

# ALiROOT szimulációk GPU alapokon



Nagy Máté Ferenc &  
Barnaföldi Gergely Gábor  
Wigner FK – ALICE Bp csoport

# TARTALOM

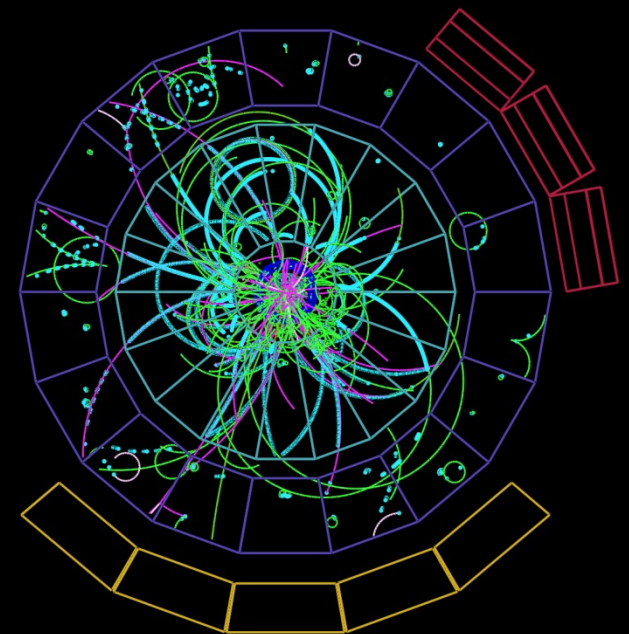
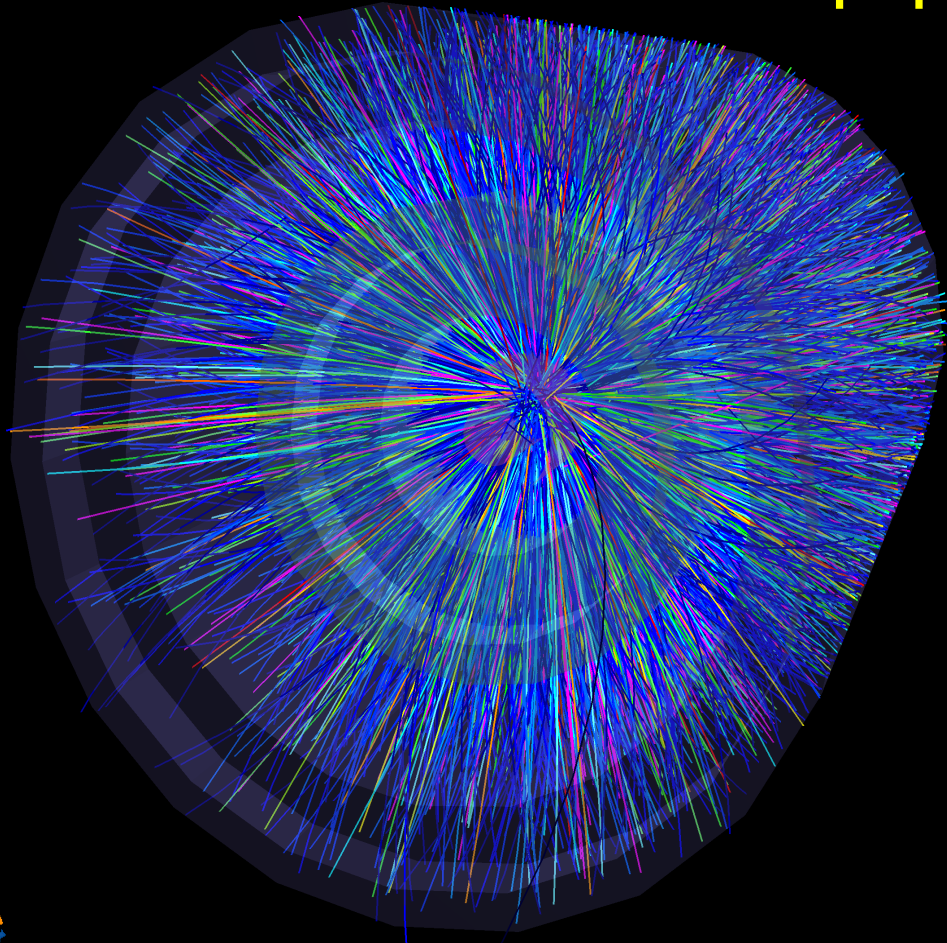
1. Az ALICE csoport és a GRID hálózat
2. Szimulációk és az AliROOT programcsomag
3. Részecskefizikai MC generátorok
4. TRandom3/4/5 benchmark eredmények
5. Proton-proton szimulációk

# ALICE és a GRID



- ALICE egyike a 3 CERN LHC óriáskísérletnek.
- A detektorokból érkező adatok feldolgozásához szuperszámítógép parkra van szükség.
- Olcsóbb és kezelhetőbb a probléma elosztott rendszerrel.
- A kísérlet tagjainak kötelező hozzájárulni a GRID rendszerhez.

# Pb-Pb ütközés vs. p-p esemény.



Azonos eseményszám mellett  
több mint 100X annyi  
valószínűség alapú folyamat.

# Szimulációk & AliROOT

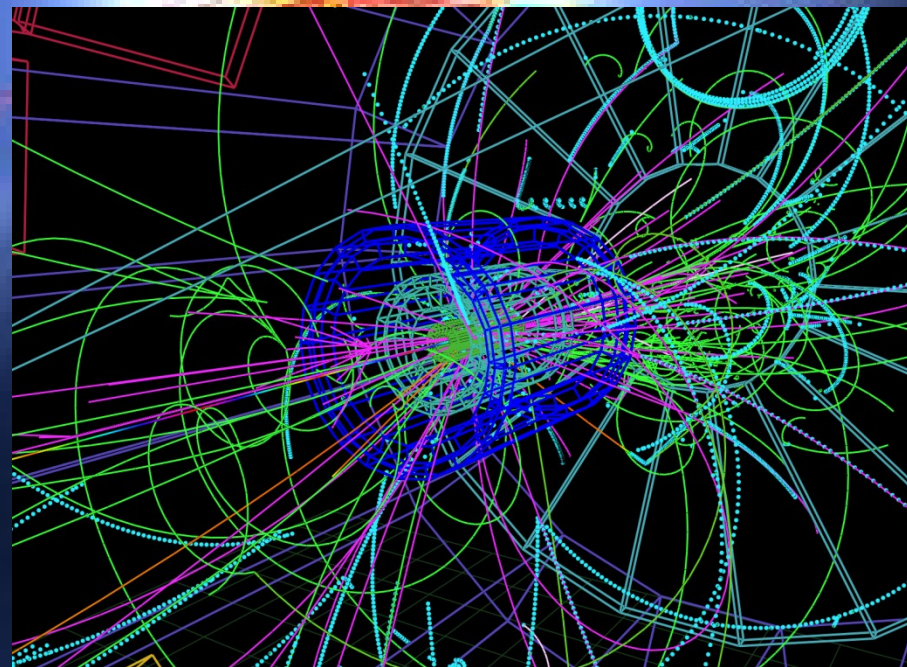
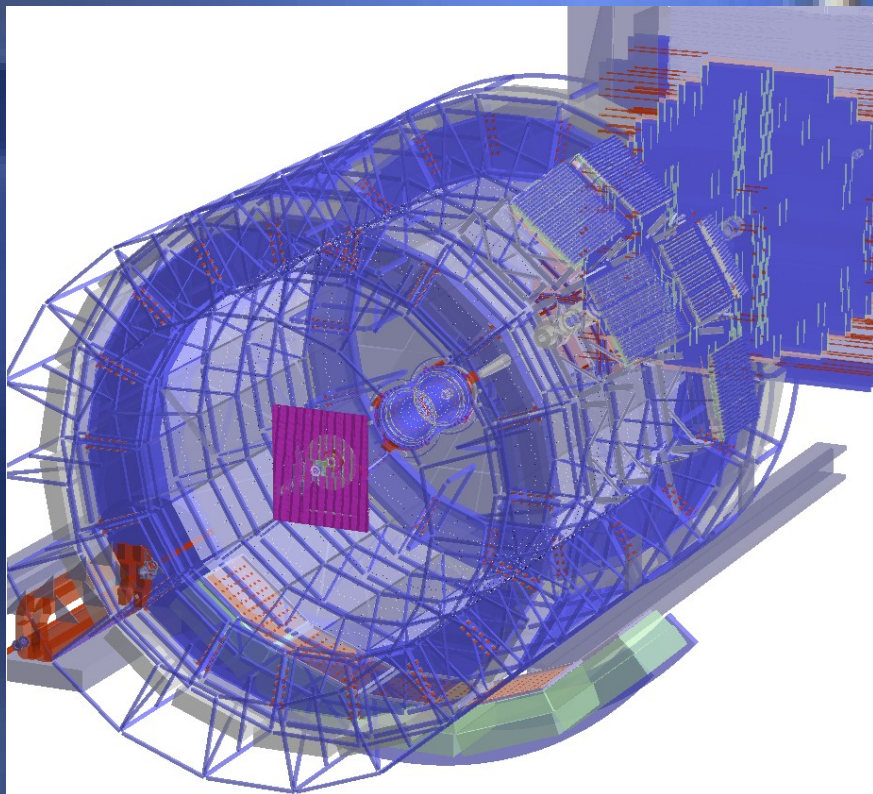
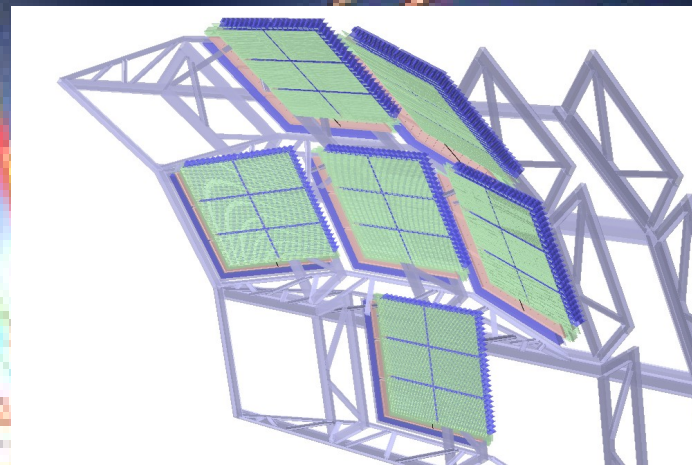
- A használt szimulációs környezet az AliROOT.
- A szimulált eredményeket összevetjük a valós mérési eredményekkel.
- Ha a modellek jóslatai egyeznek a valósággal, elfogadjuk őket.
- Cél: GPU gyorsítás



# ALiROOT

2,7GB-os keretrendszer, több  
eseménygenerátort is magába foglal.

- Mindenkori verziója tartalmazza az aktuális  
ALICE detektor geometriáját.

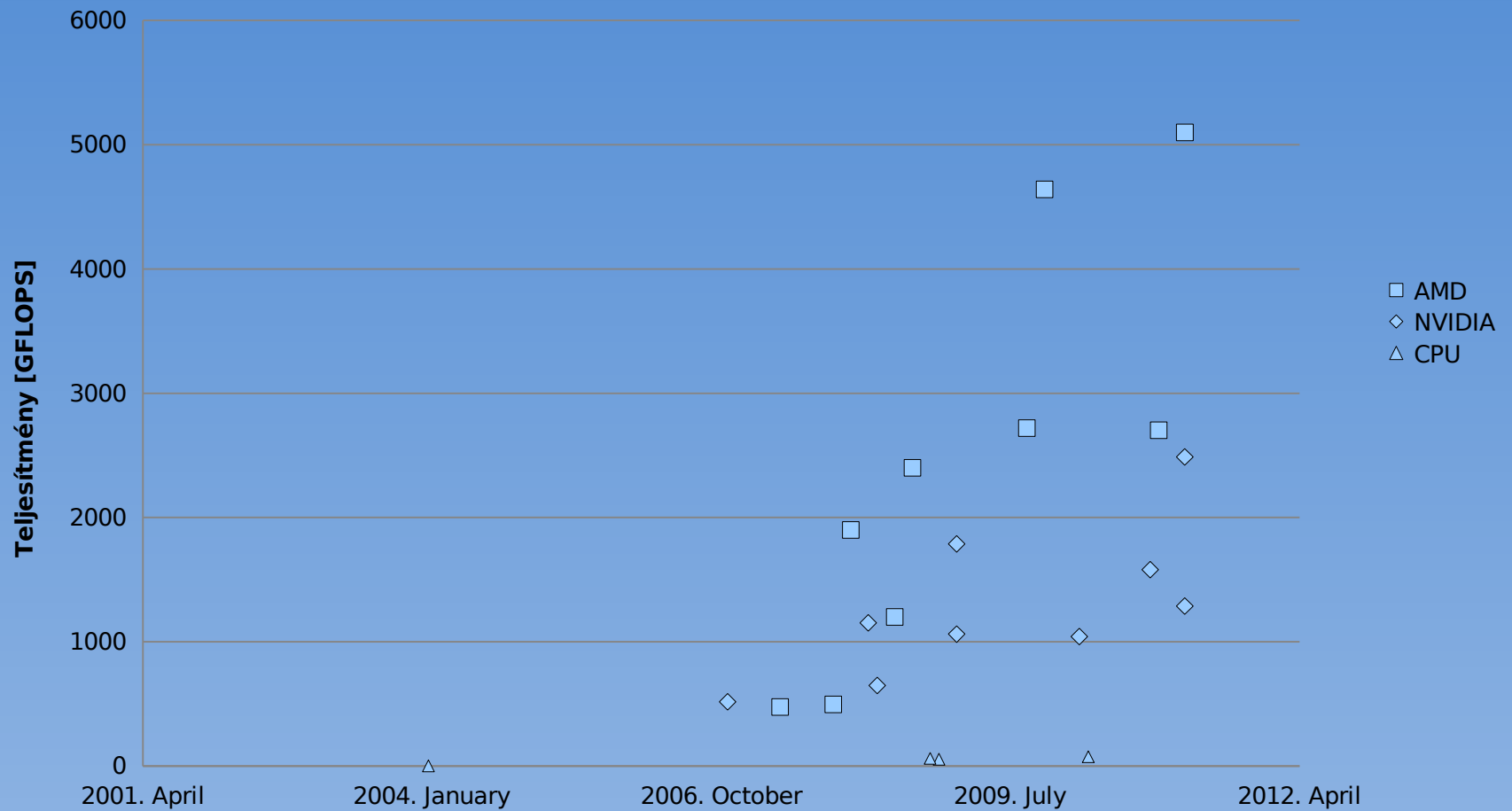


# Részecskefizikai Monte-Carlo generátorok

- Monte-Carlo szimulációnak nevezünk minden olyan eljárást, amelyben egymástól függetlenül generált véletlen számok határozzák meg a rendszer fejlődését, kimenetét.
- Az ilyen fizikai szimulációk eseményeit előállító modult hívjuk MC generátornak.
- Pszeudo random véletlen számok előnyösek a szimuláció reprodukálása miatt.
- A nagy léptékű mintavételezés miatt a gyorsaság centrális fontosságú.

# CPU vs. GPU

## CPU-GPU fejlődés

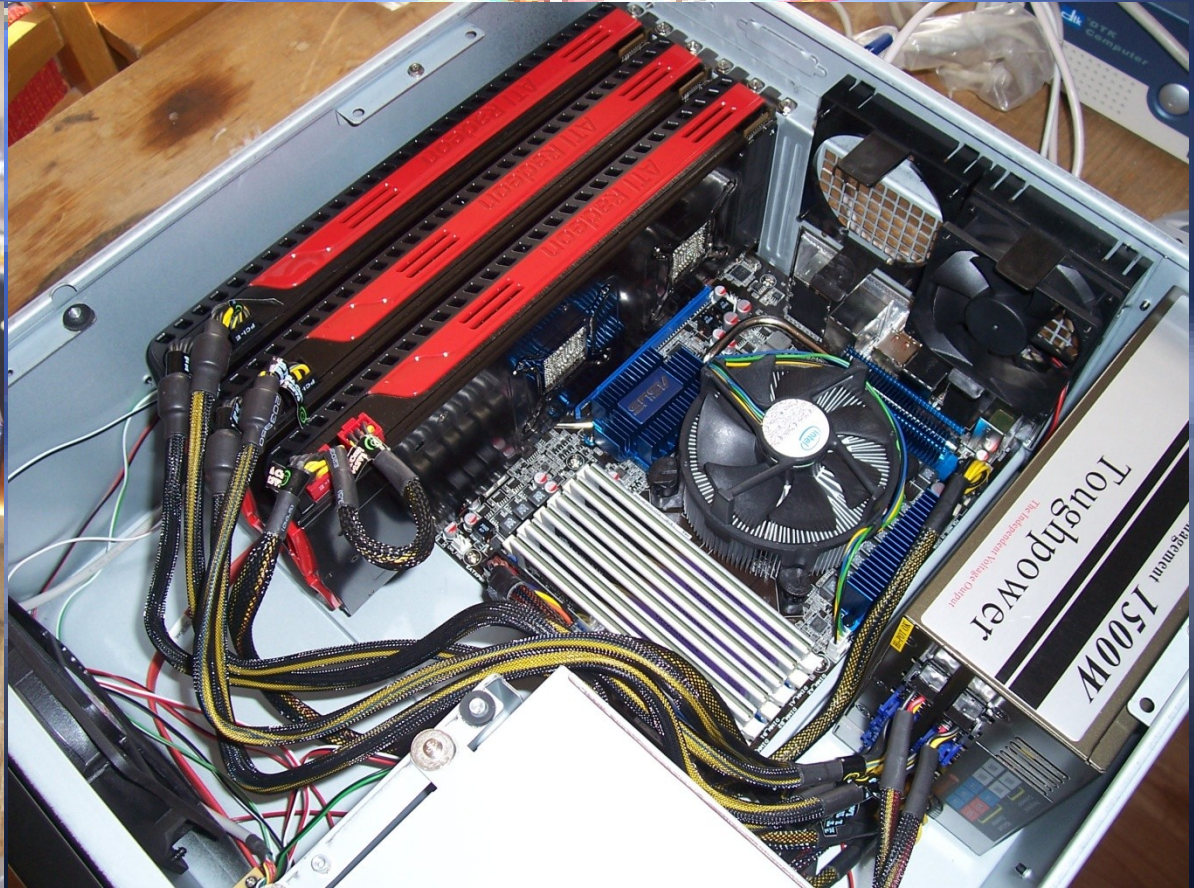




# gpu001

Intel Core-i7 2.66GHz, 12GB DDR3 1333MHz RAM

3 db Radeon 5970, 6GB GDDR5 VRAM (1GB/mag)



# AliROOT gyorsítás

- Cél: PRNG@GPU
- A ROOT matematikai osztálya 3 alap generátort definiál. Ezekből a Mersenne-Twister algoritmust használó TRandom3 a legelterjedtebb.
- GPU alapú TRandom4: Mersenne-Twister
- GPU alapú TRandom5: MWC64X

# TRandom4/TRandom5

- Konstruktor tartalmaz mindent a tényleges szám generálásig:
- Platform lekérdezés
- Context létrehozás
- Device info lekérdezés
- Kernel fordítás
- Command queue létrehozás
- Buffer létrehozás
- Random seedek device-ra küldése
- Szálazonosítók beállítása



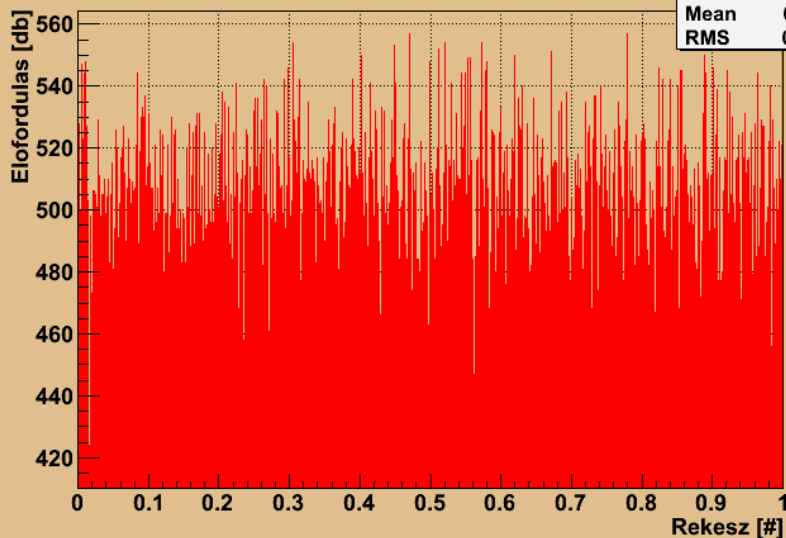
# TRandom4

- A gyorsabb működés érdekében nem lehet egyesével számot generálni.
- Egy RAM-ban lévő buffert tölt fel véletlen számokkal (~500k #). Kérésnél a bufferből másolja ki az eredményt.
- Esetszétválasztás lassítja, aszerint hogy a bufferben maradt-e annyi szám, amennyit kért a program.
- RAM-ból másolás plussz művelet a CPU implementációhoz képest.
- OpenCL miatt akár CPU-n is képes generálni.

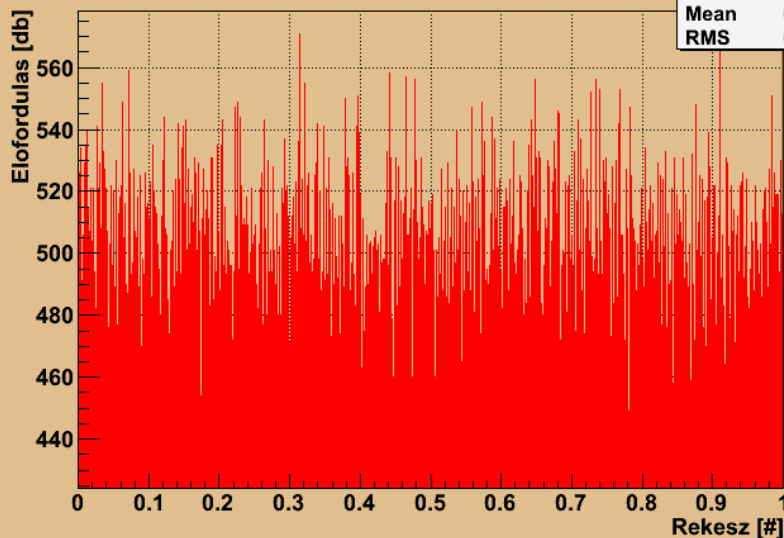


# PRNG eredmények

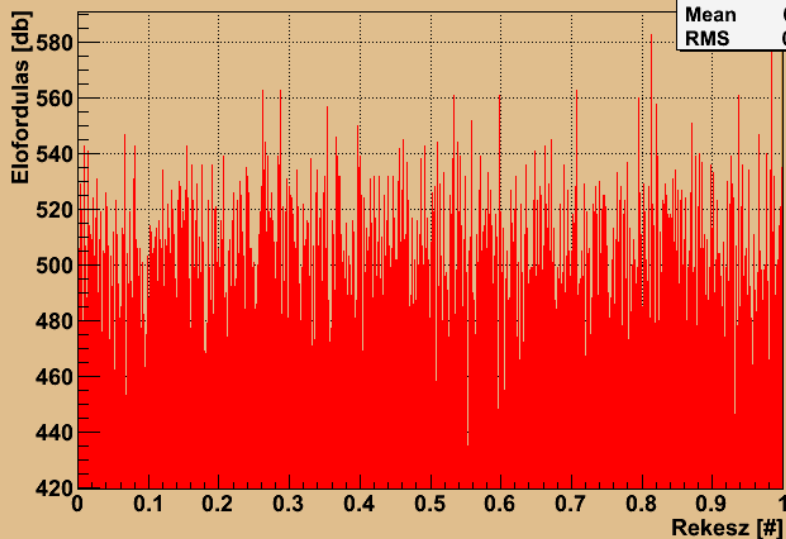
TRandom1 hisztogram



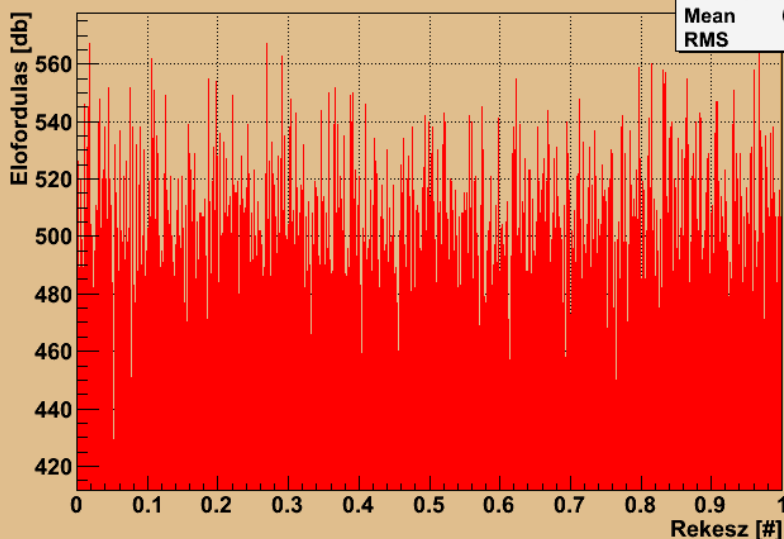
TRandom2 hisztogram



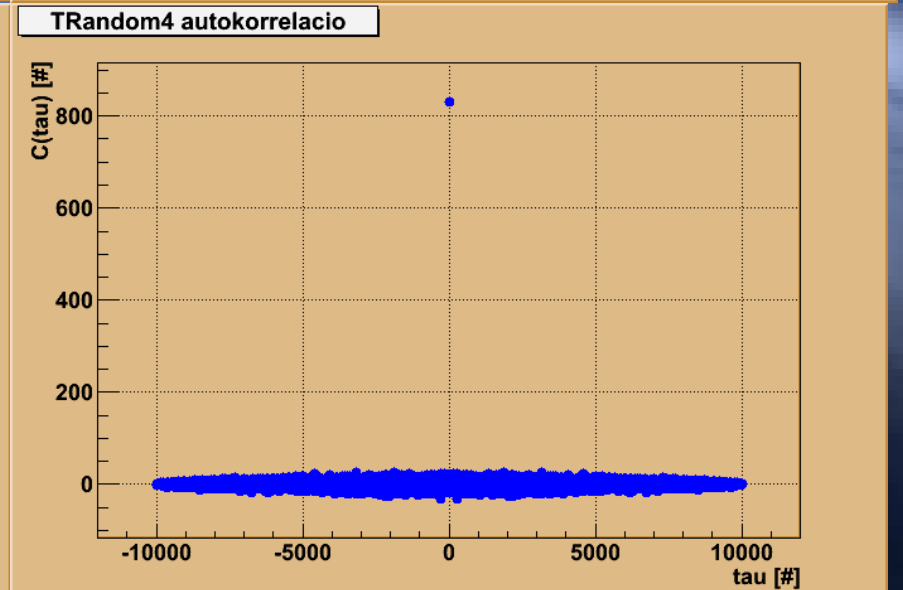
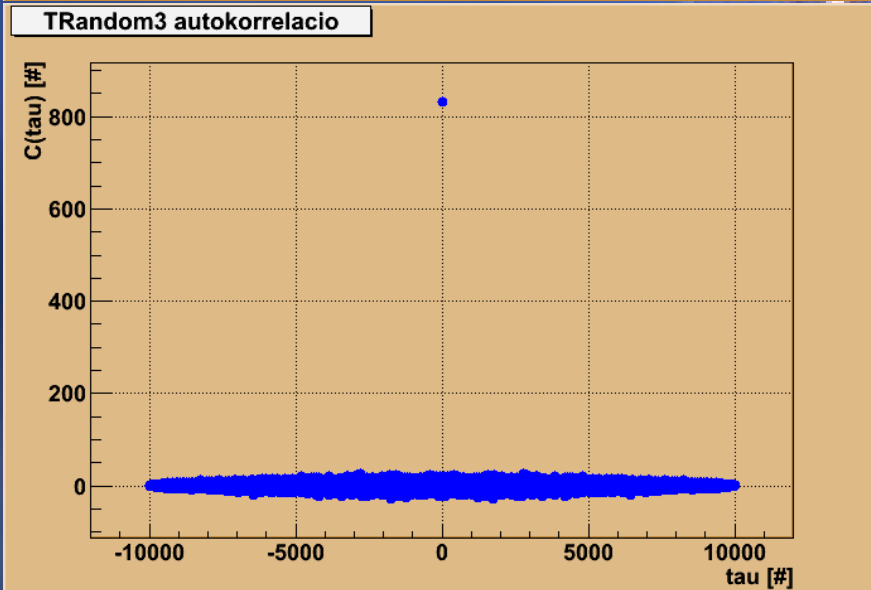
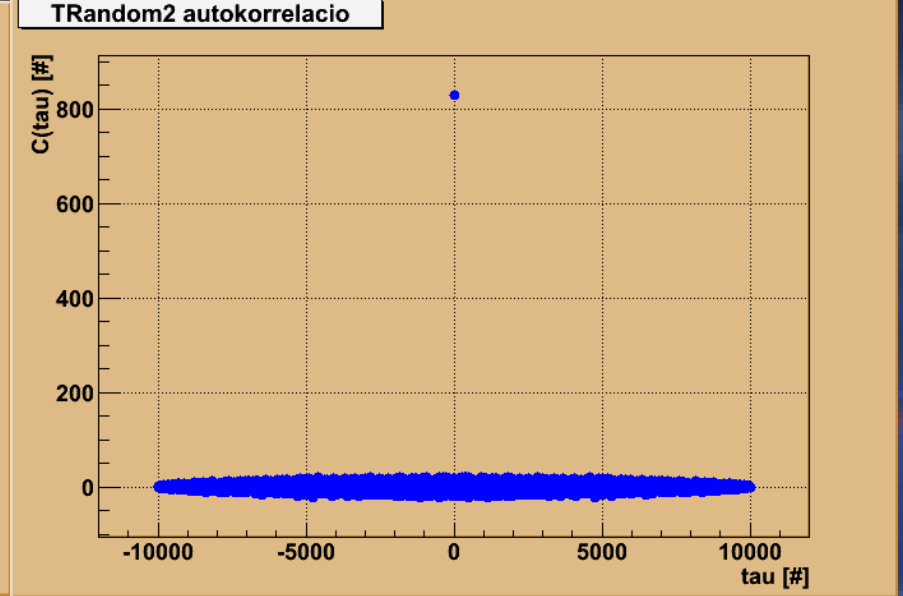
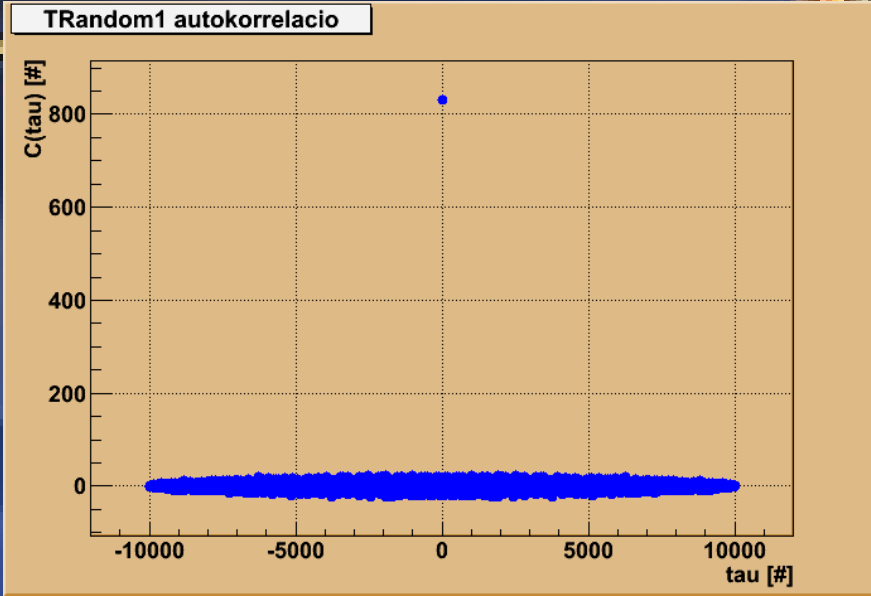
TRandom3 hisztogram



TRandom4 hisztogram



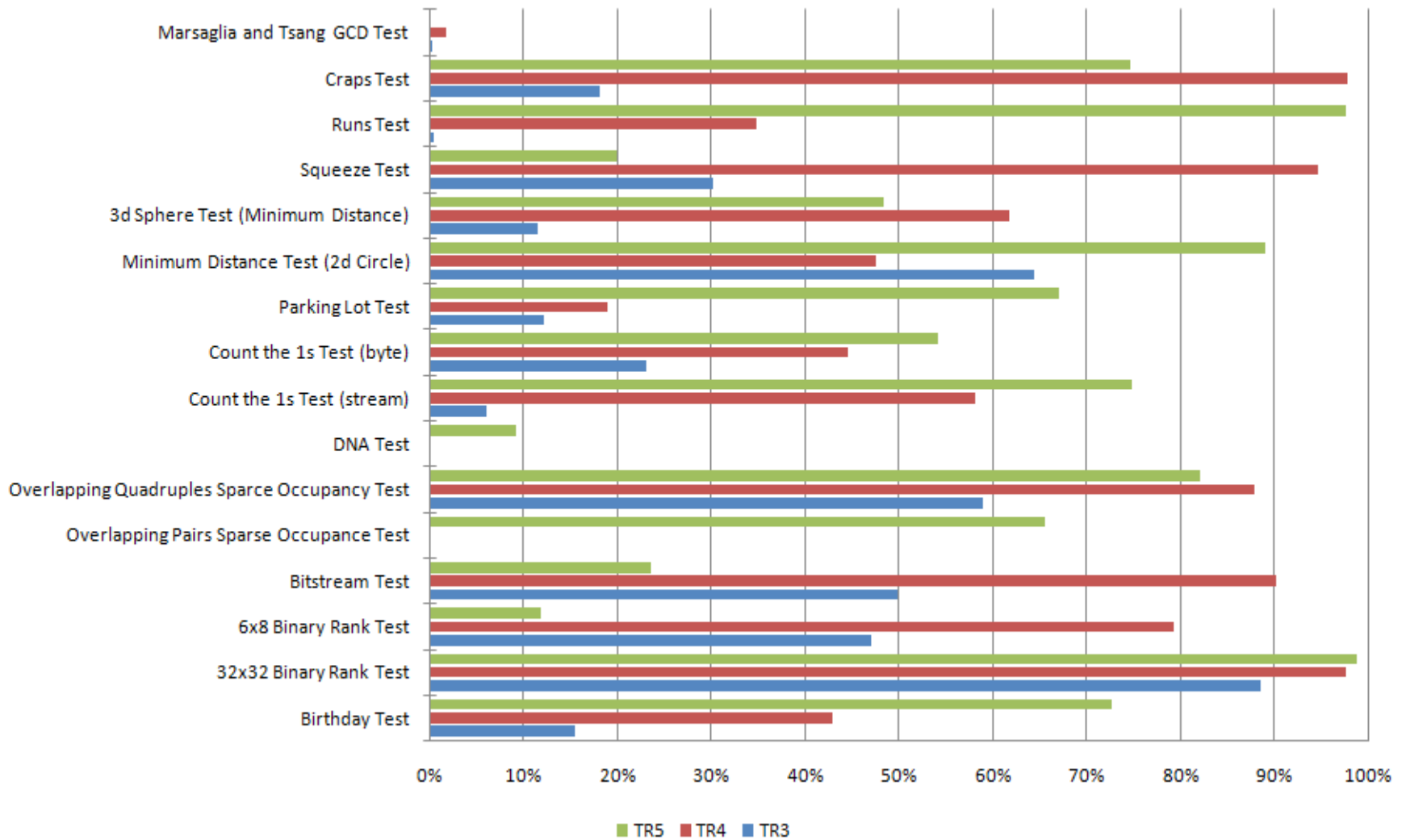
# Egyszerű teszt: auto-korreláció



# Eredmények - DieHard

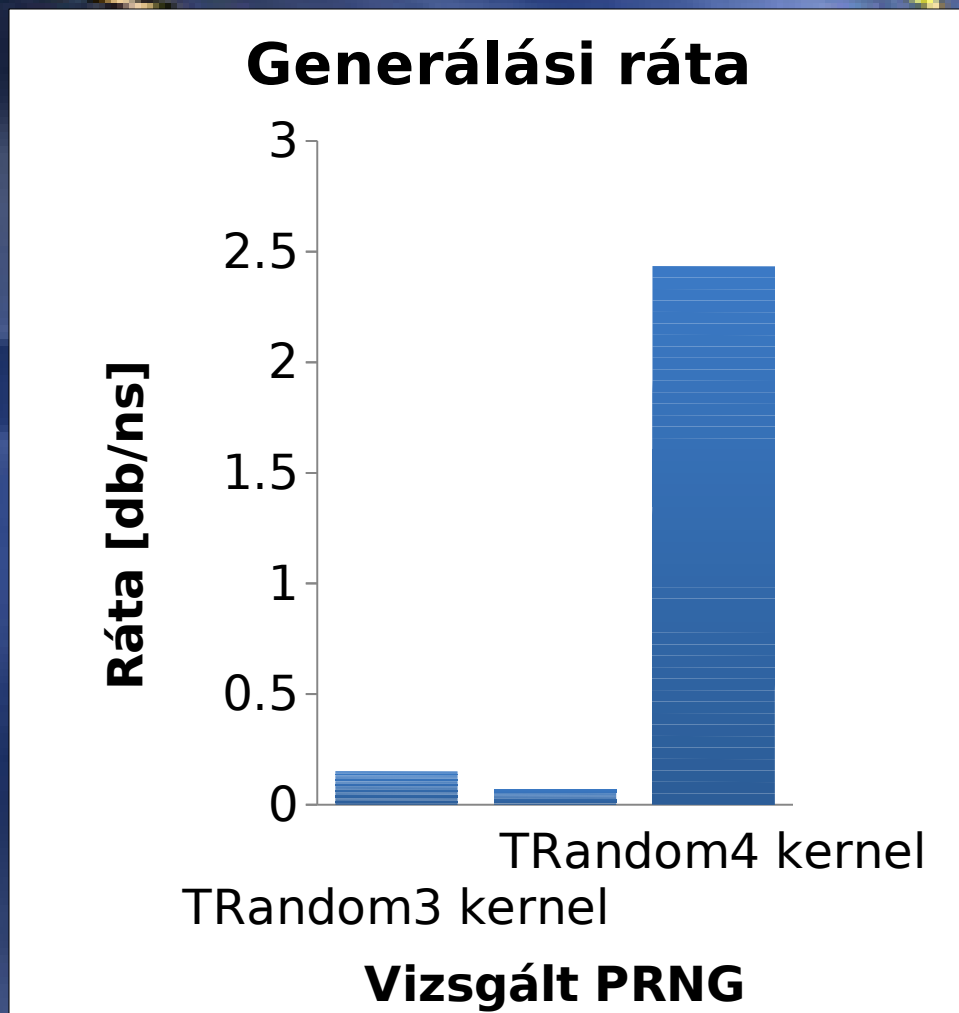
- Átfogóbb vizsgálata a PRNG-nek: DieHarder
- Lelke: Kolmogorov-Smirnov teszt, eloszlások hasonlóságáról nyilatkozik.
- Eredeti DieHard tesztek közül kettő ismerten hibásat kivéve és egy újat hozzávéve
- Trandom3/4/5 minden teszten átment!

## Results of Dieharder Tests





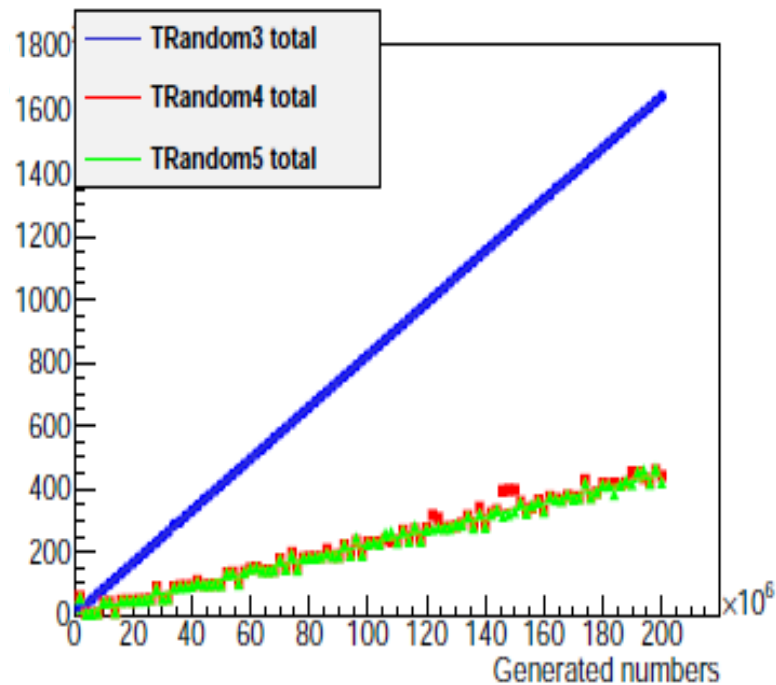
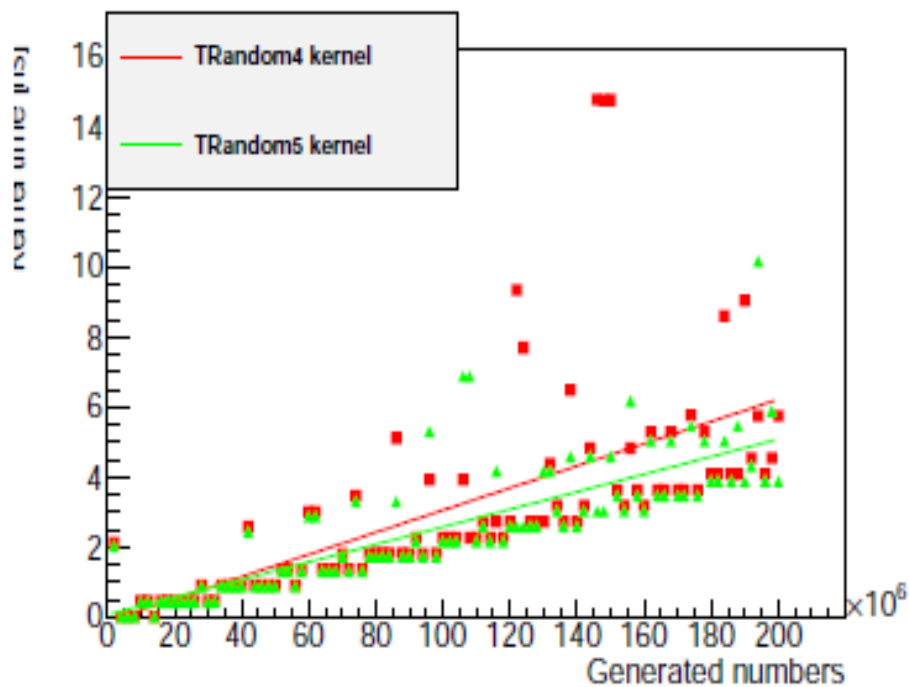
# Eredmények - Sebesség



- Teljes generálási sebesség bő kétszer lassabb.
- Ok a túlságosan nagy overhead a feladat méretéhez képest.
- Kernel idő kevesebb, mint tizenhatod része a CPU verziónak.

# Kernel idő és teljes generálási idő

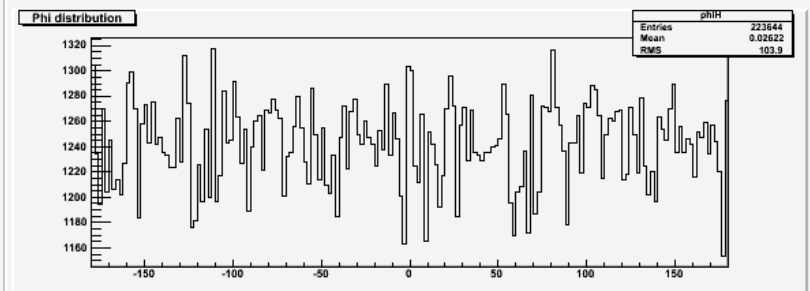
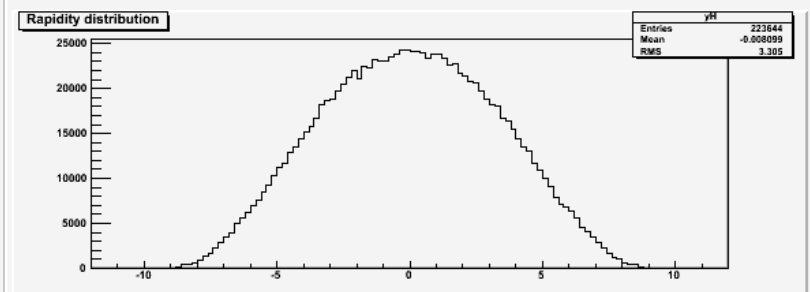
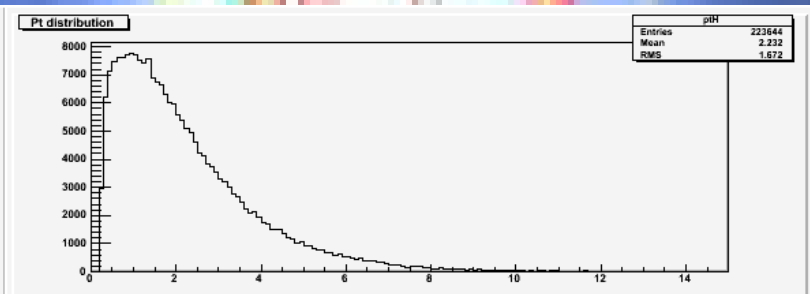
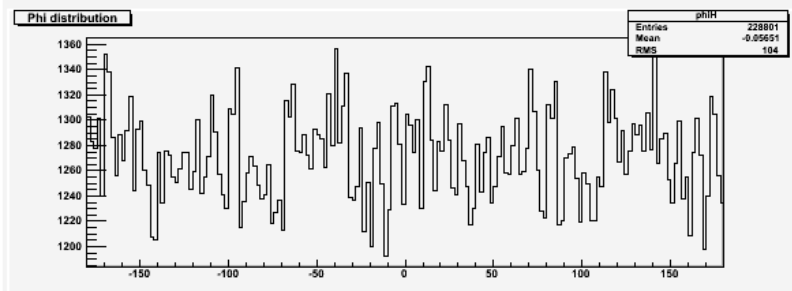
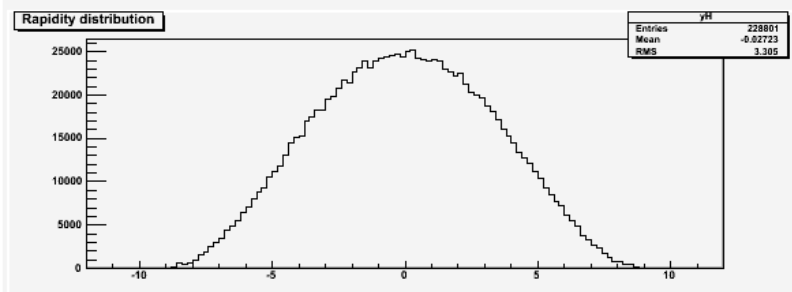
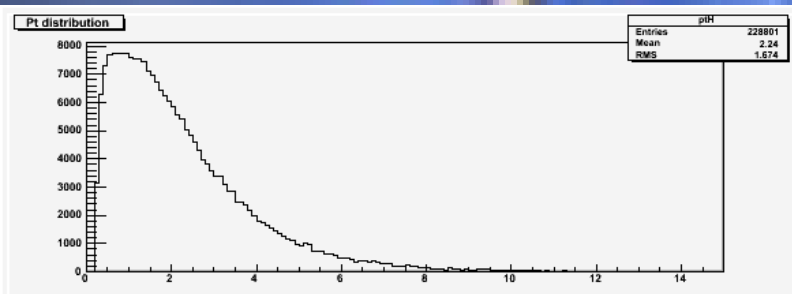
Generálási idő [ns]



# Proton-proton szimuláció

Detektor geometria nélkül, ötvenezer szimulált esemény transzverz momentum ( $p_T$ ), rapiditás ( $y$ ) és azimut szög ( $\phi$ ) szerinti eloszlásai.

TRandom3



TRandom4

# ÖSSZEFOGLALÁS

Cél:

- GPU gyorsítás írása az AliROOT keretrendszerhez

Eddigi eredmények:

- Nyílt forrású PRNG átalakítása
- Modul-szerű integráció a ROOT keretrendszerbe
- Generátor teljesítmény és eloszlás vizsgálata
- Szimuláción keresztüli igazolása a generátor helyes működésének

Konklúzió:

- Visszafelé kompatibilis modul(ok) beépítése nem lehetetlen. Az eredmények ígéretesek, érdemes folytatni az ez irányú törekvéseket.