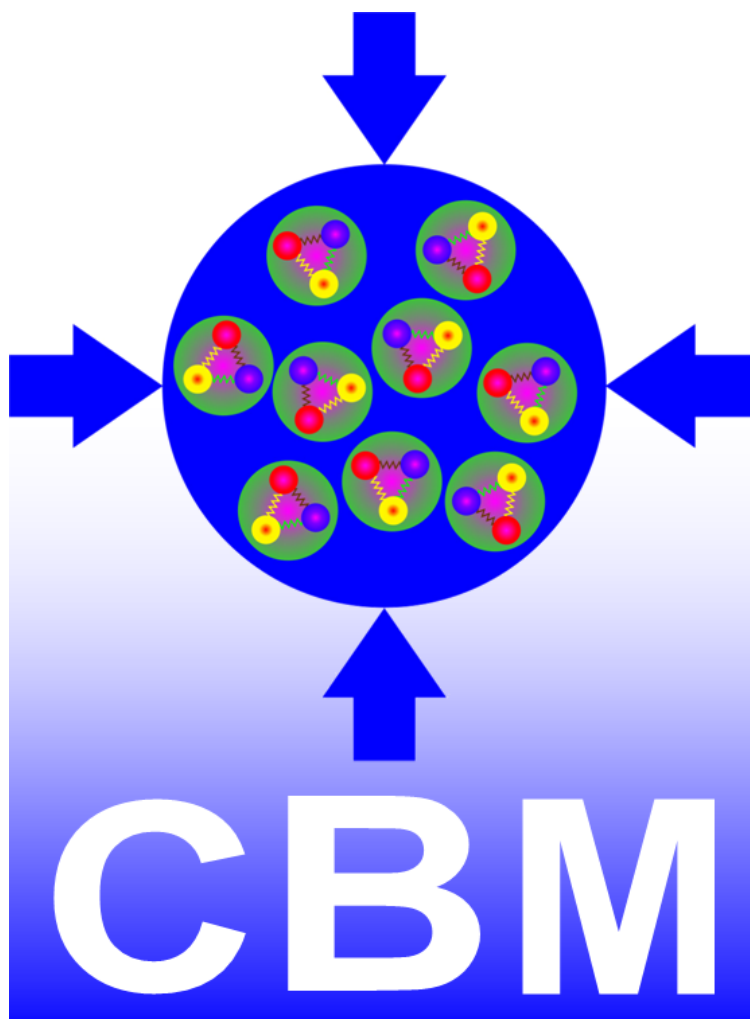


CBM szimuláció

Alex Olar

2017. szeptember 16.



Bevezető

Egy hónapot töltöttem a nyár folyamán, július hónapba, Darmstadtban a GSI nevű kutatóközpontban. Kint tartózkodásom célja az volt, hogy többet megtudjam a CBM saját szimulációjáról, amely kutató csoport már az épülő FAIR¹ része. Ezalatt a hónap alatt megismerkedtem mélyebben a ROOT ² nevű szoftverrel, a helyi cbmROOT-tal ³, valamint a C és C++ programozási nyelvekkel.

A kint létem alatt sokat tanultam a detektor technológiákról, valamint az azokban lejátszódó eseményekről és örömmel voltam részese ennek a nagyszabású projektnek és a mindennapi kutatói életnek.

Tartalomjegyzék

I. Alapok	3
I.1. QCD - BSc-s szemmel	3
I.2. CBM fizika	3
II. A CBM detektor	6
II.1. Elmélet	6
II.2. Detektor elrendezés és a szimuláció	7
III. A Φ-mezonról röviden	8
III.1. Φ -mezon rekonstrukció	8
III.2. Φ -mezon a CBM-ben	11
IV. A szimuláció	12
IV.1. Telepítés	12
IV.2. Bevezetés	13
IV.3. How-tos	15
IV.4. Φ -mezonok generálása és a kimenő adatok elemzése	16
IV.5. Összegzés	19

¹Facility for Antiproton and Ion Research

²CERN szoftver részecske analízishez

³CBM (Compressed Barionic Matter) szoftver a GSI/FAIR által fejlesztve

I.. Alapok

I.1. QCD - BSc-s szemmel

A 20. század folyamán fizikusok szembesültek azzal, hogy milyen abszolút fontos szerepet töltenek be a szimmetriák az univerzum és a körülöttünk lévő világ megismerésében. A szimmetriák vezették el őket a megmaradási tételre, valamint az antirészecskék és kvarkok felfedezéséig többek között.

A kvarkok felfedezése végre rendett teremtett a részecske állatkertben (particle ZOO), ahogy az elemi részecskék folyamatosan növekvő számára Niels Bohr szellemesen referált. Kezdetben csak három kvark volt ismert, úgy mint: u (up), d (down), s (strange). A hadronok két csoportba oszthatók szét: *mezonok* és *barionok*, amelyek rendre egy kvark-antikvark párt vagy három kvarkot tartalmaznak. A mennyiséget ami a különböző kvarkokat bizonyos szempontból jellemzi *íznek* hívjuk.

Az erős kölcsönhatás, ami a kvarkok között ható elemi kölcsönhatás, egyedi tulajdonsága a bezárás, ami megakadályozza a kvarkokat abban, hogy elszeparálva, izoláltan megtalálhatóak legyenek. Az erős kölcsönhatás töltését színnek hívjuk. A bezárás miatt, az elemi részecskék csak úgynevezett semleges színben létezhetnek, amit gyakran 'fehérnek' nevezünk. Az erős kölcsönhatást leíró alapvető elmélet a Kvantumszíndinamika (Quantum Chromo Dynamics) - QCD.

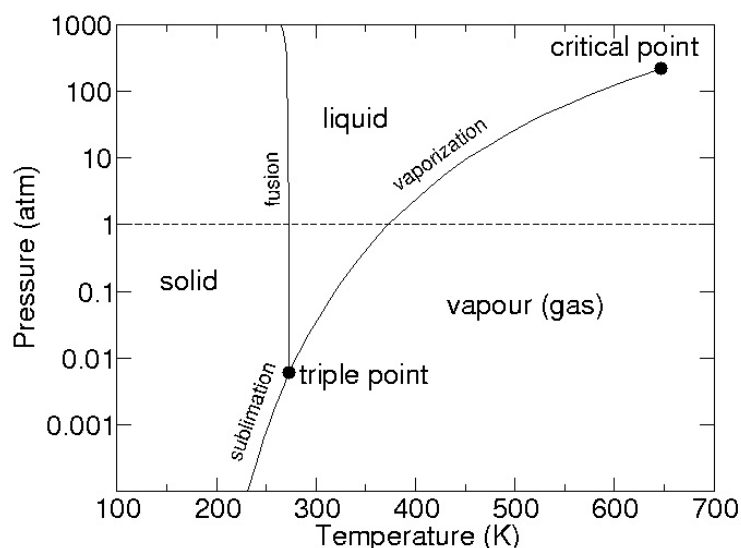
A QCD elemi részecskéi a kvarkok és antikvarkok, amelyek a gluonok által hatnak kölcsön, melyek színén színtöltést hordoznak. A gluonoknak 8 fajtájuk van, hogy minden színtranszformáció leírható legyen segítségükkel. A gluonok önmagukkal is kölcsön tudnak hatni.

I.2. CBM fizika

A barionanyaggal foglalkozva az elsődleges cél, hogy megértsük és jobban megismerjük a fázisátmenetekhez tartozó diagramot és magukat az átmeneti folyamatokat. Először is rövid bevezetőként egy kis termodinamikai áttekintéssel kezdünk a fázisokról és a fázisátalakulásokról, alapul véve a The CBM Book-ot:

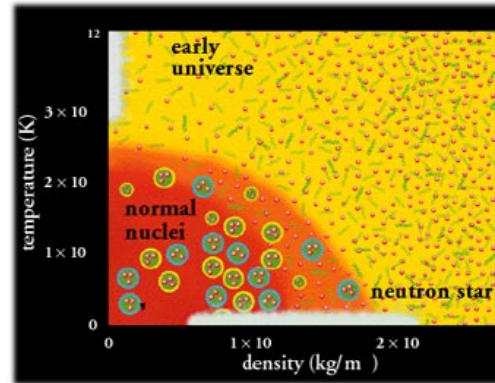
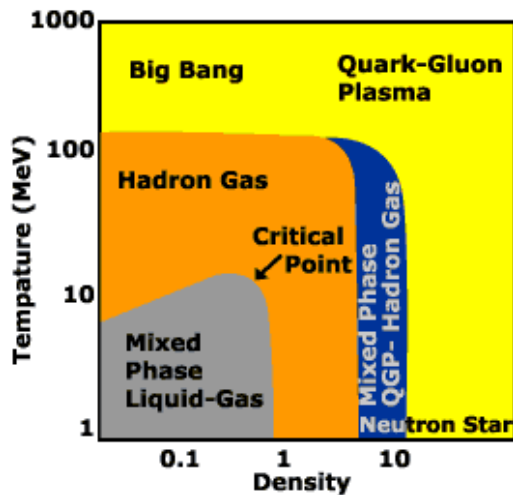
A víz fázisdiagramja megmutatja annak különböző fázisait egy nyomás-hőmérséklet rendszerben. Ismeretes, hogy adott körülmények között van egy hármaspont, amelyben a víz mindhárom halmazállapotában előfordul. A fázisok közötti vonalak mentén a víz szintén több (itt kettő) halmazállapotban előfordulhat és ezek kölcsönösen megtalálhatóak a megfelelő körülmények között. Elsőrendű fázisátmenetnek hívjuk, amikor ezen vonalakon 'áthaladva' halmazállapot-változás történik. Továbbá

megkülönböztetünk egy kritikus pontot is, amely után a fázisok nem különülnek el jelentős mértékben, ezután csak egy úgynevezett sima *crossover* figyelhető meg, nem elsőrendű fázisátmenet.



1. ábra. A víz fázisdiagramja.

Most, hogy gyorsan áttekintettem a víz fázisdiagramját, vagy legalábbis egy részét, ideje továbblépni és feltenni a kérdést, hogy mi a helyzet az erősen kölcsönható anyaggal. Az erősen kölcsönható anyag fázisdiagramja ugyanis elméleti stádiumban van, még nincs teljesen kísérletileg bizonyítva. Az ábrákon olyan különböző és elengedhetetlenül fontos fázisok vannak, amelyek a korai univerzumot jellemezték vagy éppen a neutron csillagok anyagát alkothatják. Itt mindkét ábrán egy hőmérséklet-sűrűség diagramot láthatunk.



(a) Hőmérséklet MeV-ban kifejezve, míg a előfordulását láthatjuk.
 (b) Ezen a diagramon a fázisok természetbeli előfordulását láthatjuk.
 sűrűség magsűrűségben van megadva, mindkét skála logaritmikus

A fentebbi ábrákon is látható egy fázis, amit kvark-gluon plazmának nevezünk. Ez az állapot jelen volt a Nagy Bummnál, de később nem maradt fenn, a hőmérséklet hirtelen csökkenése miatt. Látható az is, hogy a neutron csillagok belseje is kvark-gluon plazmát tartalmazhat, de azokban nem a hőmérséklet kell igen magas legyen, hanem a csillag sűrűsége.

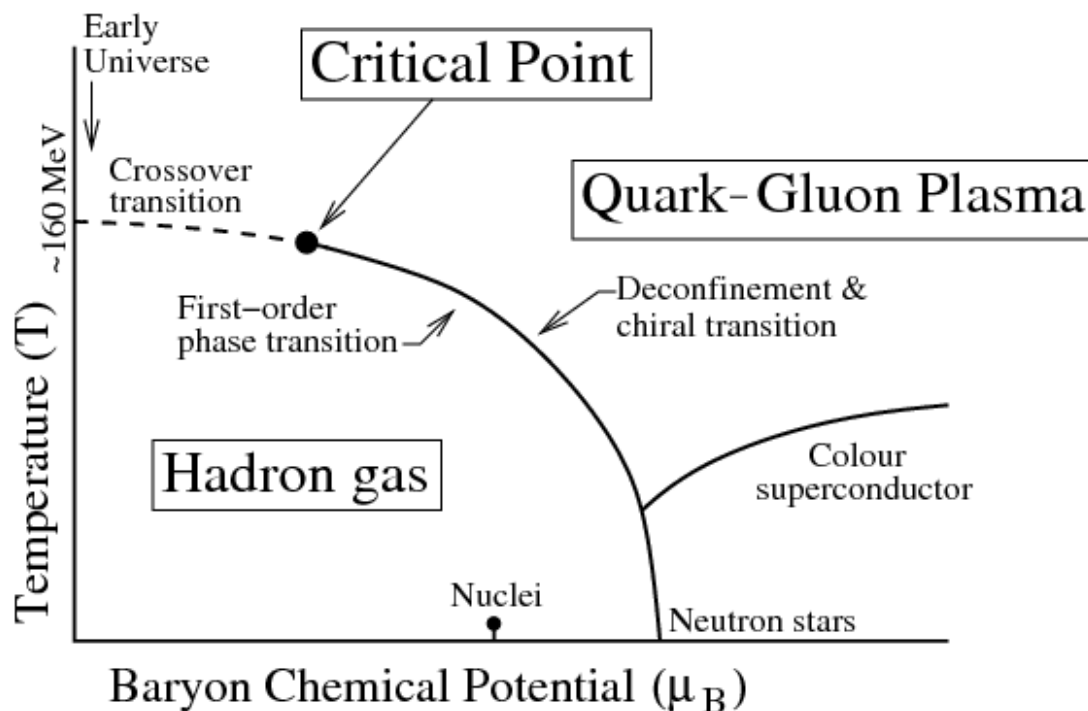
Nyilvánvalóan, a kvark-gluon plazma földi megfigyelésének egyetlen lehetőségét a nagy energiás részecske gyorsítók és azok ütköztetése biztosítja. A QCD jellemző tulajdonsága, hogy a kvarkok közti összetartás csökken, ahogy az ütközési energiát növeljük, ez a crossover jelensége. A szakirodalom aszimptotikus szabadságként hivatkozik rá. A részecske fizika egy másik fontos szimmetriája a kiralitással kapcsolatos. Ez lényegében arról beszél, hogy egy tömegtelen részecske spinje és sebességének iránya egymáshoz képest milyen irányba mutat. Hogyha azok egyirányúak, akkor a részecske jobb kezes, ha ellentétes irányúak akkor pedig bal kezes. Mivel az up és down kvarkok tömege közel azonos, ezért azt szokták mondani, hogy a QCD-nek körülbelüli királis szimmetriája van. Ellenben, ez spontán sérülhet alacsony hőmérsékleteken és sűrűségeken, ahol ez a kicsi tömegbeli különbség is jelentős lehet. Emiatt az egyik királis irány ekkor gyakoribb lesz, mint a másik, ezt nevezzük lényegében királis szimmetriasértésnek.

II.. A CBM detektor

II.1. Elmélet

A nehézionok ütközésének vizsgálata és az adatok feldolgozása egy borzasztóan komplex feladat a reakció tranziens természete miatt. Lényegében az a cél, hogy az ütközés során, 10^{-22} s-ig fennálló állapot segítségével következtessünk az erősen kölcsönható anyag fázisdiagramjára, a jelenség természetére. Az idő természetesen nagyon rövid, és az egész jelenség csak a melléktermékek révén vizsgálható.

Az elmúlt évtizedben a fő tudományos tevékenység a témában a brookhaven-i RHIC ⁴ központban és a CERN-ben található LHC ⁵ gyorsítónál zajlott. Ezek a központok nagyon fontos, és érdemleges adatot biztosítanak a fázisdiagram vizsgálatához. Ellenben ezek mind az alacsony sűrűségű régiót vizsgálják és csak a crossovert tudják feltérképezni a hadron gáz és a kvark-gluon plazma között. A FAIR projekt ezekkel szemben sokkal magasabb barion sűrűséget tervez elérni, hogy lehetőséget biztosítson az első rendű fázisátmenet vizsgálatára, valamint a kritikus pont környékének feltérképezésére.



3. ábra. A fentebb említett elméleti fázisdiagram

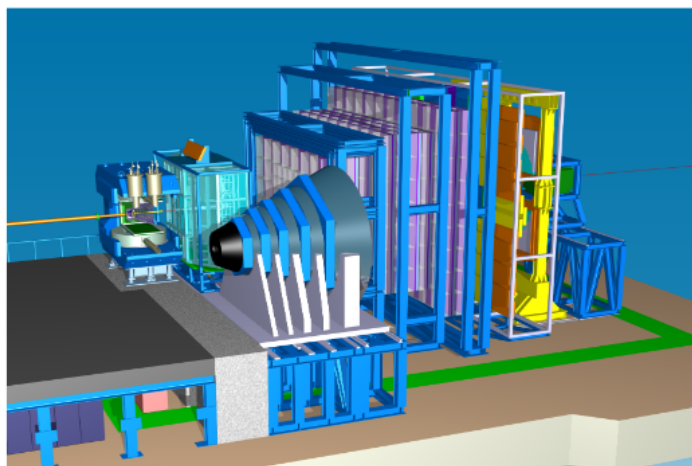
⁴Relativistic Heavy Ion Collider - Brookhaven

⁵CERN - Large Hadron Collider

II.2. Detektor elrendezés és a szimuláció

A detektor elrendezése balról jobbra haladva a következő (ábra):

- CBM szupravezető mágnes szilícium sprektrométerrel
- a micro vertex detektor (MVD) az előbbi belsejében
- a szilícium követő rendszer (STS) is
- Cserenkov-detektor (RICH - ring imaging Cherenkov detector - világos kék)
- ezt követi 4 réteg átmeneti sugázás (TRD - transition radiation detector) detektor
- és egy time-of-flight (TOF) fal
- a fő detektorok után található még egy müon sprektrométer és egy célfigyelő detektor (PSD)



4. ábra. A detektor elrendezés

A szilícium követő rendszer feladata az, hogy rekonstruálja majd a részecskék trajektóriáit. Csak töltött részecskék észlelésére képes, de képes mérni a töltés nagyságát és az impulzust is tud mérni. A TOF fal igen nagy felbontást tud elérni, nagyjából 60 ps-os felbontásra is lehetőség van.

A CBM projekt még egyelőre csak terv szintjén létezik, a szimulációt folyamatosan fejlesztik. Jövőre, vagy legkésőbb 2019-re már tervben van egy miniCBM detektor építése az esetleg később felmerülő tervezési, kivitelezési problémák kivitelezésére. A FAIR létesítmény építése idén nyáron kezdődött és az első részecskenyaláb

2022-ben várható. A miniCBM projekt a meglévő GSI gyorsítónál fog tevékenykedni az addig fennmaradó időben, ahol megpróbálják a számítógépfarmot tökéletesíteni, hogy az adatokat minél gyorsabban feldolgozhassák.

A FAIR tudósai kifejlesztettek egy több tízezer soros szimulációt, ami a ROOT-on alapszik. Ezt ők cbmROOT-nak hívják, mivel teljes egészében a CBM-hez igazodik és ingyenesen elérhető bárki számára. Sok jól ismert nehézion szimulációs eljárást használnak, amik a CBM környezetre vannak szabva, úgy mint: UrQMD ⁶, valamint PHSD ⁷. Ezek a szimulációs kódok széles körben használtak nem csak itt, hanem az egész tudomány területén.

III.. A Φ -mezonról röviden

III.1. Φ -mezon rekonstrukció

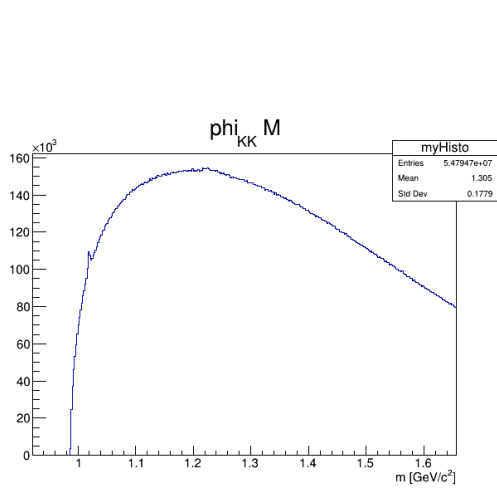
A CBM detektor egy általános célú nehézion mérési eszköz lesz, hogy az erősen kölcsönható anyag fázisdiagramját vizsgálni lehessen. A rezonanciák nagyon fontosak, hogy a sűrű anyagot vizsgálni tudjuk az ütközés során. Az ilyen rezonanciák egyike ami fontos a CBM és a fázisdiagram vizsgálatának szempontjából pedig a Φ -mezon, aminek nagyon kicsi a hadronokra vett hatáskeresztmetszete így eléggé valószínűtlen, hogy kölcsönhat a nagy mennyiségű hadronnal, ami a reakció során keletkezik, vagyis jó indikátora a sűrű, kezdeti eseménynek. A Φ -mezon egy strange és egy anti-strange kvarkot tartalmaz és a kulcsa lehet az s kvark partonikus anyagban lévő keletkezésére. A Φ -mezon K^+ , K^- párokra bomlik nagyjából 50%-os eséllyel és egyebekre (pl. dileptonokra is). Az átlagos élettartama egészen kicsi, nagyjából $1.55 \cdot 10^{-22}$ s tehát még a TOF falat sem éri el, csak a bomlástermékei lesznek detektálva már korábban is. A tömege 1.019 MeV ami a kaonok invariáns tömegével kifejezve egy rezonancia csúcsként látható az ütközés/szimuláció után kinyert adatok között.

Én a PHSD adatait vizsgáltam, amin lefuttattam a CBM szimulációt. Egy Au+Au centrális ütközést vizsgáltam $\sqrt{s} = 10$ GeV energián. A CBM szimuláció kimenetét a cbmROOT-tal rekonstruáltam. Több mint 5 millió esemény szerepelt a kezdeti .root fájlban amit a szimulációhoz használtam.

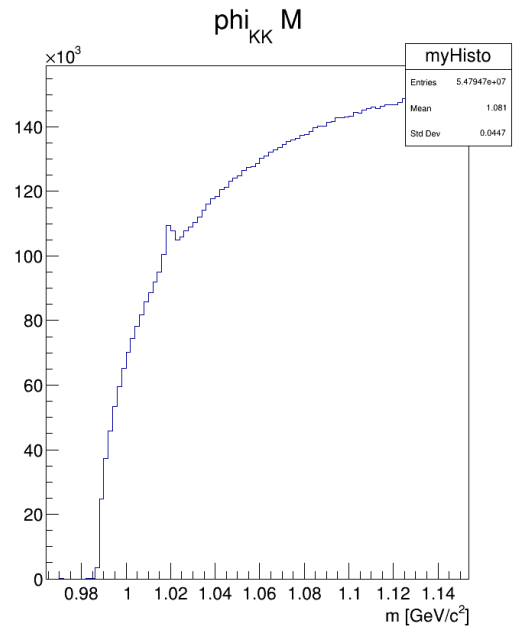
A hisztogramokon az x-tengelyen a kaon párok invariáns tömege szerepel, az y-tengelyen pedig az adott energián a 'beütések' száma. Egy apró kiugrás látható a nagy kombinatorikus háttéren nagyjából 1.02 GeV környékén ami pontosan a Φ -mezonra utal. Azok voltak a keletkezett Φ -mezonok az ütközés során.

⁶Ultra Relativistic Quantum Molecular Dynamics

⁷Parton Hadron String Dynamics



(a) A kombinatorikus háttér és egy apró, de jól látható csúcs.

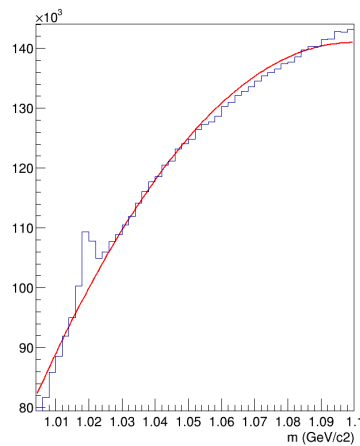


(b) A csúcs.

Egy másodfokú polinommal próbáltam becsülni a háttérrel. Az illesztés paramétereit ($ax^2 + bx + c$) :

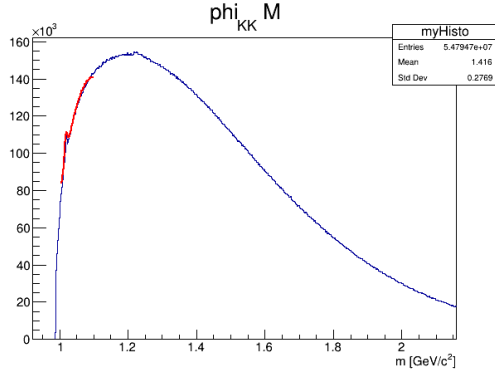
Parameter name	Value []	Error
a	-7.70559e+06	78750.5
b	1.42738e+07	150055
c	-6.49147e+06	71438.9

A háttér illesztése:

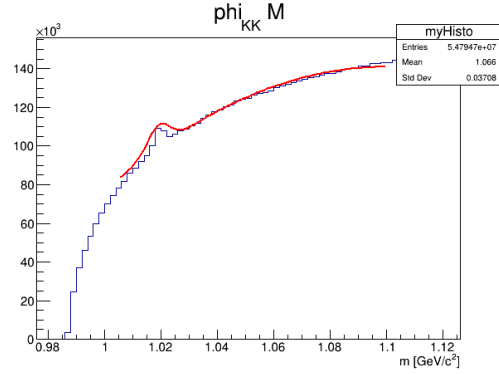


6. ábra. A háttérre vett illesztés a másodfokú polinommal.

A csúcsához más módszert alkalmaztam. Egy alacsony multiplicitású jelet használtam a csúcs alakjának becsléséhez, amit egy Gauss-függvénnyel illeszttem, majd ezt skáláztam fel a csúcsához, az állandó nagyságú háttér mellett, az eredmények a következők:



(a) A háttér és a csúcs illesztése.



(b) Ráközelítve.

Itt a ROOT makró, amit az ‘illesztéshez’ használtam:

```

1 void fit () {
2
3     TFile* srcFile = TFile::Open("
4     KFParticleFinder_phsdwocsr_auau_10gev_centra_sis100_electron_5M_ToF.
5     root");
6     TDirectory* phi = (TDirectory*)srcFile->Get("KFTopoReconstructor/
7     KFParticlesFinder/Particles/phi_{KK}/Parameters");
8     TDirectory* phi_signal = (TDirectory*)srcFile->Get("
9     KFTopoReconstructor/KFParticlesFinder/Particles/phi_{KK}/Parameters
10    /Signal");
11
12    TH1F* M = (TH1F*)phi->Get("M");
13    TH1F* Msignal = (TH1F*)phi_signal->Get("M");
14    Msignal->SetName("Msignal");
15
16    TFile* myFile = new TFile("phi_fit.root","recreate");
17    TH1F* myHisto = (TH1F*)M->Clone();
18    myHisto->SetName("myHisto");
19    TH1F* myBackground = (TH1F*)M->Clone();
20    myBackground->SetName("myBackground");
21    TH1F* mySignal = (TH1F*)Msignal->Clone();
22    mySignal->SetName("mySignal");
23    myFile->cd();
24
25    srcFile->Close();
26
27    TCanvas* canv = new TCanvas("canv","Total fit",640,480);
28
29    TF1* background = new TF1("background","pol2",1.004,1.1);

```

```

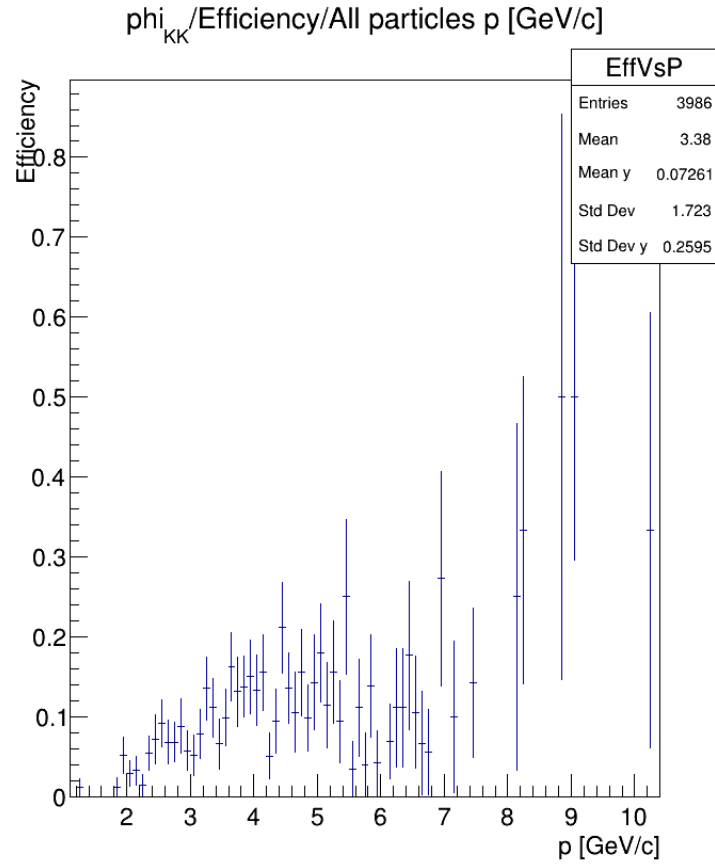
25 TF1* signal = new TF1("signal","gaus",1.011,1.033);
26
27 TF1* total = new TF1("total","gaus(0) + pol2(3)",1.005,1.1);
28
29 background->SetParameters(0.2,1.01,108000.);
30 background->SetParNames("landau_1","landau_2","landau_3");
31 myBackground->Fit("background","R+");
32
33 Double_t params[6];
34 background->GetParameters(&params[3]);
35
36 signal->SetParameters(3200., 1.021, 3000.);
37 signal->SetParNames("scale","mean","sigma");
38 mySignal->Fit("signal","R+");
39 signal->GetParameters(&params[0]);
40
41 params[0]*=250.; // rescale
42 total->SetParameters(params);
43 total->SetParNames("scale","mean","sigma",
44 "la","b","c");
45 myHisto->Draw();
46 total->Draw("same");
47
48 signal->Write();
49 background->Write();
50 total->Write();
51 myHisto->Write();
52 canv->Write();
53
54 myFile->Close();
55
56 }

```

III.2. Φ -mezon a CBM-ben

Igen nehéz feladat lesz hatékonyan detektálni a Φ -mezonokat a CBM detektorrendszerrel. A részecskék nem csak rövid életűek, de egy hatalmas háttér is nehezíti az apró csúcs megtalálását. Ezért is kell hatalmas számú eseményt vizsgálni, hogy a csúcs a statisztikában már látható legyen. Ennek ellenre határozottan mondhatjuk, hogy a CBM detektor képes lesz a Φ -mezonok detektálására és ezáltal a strange kvark termelődésének megértésére az erősen kölcsönható anyagban.

A szimuláció hatékonysági mutatókat is biztosít. Mindezeket különböző részecske impulzusok esetén. A jelzett detektálás hatékonysági értékek nem túl magasak, de eléggé stabilak adott tartományokban az észleléshez:



8. ábra. Hatékonyság az impulzus függvényében

IV.. A szimuláció

IV.1. Telepítés

A CBM szimuláció telepítésének három fő komponense van, az egyik a Fair-ROOT, majd a FairSoft és végül a cbmROOT. Bármilyen rendszerre telepíthetők az alábbi linkről: <https://redmine.cbm.gsi.de/projects/cbmroot/wiki/InstallCbmRootAuto>

Erősen ajánlott a telepítést ezt követve megtenni, mivel rengeteg apró, de akadályozó probléma előjöhethet a telepítés során. A teljes csomag tartalmazza a ROOT-ot is, így az egész nagyjából 25 GB helyet foglal.

IV.2. Bevezetés

Maga az ütközés a UrQMD és a PHSD programok segítségével játszódik le, a CBM szimuláció a detektor választ szimulálja, tehát az ezekből származó adatokat kapja meg kezdeti paraméternek. Ezek a modellek az ALICE, RHIC és LHC detektornak, valamint nem utolsó sorban a CBM detektornak lettek fejlesztve. Én főleg UrQMD adatokat használtam, de PHSD fájlokkal is találkoztam kintlétem során.

Az első lépés az, hogy le kell futtatni egy Monte Carlo szimulációt, hogy képeset legyünk a ‘valódi’ adatokat összepárosítani a keltett eseményekkel. A program ezen része arra lett tervezve, hogy kiszűrje a találatokat a detektor anyagban és olyan pontokat találjon, amelyek később trajektóriákká összeállíthatók.

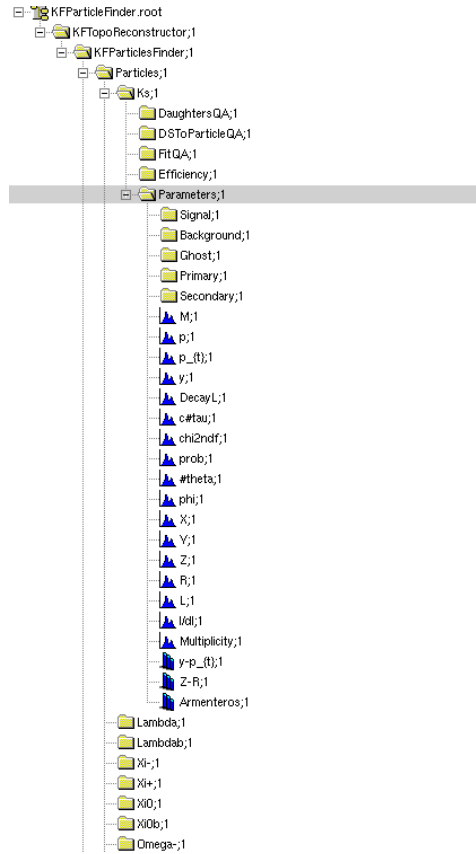
A program a Geant3 és Geant4 programokat használja, hogy a részecskék anyagon való áthaladását szimulálja. Ez is a Monte Carlo szimuláció része.

Az első makró kimenetén tehát egy szimulációs fájl van, ami az STS és az MVD detektorok által detektált találatokat tartalmazza valamint a TOF fal és egyéb detektorok adatait is. Ezeket felhasználva lép a program a második fázisba, a rekonstrukció részhez. A rekonstrukciós kód először is klasztereket próbál találni az MVD detektorban, hogy megtalálja, hogy hol volt az ütközés/ütközések kiinduló pontja. Ha ezt megtalálta továbbhalad és megpróbálja lekövetni a részecske pályákat. A töltött részecskék körpályára állnak az erős mágneses tér hatására így a pontokra köríveket próbálnak illeszteni és a legjobb illesztéssel bírót fogadják csak el (van egy százalékos határ, ami alatt hibás detektálásnak ítélik). Én főleg az MVD és STD detektorokra koncentráltam, tehát a többit most nem említem itt.

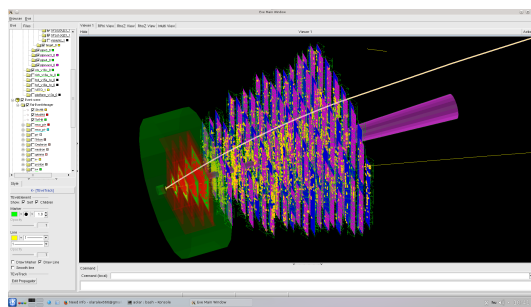
Nyilvánvalóan, a találatok és a pályákat többször próbálja meg a program helyesen megtalálni, azért, hogy elkerülje a hibákat. Kisebb az esélye így a hibás találatnak, vagy a hibásan illesztett trajektóriának. Ennek része a digitalizáció, ami lényegében azt jelenti, hogy a szimulációs program megpróbálja a detektor választ is számításba venni. Vegyük például az STS detektort. Ennek egy szálas, hálós elrendezése van, amikor egy részecske áthalad, akkor több szálon is detektálju, ezek metszéspontjában van a tényleges helye. De ha egyszerre két részecske ment át ‘ugyan azon a ponton’, akkor ezt nem láthatjuk, később a pályák illesztésénél probléma lehet. Ezért is van az, hogy ha az STS detektor több, mint 5%-a detektál, akkor a rendszer lényegében nem mér, nem szerez kiértékelhető adatokat.

A sikeres rekonstrukció után, ami a nyers adatokból létrehozta végső soron a trajektóriákat az egyetlen visszamaradó feladat a részecske felismerés és ezek pályákhoz való párosítása. Erre egy robosztus és hatékony program áll rendelkezésre, aminek a neve KFParticleFinder.

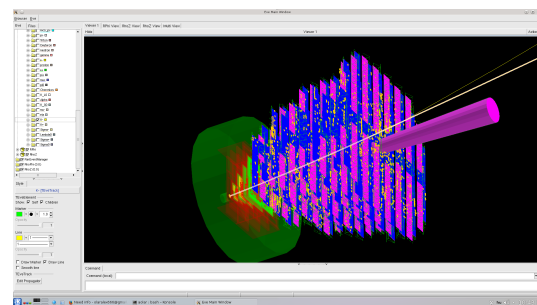
Ennek a programnak a kimenete egy .root fájl, ami rengeteg részecskét és hozzájuk tartozó adatot tartalmaz, detektálási hatékonyságról, háttérrel, armenteros diagramokkal, bemenő és kimenő jelekkel, stb. . Az szerkezete nagyjából így néz ki:



9. ábra. A ROOT fájl struktúrájának egy része.



(a) A rekonstruált pályák az MVD és STS detektorokban.



(b) Másik szögből

IV.3. How-tos

Ahogy korábban említettem először a Monte Carlo szimulációt kell használni valmilyen bemeneti fájlal. Ez egy .root fájl vagy egy egyszerű ASCII fájl is lehet, a szimulációs kód képes mindkettő fogadására. Egy ilyen fájlban részecske ID-k és impulzusuk található. A kimenete a PHSD és a UrQMD szimulációknak általában egy .root fájl, de például a HIJING sima szöveges kimenetet produkál. A CBM szimulációnál különböző függvények teszik lehetővé mindkét adattípus feldolgozását.

Megtanultam használni a jelgenerátor programot, amivel bárki, bármit küldhet a detektor szimuláció bementére. Én főként arra használtam, hogy kontrollált körülmények között, csak Φ -mezonokat küldjek be, amivel vizsgálni lehet, hogy mi lesz a program kimentén a KFPParticleFinder által kiadott .root fájlban. Ahhoz, hogy a generátor által biztosított ASCII fájlt olvasni tudja a szimuláció a következő módosítások szükségesek:

```
1 1 0 0 0
333 0.349404 0.108345 2.17087
1 2 0 0 0
333 -0.601515 -1.42376 7.32593
1 3 0 0 0
333 0.604993 0.756893 8.0675
1 4 0 0 0
333 -0.605273 0.957298 2.78006
1 5 0 0 0
333 -0.561403 -0.245707 1.30767
1 6 0 0 0
333 -0.111909 0.297546 0.780414
1 7 0 0 0
333 0.479322 0.647613 1.23907
1 8 0 0 0
333 -0.495742 -0.65654 1.05797
1 9 0 0 0
333 -0.736586 -0.211334 2.19586
1 10 0 0 0
333 0.0558235 -0.109982 3.04292
```

333 a részecske ID a Φ -mezonnál. Ezután a szimuláció tudni fogja, hogy hogyan dolgozza azt fel, és képes lesz azt elbomlasztani a megfelelő valószínűségekkel.

```
1 FairAsciiGenerator *SignalGen = new FairAsciiGenerator(inFile);
2 primGen->AddGenerator(SignalGen);
```

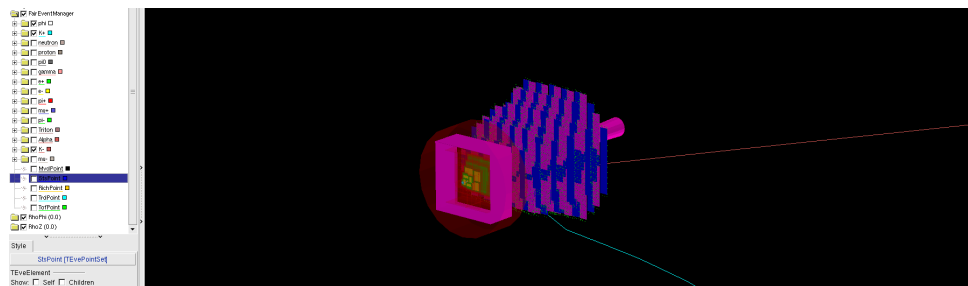
Fentebb a .root fájlokhoz használt *CbmUnigenGenerator* helyett ASCII fájlok esetén ezt kell használni. Még egy fontos lépés van itt. Ha szeretnénk vizualizálni a későbbiekben az eredményeinket, akkor engedélyeznünk kell a trajektóriák ilyen szintű mentését. Ez nyilvánvalóan nem hatékony hatalmas részecske számok esetén, de ha csak néhány részcskét küldünk be, akkor hasznos lehet látni, hogy hogyan is működik a program, esetleg hibákat is észrevehetünk.

```

1 // -Trajectories Visualization (TGeoManager Only )
2 run->SetStoreTraj(kTRUE); //->
3 //

```

Tehát a rekonstrukció után, valamint a részecske felismerés végeztével, ha bekapcsoltuk a vizualizációt képesek vagyunk vizualizálni az eseményeket. Ehhez az *eventDisplay.C* makrót kell futtatnunk. Ez a makró az egész CBM geometriát tartalmazza, tehát az egész detektort átláthatjuk vele. Megjeleníthető benne az összes trajektória és a részecskék. Néhány kép arról, ahogy egy Φ -mezon két kaonra bomlott:



11. ábra. Vizualizáció az MVD és STS detektorokban

IV.4. Φ -mezonok generálása és a kimenő adatok elemzése

A jelgenerátorral 2500 eseményt generáltam ahol a Φ -mezonok pont a céltárgy közepében helyezkedtek el, tehát minta éppen ott keletkeztek volna az ütközés során. Az ilyen adatok elemzése azért fontos, mert ekkor kontrollált körülmények között, adott részecske számmal tudjuk vizsgálni a kimenő részecskék számát, eloszlását és ebből ismeretlen kezdeti részecskénél következtethetünk a Φ -mezonok számára így a strange keltés folyamatára.

A generátor makróban a bemenő nyaláb energiáját is változtathatjuk, valamint a környezet hőmérsékletét is (mindkettő GeV-ben) és persze azt is, hogy milyen részecskét akarunk generálni.

```

1 double fSlope = .154; // temperature
2 ...
3 double eBeam = 10.; // beam energy
4 double pBeam = TMath::Sqrt(eBeam*eBeam - kProtonMass*kProtonMass);
5 ...
6 const int NParticlesPerEvent = 1;
7 const double kSignalMass[NParticlesPerEvent] = {1.019455}; // mass
8   in GeV
9 const int kSignalID[NParticlesPerEvent] = {333};
10 ...
11 for (int i=0; i<NEvent; i++){
12   // Generate rapidity, pt and azimuth

```



```

12  outputfile<<NParticlesPerEvent<<"    "<<i + 1<<"    "<<0.<<"    "<<0.<<"
    "<<0.<<endl;
13  for( int j=0;j<NParticlesPerEvent;++j) {
14  double yD    = gRandom->Gaus(fYcm, fRapSigma);
15  double ptD    = fThermal[j].GetRandom();
16  double phiD   = gRandom->Uniform(0., kTwoPi);
17
18  // Calculate momentum, energy, beta and gamma
19  double pxD     = ptD * TMath::Cos(phiD);
20  double pyD     = ptD * TMath::Sin(phiD);
21  double mtD     = TMath::Sqrt(kSignalMass[j]*kSignalMass[j] + ptD*ptD)
    ;
22  double pzD     = mtD * TMath::SinH(yD);
23
24  outputfile<<kSignalID[j]<<"    "<<pxD<<"    "<<pyD<<"    "<<pzD<<endl;
25
26  }
27  }

```

Jól látható, hogy ezt a makrórt elég könnyű személyre szabni, tehát bárki könnyedén elkészítheti magának a számára megfelelő bemeneti fájlt. A bemeneti fájl az eseménynek számával és a fájl nevével át kell adni a szimulációs programnak.

```

1 void run_mc_phi(TString inFile="Signal_phi_2500.txt", const char*
    setupName = "sis100_electron", Int_t nEvents = 2500)
2 {
3   TString outFile = "sim_phi_2500.root";
4   TString parFile = "param_phi_2500.root";
5   ...
6   // ——— Define the target geometry
    _____
7   //
8   // The target is not part of the setup, since one and the same setup
    can
9   // and will be used with different targets.
10  // The target is constructed as a tube in z direction with the
    specified
11  // diameter (in x and y) and thickness (in z). It will be placed at
    the
12  // specified position as daughter volume of the volume present there.
    It is
13  // in the responsibility of the user that no overlaps or extrusions
    are
14  // created by the placement of the target.
15  //
16  TString targetElement = "Gold";
17  Double_t targetThickness = 0.025; // full thickness in cm
18  Double_t targetDiameter = 2.5; // diameter in cm
19  Double_t targetPosX = 0.; // target x position in global c.s
    . [cm]
20  Double_t targetPosY = 0.; // target y position in global c.s
    . [cm]

```

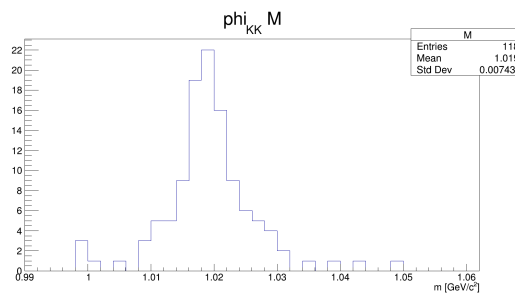
```

21 Double_t targetPosZ      = 0.;      // target z position in global c.s
    . [cm]
22 Double_t targetRotY      = 0.;      // target rotation angle around
    the y axis [deg]
23 }

```

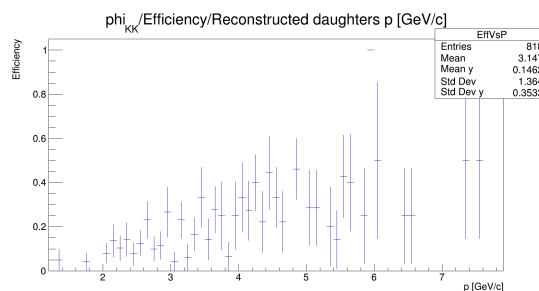
A kimenet egy .root fájl, ami mint már korábban említettem adatokat tartalmaz a beütésekkel a detektor anyagban. Fontos megemlíteni, hogy a céltárgyat is bárminek definiálhatjuk, a helyzetét is változtathatjuk, de a mi feladatunk, hogy helyesen tegyük, mert a szimuláció lefut úgy is, hogy a nyaláb el sem találja a céltárgyat.

Ezután a rekonstrukciós fájlt is kissé módosítani kell, majd ez a trajektóriákat találja meg. Majd a fizika makrót kell futtatni, hogy a *KFParticleFinder* megtalálja a pályákhoz tartozó részecskéket. A kimeneti .root fájlból néhány részlet:

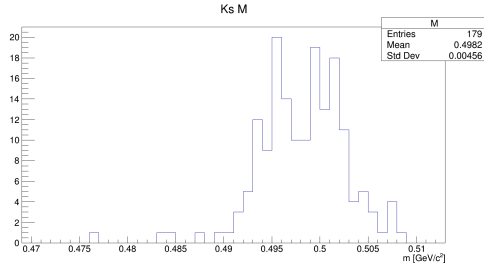


12. ábra. A kaonpárok invariáns tömegének diagramján 1.02 GeV-nél, a Φ -mezon

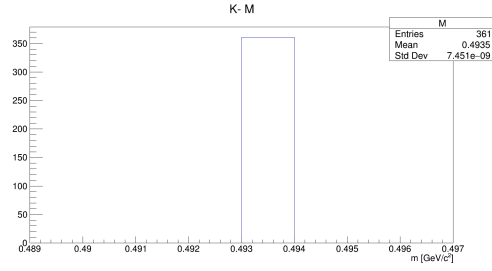
A jelben 2500 esemény volt, azaz 2500 db mezont generáltam. Nagyjából 50%-os eséllyel bomlottak el ezek kaon párokra, valamint a digitalizáció során, nagyjából 15%-os hatékonysággal tudott a program rekonstruálni, azaz nagyjából 180 db rekonstruált Φ -mezonra lehet számítani a *KFParticleFinder.root* fájlban. Mivel valószínűségekről van szó, így a fájlban lévő 120 db nem is rossz statisztikailag.



Könnyű megtalálni azokat a részecskéket is amikre a Φ -mezonok elbomlottak. Így találhatunk a bomlástermékek között pionokat, kaonokat, K_S^0 részecskéket.



(a) K_S^0 részecskék



(b) Negatív kaonok.

A pionokat itt nem tüntettem fel külön, mivel egy ilyen folyamat során azok nem adnak informatív képet, lévén, hogy nem csak a Φ -mezon tud úgy bomlani, hogy pion is van a bomlástermékek között, de a kaonok és egyéb részecskék is, így a pionok multiplicitása igen nagy.

IV.5. Összegzés

A CBM detektor képes lesz arra, hogy felismerje és megtalálja a Φ -mezon bomlásokat, ezáltal a strange termelődést és a partonikus anyagot vizsgálni tudja. A szimulációk azt sugallják, hogy a detektor minden valószínűség szerint képes lesz detektálni a szükséges részecskéket a megfelelő hatásfokokkal.

Hivatkozások

- [1] *The CBM Physics Book: Compressed Baryonic Matter in Laboratory Experiments* 2011 ed B Friman *et al* (Springer) Lect. Notes Phys.
- [2] Tapia Takaki, J. D., ALICE Collaboration 2008, Journal of Physics G Nuclear Physics, 35, 044058
- [3] V.Vovchenko I.Vassiliev I.Kisel M.Zyzak, Φ -meson production in Au+Au collisions and its feasibility in the CBM experiment, CBM Progress Report 2014
- [4] Bravina, L., Csernai, L., Faessler, A., et al. 2003, Nuclear Physics A, 715, 665
- [5] F. Wang, R. Bossingham, Q. Li, I. Sakrejda, and N. Xu, Φ -meson reconstruction in the STAR TPC, 1998
- [6] Hans Rudolf Schmidt Hyperons at CBM-FAIR, Journal of Physics: Conference Series 736