „egyenes” színátmenet: `shading=axis`
 - alapértelmezett
 - színek megadása: felső `top color=` (alapért. szürke), alsó `bottom color=` (alapért. fehér)
 - opcionális középső szín: `middle color=` – ezt adjuk meg utoljára, különben a másidk két szín felülírja!
 - színátmenet iránya: pl. `shading angle=45`

Kitöltés színátmenettel II

- például

```
\tikz{ \shade[top color=red, bottom color=green,
middle color=yellow, shading angle=30]
(0,0) rectangle (1,1);}
```



- körkörös színátmenet: shading=radial

- színek megadása: belső inner color= (alapért. szürke), külső outer color= (alapért. fehér)

- például

```
\tikz{\shade[shading=radial, outer color=red,
inner color=yellow] (0,0) rectangle (1,1);}
```



Kitöltés színátmenettel III

- gömb árnyékolás: `shading=ball`

- szín: `ball color=` (alapért. kék)
- például:

```
\tikz{\shade[shading=ball, ball color=red]  
(0,0) circle (0.5);}
```



- történetesen beamer `itemize` jele ezzel készül :)

Kitöltés mintával I

- `\usetikzlibrary{patterns}` könyvtár kell hozzá
- `\path[pattern=név]`, vagy `\pattern`
 - például `dots`, `horizontal lines`, `grid`, `fivepointed stars`, `bricks` – a listát ld. TikZ dokumentációban (26. szakasz: Pattern library)
- szín választása: `pattern color=`
- nem működik `fill`-lel együtt, felülírják egymást
 - háttérszínhez rajzoljuk le az alakzatot először kitöltőszínnel, majd a `pattern`-nel
 - ehhez hasznos (lehet) saját makróként definiálni az alakzatot

Kitöltés mintával II

- például

```
\begin{tikzpicture}
\newcommand{\utvonalam}{(0,0) rectangle (1,1)}
\fill[red] \utvonalam ;
\pattern[pattern=fivepointed stars,
pattern color=white] \utvonalam ;
\end{tikzpicture}
```



Vonalvastagság I

- az utasítások mindig az aktuális path utasítás *egészére* vonatkoznak
- vonalvastagság
 - kulcsszavakkal előre definiált vastagságok
 - normálnál vékonyabb: `thin`, `very thin`, `ultra thin`
 - normálnál vastagabb: `semithick`, `thick`, `very thick`, `ultra thick`
 - egyéni: `line width=`, mértékegység nélkül pt-ben, de bármilyen \LaTeX hossz mérték érvényes

Nyilak I

- nyilak

- a *teljes töröttvonal* legelejére és legvégére helyezi el őket
kivéve: `edge` (ld. lentebb)
- vizuálisan: `draw` utasításba balnyíl-jobbnyíl
- balnyíl és jobbnyíl helyére szimbólum vagy kulcsszó: `<`, `>`, `|`,
`stealth`, `latex`
- például

```
\tikz{\draw[<-] (0,0) -- (1,0);  
\draw[>-stealth] (0,-0.5) -- (1,-0.5);  
\draw[|->>] (0,-1) -- (1,-1);}
```

←

↘→

┆→→

- `\usetikzlibrary{arrows}` könyvtárral sok extra nyílhegy, ld.
TikZ dokumentáció (18. szakasz: Arrow tip library)

Nyilak II

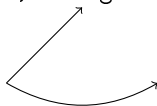
- például `\tikz{\draw[<->] (1,0) - (0,0) - (0,1); }`



- **edge utasítás:**

- edge ekvivalens a to-val (aminek speciális esete --), de *nem része a fő path-nak*
- megrajzolása után visszaugrik a kezdőpontjába(!)
- előny: külön kaphat nyilat
- hátrány: mikor kitöltéshez határozzuk meg az alakzatot, az edge-el húzott vonalak nem számítanak
- to egyéb opcióival együtt használható
- például `\tikz{\draw (0,0) edge[->] (1,1) edge[->,`

`bend right] (2,0);}`



Vonalstílus I

- vonalstílus, mint szaggatás és társai, `\draw` utasításba kulcsszóként
 - `solid`: folytonos (alapért.)
 - `dashed`: szaggatott
 - `dotted`: pontozott; ritkább `loosely dotted`, sűrűbb `densely dotted`
 - `dashdotted`: pontvonal
 - `double`: dupla vonal
 - `double=szín`: két vonal közti kitöltés színe (alapért. fehér)
 - `double distance=`: két vonal közti távolság (alapért. 0.6pt)

Vonalstílus II

- `\usetikzlibrary{snakes}` könyvtárral `snake=`, „kígyók”: hullámos vonal, cikk-cakk vonal, szimbólumsorozat vonal helyett, stb.
 - például `snake=snake` hullám, `=zigzag` cikkcakk, `=coil` „tekercs”, `=triangles` háromszögek sorozata, `=expanding waves` „erősödő hullámok”, stb.
 - teljes lista: ld. TikZ dokumentáció (31. szakasz, Snakes könyvtár)

Vonalak vége és csatlakozása I

- vastag vonaloknál látszik a különbség
- vonalak vége `line cap=`
 - `rect` (alapért.), `round` (lekerekítés), `butt` (hamarabb lecsapás)
- vonalak találkozása `line join=`
 - `miter` (alapért., vonalak végighúzása), `round` (lekerekítés), `bevel` (sarok lecsapása)
- sarok helyett körív: `rounded corners`
 - `rounded corners=hossz:` adott sugár használata, mértékegység nélkül `pt`, de bármilyen \LaTeX hossz érvényes

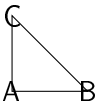
- 4 Node: címkék, csúcsok
 - Címkézés
 - Címke formázása
 - Címke vagy koordináta mentése

Címkézés I

- `node {címke}` kulcsszóval címke, szöveg helyezhető el az ábrába
 - adott koordinátahoz/csúcshoz
 - vonal mentén is

- például, adott koordinátára helyezve:

```
\tikz{ \draw (0,0) node {A} -- (1,0) node {B}  
-- (0,1) node {C} -- cycle; }
```

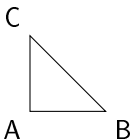


- figyeljük meg: alapértelmezésben a szöveg közepét illeszti a koordinátára

Címkézés II


- opciók megadása: `node [opciók] {címke}`
- elhelyezés: `above`, `below`, `left`, `right` kulcsszavak a `node` opciói közé
 - kombinálhatók, de csak függőleges vízszintes sorrendben, pl. `above right`
 - `auto`: TikZ megpróbálja kitalálni, hol néz ki jól
 - `swap`: második címke, tükrözött pozícióban
 - például

```
\tikz{ \draw (0,0) node [below left] {A}  
-- (1,0) node [below right] {B}  
-- (0,1) node [above left] {C} -- cycle; }
```



Címkézés III

- vonal-rajz utasítás után elhelyezve a vonal is címkézhető
- például `\tikz{\draw (0,0) to[bend left] node{hello}`

`(2,1);}` 

- alapértelmezésben a vonal felezőpontjához illeszt
 - osztóponthoz igazítás, pl. `pos=0.8` (0 és 1 közti szám)
- ehhez képest is érvényesek az `above`, `below`, stb. kulcsszavak
- `sloped` opcióval érintőirányba forgatja a szöveget
- például `\tikz{ \draw (0,0) to[bend left] node[pos=0.67,above,sloped]{hello} (2,1);}`



Címke formázása I

- címke további formázási opciói node `[opciók] {címke}`
 - vagy: `\tikz` parancsnak vagy `path`-nak adott opciók is öröklőd(het)nek
- igazítás `align=left`, vagy `=right`, `=center` (bal, jobb, közép)
 - ez engedélyezi a címkében a sortörést `\\`
- szöveg színe `text=`
- `draw` opcióval keretezi a befoglalót
 - `draw=szín`: keret színe
- a körvonalra érvényesek a vonal beállításai, vastagság, stílus, sarkok lekerekítése, stb.
- hasonlóképp `fill`, vagy `fill=color` opcióval kitölthető

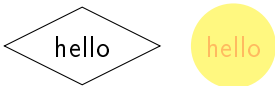
Címke formázása II

- átlátszatlanság: a színeknél látott `opacity`, `fill opacity` is érvényesek
 - a címke szövegére a `fill opacity` érvényes(!)
- befoglaló formája `shape=név`
 - téglalap `rectangle` (alapért.), vagy kör `circle`
 - `\usetikzlibrary{shapes}` használatával: négyzet `square`, ellipszis `ellipse`, rombusz `diamond`, csillag `star`, szabályos sokszög `regular polygon`, X/áthúzás `cross out`, stb.
 - teljes lista: ld. TikZ dokumentáció (30. szakasz, `Shapes library`)
 - rombusz esetén `aspect=arány` adja az x:y tengelyarányt (alapért. 1)
 - szabályos sokszög esetén kötelező `regular polygon sides=oldalszám`

Címke formázása III

- példa

```
\tikz{\draw (0,0)
node [draw,shape=diamond,aspect=2] {hello} ;
\draw (2,0) node[fill=yellow,text=red,
opacity=0.5,shape=circle] {hello};}
```



- méretezés:
- inner sep: szöveg távolsága a befoglalótól
- outer sep: befoglaló távolsága a horgonyponttól – above, right, stb. használata esetén
- minimum width, minimum height, minimum size: minimális szélesség, magasság, ill. mindkettő

Címke vagy koordináta mentése I

- miért?
 - sokszor használt pontra egyszerűbb egy névvel hivatkozni (főleg, ha nem tudjuk az abszolút koordinátáját)
 - gráf, folyamatábra csúcsát előre formázva és mentve a keretéhez fognak futni az élek és nyilak(!)
- (dekorált szöveg)címke mentése:
 - `(koord) node [opciók] (név) {címke}`,
 - vagy `node [opciók] (név) at (koord) {címke}`
 - *egyetlen* node-ból álló path parancsa `\node`
- használat: `(név)`, bárhol egy `(koord)` helyett
 - TikZ okosan a befoglalótól húzza a vonalakat(!)

Címke vagy koordináta mentése II

- például

```
\tikz{\path (0,0) node[circle,draw](A) {A}  
(1,0) node[circle,draw](B) {B};  
\node [circle,draw](C) at (3,0) {C};  
\draw (A) -- (B) to[bend right] (C);}
```



Címke vagy koordináta mentése III

- ha szöveg címkét nem szeretnénk, de valamit meg akarunk jeleníteni: a szöveg argumentuma maradjon üresen
- például

```
\tikz{\draw node[circle,fill=red,
minimum size=3pt](A) at (0,0){}
node[circle,fill=blue,
minimum size=3pt](B) at (1,0){};
\draw (A) -- (B);}
```



Címke vagy koordináta mentése IV

- „csupasz” mentett koordináta
 - nem jelenítünk meg semmit, de név szerint akarunk hivatkozni rá
 - `(koord) coordinate (név)`,
 - vagy `coordinate (név) at (koord)`
 - ekvivalens: `node[shape=coordinate](név) at (koord){}` – a `coordinate` mint `shape` nem jelenít meg *semmit*
 - *egyetlen* coordinate-ből álló path parancsa: `\coordinate`
- például

```
\tikz{\path (0,0) coordinate(A)
(1,0) coordinate(B) ++(60:1) coordinate(C);
\fill[yellow] (A) -- (B) -- (C) -- cycle; }
```



- 5 Plot: függvényábrázolás
 - Szintaxis
 - Koordináták külső fájlból
 - Képlettel megadott függvény
 - Függvényképlet szintaxisa
 - Plot mintavételezési opciói
 - Plot formázási opciói

Szintaxis I

- `plot` kulcsszó `path`-on belül használható – akár megkezdett töröttvonalat folytat egy függvénygörbével
- formázás nagy részét a `path` opcióiból öröklí
- ábrázolás forrása lehet:
 - `plot [opciók] coordinates {(koord) ... (koord)}`: helyileg adott koordináták listája – ha csak néhány pontot ábrázolunk
 - `plot [opciók] file {elérési/út/fájlnev}`: külső fájlból koordináta lista
 - `plot [opciók] függvényképlet`: formulával megadott függvény – a `pgf` matematikai motorja kiszámolja a szükséges pontokat
 - `plot [opciók] function {formula}`: `gnuplot` külső programnak átadja a formulát, az generálja a pontokat – ez nem órai anyag
 - lentebb részletezzük őket!

Ábrázolás külső fájlból I

- `plot [opciók] file {elérési/út/fájlnév}`
- a fájlban: `%` vagy `#` jellel kezdődő sorokat nem dolgoz fel
- egyébként soronként, szóközzel elválasztva ábrázolandó pontok `x` és `y` koordinátái
 - a sor folytatódhat, azt már nem dolgozza fel
- gnuplot program történetesen ilyen outputot ad

Képlettel megadott függvény I

- `plot [opciók] függvényképlet`
 - itt: gyors áttekintő; részletesen: dokumentáció 16.5 szakasz, `Plotting a function`
- változók használata
 - csak *egy változót* kezel, de függvényt, 2D vagy 3D paraméteres görbét is tud rajzolni
 - változó: makró, pl. `\x`, vagy `\t`
 - precízen: `(xképlet,yképlet)` – tipikusan `xképlet` csak `\x` (ha nem paraméteres görbe)
 - vagy `(xképlet,yképlet,zképlet)`
- képlet szintaxisa: ld. 4. szakasz
- formázás: szintén lentebb
- bevezető példa:

Függvényképlet szintaxisa I

- függvényképlet szintaxisa
- pgf matematikai motorját használja, itt gyors áttekintő az műveletekről és függvényekről
 - részletesen ld. TikZ dokumentáció 39.2 szakasz: Syntax for mathematical expressions
- alpműveletek $+$, $-$, $*$, $/$
- zárójelek, műveleti sorrend
- hatványozás $\text{pow}(x, y)$ – csak y egész számra
 - elviekben x^y -t is érti, de előjelet téveszt...
- gyök $\text{sqrt}(x)$
- abszolút érték $\text{abs}(x)$
- modulo, maradék: $\text{mod}(a, b)$
- kerekítés $\text{round}(x)$, $\text{floor}(x)$, $\text{ceil}(x)$

Függvényképlet szintaxisa II

- exponenciális $\exp(x)$
- $\min(x, y)$, $\max(x, y)$
- $\text{veclen}(x, y)$: (x, y) vektor hossza/pont távolsága az origótól
- pi : π konstans
- trigonometrikus függvények: $\sin(x)$, $\cos(x)$, $\tan(x)$ – fokban várja a szöveget
 - konverzió radiánból fokba $\text{deg}(x)$, vagy röviden $x \ r$ (ez utóbbi szorzást és osztást elvégez, más műveleteket nem)
- inverz trigonometrikus függvények $\text{asin}(x)$, $\text{acos}(x)$, $\text{atan}(x)$ – fokban adja vissza a szöveget
 - konverzió fokból radiánba $\text{rad}(x)$
- (pseUDO)véletlen számok: rnd 0 és 1 között, rand -1 és 1 között

Plot mintavételezési opciói

- **mintavételezési opciók**
- `samples=n`: hány ábrázolandó pontot vegyen fel, alapért. 25
- `domain=min:max`: ábrázolandó tartomány, alapért. -5:5
- `variable=\x` változó neve/makrója, alapértelmezésben `\x`
 - paraméteres görbénél érdemes lehet átnevezni
- `samples at={lista}`: ábrázolandó `\x` értékek listája
 - felülírja a `samples` és `domain` kulcsokat, és vice versa!

Plot formázási opciói I

- ábrázolás „módja”
- sharp plot (alapért.), pontok összekötése egyenes szakaszokkal
- smooth: folytonos görbét fektet a pontokon keresztül
 - tension=0.55: simítás mértéke – minél alacsonyabb, annál kevésbé simít, alapért. 0.55
 - smooth cycle: simított, *zárt görbét* fektet a pontokra
- ycomb: minden pontot az x tengellyel köt össze egy függőleges vonal
 - vonalvastagsággal játszva oszlopdiagram
 - analóg: xcomb
- polar comb: minden pontot az origóval köt össze egy szakasz
- only marks: csak a pontok/markerek kirajzolása; „scatterplot”

Plot formázási opciói II

- ábrázolt pontok jelölése
- `mark=marker`, ahol `marker` lehet szimbólum vagy kulcsszó
- beépítve: `*`, `+`, `x` – pont, plusz és kereszt
- `\usetikzlibrary{plotmarks}` könyvtárral sok másik, pl. `o`, `star`, `diamond`, `square`, `triangle`
- teljes lista: ld. TikZ dokumentáció 29. szakasz: `Plot mark library`
- `mark size=hossz`: a marker „sugara”, tetszőleges \LaTeX hosszmértékben
- `mark options={opciók}`, pl. szín, elforgatás megadható
- egyéb dekoráció
- tengelyek, rácsozás, feliratozás: már tanult TikZ funkcionalitás