The FLOW Analysis Package

a short writeup

January 15, 2014

Various authors, edited by Redmer Alexander Bertens (rbertens@cern.ch)

Contents

Chapter 1

Introduction

The intro to everything.

Chapter 2

A Quick Start

2.1 The flow package

The `ALICE flow package`[a] contains most known flow analysis methods. In this chapter we give a few examples how to setup an analysis for the most common cases. The chapters that follow provide more detailed information on the structure of the code and settings of the various flow methods. This write-up is however not a complete listing of the methods, for this the reader is referred to the header files.

2.2 On the fly

The macro `Documentation/examples/runFlowOnTheFlyExample.C`[b] is a basic example of how the flow package works. In this section we explain the main pieces of that macro.

1. To use the flow code the flow library needs to be loaded. In `AliROOT`:

```
gSystem->Load("libPWGflowBase");
```

In `root` additional libraries need to be loaded:

```
gSystem->Load("libGeom");
gSystem->Load("libVMC");
```

[a] The ALICE flow package is part of AliROOT, the ALICE extension of the ROOT framework, which can be obtained from http://git.cern.ch/pub/AliRoot. The flow package itself is located in the folder `$ALICE_ROOT/PWG/FLOW/`, where `$ALICE_ROOT` refers to the source directory of AliROOT.

[b] In aliroot, this macro can be found at
`$ALICE_ROOT/PWGCF/FLOW/Documentation/examples/runFlowOnTheFlyExample`

```
gSystem->Load("libXMLIO");
gSystem->Load("libPhysics");
gSystem->Load("libPWGflowBase");
```

2. We need to instantiate the flow analysis methods which we want to use. In this example we will instantiate two methods: the first which calculates the flow versus the reaction plane of the Monte Carlo, which is our reference value (see section 5.1), and second the so called Q-cumulant method (see section 5.4).

```
AliFlowAnalysisWithMCEventPlane *mcep = new
    AliFlowAnalysisWithMCEventPlane();
AliFlowAnalysisWithQCumulants *qc = new
    AliFlowAnalysisWithQCumulants();
```

3. Each of the methods needs to initialize (e.g. to define the histograms):

```
mcep->Init();
qc->Init();
```

4. To define the particles we are going to use as Reference Particles (RP's, particles used for the **Q** vector) and the Particles Of Interest (POI's, the particles of which we calculate the differential flow) we have to define two track cut objects:

```
AliFlowTrackSimpleCuts *cutsRP = new AliFlowTrackSimpleCuts();
AliFlowTrackSimpleCuts *cutsPOI = new AliFlowTrackSimpleCuts();
cutsPOI->SetPtMin(0.2);
cutsPOI->SetPtMax(2.0);
```

5. Now we are ready to start the analysis. For a quick start we make an event on the fly, tag the reference particles and particles of interest and pass it to the two flow methods.

```
for(Int_t i=0; i<nEvts; i++) {
      // make an event with mult particles
      AliFlowEventSimple* event = new AliFlowEventSimple(mult,
    AliFlowEventSimple::kGenerate);
      // modify the tracks adding the flow value v2
      event->AddV2(v2);
      // select the particles for the reference flow
      event->TagRP(cutsRP);
      // select the particles for differential flow
      event->TagPOI(cutsPOI);
      // do flow analysis with various methods:
      mcep->Make(event);
      qc->Make(event);
}
```

6. To fill the histograms which contain the final results we have to call Finish for each method:

```
mcep->Finish();
qc->Finish();
```

7. This concludes the analysis and now we can write the results into a file:

```
TFile *outputFile = new TFile("AnalysisResults.root","RECREATE")
    ;
mcep->WriteHistograms();
qc->WriteHistograms();
```

2.3 What is in the output file?

Now we have written the results into a file, but what is in there?

2.4 Reading events from file

The macro `Documentation/examples/runFlowReaderExample.C` is an example how to setup a flow analysis if the events are already generated and for example are stored in ntuples.

2.5 A simple flow analysis in ALICE using Tasks

The macro `Documentation/examples/runFlowTaskExample.C` is an example how to setup a flow analysis using the full ALICE Analysis Framework.

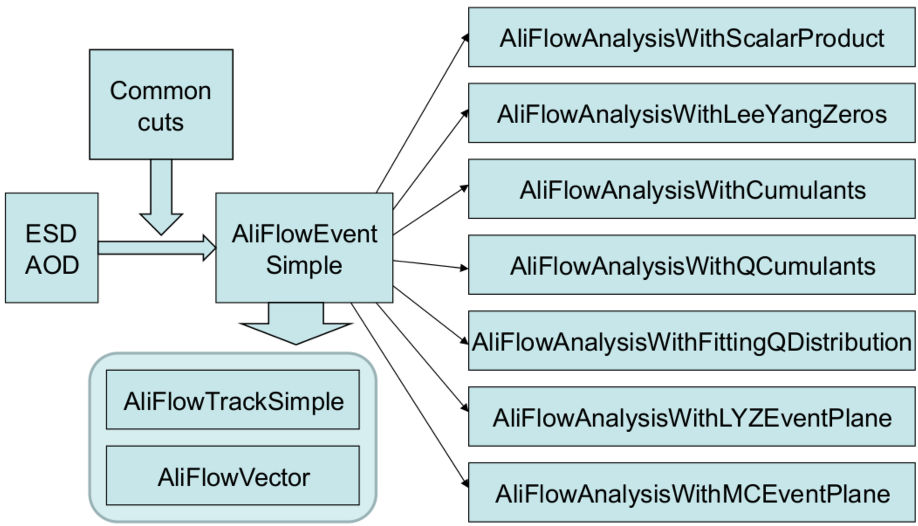
Chapter 3

The Flow Event

Here we describe the flowevent, flowtracks, general cuts and cuts for RPs POIs. OntheFly, AfterBurner. Filling with ESD, AOD, Ntuples, etc.

Chapter 4

The Program



Here we describe the program.

Chapter 5

Methods

The flow package aims at providing the user with most of the known flow analysis methods. Detailed theoretical overview of the methods can be found in the following papers, which are included in the folder `$ALICE_ROOT/PWGCF/FLOW/Documentation/otherdocs/`

- The Monte-Carlo Truth
- Scalar Product Method
  `EventPlaneMethod/FlowMethodsPV.pdf`
- Generating Function Cumulants
  `GFCumulants/Borghini_GFCumulants_PracticalGuide.pdf`
- Q-vector Cumulant method
  `QCumulants/QCpaperdraft.pdf`
- Lee-Yang Zero Method
  `LeeYangZeroes/Borghini_LYZ_PracticalGuide.pdf`
- Lee-Yang Zero Method
  `LeeYangZeroesEP/LYZ_RP.pdf`

The structure of this chapter is as follows: of each of the available methods a short description is given in the `theory` subsection (for more detailed information, see the papers listed above) followed by details which are specific to the implementation in the subsection `implementation`. Caveats, possible issues, etc, are listed in the `caveats` subsections.

5.1 The Monte-Carlo Truth

Here we describe the implementation of the monte-carlo truth.

5.2 Scalar Product Method

5.2.1 Theory

The scalar product method

The scalar product method - as well as the Q-cumulant method which will be described later - does not depend on explicit construction of an (sub)event plane, but estimates v_n directly from multi-particle correlations.

To do so, firstly all particles in an event are labeled either as *reference particles* (RP's) or *particles of interest* (POI's). The RP and POI selections are in turn divided into sub-events, which are again taken from different η ranges, in analogy to the approach taken for the event plane method. Each POI is correlated with a sub-event Q-vector from the RP selection, which allows for the calculation of v_n without any explicit reconstruction of an event plane[**?**].

The reason for the division into RP's and POI's is the fact that the two particle correlator of POI's,

$$v_n^{POI} = \sqrt{\left\langle e^{in(\phi_i^{POI} - \phi_j^{POI})} \right\rangle} \tag{5.2.1.1}$$

is generally not stable statistically. Introducing reference flow, 5.2.1.1 can be rewritten as

$$v_n^{POI} = \frac{\left\langle e^{in(\phi_i^{POI} - \phi_j^{RP})} \right\rangle}{\left\langle e^{in(\phi_i^{RP} - \phi_j^{RP})} \right\rangle} = \frac{v_n^{POI} v_n^{RP}}{\sqrt{v_n^{RP} v_n^{RP}}}. \tag{5.2.1.2}$$

By taking an abundant particle source as RP's - in the case of this study the RP selection comprises all charged particles - both correlators in 5.2.1.2 are statistically stable.

The scalar product method In the scalar product method, POI's u_k,

$$u_k = e^{in\phi_k}, \tag{5.2.1.3}$$

are correlated with Q_a^*, the complex-conjugate Q-vector built from RP's in a given sub-event a. First, the scalar product of u_k and Q_a^* is taken,

$$u_k \cdot \sum_{\substack{j=1,\\ j\neq k}}^{M_{RP,a}} u_j^* \tag{5.2.1.4}$$

where $M_{RP,a}$ denotes RP multiplicity for a given sub-event a and the inequality $j \neq k$ removes auto-correlations. From this, differential v_n of POI's (v_n') and v_n

of RP's (v_n^a) in sub-event a can be obtained in a straightforward way from the correlation of POI's and RP's:

$$\langle u\cdot Q_a^*\rangle = \frac{1}{M_{RP,a}-k}\sum_{i=k}^{M_{RP,a}}\left(u_k\sum_{\substack{j=1,\\ j\neq k}}^{M_{RP,a}} u_j^*\right) \tag{5.2.1.5}$$

where POI multiplicity is expressed in terms of $M_{RP,a}$; $M_{POI} = M_{RP,a} - k$. Since for any function $f(x)$ and constant a

$$\sum af(x) = a\sum f(x) \tag{5.2.1.6}$$

5.2.1.5 can be rewritten as

$$\begin{aligned}\langle u\cdot Q_a^*\rangle &= \frac{1}{M_{RP,a}-k}\sum_{i=k}^{M_{RP,a}} e^{in[\phi_k-\Psi_n]}\sum_{j=1}^{M_{RP,a}} e^{-in[\phi_j-\Psi_n]} \\ &= M_{RP,a}v_n' v_n^a\end{aligned} \tag{5.2.1.7}$$

where in the last step of 5.2.1.7 it has been used that

$$v_n = \frac{\sum_i^M e^{in[\phi_i-\Psi_n]}}{M}. \tag{5.2.1.8}$$

To obtain the estimate of v_n, one must still disentangle the reference flow contribution from the event averaged correlation given in 5.2.1.5. Proceeding in a fashion similar to that presented in equation 5.2.1.5, it can be shown that

$$\left\langle \frac{Q_a}{M_a}\cdot\frac{Q_b^*}{M_b}\right\rangle = \left\langle v_n^a v_n^b\right\rangle \tag{5.2.1.9}$$

where Q_a, Q_b are the Q-vectors of RP's in sub-event a, b. Under the assumption that

$$\left\langle v_n^2\right\rangle = \left\langle v_n\right\rangle^2, \tag{5.2.1.10}$$

- an assumption which will be spoiled in the case of flow fluctuations - and requiring that the v_n estimates in both sub-events are equal, one simply evaluates

$$v_n' = \frac{\left\langle\left\langle u\cdot\frac{Q_a^*}{M_a}\right\rangle\right\rangle}{\sqrt{\left\langle\frac{Q_a}{M_a}\cdot\frac{Q_b^*}{M_b}\right\rangle}} \tag{5.2.1.11}$$

to obtain v_n^a. For equal multiplicity sub-events $M_a = M_b$, 5.2.1.11 is simplified to

$$v_n' = \frac{\langle\langle u\cdot Q_a^*\rangle_t\rangle}{\sqrt{\langle Q_a\cdot Q_b^*\rangle}}. \tag{5.2.1.12}$$

v_n^b can be obtained by switching indices a and b in expressions 5.2.1.11 and 5.2.1.12, and should equal v_n^a. This principle can be generalized straightforwardly to allow for a selection of RP's which has been divided into three subevents.

$$v_n^a = \frac{\left\langle\left\langle u \cdot \frac{Q_a^*}{M_a}\right\rangle\right\rangle}{\sqrt{\langle v_n'^a v_n'^b\rangle \langle v_n'^a v_n'^c\rangle / \langle v_n'^b v_n'^c\rangle}} = \frac{\left\langle\left\langle u \cdot \frac{Q_a^*}{M_a}\right\rangle\right\rangle}{\sqrt{\left\langle \frac{Q_a}{M_a} \cdot \frac{Q_b^*}{M_b}\right\rangle \left\langle \frac{Q_a}{M_a} \cdot \frac{Q_c^*}{M_c}\right\rangle / \left\langle \frac{Q_b}{M_b} \cdot \frac{Q_c^*}{M_c}\right\rangle}} \tag{5.2.1.13}$$

where cyclic permutation of a, b, c (in analogy to the switching of indices in **??** gives the estimates of v_n^b and v_n^c.[insert some discussion here: is this result actually true, and some light on va, vb, (vc)]

5.2.2 Implementation

Extension to Event Plane method

As explained earlier, the event plane analysis results in this study are actually obtained by normalizing the Q-vectors in the scalar product by their length $|Q_n|$. Consider the following:

$$\frac{Q_n^*}{|Q_n^*|} = \frac{|Q_n^*| e^{-in\Psi_{Q_n}}}{|Q_n^*|} = e^{-in\Psi_{Q_n}}. \tag{5.2.2.1}$$

For a full event, the enumerator of 5.2.1.11 can be expressed as

$$\langle\langle u \cdot e^{-in\Psi_{Q_n}}\rangle\rangle = \langle\langle e^{in\phi_i} \cdot e^{-in\Psi_{Q_n}}\rangle\rangle = \left\langle\left\langle e^{in(\phi_i - \Psi_{Q_n})}\right\rangle\right\rangle = \langle\langle \cos(n[\phi_i - \Psi_{Q_n}])\rangle\rangle$$

which corresponds to the all-event average of **??**. As shown in the previous subsection this expression equals v_n^{obs}.

For normalized Q-vectors, the denominator of 5.2.1.11 reads (using 5.2.2.1):

$$\sqrt{\left\langle \frac{Q_a}{|Q_a|} \cdot \frac{Q_b^*}{|Q_b^*|}\right\rangle} = \sqrt{\left\langle e^{in[\Psi_{Q_{n_a}} - \Psi_{Q_{n_b}}}\right\rangle} = \sqrt{\left\langle \cos(n[\Psi_{Q_{n_a}} - \Psi_{Q_{n_b}}]\right\rangle)} \tag{5.2.2.2}$$

from which the event plane resolution can be calculated using **??** or **??**.

Caveats

5.3 Generating Function Cumulant Method

Here we describe the generating function cumulant method and how it is implemented.

5.4 Q-vector Cumulant Method

5.4.1 Theory

The Q-cumulant (QC) method[a] uses multi-particle correlations to estimate v_n estimates for RP's and POI's, but does not limit itself to two-particle correlations. Although higher-order Q-cumulant calculations are available, this section will discuss the method using two- and four-particle correlations.

Multi-particle correlations in the QC method are expressed in terms of cumulants, which are the the expectation values of correlation terms in joint probability density functions. Consider the following: if two observables f for particles x_i and x_j are correlated, the joint probability $f(x_i, x_j)$ is the sum of the factorization of the constituent probabilities and a covariance term:

$$f(x_i, x_j) = f(x_i)f(x_j) + f_c(x_i, x_j) \tag{5.4.1.1}$$

When taking as an observable azimuthal dependence,

$$x_i \equiv e^{in\phi_i}, \qquad x_j \equiv e^{in\phi_j} \tag{5.4.1.2}$$

the two-particle cumulant is expressed as the covariance of the expectation value:

$$E_C(e^{in[\phi_i-\phi_j]}) = E(e^{[in(\phi_i-\phi_j]}) - E(e^{in[\phi_i]})E(e^{in[-\phi_j]}). \tag{5.4.1.3}$$

Symmetry arguments (along the lines of those given in appendix **??**) dictate that the product of separate expectation values is equals zero, from which a familiar expression for the two-particle correlation is obtained:

$$\begin{aligned} E_C(e^{in[\phi_i-\phi_j]}) &= E(e^{in[\phi_i]})E(e^{in[-\phi_j]}) \\ &= \left\langle e^{in[\phi_i]} \right\rangle \left\langle e^{in[-\phi_j]} \right\rangle \\ &= \left\langle e^{in[\phi_i-\phi_j]} \right\rangle \\ &= \langle 2 \rangle , \end{aligned} \tag{5.4.1.4}$$

the all-event average of which is denoted by

$$c_n\{2\} = \langle\langle 2 \rangle\rangle \tag{5.4.1.5}$$

where $c_n\{2\}$ is called the two-particle cumulant. For the four-particle case, one proceeds likewise:

$$\begin{aligned} E_c(e^{in[\phi_i+\phi_j-\phi_k-\phi_l]}) &= E(e^{in[\phi_i+\phi_j-\phi_k-\phi_l]}) \\ &- E(e^{in[\phi_i-\phi_k]})E(e^{in[\phi_j-\phi_l]}) \\ &- E(e^{in[\phi_i-\phi_l]})E(e^{in[\phi_j-\phi_k]}). \end{aligned} \tag{5.4.1.6}$$

[a] The overview given in this section is inspired by [?], for further reading the reader is referred there. A full derivation of results that are relevant in this study is given in appendix **??**.

The four-particle cumulant can be expressed in terms of two- and four-particle correlations as well,

$$c_n\{4\} = \langle\langle 4\rangle\rangle - 2\,\langle\langle 2\rangle\rangle^2 \,. \tag{5.4.1.7}$$

From 5.4.1.5 and 5.4.1.7 it follows that v_n harmonics are related to cumulants following

$$\begin{aligned} v_n\{2\} &= \sqrt{c_n\{2\}} \\ v_n\{4\} &= \sqrt[4]{-c_n\{4\}}. \end{aligned} \tag{5.4.1.8}$$

where $v_n\{2\}$, $v_n\{4\}$ denote flow estimates obtained from two- and four-particle correlations.

In a fashion similar to that explained in the previous subsection, the Q-cumulant method uses reference flow to obtain a statistically stable estimate of the differential flow of POI's. Differential POI flow, for the two- and four-particle case, can be expressed as

$$\begin{aligned} d_n\{2\} &= \langle\langle 2'\rangle\rangle \\ d_n\{4\} &= \langle\langle 4'\rangle\rangle - 2\cdot \langle\langle 2'\rangle\rangle \langle\langle 2\rangle\rangle \end{aligned} \tag{5.4.1.9}$$

where $d_n\{2\}, d_n\{4\}$ denotes the two-, four-particle differential flow and the $'$ is used as an indicator for differential (p_t dependent) results. Disentangling from this the reference flow contributions, one is left with the final expression for the estimate of differential v_n for POI's:

$$\begin{aligned} v'_n\{2\} &= \frac{d_n\{2\}}{\sqrt{c_n\{2\}}} \\ v'_n\{4\} &= -\frac{d_n\{4\}}{(-c_n\{2\})^{3/4}}. \end{aligned} \tag{5.4.1.10}$$

5.4.2 Implementation

Here we describe the Q-vector cumulant method and how it is implemented.

5.5 Lee-Yang Zero Method

Here we describe the Lee-Yang Zero method and how it is implemented.

5.6 Lee-Yang Zero Method

Here we describe the Lee-Yang Zero method and how it is implemented.

5.7 Fitting the Q-vector Distribution

Here we describe how the fitting of the Q-vector distribution is implemented.

Chapter 6

Summary

This sums it all up.

Chapter 7

Bibliography

[1] J. Y. Ollitrault, Phys. Rev. D **46** (1992) 229.

[2] P. Danielewicz, Nucl. Phys. A **661** (1999) 82.

[3] D. H. Rischke, Nucl. Phys. A **610** (1996) 88C.

[4] J. Y. Ollitrault, Nucl. Phys. A **638** (1998) 195.

[5] S. Voloshin and Y. Zhang, Z. Phys. C **70** (1996) 665.

[6] K. H. Ackermann *et al.* [STAR Collaboration], Phys. Rev. Lett. **86** (2001) 402

[7] C. Adler *et al.* [STAR Collaboration], Phys. Rev. Lett. **87** (2001) 182301

[8] T.D. Lee *et al.*, New Discoveries at RHIC: Case for the Strongly Interacting Quark-Gluon Plasma. Contributions from the RBRC Workshop held May 14-15, 2004. Nucl. Phys. A **750** (2005) 1-171

Appendix I

Here we put short pieces of code.

Index