

# Szimulációk a statisztikus fizikában (elő)gyakorlat, 2011. szept. 12.

Balogh László, Rácz Éva és Stippinger Marcell  
Honlap: [http://www.phy.bme.hu/~balogh/...](http://www.phy.bme.hu/~balogh/)

# Célkitűzés

- Felfrissítjük a programozásbeli alapokat, a fennálló hiányosságokat csökkentjük
- Az előadáson előforduló elveket/módszereket egyszerű példaprogramokon keresztül gyakoroljuk
- Egyéb hasznos szoftvereket is bemutatunk, melyek a dokumentációt/publikációt segítik.
- Kis HF-ek az otthoni gyakorlás céljára

# Célkitűzés

- Az előbbi pontok egymáshoz viszonyított arányát majd a hallgatói igényeknek megfelelően alakítjuk, hisz a cél egy *hasznos* tárgy megalkotása.
- A tárggyal kapcsolatos észrevételekre, javaslatokra nyitottak vagyunk.

# Elérhetőségek

F2 alagsor, doktoranduszszoba

Tel.: 1-463-2691

- Balogh László: [blaci947@yahoo.co.uk](mailto:blaci947@yahoo.co.uk)
- Rácz Éva: [racze@phy.bme.hu](mailto:racze@phy.bme.hu)
- Stippinger Marcell: [stippingerm@gmail.com](mailto:stippingerm@gmail.com)

# Linux

- Sokan használnak már Linuxot, a többieknek is javasoljuk, mert:
  - Jól működik & ingyenes
  - A gyakorlaton használt programok egyszerűbben kezelhetők/telepíthetők Linux alatt
  - Előbb-utóbb mindenki belefut, aki a pályán marad.

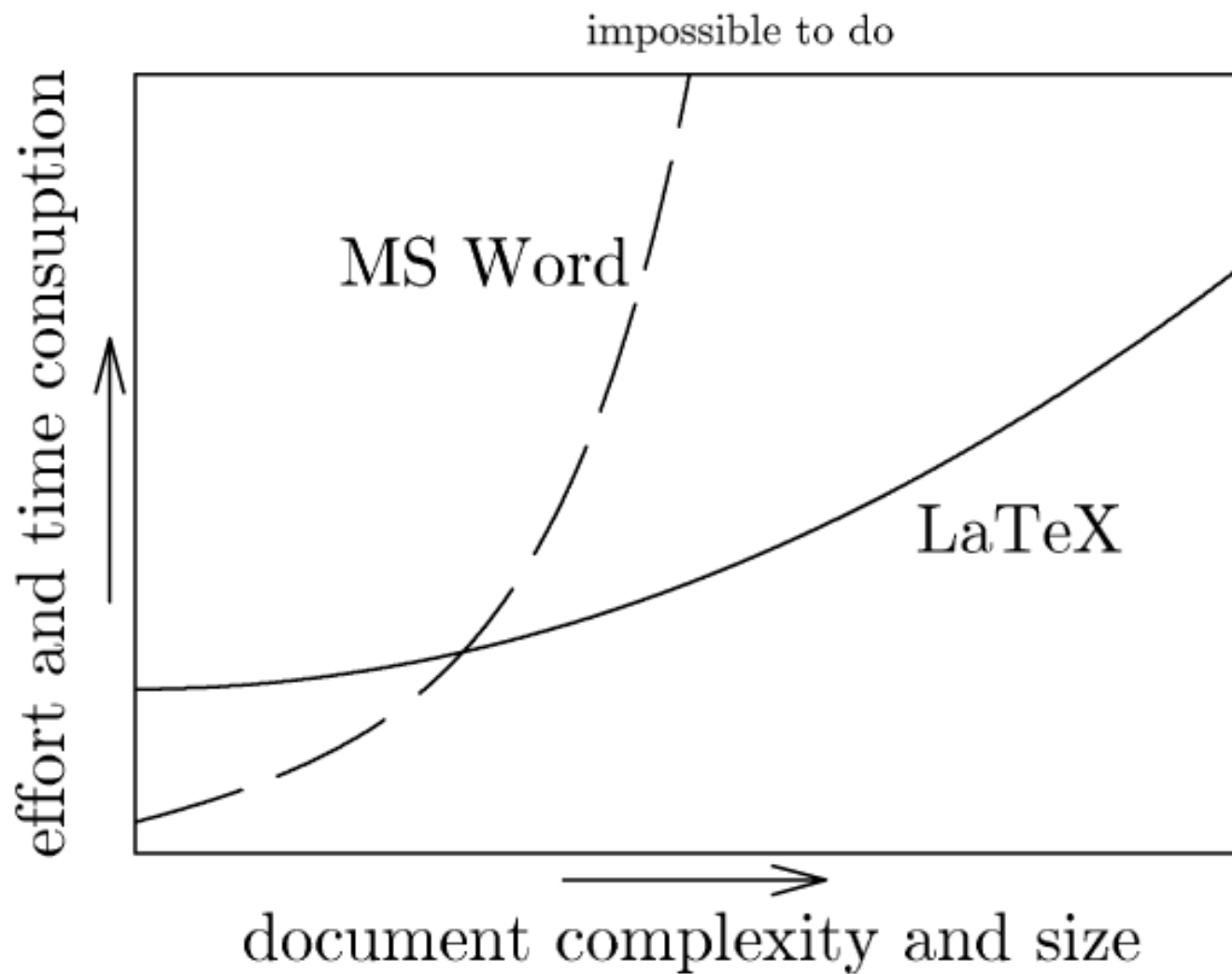
# C-fordító + kiegészítők

- Fordító: a kompatibilitási problémák elkerülése végett a **gcc**-t ajánljuk, Windows alatt ez Dev-C++-szal, MinGW-vel vagy Cygwin-nel oldható meg.
- A gyakorlaton a C-t fogjuk használni, kerüljük a C++ nyelvi elemeket (akit érdekel az objektumorientált programozás, egy jó bevezető <http://www.cplusplus.com/doc/tutorial/> , ezen a honlapon sima C-re is van referencia)
- Debug: **gdb** (<http://www.gnu.org/s/gdb/>)

# C-fordító + kiegészítők

- Memóriakezelési problémák felderítése (és még sok más): **valgrind** (<http://valgrind.org>)
- Automatikus dokumentációgenerálás forráskódból: **doxygen** (<http://www.stack.nl/~dimitri/doxygen/>)
  - példa a kishf-ben
- Forráskód szerkesztése
  - Szövegszerkesztő (pl. Notepad++ vagy Kwrite)
  - IDE (pl. Dev-C++ vagy Eclipse)

# LaTeX+kiegészítők



Forrás: <http://www.pinteric.com/miktex.html>



# LaTeX+kiegészítők

- Latex
  - Linux: texlive\*
  - Windows: MikTeX (<http://miktex.org/>)
  - Hasznos: <http://tobi.oetiker.ch/lshort/lshort.pdf>
- Szerkesztő: bármilyen szövegszerkesztő, vagy latex IDE (l: kile, w: TeXnicCenter)
- Ábrázolás: **gnuplot** (egyszerű szkriptelhetőség, parancssori használat: pl. C-kódon belül is meghívható)

# Kódrészlet 1.

**main.cc:**

```
... /*fejléc*/  
double x[50];  
double y[50];  
FILE* pFile = fopen("adatok.txt", "w");  
... /* hibaellenőrzés, feltöltjük a tömbjeinket*/  
for(i = 0; i < 50; ++i)  
    fprintf(pFile, "%f\t%f\n", x[i], y[i]);  
fclose(pFile);  
system("gnuplot abrazolas.plt");  
... /*program lezárása*/
```

# Kódrészlet 2.

**abrazolas.plt:**

```
# a kimeneti terminál megadása (nagyon sok  
# lehetőség van, ld. gnuploton belül:  
# help terminal)  
  
set terminal postscript enhanced color  
  
# kimeneti fájl beállítása  
set output "kimenet.eps"  
  
# az ábrázolandó grafikon és az ábr. stílus  
# megadása  
plot "adatok.txt" with linespoints title "adatok"  
  
unset output
```

# +Megjegyzések

- Akármiről is legyen szó, érdemes a programot előre megtervezni. Ha azonnal a kódolásnak esünk neki, eredményül nehezen kezelhető, sok szempontból szuboptimális kimenetet kapunk.
- Bármilyen környezetben dolgozunk, a forrás átláthatósága alapvető:
  - Lehet, hogy néhány hónappal később is meg szeretnénk érteni, mit csinálunk ma.
  - Előbb-utóbb többrésztvevős projekteken is fogunk dolgozni, ahol a követhetőség elengedhetetlen.
  - A nagyhf-et is csak korrekt, rendezett kóddal fogadjuk el :)