

# Bevezetés a számítástechnikába

## L<sup>A</sup>T<sub>E</sub>X- 2

Siklósi Bálint

Pázmány Péter Katolikus Egyetem - Információs Technológiai és Bionikai Kar

*siklosi.balint@itk.ppke.hu*

2020. november 24-26.

1 Matematikai kifejezések

2 Ábrák

3 Táblázatok

4 Utalások

5 Kódmegjelenítés

6 Diasor

# Matematikai módok

Ismétlés:

- sorközi: `\(...\)` vagy `$...$`
- számozatlan: `\[...\]` vagy `\begin{equation*}...\end{equation*}`

Új:

- számozott: `\begin{equation}...\end{equation}`

```
\begin{equation}
  \sum_{i=0}^{\infty}
  \sqrt[5]{a^{(i+1)^2}_3}
\end{equation}
```

$$\sum_{i=0}^{\infty} \sqrt[5]{a_3^{(i+1)^2}} \quad (1)$$

# Több soros egyenletek és igazítás - split

```
\begin{equation}
  a = b+c-d\\
    = e-f\\
    = g
\end{equation}
```

$$\begin{aligned}
 a &= b + c - d \\
 &= e-f \\
 &= g \quad (2)
 \end{aligned}$$

# Több soros egyenletek és igazítás - split

```
\begin{equation}
  a = b+c-d\\
    = e-f\\
    = g
\end{equation}
```

$$\begin{aligned}
 a &= b + c - d \\
 &= e-f \\
 &= g \quad (2)
 \end{aligned}$$

```
\begin{equation}
  \begin{split}
    a \&= b+c-d\\
      \&= e-f\\
    \&= g
  \end{split}
\end{equation}
```

$$\begin{aligned}
 a &= b + c - d \\
 &= e - f \\
 &= g
 \end{aligned} \tag{3}$$

# Több soros egyenletek és igazítás - split

## Magyarázat:

- `\begin{split}...\end{split}`  
a split környezet, amiben már lehet új sort kezdetni és a sorokat egymáshoz igazítani, és az egész matematikai kifejezésnek csak egyetlen címkéje (sorszám) lesz
- `\\`  
új sor kezdése
- `&`  
ezek jelölik az igazítási pontokat, amelyek egymás alá kerülnek

# Több soros egyenletek és igazítás - align

```
\begin{align}
```

```
  a_{11} &= b_{11} &
```

```
  a_{12} &= b_{12} \\
```

```
  a_{21} &= b_{21} + c_{21} &
```

```
  a_{22} &= b_{22}
```

```
\end{align}
```

$$a_{11} = b_{11} \qquad a_{12} = b_{12} \quad (4)$$

$$a_{21} = b_{21} + c_{21} \quad a_{22} = b_{22} \quad (5)$$

Magyarázat: az `align` környezet lehetőséget ad egyenletek folyamatos számozására és egymáshoz igazítására, soronkénti számozással

# Szöveg beszúrása matematikai módban

Ha szöveget akarunk beilleszteni a képletbe, akkor a `\text{...}`-et kell használnunk, különben a szöveg is dőlt betűvel lesz szedve.

$$\text{az } e^{i\pi} - 1 = 0 \text{ egyenlet alapján} \quad (6)$$

```
\begin{equation}
  \text{az } e^{i\pi} - 1 = 0 \text{ egyenlet } \text{alapjan}
\end{equation}
```



# Zárójelezés

A következő táblázatban láthatók a különböző zárójelező operátorok és a közrefogott képlethez viszonyított magasságuk:

jel	( )	\left( \right)		
eredmény	$(a)(\sum_{i=0}^{\infty})$	$(a)\left(\sum_{i=0}^{\infty}\right)$		
jel	\bigl( \bigr)	\Bigl( \Bigr)	\biggl( \biggr)	\Biggl( \Biggr)
eredmény	$(a)(\sum_{i=0}^{\infty})$	$(a)(\sum_{i=0}^{\infty})$	$(a)(\sum_{i=0}^{\infty})$	$(a)(\sum_{i=0}^{\infty})$

Általában `\left` és `\right` változatot, vagy a magában álló zárójelet használjuk.

# Zárójelezés

Ha nem sima zárójelezést használjuk, hanem a `\left \right` operátorokat, akkor kötelező kitenni *mindkettőt*. Amennyiben az egyiket elhagyjuk, fordítási hibát kapunk.

Természetesen, ha egy zárójelnek "nincs párja" és a `\left, \right` konstrukciót szeretnénk használni, akkor a kihagyandó zárójel helyére az operátor után `.-ot` kell tennünk. Itt pl. `\left. és \right\}`-ot használunk:

$$\left. \begin{aligned} B' &= -\partial \times E \\ E' &= \partial \times B - 4\pi j \end{aligned} \right\} \text{Maxwell's equations} \quad (7)$$

# Betűtípusok

Ahol csak lehet, kerüljük ezek használatát, de ha mégis szükséges, a következő parancsokkal lehet a betűtípust megváltoztatni:

- félkövér betűk, szimbólumok  $\backslash\mathrm{ABC}\mathrm{Rabc}\backslash\pi$   
 $\mathrm{ABC}\mathrm{Rabc}\pi$
- kalligrafikus betűk (csak nagy)  $\backslash\mathrm{mathcal}\{\mathrm{ABC}\mathrm{Rabc}\backslash\pi$   
 $\mathcal{ABC}\mathcal{R}\neg\lfloor\rfloor\pi$
- kettős húzott betűk (csak nagy)  $\backslash\mathrm{mathbb}\{\mathrm{ABC}\mathrm{Rabc}\backslash\pi$   
 $\mathbb{ABC}\mathbb{R}\ni\pi$
- gótikus hatású betűk  $\backslash\mathrm{mathfrak}\{\mathrm{ABC}\mathrm{Rabc}\backslash\pi$   
 $\mathfrak{ABC}\mathfrak{Rabc}\pi$

ahol nincs félkövér a  $\backslash\mathrm{mathbf}\{\}$ -el, ott a  $\backslash\mathrm{boldsymbol}\{\}$ -t használjuk.  
 pl.  $\pi$

# Mátrix környezetek

A következő mátrix környezeteket definiálja az amsmath csomag:

- pmatrix  $()$
- bmatrix  $[]$
- Bmatrix  $\{ \}$
- vmatrix  $| |$
- Vmatrix  $|| ||$

pl.:

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

A mátrix környezetek matematikai módban használandók!



# Ábra környezet

- Ábrák beillesztése mindig problémás: Hova kerüljön a szövegen belül? Mekkora margókat hagyjunk a kép körül? Hogyan érhetjük el, hogy minden kép az egyes oldalakon azonos pozícióba kerüljön?
- $\text{\LaTeX}$  környezetben ezeket a funkciókat a  $\text{\LaTeX}$  fordító kezeli.
- Az ábra környezet segítségével képeket, ábrákat helyezhetünk el a dokumentumban.

# Ábra környezet

Közvetlenül beszúrhatunk PDF, PNG, JPEG formátumú képeket (gyakorlatilag ezek azok, amiket a PDF formátum támogat, mint beágyazható formátumokat) a következő parancsok segítségével:

```
\begin{figure}
  \centering
  \includegraphics{figure.png}
  \caption{Ez egy beszurt kep}
\end{figure}
```

A parancsok működéséhez a `graphicx` csomag betöltése szükséges (`\usepackage`)

# Ábrák elhelyezése oldalon belül

Alapvetően a  $\text{\LaTeX}$  fordítóra van bízva az ábrák elhelyezése, de lehetőségünk van kívánalmakat megfogalmazni:

- h A fordító törekszik a képet ott elhelyezni, ahol azt definiáltuk a dokumentumban, ha talál megfelelő méretű helyet az oldalon
- t Az adott (vagy azt követő) oldal *tetején* helyezi el az ábrát.
- b Az adott (vagy azt követő) oldal *alján* helyezi el az ábrát.
- p Olyan oldalon helyezi el, ahol nincs szöveg, csak további ábrák.
- ! Még nyomatékosabban kérjük a  $\text{\LaTeX}$ et, hogy az általunk megadottak szerint próbálja elhelyezni az ábrát.

Példa:

```
\begin{figure}[!ht]
```



# Ábra méretének meghatározása

- Az `includegraphics` paramétereként lehet megadni, hogy mekkora méretben szeretnénk beilleszteni a képünket.
- Például az `\includegraphics[width=0.48\textwidth]` megadásával a szövegszélesség 48%-a lesz a képszélesség.
- A `width=1in` esetén egy inch szélességűre állítjuk a képszélességet. (Használható cm, vagy mm is a méret megadásához.)
- Az `\includegraphics[scale=0.5]` parancs segítségével nagyíthatunk/kicsinyíthetünk. (Ebben a példában a felére kicsinyítünk.)
- A `width` mellett a `height` is használható a képméret megadására. Mindkét esetben a nem megadott oldal méretét a képarányok megőrzésével határozza meg.



# A tabular környezet

Táblázatokat a `tabular` környezettel helyezhetünk el a dokumentumban:

```
\begin{tabular}{|l||rc}
  narancs & k r t e & szilva & \\
  aa & bbbbbb& ccccc & \\
  \hline
  sz l & KV & f g e
\end{tabular}
```

Eredménye:	narancs		körte	szilva
	aa		bbbbbb	cccc
	szőlő		KV	füge

# A table környezet

Ha a táblázatot is szeretnénk címmel ellátni (és a `figure` környezethez hasonlóan az automatikus elhelyezést is igénybe venni), akkor a `table` környezetet kell használni:

```
\begin{table}[h!]  
  \centering  
  \begin{tabular}{|l||rc}  
    narancs & k  rte & szilva  \\  
    aa & bbbbbb& cccccc \\  
    \hline  
    sz l    & KV & f ge  
  \end{tabular}  
  \caption{Az elso tablazatom}  
\end{table}
```

Ebben a környezetben is használhatók a `figure` környezetnél megismert elhelyezéssel kapcsolatos opciók (`h`, `t`, `!`, stb.).



# Címkék

- Ahhoz, hogy egy struktúrális elemre (fejezetre, alfejezetre, stb.), ábrára, táblázatra hivatkozni tudjunk, az adott elemnél el kell helyeznünk a `\label{cimke}` parancsot. A `cimke` bármilyen szöveg lehet, ezt fogjuk használni a későbbi hivatkozás során.
- Fontos, hogy a címkét mindig az egység belsejében, a sorszám növelése után tegyük meg. Például `\section{}` címkézése:  
`\section{Cim}\label{cimke}`
- Ez után a `\ref{cimke}` használatával tudunk hivatkozni.
- Pl.: Ehhez hasonló módszert láthatunk a [23.](#) dián is. (És ez a sorszám akkor is helyesen készül el, ha beszúrunk közben még egy rakás egyéb diát.)

# Címkék (lebegő környezetek)

```
\begin{figure}  
  \centering  
  \includegraphics{figure.png}  
  \caption{Ez egy beszurt kep}  
  \label{cimke}  
\end{figure}
```

A címkézés hasonlóan tehető meg a `table` környezet esetén is.

# aref és Aref

- A `\usepackage[magyar]{babel}` használata esetén használhatóak az `\aref{}` és `\Aref{}` referenciakezelő parancsok is, amelyek a referált szám elé kiteszik a névelőt is (kisbetővel az `\aref{}` esetében és nagybetővel az `\Aref{}` esetében), így nem kell azzal sem foglalkoznunk, hogy a hivatkozott szám elé *a* vagy *az* névelőt kell tenni.
- Gondoljunk bele mi történik, ha beszúrunk egy sorszámozott részt a dokumentumba! Megváltoznak a sorszámok, egyes esetekben *a* névelő kell az helyett vagy fordíva. Az `\aref{}` és `\Aref{}` megoldja ezt a problémát.



# Oldalszámra hivatkozás

A `\ref{}`, `\aref{}`, `\Aref{}` sorszámra hivatkozó parancsok mellett lehetőség van arra is, hogy a hivatkozott elem oldalszámát jelentsük meg, amit a `\pageref{}`, `\apageref{}`, `\Apageref{}` parancsokkal tehetünk meg. A névelős változatok ugyanúgy működnek, mint azt az előzőekben megismertük.

# Irodalomjegyzék

- Ha hivatkozni szeretnénk egy cikkre, (pl.: hetero cikk[1]) akkor keressük meg annak a bibTeX összefoglalóját.
- Pl.: cikk → cite this → BibTeX
- Az összes hivatkozni kívánt cikk BibTeX leírását másoljuk be pl. egy *hivatkozasok.bib* nevű fájlba (Mindegyik bejegyzés első eleme legyen egy leíró azonosító. pl.: *hetero\_cikk*)
- Preambulumba: `\usepackage[style = ieee]{biblatex}`  
`\addbibresource{hivatkozasok.bib}`
- Végül: `\printbibliography`



B. Siklosi, I. Z. Reguly, and G. R. Mudalige, „Heterogeneous cpu-gpu execution of stencil applications,” in *2018 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, pp. 71–80, Nov 2018.



# lstlisting

- `\usepackage{listings}`
- `\begin{lstlisting}[language=Python]...\end{lstlisting}`

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            do_stuff()
    return M
```



# Beamer

- `\documentclass{beamer}`
- `\begin{frame}{Frame Title}...\end{frame}`
- `\only<3>, \pause, \onslide<3->`

**STAY STRONG**



**THE SCHOOL YEAR IS  
ALMOST OVER!**

mememaker.net