

CZECH TECHNICAL UNIVERSITY
FACULTY OF NUCLEAR SCIENCES AND PHYSICAL ENGINEERING
DEPARTMENT OF PHYSICS

Research Work

**Study of the methodics of measuring the p/\bar{p} ratio
and the number of protons in nucleus-nucleus
collisions in the central region of the ALICE detector**

Michal Petráň

Supervisor: RNDr.Vojtěch Petráček, CSc. Prague, Sept. 5, 2007

Contents

Abstract	ii
1 Introduction	1
2 Particle production mechanism	2
2.1 Initial conditions and deconfinement	2
2.2 Statistical approach	3
2.2.1 Implementation of conservation laws	5
2.3 Kinetics of time evolution and equilibration of charged particles .	6
2.3.1 Kinetic master equation for probabilities	6
2.4 Unified conditions of particle freeze-out	8
3 Baryon Number Flow	12
3.1 Baryon number distribution	13
3.1.1 BN annihilation at high energies	13
3.1.2 BN asymmetry	14
3.1.3 BN flow	15
3.2 Heavy ion collisions	15
4 AliROOT	18
4.1 Framework	18
4.2 Installation	18
4.2.1 Environmental Variables	19
4.2.2 Obtaining packages	19
4.3 Simulation	20
4.3.1 Event generation	21
4.3.2 Transport	22
4.4 Reconstruction framework	25
4.5 Analysis	29
4.5.1 Analysis Tools	29
5 Data analysis	32
5.1 PYTHIA	32
5.2 ROOT	32
5.3 My ROOT macro	33
5.4 Results	33
5.4.1 Asymmetry in pp collisions at $\sqrt{s_{NN}} = 900 \text{ GeV}$	33
5.4.2 Asymmetry in pp collisions at $\sqrt{s_{NN}} = 14 \text{ TeV}$	34
6 Conclusion and Outlook	38
7 Appendix A	40

Abstract

Název práce:

Studium metodiky měření poměru p/\bar{p} a počtu protonů ze srážených jader v centrální oblasti detektoru ALICE

Autor: Michal Petráň

Obor: Jaderné inženýrství

Druh práce: Výzkumný úkol

Vedoucí práce: RNDr. Vojtěch Petráček, CSc. Katedra fyziky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze

Abstrakt: Při srážkách těžkých jader dochází k intenzivnímu zbrždění nukleonů srážejících se jader. Rapiditní rozložení protonů participujících při srážce je důležitým signálem charakterizujícím průzračnost jaderné hmoty. Množství participujících nukleonů v centrální oblasti rovněž určuje podmínky pro přechod jaderné hmoty do dekonfinované fáze. Při srážkách při těžištové energii 5.5 TeV/A bude centrální oblast téměř prosta nukleonů ze srážených jader. Převážná část baryonů bude produkována párovou produkcí, takže poměr p/\bar{p} se bude blížit 1 a rozdíl $p - \bar{p}$, charakterizující počet protonů ze srážených jader, se bude blížit 0. Pro správné měření těchto veličin je třeba prostudovat systematiku měření p a \bar{p} v detektoru ALICE. Bude provedena simulace v simulátoru ALIROOT a to jak pro případ jaderných srážek, tak pro případ p-p srážek při 900 GeV, při nichž bude možno ověřit přesnost prováděných simulací srovnáním s prvními daty z experimentu. .

Klíčová slova: Kvark-gluonové plazma, Srážky těžkých iontů, ALICE, ALIROOT.

Abstract

Title:

Study of the methodics of measuring the p/\bar{p} ratio and the number of protons in nucleus-nucleus collisions in the central region of the ALICE detector.

Author: Michal Petráň

Abstract: In heavy-ion collisions the nucleons from the colliding nuclei are intensively breaking. Rapidity distribution of the protons participating in the collision is a very important signal characterizing the transparency of the nuclear matter. The quantity of participating nucleons in the central region defines the conditions for the transition of nuclear matter to the deconfined phase. In the collisions at 5.5 TeV/A, the central region will be almost free of nucleons from the colliding nuclei. The majority of the baryons is produced by the pair production mechanism, therefore the p/\bar{p} ratio should approach 1 and the difference $p - \bar{p}$, characterizing the number of protons from the colliding nuclei, should approach 0. For proper measurement of these variables, it is necessary to study the systematics of the p and \bar{p} in the ALICE detector. A simulation in the AliROOT simulator will be done for both the nucleus-nucleus collision and pp collision at 900 GeV. In the near future, the precision of simulated data will be compared to the real data from the ALICE detector.

Key words: Quark-gluon plasma, Heavy-ion collisions, ALICE, AliROOT.

Prohlášení

Prohlašuji, že jsem svůj výzkumný úkol vypracoval samostatně a použil jsem pouze podklady a zdroje (literatury, projekty, SW, atd.) uvedené v příloženém seznamu.

Nemám žádný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne:

Acknowledgement

First of all, I would like to thank my supervisor Vojtěch Petráček for his invaluable help, motivation and guidance through the preparation of this work. Last but not least I would like to thank him for the much needed language corrections.

I also express my gratitude to others from Dept. of Physics at Faculty of Nuclear Sciences and Physical Engineering, CTU for their occasional advice in different topics.

Finally, I am grateful to my parents for their patience and support during the whole time of my studies.

1 Introduction

In this work I concentrate on the possible mechanisms of particle production and baryon stopping in heavy-ion collisions. I think, that statistical model of particle production will provide satisfactory description for particle ratios, chemical potential and kinetic freeze-out temperature from the new phase of matter, the quark-gluon plasma (QGP) to hadron gas (HG). Later, the string junction model is discussed as a cause of baryon stopping and therefore being a possible explanation of the proton/antiproton asymmetry, which is observed in different experiments and, as I believe, will be observed in the upcoming ALICE experiment at LHC. A description of the ALICE offline analysis framework (AliROOT) is given. I have done a simulation with PYTHIA event generator and analyzed the events for proton asymmetry and ratios.

2 Particle production mechanism

QCD predicts, that strongly interacting matter undergoes a phase transition from hadron gas (HG) to quark-gluon plasma (QGP) in heavy ion collisions. After the collision, the same phase transition but in the opposite direction takes place. It is particularly important to find experimental probes to check whether the produced medium really was QGP in its early stage after the collision. Different probes have been studied with different experiments. The most promising signals of deconfinement are related to transverse momentum spectra of photons, di-leptons and hadrons.

Hadron multiplicities and their correlations could provide information about the nature, composition and size of the medium from which they are originating. Very interesting measurement is the extent of equilibration in particle yields. As the QGP is a partonic medium at (or almost at) local thermal equilibrium, its appearance should cause hadronic constituents to approach chemical equilibrium at the time of hadronization. To better understand this, let us discuss the particle production mechanisms as proposed by statistical models and consult them with experimental data from heavy ion experiments (RHIC, SPS,...).

2.1 Initial conditions and deconfinement

There are two things in relativistic heavy ion collisions; the critical energy density ϵ_c and the equation of state (EoS). The energy density is needed to establish the necessary initial conditions to create QGP and the EoS is needed to describe the space-time evolution of the system. Both of these can be obtained from Lattice QCD calculations. The system evolution seems to have typical evolution for a phase transition, an abrupt change around very narrow temperature range around the critical temperature T_c . The Lattice Gauge Theory (LGT) gives us in two flavor QCD critical temperature $T_c = 173 \pm 8 \text{ MeV}$ and a corresponding energy density $\epsilon_c = 0.6 \pm 0.3 \text{ GeV}/\text{fm}^3$ for the phase transition. The initial energy density ϵ_0 can be estimated from the transverse energy E_T as:

$$\epsilon_0(\tau_0) = \frac{1}{\pi R^2} \frac{1}{\tau_0} \frac{dE_T}{dy}, \quad (1)$$

where the initially produced fireball has the shape of a cylinder of transverse radius $R \sim A^{1/3}$ and length $dz = \tau_0 dy$. Assuming that πR^2 is the overlap area of the colliding Pb nuclei and initial time $\tau_0 \simeq 1 \text{ fm}/c$ and using the average transverse energy at midrapidity measured by SPS ($\sqrt{s} = 17.3 \text{ GeV}$) to be 400 GeV , we obtain

$$\epsilon_0^{SPS}(\tau_0 \simeq 1 \text{ fm}/c) = 3.5 \pm 0.5 \text{ GeV}/\text{fm}^3 \quad (2)$$

At RHIC ($\sqrt{s} = 130 \text{ A GeV}$ for Au-Au) with the same thermalization time as at SPS, they obtained ϵ_0 higher by a factor of 1.5–1.6. However, the thermalization time was discussed to be 3–5 times shorter using different models. The expected value for LHC is in the range $400 < \epsilon_0^{LHC} < 1300 \text{ GeV}/\text{fm}^3$, which corresponds to an initial temperature in the range $1 \text{ GeV} < T^{LHC} < 1.2 \text{ GeV}$. Both these values largely exceed the critical values ϵ_c and T_c respectively. Main constituents of the partonic medium created in nucleus-nucleus collision are gluons. The initially created gluons are far from being thermally distributed, the

system needs time to equilibrate, so despite large energy density, the QGP does not need to be created. But at RHIC and LHC, it was shown, that the equilibration of partons should definitely happen.

Let us assume the formation of QGP and its thermal nature. We can then expect the thermal nature to be conserved during the hadronization and in this case we should expect the distribution of the final particles to be thermal too.

2.2 Statistical approach

First, we have to define the equilibrium. An ensemble is in thermodynamic equilibrium, when it occupies uniformly the available phase space. This condition is necessary and sufficient. The statistical operator, which describes the system, depends only on the temperature T and baryon chemical potential μ_B . Plus we need to use grand-canonical formalism to describe the system as the number of particles is not constant. To describe the system composition, we will need the Grand Canonical partition function

$$Z^{GC}(T, V, \mu_Q) = Tr \left[e^{\beta(-\sum_i \mu_{Q_i} Q_i)} \right], \quad (3)$$

where H is the system Hamiltonian, Q_i are the conserved charges and μ_{Q_i} that guarantee the conservation of the charges Q_i in the average over the whole system. $\beta = 1/T$ is the inverse temperature. The Hamiltonian usually used is that of hadron resonance gas. For practical reasons, the hadron mass spectrum has been cut off to $\sim 1.5 GeV$ for mesons and $\sim 2 GeV$ for baryons. In these ranges the resonances and decays are well described. This limits the maximum temperature to $T_{max} \cong 200 MeV$, so up to this temperature this model (with mass cut off) can be used and considered trustworthy. Heavier resonances are not negligible at higher temperatures. They are usually implemented to the model by introducing hard core repulsion, i.e. a Van der Waals-type interaction. The used Hamiltonian contains all relevant degrees of freedom of confined strongly interacting medium and includes interactions that result in resonance formation. What's more, this model is consistent with the equation of state predicted by the LGT below critical temperature.

In such medium we want to conserve electric charge Q , baryon number B and strangeness S . Therefore we introduce $\vec{\mu} = (\mu_B, \mu_S, \mu_Q)$ with the chemical potentials μ_i related to baryon number, strangeness and electric charge respectively. Now we can write the partition functions as a sum of

$$\ln Z(T, V, \vec{\mu}) = \sum_i \ln Z_i(T, V, \vec{\mu}) \quad (4)$$

partition functions of all hadron resonances. For a particle i , we write its energy as $\epsilon_i = \sqrt{p^2 + m_i^2}$, its strangeness S_i , baryon number B_i , electric charge Q_i and spin-isospin degeneracy factor g_i . Then its partition function reads:

$$\ln Z_i(T, V, \vec{\mu}) = \frac{V g_i}{2\pi^2} \int_0^\infty \pm p^2 dp \ln (1 \pm \lambda_i e^{\beta \epsilon_i}) \quad (5)$$

with (+) for fermions and (-) for bosons and the fugacity factor

$$\lambda_i(T, \vec{\mu}) = \exp \left(\frac{B_i \mu_B + S_i \mu_S + Q_i \mu_Q}{T} \right). \quad (6)$$

Next, we can expand the logarithm and integrate over the momentum. We have

$$\begin{aligned} \ln Z_i(T, V, \vec{\mu}) &= \frac{Vg_i}{2\pi^2} \int_0^\infty \pm p^2 dp \left[\sum_{n=0}^\infty \frac{(-1)^n}{n+1} (\pm \lambda_i e^{\beta \epsilon_i})^{n+1} \right] = \dots \\ \dots &= \frac{VTg_i}{2\pi^2} \sum_{k=1}^\infty \frac{(\pm 1)^{k+1}}{k^2} \lambda_i^k m_i^2 K_2 \left(\frac{km_i}{T} \right), \end{aligned} \quad (7)$$

where $K_2(x)$ is the modified Bessel function, (+) for bosons and (-) for fermions. The first term in (7) corresponds to Boltzmann approximation. The density of particle i is obtained from (7) as

$$n_i(T, \vec{\mu}) = \frac{N_i}{V} = \frac{Tg_i}{2\pi^2} \sum_{(k=1)}^\infty \frac{(\pm 1)^{k+1}}{k} \lambda_i^k m_i^2 K_2 \left(\frac{km_i}{T} \right) \quad (8)$$

The partition function (4) is the basic quantity that allows description of all thermodynamical properties of the fireball of hadrons and resonances in thermal and chemical equilibrium.

We will concentrate on particle density in view of further applications to the description of particle production in heavy ion collisions. Here we take into account the resonances and their decays to lighter particles. The average number of particles in volume V of temperature T that carries strangeness S_i , baryon number B_i and electric charge Q_i is

$$\langle N_i \rangle(T, \vec{\mu}) = \langle N_i \rangle^{th}(T, \vec{\mu}) + \sum_j \Gamma_{j \rightarrow i} \langle N_j \rangle^{th,R}(T, \vec{\mu}), \quad (9)$$

where the first term describes the thermal average number and the second term describes contributions from particles j that decays into particle i with branching ratio $\Gamma_{j \rightarrow i}$. The thermal terms are obtained from equation (8). At high temperature (or density), the "decay" term dominates over the thermal. In high density regime (high temperature T and/or chemical potential μ_B the repulsive interactions (such as the hard core repulsion for short distance repulsion) should be included in the partition function (4). Usually this is done by implementing excluded volume corrections, which leads to a shift of the baryo-chemical potential μ_B . These corrections are important when we discuss observables of density type, however, the particle density ratios are only weakly affected.

There are five parameters in the model, but only three of them are independent. The isospin asymmetry fixes the charge chemical potential μ_Q and the strangeness neutrality condition eliminates the strange chemical potential. For the particle multiplicity ratios, we are left with only two independent parameters μ_B and T . The relation $\mu_S = \mu_S(T, \mu_B)$ on fig.(1) is obtained from strangeness neutrality condition. At high temperature μ_S is almost linear with μ_B as $\mu_S \sim \frac{1}{3}\mu_B$.

On the other hand, at lower energies ($T < 100 MeV$), the widths of resonances have to be included in eq.(9), because the number of light particles coming from the decays of resonances is increased by the finite width. Approximation of the resonance width by a δ -function is not justified. The partition

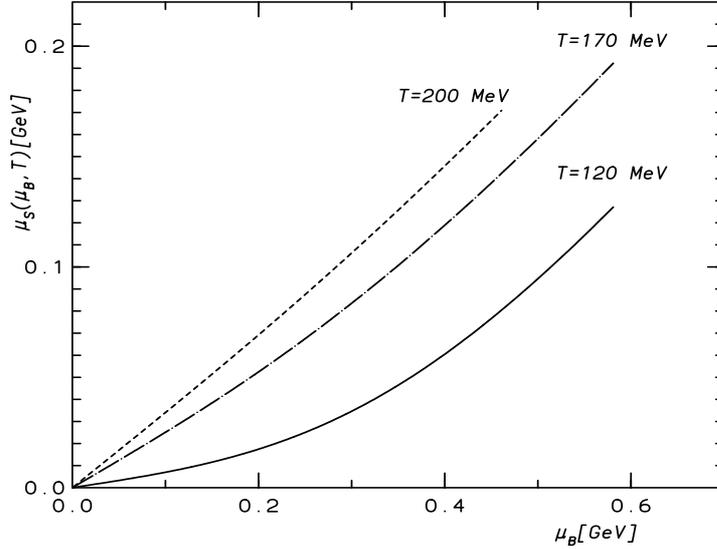


Figure 1: The strange chemical potential μ_S as a function of baryon-chemical potential for $T=120,170$ and 200 MeV. The results are obtained by imposing the strangeness neutrality condition in a hadron resonance gas.(figure from [8])

function then reads:

$$\ln Z_R = N \frac{V d_R}{2\pi^2} T \exp \left(\frac{B_R \mu_B + Q_R \mu_Q + S_R \mu_S}{T} \right) \cdot \int_{S_{min}}^{S_{max}} ds s K_2 \left(\frac{\sqrt{s}}{T} \right) \frac{1}{\pi} \frac{m_R \Gamma_R}{(s - m_R^2)^2 + m_R^2 \Gamma_R^2} \quad (10)$$

where s_{min} is the threshold value for the resonance decay and $s_{max} \sim m_R + 2\Gamma_R$. This statistical model was applied to describe particle yields in heavy ion collisions. Hadron multiplicities ranging from pion to Ω baryons and their ratios were used to verify that there is a set of parameters (T, μ_B) , which reproduces all measured yields and such parameters were successfully found.

2.2.1 Implementation of conservation laws

Particle yields in central heavy ion collisions are well described assuming a complete thermalized state at fixed temperature and baryo-chemical potential in energy from AGS up to LHC. In central heavy ion collision, the statistical hadron resonance gas model accounts for conservation laws of baryon number, strangeness and electric charge in GC ensemble. For description of either peripheral heavy ion, hadron - hadron or low energy collisions, the statistical operator (3) would need to be changed.

In the statistical approach, particle production can be only described using grand-canonical ensemble with respect to conservation laws, if the number of produced particles carrying a conserved charge is sufficiently large. The net

value of given charge fluctuates from event to event. These fluctuations can be neglected if particles carrying the charge are abundant. On the other side, if the particle production is low, the fluctuations can be as high as the event averaged value. In this case, one has to implement the exact charge conservation in each event.

The exact conservation of quantum numbers introduces a constraint on a thermodynamical system. Therefore the time dependency and equilibrium distribution of particle multiplicity can differ from the expected one in the Grand Canonical (GC) limit. In order to see the differences, one needs to formulate the time equations for particle production and evolution. In the partonic medium this requires, in general, formulation of the transport equation. In the hadronic medium, the charge conservations related to $U(1)$ internal symmetry need to be taken into account.

2.3 Kinetics of time evolution and equilibration of charged particles

In order to study chemical equilibrium in a hadronic medium, we will introduce a kinetic model that includes the production and annihilation of particle-antiparticle pairs $c\bar{c}$ carrying $U(1)$ quantum numbers like strangeness or charm. Furthermore, it is assumed that such particles are produced via a binary process $ab \rightarrow c\bar{c}$ and the particle momentum distributions are thermal and described by the Boltzmann statistics.

The a, b particles are charge neutral constituents of the fireball of temperature T and volume V . We will describe time evolution and equilibration of particles c and \bar{c} inside the fireball as it is limited by the constraints imposed by the $U(1)$ symmetry. Next, we will formulate a general master equation for the probability distribution of particle multiplicity in a medium with vanishing net charge and consider its properties and solutions for particular cases.

2.3.1 Kinetic master equation for probabilities

Let us note $P_{N_c}(\tau)$ the probability to find N_c particles c , where $0 \leq N_c \leq \infty$. This probability will change with time as the creation $ab \rightarrow c\bar{c}$ and the annihilation $c\bar{c} \rightarrow ab$ processes will take place.

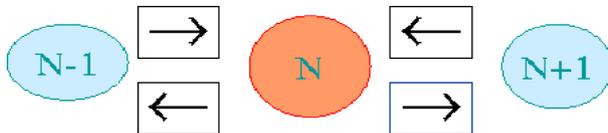


Figure 2: A schematic view of the master equation for the probability P_N due to $ab \rightarrow c\bar{c}$ and the inverse process

Two main terms will play role in the rate equations; the magnitude of transition probability per unit time of production G/V and absorption L/V of $c\bar{c}$

pairs. The gain ($G = \langle \sigma_{ab \rightarrow c\bar{c}} v_{ab} \rangle$) and the loss ($L = \langle \sigma_{c\bar{c} \rightarrow ab} v_{c\bar{c}} \rangle$) terms represent the momentum average of particle production and absorption cross sections. The transition $N_c + 1 \rightarrow N_c$ probability per unit time is given by the product of the probability L/V and number of possible reactions, which is $(N_c + 1)(N_{\bar{c}} + 1)$, assuming that $c\bar{c}$ symmetry is locally exactly conserved ($N_c = N_{\bar{c}}$). Then this factor can be written as $(N_c + 1)^2$. The transition probability of $N_c \rightarrow N_c + 1$ is described by $G \langle N_a \rangle \langle N_b \rangle / V$, where we assume particles a and b not to be correlated, and their multiplicity is governed by the thermal averages. We assume also, that their multiplicities are not significantly affected by either of the processes $ab \rightarrow c\bar{c}$ and $c\bar{c} \rightarrow ab$. Then the master equation of probability $P_{N_c}(\tau)$ can be written as:

$$\begin{aligned} \frac{dP_{N_c}}{d\tau} &= \frac{G}{V} \langle N_a \rangle \langle N_b \rangle P_{N_c-1} + \frac{L}{V} (N_c + 1)^2 P_{N_c+1} - \\ &- \frac{G}{V} \langle N_a \rangle \langle N_b \rangle P_{N_c} - \frac{L}{V} N_c^2 P_{N_c} \end{aligned} \quad (11)$$

First two terms describe increase of P_{N_c} and the second two the decrease of P_{N_c} . The thermal averaged cross sections under Boltzmann approximation of the process $ab \rightarrow c\bar{c}$ are obtained from

$$\langle \sigma_{ab \rightarrow c\bar{c}} v_{ab} \rangle = \frac{\beta \int_{t_0}^{\infty} dt \sigma_{ab \rightarrow c\bar{c}}(t) [t^2 - (m_{ab}^+)^2] [t^2 - (m_{ab}^-)^2] K_1(\beta t)}{8 m_a^2 m_b^2 K_2(\beta m_a) K_2(\beta m_b)}, \quad (12)$$

where K_1, K_2 are the modified Bessel functions of the second kind, $m_{ab}^+ = m_a + m_b$, $m_{ab}^- = m_a - m_b$, $t = \sqrt{s}$ is the CMS energy, $\beta = 1/T$ and

$$v_{ab} = \frac{(k_a k_b)^2 - m_a^2 m_b^2}{E_a E_b}$$

is the relative velocity of the incoming particles and the integration limit $t_0 = \max[(m_a + m_b), (m_c + m_{\bar{c}})]$. The equation (11) allows one to calculate the time evolution of the momentum averages of particle multiplicities and their arbitrary moments. After multiplying the above equation by N_c and summing over it, we obtain general kinetic equation for the time evolution of the average number $\langle N_c \rangle = \sum_{N_c=1}^{\infty} N_c P_{N_c}(t)$ of particles c in the system. The equation reads

$$\frac{d\langle N_c \rangle}{d\tau} = \frac{G}{V} \langle N_a \rangle \langle N_b \rangle - \frac{L}{V} \langle N_c^2 \rangle \quad (13)$$

This equation connects multiplicity $\langle N_c \rangle$ with its second moment $\langle N_c^2 \rangle$ and therefore cannot be solved analytically. However, we can look at two limiting situations.

- i). for abundant production of c particles, $\langle N_c \rangle \gg 1$ and
- ii). when c particles are very rare, when $\langle N_c \rangle \ll 1$

Since

$$\langle N_c^2 \rangle = \langle N_c \rangle^2 + \langle \delta N_c^2 \rangle,$$

where $\langle \delta N_c^2 \rangle$ represents fluctuations of numbers of particles c .

i). For $\langle N_c \rangle \gg 1$, $\langle N_c^2 \rangle \approx \langle N_c \rangle^2$ and equation (13) reduces to

$$\frac{d\langle N_c \rangle}{d\tau} = \frac{G}{V} \langle N_a \rangle \langle N_b \rangle - \frac{L}{V} \langle N_c \rangle^2 \quad (14)$$

ii). on the other hand, when $\langle N_c \rangle \ll 1$, particles c and \bar{c} strongly correlated and thus $\langle N_c^2 \rangle \approx \langle N_c \rangle$ and

$$\frac{d\langle N_c \rangle}{d\tau} = \frac{G}{V} \langle N_a \rangle \langle N_b \rangle - \frac{L}{V} \langle N_c \rangle \quad (15)$$

We can see that depending on the thermal conditions in the system (the volume and the temperature), we are getting different results for the equilibration and for the time evolution of the number of produced particles c . The above equations (14,15) have the following solutions

$$\langle N_c(\tau) \rangle = \langle N_c \rangle_{eq} \tanh\left(\frac{\tau}{\tau_0}\right)$$

$$\langle N_c(\tau) \rangle^C = \langle N_c \rangle_{eq}^C \left(1 - e^{-\frac{\tau}{\tau_0^C}}\right)$$

respectively. The constants $\langle N_c \rangle_{eq}$ and τ_0 , and $\langle N_c \rangle_{eq}^C$, τ_0^C respectively denote the equilibrium number of particles c and the relaxation time constant. In the second case the index C stands for Canonical ensemble as for $\langle N_c \rangle \ll 1$ we get to Canonical limit of the Grand Canonical ensemble approach we are working with. These constants are given by:

$$\langle N_c \rangle_{eq} = \sqrt{\varepsilon} \quad ; \quad \tau_0 = \frac{V}{L\sqrt{\varepsilon}}$$

$$\langle N_c \rangle_{eq}^C = \varepsilon \quad ; \quad \tau_0^C = \frac{V}{L}.$$

We can conclude that the two limits (i) and (ii) are essentially determined by the size of the fluctuation $\langle \delta N_c^2 \rangle$. The Grand Canonical result corresponds to small fluctuations, $\frac{\langle \delta N_c^2 \rangle}{\langle N_c \rangle^2} \leq 1$ and the canonical result, on the other hand, apply with large fluctuations $\frac{\langle \delta N_c^2 \rangle}{\langle N_c \rangle^2} \geq 1$.

2.4 Unified conditions of particle freeze-out

From data analysis of heavy-ion collisions from SIS, AGS, SPS and RHIC it is clear, that the Canonical or Grand Canonical statistical model reproduces the most of the measured hadron yields. Going from lower energies at SIS to higher energies at RHIC, the freeze-out temperature raises and the baryo-chemical potential decreases (see figure 3).

All three points have one common feature, the average energy $\langle E \rangle$ per average number of hadrons $\langle N \rangle$ is approximately 1 GeV. A chemical freeze-out is reached, whenever the ratio $\langle E \rangle / \langle N \rangle$ drops below 1 GeV.

In cold nuclear matter the $\langle E \rangle / \langle N \rangle$ is approximately determined by the nucleon mass. For thermally excited matter, we can use the non-relativistic approximation

$$\frac{\langle E \rangle}{\langle N \rangle} \simeq \langle m \rangle + \frac{3}{2}T \quad (16)$$

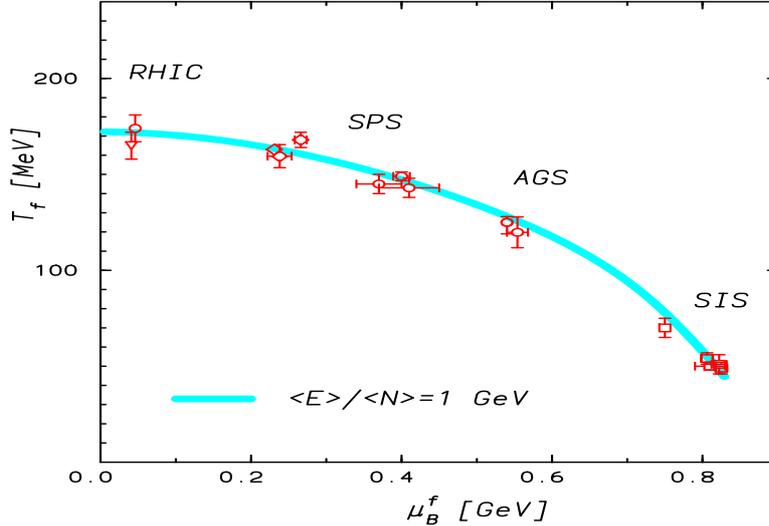


Figure 3: A compilation of chemical freeze-out parameters appropriate for A–A collisions at different energies; The full line represents the phenomenological condition of a chemical freeze-out at the fixed energy/particle $\simeq 1.0$ GeV

where $\langle m \rangle$ is thermal average mass in the collision fireball. This is consistent with relatively low temperature $T \approx 53 \text{ MeV}$ at SIS and higher temperatures at SPS and RHIC. Same for the particle yields (and therefore $\langle m \rangle$) which differ at freeze-out.

The freeze-out condition of fixed energy/particle is justified by the UrQMD model and is interpreted as the condition of inelasticity in heavy ion collisions. At lower energies, however, this condition overestimates the temperature as seen from figure 4.

It may be caused by composite objects (like He or Li nuclei), which are not proved to be of thermal origin at SIS energy. For phenomenological determination of freeze-out parameters for different energies, we can use the requirement $\langle E \rangle / \langle N \rangle \simeq 1 \text{ GeV}$.

The above condition provides relation between the temperature T and the chemical potential μ_B at all collision energies. So it is sufficient to have only one measured ratio (e.g. ratio pion/participants) to establish the energy dependence of the two thermal parameters T and μ_B . Thus, by interpolation and/or parametrization of the energy dependence of the μ_B and then using the unified freeze-out condition $\langle E \rangle / \langle N \rangle \simeq 1 \text{ GeV}$ we can get energy dependence of T . It

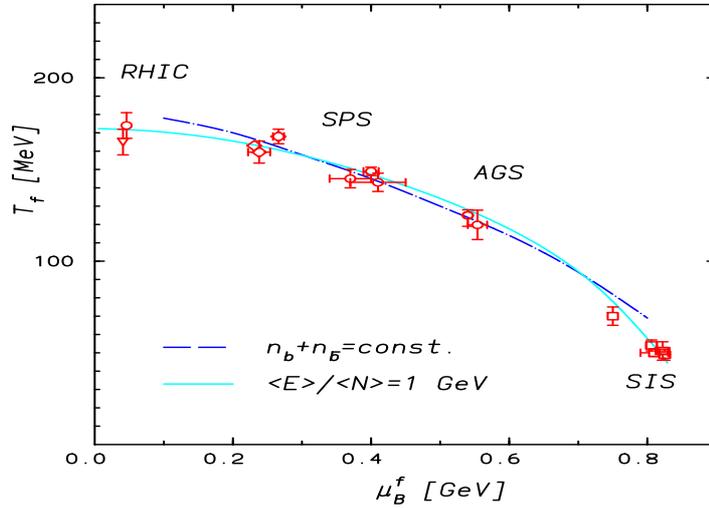


Figure 4: The broken line describes the chemical freeze-out conditions of fixed total density of baryons plus antibaryons, $n_b + n_{\bar{b}} = 0.12/\text{fm}^3$. The full line represents the condition of the fixed energy/particle $\simeq 1.0 \text{ GeV}$

was shown to be well parameterized as:

$$\mu_B(s) = \frac{a}{1 + \frac{\sqrt{s}}{b}}, \quad (17)$$

where $a \simeq 1.27 \text{ GeV}$ and $b \simeq 4.3 \text{ GeV}$.

The knowledge of energy dependance of μ_B allows us to predict particle ratios at different energies.

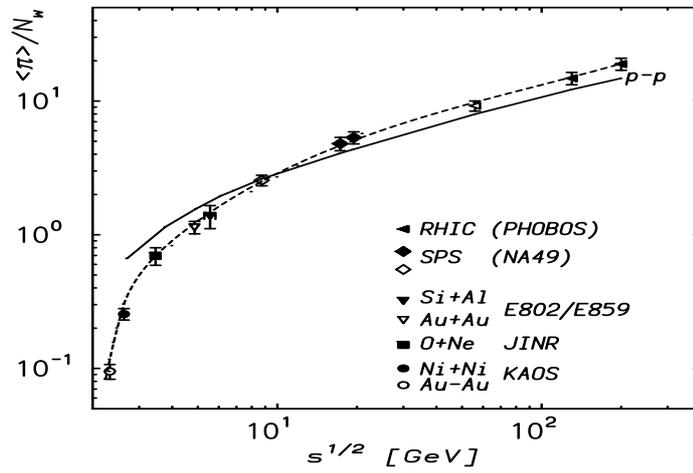


Figure 5: The total number of pions per wounded nucleon ($\langle \pi \rangle / N_w$) versus the center-of-mass energy. The short-dashed and dashed lines are a fit to the data.

3 Baryon Number Flow

In heavy-ion collisions, it is not obvious what carries the baryon number in a proton. The baryon number (BN) itself is defined as the number of quarks minus the number of antiquarks divided by three. In a conserved system, this quantity is conserved. Defined like this, the baryon number could be associated with the density of valence quarks, which is defined as $q_i^v(x) = q_i(x) - \bar{q}_i(x)$ where i is the flavor of the quark and x its momentum fraction. Immediately, it follows for a proton:

$$\sum_i \int_0^1 dx [q_i(x) - \bar{q}_i(x)] = \sum_i \int_0^1 dx q_i^v(x) = 3, \quad (18)$$

which is the same as the BN of a proton (after the division by three). This could lead to a not correct conception that the valence quarks are carrying the BN. Let us take the reaction

$$\pi^- + p \longrightarrow \Omega^- + K^+ + 2K^0.$$

The baryon number is conserved, but either of valence quarks of the initial proton is a valence quark in the Ω^- . The gluon fields have created three $s\bar{s}$ pairs, the \bar{s} together with $2d$ and u formed the K mesons and the three s quarks formed the Ω^- . This shows, that some partons other than valence quarks carry the baryon number. Probably gluons are responsible [9].

Another example that speaks against the valence quarks carrying the baryon number are heavy ion collisions at very high energies. In the mid-rapidity region, there should be naively zero net baryon number, because the valence quarks are very difficult to stop. The energy loss of a quark that propagates through a heavy nucleus is rather small, $\Delta E \sim 10$ GeV, and energy independent. In this image, the baryon number would be carried in the straight forward directions, it means to large positive and negative rapidity regions leaving the mid-rapidity free of the net-baryon number. The picture is correct for the behavior of valence quarks, but not for the BN.

The valence quarks carries substantial part of the colliding nucleus energy. The rest of the energy is carried by softer quarks, anti-quarks and gluons. High-energy quark cannot be stopped via soft interactions it sustains passing through a heavy nucleus. It loses only small part of its initial energy via gluon radiation induced by those soft collisions. Many gluons and $q\bar{q}$ pairs are left behind in the collision region. Sometimes, the energy density left behind by such quarks is sufficiently high, that a quark-gluon plasma is created.

The valence quarks after propagation through the heavy nucleus forgets its nature as a constituent of a nucleon, so it emerges independently in the beam direction as a fragmentation jet. This second picture shows, that the BN is not carried by the high energetic valence quarks in the beam direction, but is stuck in the glue near mid-rapidity.

So far three main ideas were mentioned:

1. The valence quarks survive the collision as fragmentation jets correlated with the beam direction, while the BN does not. So the valence quarks are not the carriers of the BN.

2. BN stopping does not necessarily correspond to energy stopping. The energy carried by the valence quarks can penetrate through the colliding nucleus, while the BN does not. On the other hand, when (with very small probability) the energy is stopped in the central region, the BN is stopped as well.
3. It appears that the BN is stopped along with the glue, therefore the gluons are considered to be the carriers of the BN.

Before, it was believed that in a heavy ion collisions, the valence quarks passed through the Lorentz contracted nucleus would carry most of the BN. Leaving behind a region of very hot region of strongly interacting matter with very low BN. But experimental results at CERN SPS shows different. There were much more baryons at mid-rapidity than expected assuming the BN associated with the valence quarks. Moreover, a surprising number of them had a strange valence quark. From there it followed immediately the idea, that the BN is not tied to valence quarks.

3.1 Baryon number distribution

For studying the baryon number variable, we need suitable probes for it. The earliest probe used to investigate BN was $p\bar{p}$ reaction where the annihilation cross section was measured. Another way to study the x distribution of BN is to measure BN asymmetry of produced particles. Let us assume, that the BN of the final hadrons are strongly correlated in rapidity to the initial partons, from which they are produced.

3.1.1 BN annihilation at high energies

In 1970s, experiments on BN annihilation via $p\bar{p} \rightarrow mesons$ were made. Here are summarized some of the main observations. If the BN is associated with the gluons, then it should be evenly distributed over the entire rapidity scale as the gluon is a vector particle. Processes mediated by gluon exchange are energy independent, for example all hadronic total cross sections show themselves to be the same over the energy scale. The energy independence leads also to the fact, that $p\bar{p}$ annihilation cross section does not vanish at high energy. We can employ the string junction model (see section 3.1.3) for the annihilation process. Two different configurations of the gluon strings are possible. String junction and string anti-junction can overlap (Fig 6 bottom), or the proton (resp. anti-proton) consists of di-quark (resp. di-anti-quark) pair and a quark (resp. anti-quark) bound to it (Fig 6 top).

In this approach, we will assume the string length $\sim 0.2 - 0.3 fm$. This leads to a cross section $\sigma_{ann}^{p\bar{p}} \approx 1 - 2 mb$. Perturbative QCD treatment of annihilation perfectly confirms this result.

There is a slight difference between the $p\bar{p}$ and pp cross section. The difference in multiplicity is due to a specific three-string topology of the events with string junction exchange. In the pp collision, the multiplicity is greater 1.5 times than the non-annihilation $p\bar{p}$ events. The results of the analysis gives $\sigma_{ann}^{p\bar{p}} \approx 1.5 \pm 0.1 mb$, which agrees with theoretical expectations. The experiments used to measure this cross section ranged from 10 to 1480 GeV in lab energy, which confirms energy independence of this mechanism. That means that the BN

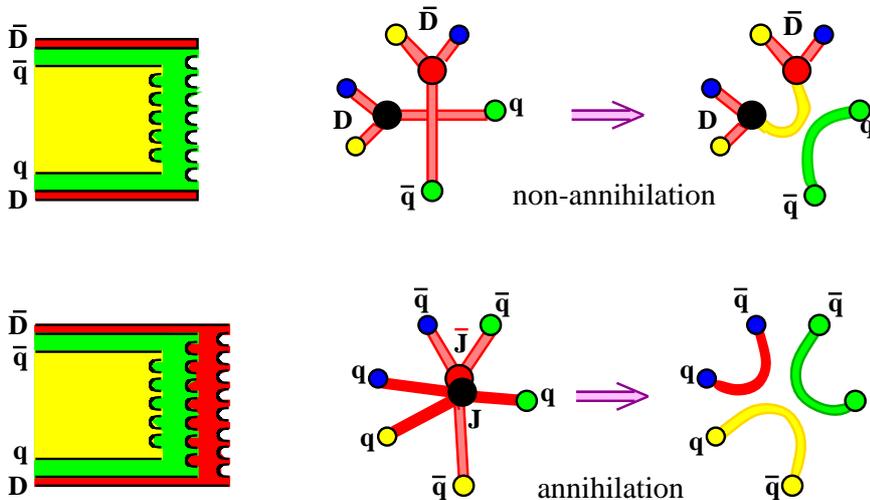


Figure 6: The cartoon shows interaction of a proton consisted of a diquark (D) and a quark with an antiproton.

transport over large rapidity intervals is rapidity independent. Therefore the BN distribution at small x is proportional to $1/x$ [9].

3.1.2 BN asymmetry

Another probe to BN distribution is BN asymmetry defined as

$$A_{BN}(x) = 2 \cdot \frac{N_{BN} - N_{\overline{BN}}}{N_{BN} + N_{\overline{BN}}}, \quad (19)$$

where $N_{BN(\overline{BN})}$ is number of baryons (anti-baryons) created in the event and which is function of Björken x . The BN asymmetry can be interpreted as the ratio of production rate of BN stopped in the interaction region and the BN created from the vacuum. Unfortunately, this is correct only if the effect of stopping is very small. We assume, that the BN asymmetry is caused by the asymmetry of parton distribution in the projectile proton. Therefore, we can say, that any BN excess in lower rapidity, then that of the projectile can be looked at as the measure of BN at small x ($\sim 10^{-3}$) in the partonic distribution of the proton.

Let us take an example of a proton-meson collision in the rest frame of the proton. The high energetic incident meson creates a parton cloud containing many baryon-antibaryon pairs. Despite the fact, that the partonic cloud is symmetric, the BN of the target proton can annihilate with anti-BN of an antibaryon fluctuation from the cloud. And hence, the BN is now carried by a baryon fluctuation of the incident meson. The asymmetry in this case is given by:

$$A_{BN}(x) = \frac{\sigma_{ann}(s = m_N^2/x)}{\sigma_{in}^{MN}}, \quad (20)$$

where σ_{in}^{MN} is the inelastic meson-nucleon cross section. When we use the asymptotic value of $\sigma_{ann}^{\bar{p}p} = 1.5 mb$ and $\sigma_{in}^{MN} = 20 mb$, the asymmetry $A_{BN} =$

7% is predicted [9].

The H1 collaboration measured the asymmetry using the $\gamma - -p$ interaction. The preliminary results $A_{BN}(x = 10^{-3}) = 8 \pm 1 \pm 2.5\%$ are in a good agreement with the prediction mentioned above.

Next, we can define the asymmetry for each particular species of baryon B;

$$A_B(x) = 2 \cdot \frac{N_B - N_{\bar{B}}}{N_B + N_{\bar{B}}}. \quad (21)$$

The produced BN is realized via many different baryons with corresponding relative branching ratios. The contribution of each type of baryon is different, for some it will approach the maximal value of 2 and the BN asymmetry will still be rather small. At this point we cannot characterize the BN flow with A_B .

3.1.3 BN flow

The string configuration is quite different in mesons and in baryons. In mesons, there is a $q\bar{q}$ pair connected by a color flux tube. In baryons, there are three quarks connected by three color strings in a shape of Mercedes sign. The point in which the strings are connected is called string junction. Analogically for anti-baryons and anti-junction. When junction and anti-junction interact, they can annihilate into mesons. The association of BN with the topology of gluonic fields rather than with the quarks is not new. It is compatible with the BN assigned to gluons as presented earlier in this work. The gluons themselves don't have any BN, the dynamical association of BN with the gluonic fields is explained in the following paragraphs.

In the infinite momentum frame, the string junctions share the proton momentum and therefore can be given a partonic interpretation. Considering the Fock decomposition of the proton containing sea quarks $q\bar{q}$

$$|p\rangle = |3q^v\rangle + |3q^v q^s \bar{q}^s\rangle + |3q^v 2q^s 2\bar{q}^s\rangle + \dots,$$

the sea quarks can form a baryon by extending the processes illustrated in figure (7). It is thus conceivable that BN might have the distribution $\propto 1/x$ at small x similar to that of gluons and sea quarks.

It is not very probable, though, that the BN is transported to low x by 3 $q\bar{q}$ chains. It requires the production of a $3q$ color decuplet state, which is the cause of the low probability in hadronic collisions. It is needed to remark, that if the baryon is created at low x , it does not need to carry the same flavor as the initial valence quarks.

3.2 Heavy ion collisions

It appears that at high energies the momentum of the projectile valence quarks survive multiple interactions in the collision, but lose their identity as baryonic constituents. In other words, the BN moves from the fragmentation regions and accumulates at central rapidities.

The BN observed so far at SPS remains primarily in the fragmentation regions and only a very small fraction is found at smaller rapidities, because of the spread in the momentum fractions of the valence quarks. On the other hand, in for Pb-Pb collisions it was predicted that almost all BN is stopped. Due to

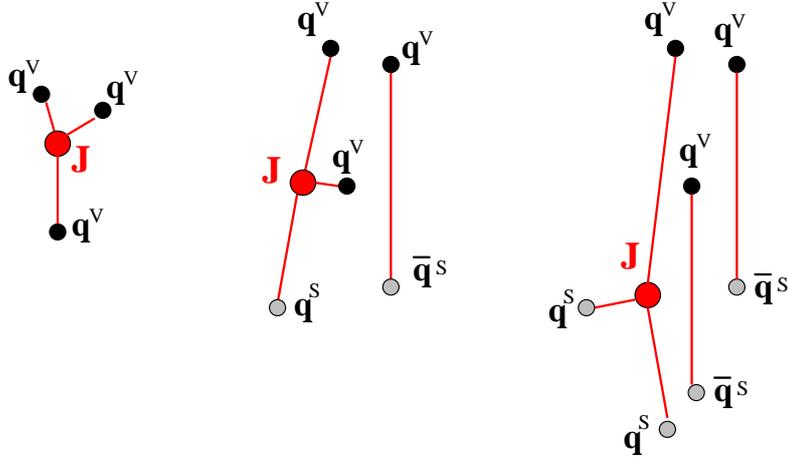


Figure 7: String configurations corresponding to different terms in Fock state decomposition of a baryon.

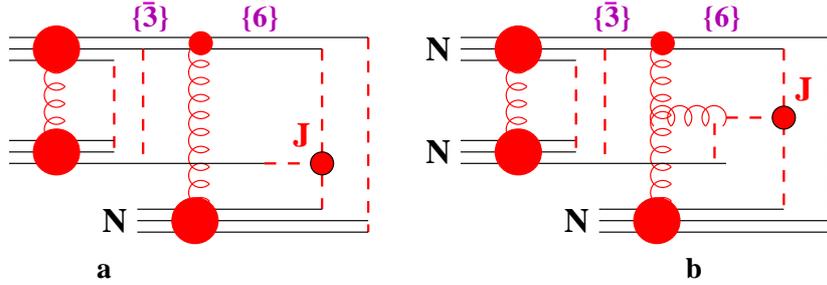


Figure 8: The BN acquires the rapidity of a valence quark in the target nucleon (a) or a radiated gluon (b).

multiple interactions in the colliding nuclei the BN escapes the fragmentation region with probability close to one. These predictions were confirmed by the NA49 collaboration. Let us assume that BN liberated via multiple interactions in heavy ion collisions move to the rapidities of valence quarks similar to NN collisions. This leads to the fact, that the probability to stop BN at central rapidity would decrease with energy as $s^{-1/4}$, in other words one should expect less BN stopping at RHIC and even less at LHC compared to SPS. But there is another energy independent mechanism that was not significant at energies up to SPS. The diquark and quark from the projectile nucleus lose their coherence after the first inelastic interaction on the surface of the nucleus. The projectile BN associated with the diquark and liberated right after in multiple interactions should acquire the rapidity of a valence quark of the target or a gluon radiated at mid rapidities (see Fig. 8).

Although the total probability of BN flow to mid rapidities is energy independent, the distribution between these two mechanisms depends on energy. At high energies it is more probable that the BN will be stuck with one of numerous radiated gluons, while at medium high energies the contribution of valence quarks may be important. It follows from the above consideration, that

almost all BN should be stopped in central Au-Au collisions at RHIC and Pb-Pb collisions at LHC. The BN should be spread over the whole rapidity range. A sensitive test for BN stopping is the fraction of hyperons produced at mid rapidities, especially cascades.

4 AliROOT

As every experiment, ALICE has its offline framework for data analysis, which is called AliROOT. It uses the ROOT [7] system as a foundation on which the framework for simulation, reconstruction and analysis is built. Except for large existing libraries, such as Pythia6 and HIJING, and some remaining legacy code, this framework is based on the Object Oriented programming paradigm and is written in C++.

Fully operational AliROOT contains of a few separate packages. The ROOT [7] system, AliROOT [5], and one has to install at least one of the particle transport packages GEANT 3, FLUKA or GEANT 4 depending on whichever suits one's needs the best.

4.1 Framework

In high energy physics a framework consists of a set of software tools able to process data. The primary interaction is simulated by an *event generator* (which doesn't need to be part of the framework, but usually is). The event generator produces a set of particles with their momenta. They are being stored on a form of kinematic tree of mother-daughter relationships and production vertices. The *transport* package does the transport of particles through a set of detectors and produces *hits*, which means in ALICE terminology energy deposit at certain point. In some detectors, the energy deposit is used only to be compared with a given threshold (TOF or ITS), in others the energy deposit is used for particle identification as well. Having this done, the detector response is transformed into *digits*. They are two types of digits. *Summable digits* are generated with low thresholds and the result is additive. It is at this point, where one can combine data, e.g. different backgrounds with isolated signals, or data from different event generators. The other type is called *digits* and the real thresholds are used. They are very similar to the real data one gets from the detector. They are both conversions chains provided in AliROOT, $hits \rightarrow summable\ digits \rightarrow digits$ and $hits \rightarrow digits$. The two main differences between the raw data and the digits are, that raw data is stored in a binary format and the digits in AliROOT classes and secondly the digits remember their Monte Carlo information. The raw data doesn't remember it. The reconstruction can be done with both digits and raw data. After digits are created, the reconstruction and analysis chain can be activated. One can then evaluate the detector and software performance. The whole system is made modular, so user can replace any part of the code with his own. But still there are parts of the system that are not intended to be replaced, such as user interface and such.

4.2 Installation

A more detailed guide to install AliROOT is [5] or more recent version on the web.

The main platform for AliROOT is Linux system running on Intel 32-bit with gcc compiler. Other platforms are known to be compatible as well, such as 64-bit Intel, 64-bit AMD, 32-bit AMD, other Unix systems, icc and cc compilers and even MacOS. Further installation remarks can be found also in [6].

4.2.1 Environmental Variables

Before the installation itself, several environmental variables must be set. In the following examples the user is working on Linux and the default shell is bash. It is enough to add to the `.bash_profile` file a few lines as shown below:

```
# ROOT
export ROOTSYS=/home/mydir/root
export PATH=$PATH\:$ROOTSYS/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ROOTSYS/lib

# AliROOT
export ALICE=/home/mydir/alice
export ALICE_ROOT=$ALICE/AliRoot
export ALICE_TARGET='root-config --arch'
export PATH=$PATH\:$ALICE_ROOT/bin/tgt_${ALICE_TARGET}

# GEANT 3
export PLATFORM='root-config --arch'
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ALICE/geant3/lib/tgt_${ALICE_TARGET}
```

where `"/home/mydir"` has to be replaced with the actual directory path. The meaning of the environment variables is the following:

`ROOTSYS` – the place where the ROOT package is located;

`ALICE` – top directory for all the software packages used in ALICE;

`ALICE_ROOT` – the place where the AliROOT package is located, usually as sub-directory of ALICE;

`ALICE_TARGET` – specific platform name. Up to release v4-01-Release this variable was set to the result of `'uname'` command. Starting from AliRoot v4-02-05 the ROOT naming schema was adopted, and the user has to use `'root-config -arch'` command.

`PLATFORM` – the same as ALICE TARGET for the GEANT 3 package. Until GEANT 3 v1-0 the user had to use `'uname'` to specify the platform. From version v1-0 on the ROOT platform is used instead `'root-config -arch'`.

4.2.2 Obtaining packages

It is necessary to install ROOT as the first package. This can be done in different ways. The user can download a pre-compiled package, or download the source code from a CVS (Concurrent Version System) repository, which comprises the following:

1. Login to the ROOT CVS repository

```
% cvs -d :pserver:cvs@root.cern.ch/user/cvs login
% CVS password: cvs
```
2. Downloading the needed version of ROOT (v5-08-00 in the example)

```
cvs -d :pserver:cvs@root.cern.ch:/user/cvs co -r v5-08-00 root
```

A list of matching versions of ROOT, GEANT 3 and AliROOT releases can be found at <http://aliweb.cern.ch/Offline/releases.html>

3. Now the source of ROOT is stored in a directory called `root`. The user have to go there and set up an environment variable `ROOTSYS` to the full path of this directory. For example:

```
cd root
export ROOTSYS='pwd'
```

4. Now you can compile the ROOT running:

```
make
```

Then it is necessary to install GEANT 3 using the following commands:

```
cvs -qz9 -d :pserver:cvs@root.cern.ch:/user/cvs co -r v1-6 geant3
```

Then install it typing:

```
cd geant3
make
```

And as the final step one can obtain the AliROOT package from the CVS server at `alisoft.cern.ch:/soft/cvsroot` using the following commands:

```
cvs -d :pserver:cvs@alisoft.cern.ch:/soft/cvsroot login
CVS password: cvs
cvs -qz9 -d :pserver:cvs@alisoft.cern.ch:/soft/cvsroot \
co -r v4-04-Release AliRoot
```

And then install it with:

```
cd AliRoot
make
```

Further options (as cleaning installation, separate module compilation and others) are neatly specified in [5].

Now the AliRoot should be correctly installed and ready for use. By typing in `alroot` the AliROOT C/C++ interpreter should start. Like in the ROOT system, single lines of the code can be used as well as running scripts and macros from external files.

4.3 Simulation

Heavy ion collisions are quite a challenge for both the reconstruction and analysis algorithms. The models predict about 1400 to 8000 particles in the central unit of rapidity in the final state. The development of these algorithms requires a precise simulation of the detector response. Even though recent experiments (RHIC) show, that the number of particles in the final state will be near the lower end of the interval mentioned above, the detectors need to be ready for the highest multiplicity, because the extrapolation to the LHC energy is so far

away from the SPS and RHIC energies (20AGeV and 200AGeV respectively) and therefore is not very reliable. Moreover, the predictions from different heavy ion collisions generators differ substantially at LHC energies. Still, we use several of them and compare the results to have a range of possible results.

The AliRoot environment provides an access to external generators like PYTHIA, HIJING, DPMJET, is able to assemble events from signals from different generators and add signal-free background at the level of primary particles (cocktail) or at the summable digits level (merging).

4.3.1 Event generation

To facilitate the use of different generators, an abstract generator interface class `AliGenerator` has been created. Several event generators were already accessible through a ROOT class `TGenerator`. Through the implementations of this abstract base class we wrap FORTRAN Monte Carlo codes like PYTHIA, HERWIG and HIJING that are thus accessible from the AliRoot classes (see Fig. 9).

Figure 9: The `AliGenerator` is the base class, which is responsible for primary particle generation of an event. Some realizations of this class do not generate particles themselves, but delegate the task to an external generators.

PYTHIA6

PYTHIA is used for simulation of proton-proton interaction and for generation of jets in case of event merging.

HIJING

HIJING (Heavy-Ion Jet Interaction Generator) combines a QCD-inspired model of jet production with the Lund model for jet fragmentation. Hard or semi-hard parton scattering with transverse momentum of a few GeV are expected to dominate the high-energy heavy-ion collisions. The HIJING model was developed with special emphasis on the role of mini jets in pp, pA and A-A reactions at collider energies.

The Lund FRITIOF model and the Dual Parton Model (DPM) have lead to the formulation of HIJING for soft nucleus-nucleus reactions at intermediate energies, $\sqrt{s_{NN}} \approx 20 \text{ GeV}$. The hadronic-collision model has been inspired by the successful implementation of perturbative QCD processes in PYTHIA. Binary scattering with Glauber geometry for multiple interactions are used to extrapolate to pA and A-A collisions.

Besides these two, there are a few more generators for specific studies such as:

- MEVSIM – developed for STAR to quickly produce large number of A-A collisions. General observables like particle momentum and rapidity distributions, particle multiplicities and even flow can be measured. Developed originally in FROTRAN.

- GeVSim – based on MEVSIM, but written completely in C++ and designer for higher energies. It is a general event generator for observing similar observables as the MEVSIM package.
- HBT processor – introduces two particle correlation into generated events by another generator. Normally, the correlation functions are flat at the region of small relative momenta. It shifts the momenta of each particle so it fits a correlation function of a pre-selected model. More than a generator, the HBT processor is a so called afterburner.

Combination of generators

In the AliRoot environment, the user can combine these generators via an abstract class `AliGenCocktail`, which is an example of `AliGenerator`, which does not generate the events itself, but delegates the task to others, in this case several generators at the same time. Each of those can be connected as `AliGenCocktailEntry` at run time.

4.3.2 Transport

It is crucial to minimize the amount of material in the detector region. The simulation was instrumental in optimizing the detectors' design to save costs without a negative impact on the physics.

L3 and dipole magnets

The L3 and dipole magnets are described both their magnetic fields and material distributions. The magnetic field description includes their interference between the two. The fields are described for three independent maps for 0.2, 0.4 and 0.5T solenoid L3 magnetic field strengths. For faster transport of the particles, there is a parametrization of the magnetic field, which is constant solenoidal field inside the barrel and a dipole field which varies along the z axis in the muon arm.

The support frame of the barrel detectors is described according to its final design.

The design of the beam pipe has been finalized, so every piece of it is represented in the simulation. Basically every detector has both a detailed version of its geometry, used to study its performance, and a coarse version that provides the correct material budget with minimal details. Its used to study influence on other detectors without any detector response. All of the detector geometries are already implemented in AliRoot.

ITS

The ITS has both a coarse and detailed version. The detailed geometry is crucial for determining the impact parameter. On the other hand the coarse geometry is much faster whenever ITS hits are not needed.

TPC

For TPC there are three possible geometry configurations. Version 0 is a coarse geometry, without any sensitive element present. It is used for a material budget for outer detectors. Version 1 is the geometry for the Fast Simulator. The sensitive volumes are thin gaseous strips placed in the Small (S) and Large (L) sectors at the pad-row centers. The hits are created whenever a track crosses the sensitive volume. The energy loss is not taken into account. Version 2 is the geometry version for the slow simulator. The sensitive volumes are S and L sectors. The user can even specify either all or only a few of them to be used. The hits are produced after every ionizing collision. The transport step is calculated for every collision from an exponential distribution. The energy loss is calculated from an $1/E^2$ distribution and the response is parameterized by a Mathieson distribution.

TRD

The TRD geometry is now quite complete, including the correct material budget for electronics and cooling pipes. The full response and digitization are implemented. The transition-radiation photon yield is approximated by an analytical solution for a foil stack, with adjustment of the yield for a real radiator, including foam and fiber layers from test beam data. This is quite a challenging detector to simulate, as both normal energy loss in the gas and absorption of transition-radiation photons have to be taken into account.

During the signal generation several effects are taken into account: diffusion, 1-dimensional pad response, gas gain and gain fluctuations, electronics gain and noise, as well as conversions to ADC values. Absorption and $\mathbf{E} \times \mathbf{B}$ effects will be introduced.

TOF

The TOF detector covers a cylindrical surface of polar acceptance $|\theta - 90^\circ| < 45^\circ$. Its total weight is 25 tons and it covers an area of 160 m² with about 160 000 total readout channels and an intrinsic resolution of 60ns. It has a modular structure corresponding to 18 sectors in φ and to 5 segments in z . All modules have the same width of 128cm and increasing lengths, adding up to overall TOF barrel length of 750 cm.

Inside each module the strips are tiled, thus minimizing the number of multiple partial-cell hits due to the obliqueness of the incidence angle. The double stack-strip arrangement, the cooling tubes, and the materials for electronics have been described in detail. During the development of TOF design several different geometry options have been studied, all highly detailed.

HMPID, PHOS, ZDC, Muon arm and others

The HMPID detector also poses a challenge in the simulation of the Cherenkov effect and the secondary emission of feedback photons. A detailed simulation has been introduced for all these effects and has been validated both by test-beam data and with the ALICE RICH prototype that has been operating in the STAR experiment.

The PHOS has also been simulated in detail. The geometry includes the

Charged Particle Veto (CPV), crystals (EMC), readout (PIN or APD) and support structures. Hits record the energy deposition in one CPV-EMC cell per entering particle. In the digits the contribution from all particles per event are summed up and noise is added.

The simulation of ZDC in AliRoot requires transport of spectator nucleons with Fermi spread, beam divergence and crossing angle for over 100m. The HIJING generator is used for these studies taking into account the correlations with transverse energy and multiplicity.

The muon spectrometer is composed of 5 tracking stations and 2 trigger stations. For stations 1-2 a conservative material distribution is adopted, while for station 3-5 and for the trigger stations a detailed geometry is implemented. Supporting frames and support structures are still coarse or missing, but they are not very important in the simulation of the signal. The muon chambers have a complicated segmentation that has been implemented during the signal generation via set of virtual classes. This allows changing the segmentation without modifying the geometry. Summable digits are generated taking into account the Mathieson formalism for charge distribution, while work is ongoing on the angular dependence, Lorenz angle and charge correlation.

The complex T0-FMD-V0-PMD forward detector system is still under development and optimization. There are several options provided to study their performance.

ALICE geometry and generation of simulated data is in place to allow full event reconstruction including the main tracking devices. The framework allows comparison with test-beam data that has already been performed. The early availability of a complete simulation has been an important point for the development of reconstruction and analysis code and user interfaces, now the focus of the development. The ALICE geometry is implemented in a ROOT class TGeo. A detailed description of this class is available in ROOT User's Guide [7].

Simulation of detector response

Much of the activity described in this work is a large virtual experiment where thousands of events are generated and analyzed in order to produce the result presented. This has the objective of studying in detail the ALICE physics capabilities, to clarify the physics goals of the experiment, and of verifying the functionality of our software framework from (simulated) raw data to physics. To carry out this double objective, it is important to have a high-quality and reliable simulation. One of the most common programs for full detector simulation is GEANT 3 which, however, is a 20-years old FORTRAN program officially frozen since 1993. We are waiting for GEANT 4 to become available for production for the LHC and we also intend to evaluate FLUKA as a full detector simulation program. Therefore, it was decided to build an environment that could profit from the maturity and solidity of GEANT 3 and, at the same time, protect the investment in the user code when moving to new Monte Carlo. Combining the above immediate needs and long term requirements into a single

framework, the GEANT3 code was wrapped into a C++ class (`TGeant3`) and a Virtual Monte Carlo abstract interface was implemented in the AliRoot. This has proved very satisfactory. Using this class, user can control the physics processes used during the generation without specifying which Monte Carlo will be actually used.

Conclusion of the Simulation

The simulation process can be described step by step as follows:

- Event *generation* of final particles is carried on by an event generator code (or parametrization) and the final state particles are fed to the transport program.
- The particles emerging from the interaction of the beam particles are *transported* in the material of the detector, simulating their interaction with it and the energy deposition that generates the detector response (hits).
- From the *detector response*, the *signal is generated*. The detector response is based on the energy deposition from the particles passing through it. This is the ideal detector response before the conversion to digital signal and formatting by the front-end electronics.
- Then the ideal detector signal is *digitized* and formatted according to the output of the front-end electronics and the data acquisition system. The results should resemble the real data produced by the detector.

After all these steps comes the time for software analysis and techniques development. From these data we will go backwards and reconstruct the full event from the detector response, which is the most important part, and see how effective and precise our analysis is comparing it with original Monte Carlo output.

4.4 Reconstruction framework

At first, let us agree on some terminology used in the ALICE environment both hardware and software.

Whenever it is not specified differently, we refer to the "global ALICE coordinate system". It is a right-handed coordinate system with the z axis coinciding with the beam-pipe in the opposite direction than the muon arm is situated, the y axis point upwards, the x axis pointing to the center of LHC and with the origin of coordinates defined by the intersection of z axis with the central-membrane plane of the TPC.

In the following text, we use also the following terms:

- *Digit*: This is a digitized signal (ADC count) obtained by a sensitive pad of a detector at a certain time.
- *Cluster*: This is a set of adjacent (in space and/or time) digits that were presumably generated by the same particle crossing the sensitive element of a detector.

- Reconstructed *space point*: This estimation of the position where a particle crossed the sensitive element of a detector (often found as the center of gravity of a cluster).
- Reconstructed *track*: This is a set of five parameters (such as the curvature and the angles with respect to the coordinate axes) of particle's trajectory together with the corresponding covariance matrix estimated at a given point of space.

The input to the reconstruction framework are digits in a root tree format or raw data format. First a local reconstruction of clusters is performed in each detector. Then vertexes and tracks are reconstructed and particles types are identified. The output of the reconstruction is the Event Summary Data (ESD). The `AliReconstruction` class provides a simple user interface to the reconstruction framework which is explained in the source code and in the on-line html documentation.

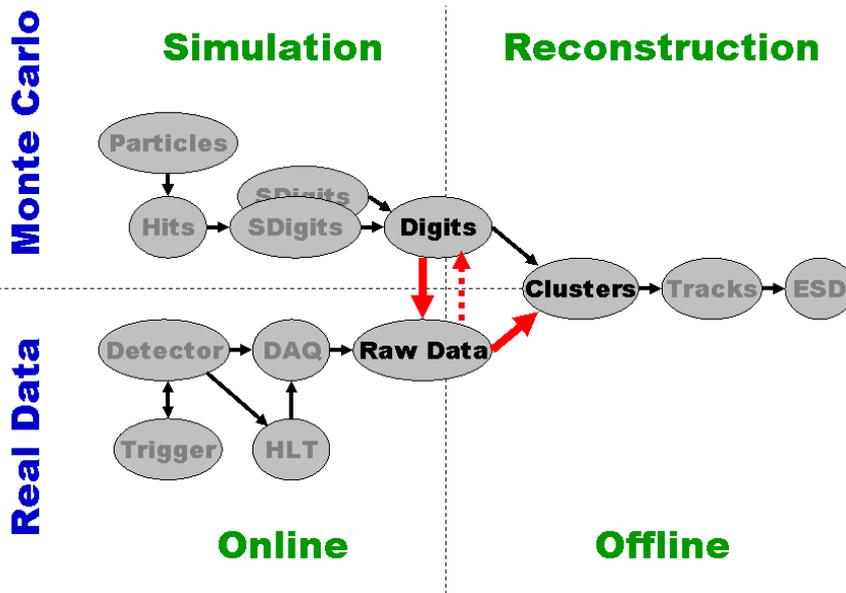


Figure 10: The ALICE reconstruction framework scheme.

The main interface to specific reconstruction in each detector is realized via the base class `AliReconstructor`. For each detector there is a derived reconstructor class. Options of the detector in a string format can be obtained using the `GetOption()` method inside the reconstructor. Every detector is created via a plugin, Therefore it has to have a default constructor. If not specified differently, it is assumed, that the name of the reconstructor for a detector DET is `AliDETReconstructor` and that it is located on the library `libDETrec.so` (or `libDET.so`).

Input Data

If the data is provided in format of root trees, either the loader or directly the trees are used to access the data. In case of raw data input, the digits are accessed via a raw reader.

If a `galice.root` file exists, the run loader will be retrieved from it. Otherwise the run loader and the headers will be created from the raw data. The reconstruction can not work if there is no `galice.root` file and no raw data input.

Output Data

The clusters (reconstructed points) are considered as intermediate output and are stored in root trees handled by the loaders. The final output of the reconstruction is a tree with objects of type `AliESD` stored in the file `AliESDs.root`. This Event Summary Data (ESD) contains list of reconstructed tracks/particles and global event properties.

During the reconstruction, there is the primary vertex (PV) to be found first. In the first step, the ITS signal is used for the first approximation. Then, when tracking of all primary particles is done, the PV is evaluated again using all the tracks of primary particles to find the final reconstructed primary vertex using several classes derived from the `AliVertexer` base class (i.e. `AliITSVertexerZ` or `AliITSVertexerTracks`).

Next step is tracking of all particles. This is done in three steps. All the classes used to do this are derived from `AliTracker` base class. First tracking finds every track in TPC and goes in the inward direction through the ITS to the primary vertex (`Clusters2Tracks`). Second step goes from the PV in the outward direction (`PropagateBack`) propagating tracks from ITS in all the detectors. And lastly, the tracks are refitted inwards (`RefitInward`) from TRD through TPC and ITS. All these three methods have `AliESD` object as an argument, which is used to exchange track information between detectors without introducing dependencies between the code of the detector trackers.

There are two different methods used for the tracking. Global ones that let the tracking algorithms find all tracks from all clusters and after all the tracks are found and identified, filters can be applied. Advantages of this method are its stability with respect to noise and mismeasurements and the possibility to operate directly on raw data. On the other hand this method requires a precise global track model. Such a model can sometimes be unknown at the time of reconstruction or even does not have to exist at all.

Local methods though don't require a global model, track parameters are estimated locally at a given point in space. The decision to accept or reject a measurement is made using the local information or the information coming from the previous history of this track. With these methods, all the local track peculiarities are taken into account (detector geometry, non-linearities, etc.). Unfortunately, the local methods rely on sophisticated point reconstruction algorithms. They are sensitive to noise, wrong or displaced measurements and the precision of space point error parametrization. The most advanced kind of local track-finding methods is Kalman filtering (for details see section 4.1 of [6])

and references mentioned there). This algorithm have several advantages;

- Simultaneous track recognition and fitting.
- Possibility to reject incorrect space points "on the fly", during the only tracking pass. These points can appear as a consequence of the imperfection of the cluster finder, they may be due to noise or they may be points belonging to other track accidentally captured in the list of points belonging to the track under consideration.
- In case of multiple scattering, track measurements are correlated and therefore large matrices need to be inverted during a global fit. In the Kalman filter procedure, only small matrices (5×5) have to be manipulated, which is faster.
- It is a natural way to extrapolate tracks from one detector to another.

For simplifying the extrapolation tracks from one detector to another, a common coordinates are used for each of the detectors. The origin and z axis coincide with the global ALICE coordinate system, the x axis is perpendicular to the sensitive plane of the detector and y axis completes right-handed Cartesian coordinate system.

Once the tracks are reconstructed and stored as `AliESD` object, further information is added to the ESD, such as reconstructed primary vertex coordinates, particles are identification, etc.

Event Summary Data

The classes needed to process and analyze the ESD are packed together in a standalone library (`libESD.so`) which can be used separately from the AliROOT framework. The main class is `AliESD`, which contains all information needed during the physics analysis:

- field to identify the event such as run number, event number, trigger word, version of the reconstruction, etc.;
- reconstructed ZDC energies and number of participants;
- primary vertex;
- T0 estimation of the primary vertex;
- array of ESD tracks;
- array of HLT tracks both from the conformal mapping and from the Hough transform reconstruction;
- array of MUON tracks;
- array of PMD tracks;
- array of reconstructed V^0 vertices, cascade decays and kinks
- indices of the information from PHOS and EMCAL detectors in the array of the ESD tracks.

4.5 Analysis

The analysis of experimental data is the final step of event processing and it usually repeated many times. Analysis is a very diverse activity, where the goals of each particular analysis pass may differ significantly.

The ALICE detector is optimized for the reconstruction of heavy-ion collisions. In addition to that, ALICE has a broad program for p-p and p-A interactions. Main points of the ALICE heavy-ion program can be divided into four programs;

- **global event characteristics:** particle multiplicity, centrality, energy density, nuclear stopping;
- **soft physics:** chemical composition (particle and resonance production, particle ratios and spectra, strangeness enhancement), reaction dynamics (transverse and elliptic flow, HBT correlations, event-by-event dynamical fluctuations);
- **hard probes:** jets, direct photons;
- **heavy flavors:** quarkonia, open charm and beauty production.

Each of this points is done by a Physics Working Group (PWG) numbered 1-4.

The analysis program is divided into two main types; the scheduled and chaotic analysis. They differ in almost all main aspects, such as data access pattern, frequency of code changes, storage and registration of their results.

Scheduled analysis typically uses all the available data from the given period and stores the results on the Grid. The procedure is centralized and can be often considered as data filtering. Its results can be used for subsequent analysis. The scheduled analysis is preferred by the Physics Board and computing and storage resources are allocated accordingly. The analysis code is developed in advance, tested and released before the beginning of the data processing.

On the other hand, the **chaotic analysis** is focused on a single physics task and typically is based on the filtered data from the scheduled analysis. Every physicist can also access large number of events to look for a rare signal. Usually, the user develops the code using small sample of data and changes the algorithm and criteria frequently. The output of this analysis is typically only a set of histograms. Such tuning of the analysis code can be done on the Grid or locally on a data sample. The final version of the analysis can be then submitted to the Grid and can access large portions of or even all the ESD files. The activity is in most cases coordinated inside the PWGs via the definition of priorities.

4.5.1 Analysis Tools

For both simulation and analysis, the main infrastructure tool is a distributed network called Grid. The Grid middleware is hidden by an interface to it called

gShell, which is a single working shell. The gShell package contains all the commands a user may need for catalog queries, job submission, registration and removal of files and process monitoring. The service is implemented as a pool of preforked server daemons, which serve single-client requests. The client-server protocol implements a client state, which consists of a current working directory, session ID and time-dependent symmetric cipher on both ends to guarantee client security and privacy. The server daemons execute the client calls with the client identity.

The Parallel ROOT Facility (**PROOF**) has been specially designed and developed to allow the analysis and mining of very large data sets, minimizing response time. It makes use of the inherent parallelism in event data and implements an architecture that optimizes I/O and CPU utilization in heterogeneous clusters with distributed storage. The system provides transparent and interactive access to terabyte-scale data sets. Being part of the ROOT framework, PROOF inherits the benefits of a performing object storage system and a wealth of statistical and visualization tools. The most important design features of the PROOF are:

- transparency – no difference between the local ROOT and a remote parallel PROOF session;
- scalability – no implicit limitations on number of computers used in parallel;
- adaptability – the system is able to adapt to variations in the remote environment.

PROOF is based on multi-tier architecture: the ROOT client session, the PROOF master server, optionally a number of PROOF sub-master servers, and the PROOF worker servers. The user connects from the ROOT session to a master server on a remote cluster and the master server creates sub-masters and worker servers on all the nodes in the cluster. All workers process queries in parallel and the results are presented to the user as coming from a single server. PROOF can be run either in a purely interactive way, with the user remaining connected to the master and worker servers and the analysis results being returned to the user's ROOT session for further analysis, or in an "interactive batch" way where the user disconnects from the master and workers (see Fig.11). By reconnecting later to the master server the user can retrieve the analysis results for that particular query. This last mode is useful for relatively long running queries (several hours) or for submitting many queries at the same time. Both modes will be important for the analysis of ALICE data.

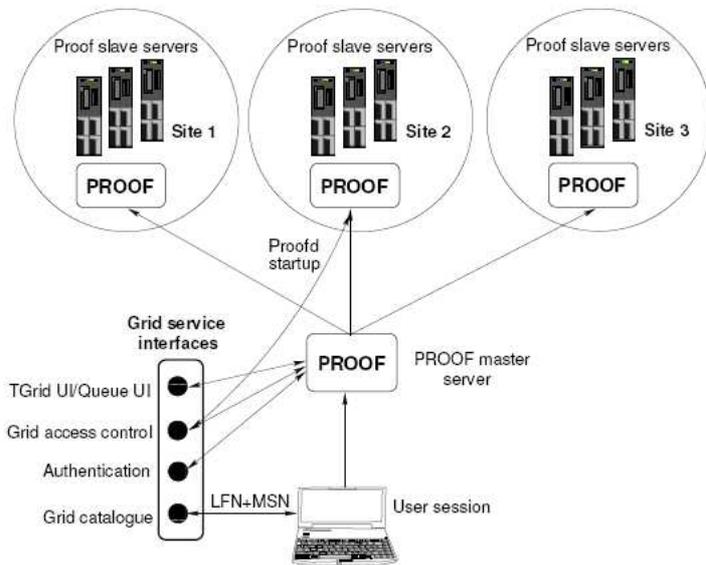


Figure 11: Setup and interaction with the Grid middleware of a user PROOF session distributed over many computing centers.

5 Data analysis

Main goal of my analysis was to simulate a pp collision at two different collision energies, 900 GeV and 14 TeV in CMS and to analyze these data. I have used for this purpose the ROOT 5.14 environment with PYTHIA 6.409 as the event generation.

5.1 PYTHIA

PYTHIA is a program for the generation of high-energy physics events, i.e. for the description of collisions at high energies between elementary particles such as e^+ , e^- , p and \bar{p} in various combinations. It contains theory and models for a number of physics aspects, including hard and soft interactions, parton distributions, initial- and final-state parton showers, multiple interactions, fragmentation and decay. It is largely based on original research, but also borrows many formulae and other knowledge from the literature.

Development of JETSET, the first member of the "Lund Monte Carlo" family, was begun by members of the Lund theory group in 1978, and has continued since then, on and off. A number of people have contributed to this and other programs based on it. The most extensive of these is PYTHIA. Over the years, these two programs have more and more come to be maintained in common. In 1997 they were therefore merged to one, under the PYTHIA label. Up until recently the current version was PYTHIA 6.4.

A few years ago a rewriting (from Fortran 77) to C++ was begun, and with the release of PYTHIA 8.1 this new code becomes the official current version. For some time to come both versions will be maintained, however, since 8.1 is not yet fully developed. It already offers some new features not found in 6.4, on the other hand, and will gradually pull ahead as further physics is introduced in it.

Despite the new version 8 was available, when I started my analysis, I used older, but still widely used fortran code, e.g. PYTHIA version 6.409.

5.2 ROOT

The ROOT system provides a set of Object Oriented frameworks with all the functionality needed to handle and analyze large amounts of data in a very efficient way. Having the data defined as a set of objects, specialized storage methods are used to get direct access to the separate attributes of the selected objects, without having to touch the bulk of the data. Included are histogramming methods in 1, 2 and 3 dimensions, curve fitting, function evaluation, minimization, graphics and visualization classes to allow the easy setup of an analysis system that can query and process the data interactively or in batch mode.

Thanks to the built-in CINT C++ interpreter the command language, the scripting, or macro, language and the programming language are all C++. The interpreter allows for fast prototyping of the macros since it removes the time consuming compile/link cycle. It also provides a good environment to learn C++. If more performance is needed the interactively developed macros can be compiled using a C++ compiler.

The system has been designed in such a way that it can query its databases in parallel on multi-processor platform (MPP) machines or on clusters of work-

stations or high-end PC's. ROOT is an open system that can be dynamically extended by linking external libraries. This makes ROOT a premier platform on which to build data acquisition, simulation and data analysis systems.

I have chosen the version 5.14, which was up to date when I started to develop my analysis code. It has the PYTHIA generator mentioned above included and wrapped, so it's possible to use it via the ROOT CINT C++ code.

5.3 My ROOT macro

I have done the analysis on 1 million of generated pp collisions at 900 GeV and 14 TeV available energy in CMS. The macros used are pretty much the same except for the initialization of PYTHIA.

After resetting the ROOT environment to default values with `gROOT->Reset()` command, the macro opens a `.ROOT` file and creates histograms, which will be filled during the macro run. The main loop over all events does the following for each event:

- resets counters for both protons and antiprotons;
- counts p_T and η for every particle;
- checks whether the particle is in the acceptance of the TPC ($|\eta| < 0.9$)
- scans if the particle is a proton or an antiproton and fills the appropriate histograms;
- gives back a summary of the "run";
- draws p_T and η distribution;
- draws the proton-antiproton asymmetry and difference histograms;
- closes the file and ends.

The macro can be found in Section 7.

5.4 Results

5.4.1 Asymmetry in pp collisions at $\sqrt{s_{NN}} = 900 \text{ GeV}$

After an analysis of one million generated events, I have received proton and antiproton pseudorapidity distributions (see Fig 12 top left and top right). One can clearly see the forward excess multiplicity of spectator protons. Therefore, in the forward regions ($|\eta| > 4$) the asymmetry is approaching the upper limit value 2. The interesting region is located near the xy plane, where I could as well apply the TPC acceptance cut ($|\eta| < 0.9$). After applying this cut, the forward spectator protons are eliminated and I obtained a result for the asymmetry in the mid-rapidity region, which is $A = 0.05 \pm 0.02\%$.

This result is in accordance with results of others from the ALICE collaboration (~ 0.07), which were presented at Alice Physics Week 2007 in Münster, Germany. Regarding the p_T distribution of protons and antiprotons, one can find them in Fig. 13, top. It seems for their ratio (Fig. 13, bottom), that it

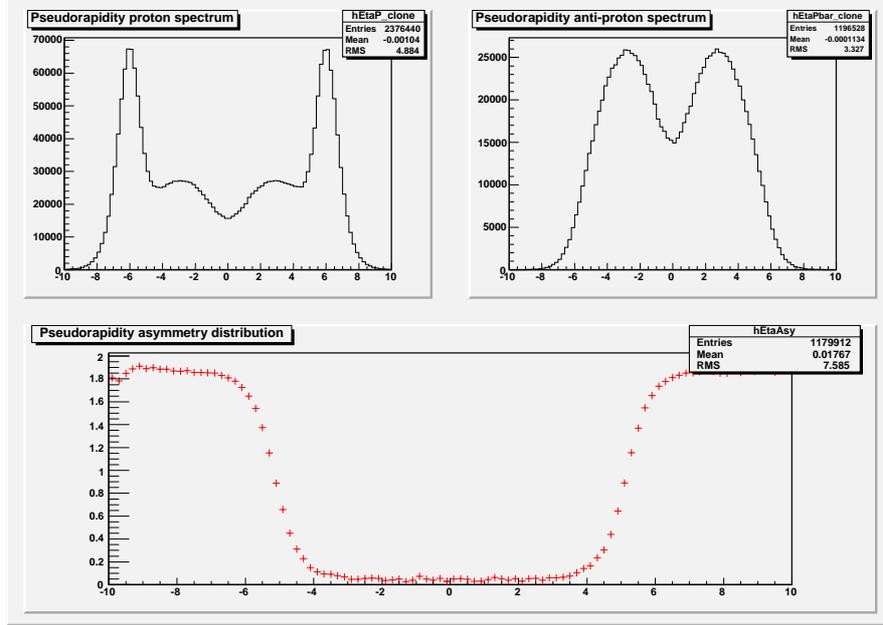


Figure 12: Top: pseudorapidity distribution of protons (left) and antiprotons (right). Bottom: Asymmetry distribution in pseudorapidity; for $\sqrt{s_{NN}} = 900 GeV$

approaches the value of $N_p/N_{\bar{p}} \simeq 2.8$ at high p_T . Of course, at high p_T the number of particles decreases, so their fluctuations and fluctuations of their ratio are getting more and more significant.

5.4.2 Asymmetry in pp collisions at $\sqrt{s_{NN}} = 14 TeV$

After an analysis of one million generated events at $\sqrt{s_{NN}} = 14 TeV$, I have received the following proton and antiproton pseudorapidity distributions (see Fig 14 top left and top right). As in the lower energy case, we can see the forward and backward regions full of spectator protons. On the other hand, the mid-rapidity region multiplicities are enhanced due to higher energy. We have in this case better statistics, relative to the previous case and we can extend our measurement to larger pseudorapidity region to about $|\eta| < 6$ in which the asymmetry appears to be constant. It fluctuates around the value of $A = (0.005 \sim 0.01) \pm 0.01$. I would say, that the asymmetry A would approach zero, when we go to higher and higher energies. But that's something future experiments will show.

As in the previous case, I have studied the p_T distribution of both protons and antiprotons (see Fig. 15 top). The protons have excess production at lower p_T , but at higher $p_T \sim 2 GeV/c$, the number of produced protons and antiprotons approaches each other, which one can observe in Fig. 15 bottom, as the ratio $N_p/N_{\bar{p}}$ slowly approaches 1 at high p_T .

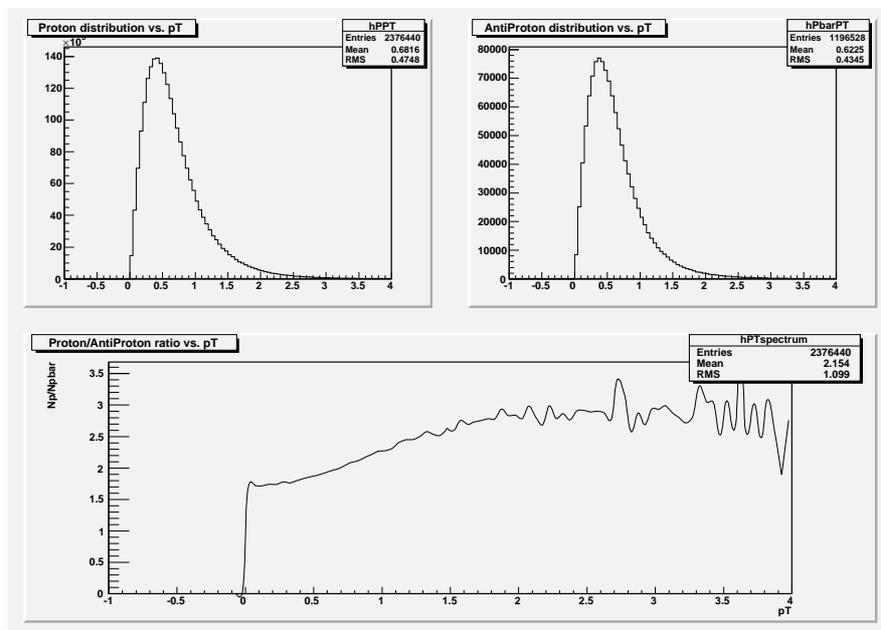


Figure 13: Top: p_T distribution of protons (left) and antiprotons (right). Bottom: Proton/antiproton ratio vs. p_T ; for $\sqrt{s_{NN}} = 900 \text{ GeV}$

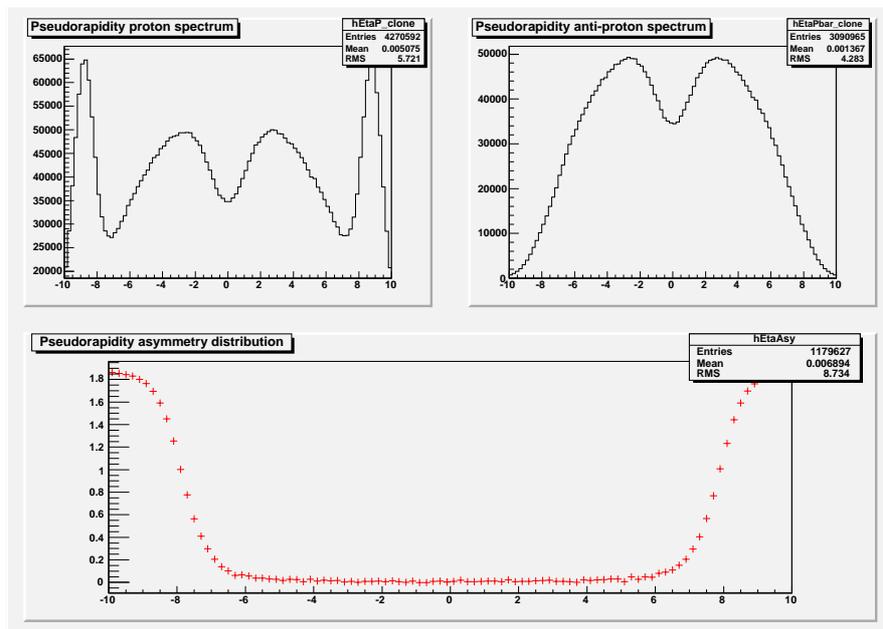


Figure 14: Top: pseudorapidity distribution of protons (left) and antiprotons (right). Bottom: Asymmetry distribution in pseudorapidity; for $\sqrt{s_{NN}} = 14 TeV$

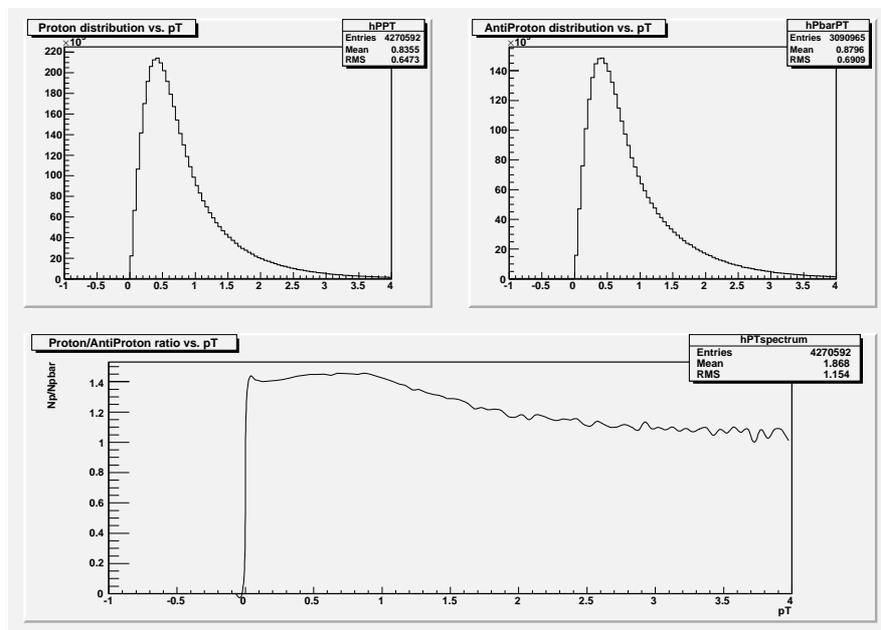


Figure 15: Top: p_T distribution of protons (left) and antiprotons (right). Bottom: Proton/antiproton ratio vs. p_T ; for $\sqrt{s_{NN}} = 14 TeV$

6 Conclusion and Outlook

The statistical model of particle production is a very good tool for describing heavy-ion collisions, as the number of particles produced is getting higher and higher with the energy. It provides great predictions for global observables, like the baryo-chemical potential μ_B and the kinetic freeze-out temperature T_c . However, we need some other explanation for proton/antiproton number asymmetry. Therefore we use the string junction description to explain the baryon stopping in the collisions. A framework and structure of the offline analysis in the ALICE experiment is being prepared and developed. I did a simulation of pp collisions at two different energies, which will be used in LHC, $\sqrt{s_{NN}} = 900 GeV$ and $\sqrt{s_{NN}} = 14 TeV$. At lower energy, the asymmetry in mid-rapidity region appears to be higher ($A \sim 0.05$) than in the higher energy ($A \sim 0.005$). The same is true for proton/antiproton ratio at high p_T . at the lower energy, the ratio tend to the value of 2.8 and at the higher energy to 1.1.

In the future, I want to concentrate on the statistical model and description of the global observables in heavy-ion collisions.

References

- [1] Technical design report of the Inner Tracking System (ITS), 1999, *CERN/LHCC/99-12*
- [2] Jean Letessier & Johann Rafelski, 2002, Hadrons and Quark-Gluon Plasma, *Cambridge University Press*
- [3] J.Kapusta, B.Muller, J.Rafelski, Quark-gluon Plasma, *Elsevier, 2003*
- [4] Alice Offline Pages, <http://aliceinfo.cern.ch/Offline>
- [5] Markus Oldenburg, AliROOT Offline Tutorial, http://oldi.web.cern.ch/oldi/AliceTutorial/OfflineTutorial_I.ppt, May 8, 2007
- [6] P.Hristov, AliROOT Primer, <http://aliceinfo.cern.ch/static/offlineold/primer.pdf>
- [7] René Brun (CERN), Fons Rademakers (CERN), Philippe Canal(FNAL), Ilka Antcheva (CERN), Damir Buskulic (LAPP), Users Guide 5.16, ftp://root.cern.ch/root/doc/Users_Guide_5_16.pdf, July 2007
- [8] P.Braun-Munzinger, K.Redlich, J.Stachel, Particle production in heavy ion collisions, *arXiv:nucl-th/0304013v1*, 3 Apr 2003
- [9] G.T.Garvey, B.Z.Kopeliovich, B.Povh, Baryon Number Flow in High-Energy Collisions, *arXiv:hep-ph/0006325v3*, 21 Jan 2002

7 Appendix A

ROOT macro

```
{
gROOT->Reset();

#include <cmath>

Int_t nEvents = 1000000; Int_t firstEvent = 1; char
*fnRes="PPbar14.root";

TFile *fileRes = new TFile(fnRes,"RECREATE");

// create histograms
TH1F *hPPbarDiff = new TH1F("hPPbarDiff",
    "PPbar - Ppbar in acceptance",510,-2.5,2.5);
TH1F *hPPbarAsy = new TH1F("hPPbarAsy",
    "PPbar Asymmetry",510,-2.5,2.5);
TH1F *hPPbarAsyCut = new TH1F("hPPbarAsyCut",
    "PPbar Asymmetry w/o TPC acceptance ( |eta| < 0.9 )",510,-2.5,2.5);
TH1F *hParticleNumber = new TH1F("hParticleNumber",
    "Number of particles in the TPC acceptance",499, 0, 500);

TH1D *hPPT = new TH1D("hPPT",
    "Proton distribution vs. pT",100,-1.,4.);
TH1D *hPbarPT = new TH1D("hPbarPT",
    "AntiProton distribution vs. pT",100,-1.,4.);
TH1D *hPTSpectrum = new TH1D("hPTSpectrum",
    "Proton/AntiProton ratio vs. pT",100,-1.,4.);

TH1D *hEtaP = new TH1D("hEtaP",
    "Pseudorapidity proton spectrum",100,-10.,10.);
TH1D *hEtaPbar = new TH1D("hEtaPbar",
    "Pseudorapidity anti-proton spectrum",100,-10.,10.);
TH1D *hEtaAsy = new TH1D("hEtaAsy",
    "Pseudorapidity asymmetry distribution",100,-10.,10.);

Float_t eta_cut = .9; // pseudorapidity cut for ALICE TPC
Double_t eta = 0.;
Float_t asy,asy;

Int_t Np = 0,Npbar = 0; //number of protons and antiprotons respectively
Int_t Npc = 0,Npbarc = 0; //number of protons and antiprotons respectively in the cut
Int_t Ngood = 0, NgoodCelkem = 0; //number of particles in event
Int_t Nforward = 0; // number of particles with pT = 0
Int_t p_code = 0; // variable for temporary saving the pythia.GetK(iPart,1)
Double_t px,py,pz,pT,E,m,theta;

const Float_t c = 1;

//gSystem->Load("li2bEG");
//gSystem->Load("libPythia6");
//gSystem->Load("libEGPythia6.so");

TPythia6 *pythia = new TPythia6(); // inicializace Pythia6

pythia.SetMSEL(1); // 1 is default

pythia.Initialize("cms", "p", "p", 14000.);
cout << endl;

/*****
* Main program body, simaltion and evaluation of variables *
*****/

for (Int_t eventNr = firstEvent; eventNr < firstEvent+nEvents; eventNr++) {
```

```

//cycle over all events

// reinitialize variables...
Npc = 0;
Npbarc = 0;
Np = 0;
Npbar = 0;
Ngood = 0;

// information about progress each 10^n event

if((eventNr%100 == 0)) cout << "event " << eventNr << endl;

pythia.GenerateEvent();
Int_t nPart = pythia.GetNumberOfParticles();

for (Int_t iPart=2; iPart < nPart; iPart++) {
// cycle over all particles in one event

if(((p_code = pythia.GetK(iPart,1)) < 11) && (p_code > 0))
// Pythia manual says, that particles got codes 1-10...
// therefore this condition
{

px = pythia.GetP(iPart,1);
py = pythia.GetP(iPart,2);
pz = pythia.GetP(iPart,3);
pT = sqrt(px*px + py*py);

if ((pT == 0. || pz == 0.) {
printf(" !!!! je tam nula, zadna munice :-((( \n " );
Nforward++;
cout << " ---- pT = " << pT << "      pythia rika : "
<< pythia.GetK(iPart,1) << endl;
cout << " ---- px = " << px
<< "
<< "      py = " << py << endl;
cout << " ---- pythia identifikace : "
<< pythia.GetK(iPart,1) << endl;
cout << " ===== "
<< endl;
continue; }
theta = atan2(pT,pz);

eta = - log( tan (0.5 * theta));

// identifying proton (within and out of the acceptance cut)
if (abs (eta) < eta_cut) {Ngood++; if (pythia.GetK(iPart,2) == 2212) Npc++;}
if (pythia.GetK(iPart,2) == 2212) { Np++; hPPT->Fill(pT); hEtaP->Fill(eta);

}

// identifying anti-proton (within and out of the acceptance cut)
if (abs (eta) < eta_cut) {if (pythia.GetK(iPart,2) == -2212) Npbarc++;}
if (pythia.GetK(iPart,2) == -2212)
{ Npbar++; hPbarPT->Fill(pT); hEtaPbar->Fill(eta);

}

}/*if each particle*/
} /* for each particle */

if( (Np+Npbar) > 0 ) hPPbarAsy->Fill(asy = (2.0*(Np-Npbar)/(Np+Npbar)) );
else {
hPPbarAsy->Fill(0);
}

if( (Npc+Npbarc) > 0 ) hPPbarAsyCut->Fill( asyc = (2.*(Npc-Npbarc)/(Npc+Npbarc)) );
else {
hPPbarAsyCut->Fill(0);
}

```

```

hParticleNumber->Fill(Ngood);
NgoodCelkem = NgoodCelkem + Ngood;

    //if ((Npbarc != Npbar) || (Np != Npbar)) {cout << " ---- nestejne "
    // << endl; hPPbarDiff->Fill(asy);Nforward++;}
    //cout << " == asy : " << asy << " == asyC : " << asyc << endl;
} //konec cyklu pres vsechny eventy

cout << " ===== " << endl;
cout << " - Deleni nulou: " << Nforward << endl;
cout << " - Castic v akceptanci: " << NgoodCelkem << endl;
cout << " ===== " << endl;

// ----- DRAWING pT DISTRIBUTIONS
TCanvas *c4 = new TCanvas("c4","Proton and AntiProton pT spectra", 800, 600);
c4->cd();

TPad *pr1 = new TPad("pr1","AntiProton/Proton Ratio",0.02,0.02,0.98,0.48);
TPad *pr2 = new TPad("pr2","Proton pT spectrum",0.02,0.52,0.48,0.98);
TPad *pr3 = new TPad("pr3","AntiProton pT spectrum",0.52,0.52,0.98,0.98);

pr1->Draw();
pr2->Draw();
pr3->Draw();

pr2->cd();
hPPT->DrawCopy();

pr3->cd();
hPbarPT->DrawCopy();

hPTSpectrum->Divide(hPPT,hPbarPT,1,1,"");
hPTSpectrum->SetYTitle("Np/Npbar");
hPTSpectrum->SetXTitle("pT");

pr1->cd();
hPTSpectrum->DrawCopy("C"); //draw with smooth line through the bins

// ----- drawing ETA distribution
TCanvas *c5 = new TCanvas("c5","Proton and AntiProton pseudorapidity spectra", 800, 600);
c5->cd();

TPad *pad1 = new TPad("pad1","AntiProton/Proton Asymmetry vs. Pseudorapidity",0.02,0.02,0.98,0.48);
TPad *pad2 = new TPad("pad2","Proton Eta spectrum",0.02,0.52,0.48,0.98);
TPad *pad3 = new TPad("pad3","AntiProton Eta spectrum",0.52,0.52,0.98,0.98);

pad1->Draw();
pad2->Draw();
pad3->Draw();

pad2->cd();
TH1D *hEtaP_clone = (TH1D*) hEtaP->Clone();
hEtaP_clone->SetName("hEtaP_clone");
hEtaP_clone->DrawCopy();

pad3->cd();

TH1D *hEtaPbar_clone = (TH1D*) hEtaPbar->Clone();
hEtaPbar_clone->SetName("hEtaPbar_clone");
hEtaPbar_clone->DrawCopy();

hEtaP->Add(hEtaP,hEtaPbar,1,1);
hEtaPbar->Add(hEtaP,hEtaPbar,1,-2);
hEtaAsy->Divide(hEtaPbar,hEtaP,2,1);

pad1->cd();

```

```

hEtaAsy->SetMarkerStyle(2); // crosses
hEtaAsy->SetMarkerColor(2);
hEtaAsy->DrawCopy("P"); // draw with markers

// ----- drawing asymmetry

TCanvas *c3 = new TCanvas("c3","Particles",10,10,1000,800);
//c3->Divide(2,2);
c3->cd();

cout << " dbg ---- Canvas c1 created ---- " << endl;

TPad *p1 = new TPad("p1","Number of particles in events",
                    0.02,0.02,0.98,0.48);
TPad *p2 = new TPad("p2","Asymmetry",0.02,0.52,0.48,0.98);
TPad *p3 = new TPad("p3","Difference",0.52,0.52,0.98,0.98);

cout << " dbg ---- Pads p1-p3 created ---- " << endl;

p1->Draw();
p2->Draw();
p3->Draw();

cout << " dbg ---- Pads p1-p3 drawn ---- " << endl;
p1->cd();

hParticleNumber->SetXTitle("N of particles");
hParticleNumber->SetYTitle("N of events");
hParticleNumber->DrawCopy();
cout << " dbg ---- histogram Particles drawn ---- " << endl;
hPPbarDiff->Add(hPPbarAsy,hPPbarAsyCut,1,-1);

hPPbarDiff->SetFillColor(6);
//hPPbarDiff->SetLineColor(1);
hPPbarAsyCut->SetLineColor(2);
//hPPbarAsyCut->SetLineWidth(1);
//hPPbarAsy->SetLineWidth(1);
hPPbarAsy->SetLineColor(4);

hPPbarAsy->SetXTitle("A");
//hPPbarAsy->SetYTitle("Nevt");

hPPbarAsyCut->SetXTitle("A");
//hPPbarAsyCut->SetYTitle("Nevt");

hPPbarDiff->SetXTitle("Diff");
//hPPbarDiff->SetYTitle("Nevt");

p2->cd();
hPPbarAsyCut->DrawCopy();

hPPbarAsy->DrawCopy("same");

//hPPbarDiff->SetMaximum(hPPbarAsy->GetMaximum());
p3->cd();
hPPbarDiff->DrawCopy();

// ----- DRAWING BIG ASYMMETRY HISTOGRAMS

TCanvas *c2 = new TCanvas("c2","Asymmetry",1000,800);

c2->cd();
hPPbarAsyCut->DrawCopy();
hPPbarAsy->DrawCopy("same");

TCanvas *c12 = new TCanvas("c12","Difference",1000,800);
c12->cd();
hPPbarDiff->DrawCopy();

// ----- CLOSING FILE

//fileRes->cd();

```

```
    fileRes->Write();  
    fileRes->Close();  
}
```