The FLOW Analysis Package

a short writeup

25-10-2010

Contents

Chapter 1

A Quick Start

The *ALICE flow package*[1] 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.

1.1   On the fly

The macro `Documentation/examples/runFlowOnTheFlyExample.C` 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("libPWG2flowCommon");
   ```
   In *root* additional libraries need to be loaded:
   ```
   gSystem->Load("libGeom");
   gSystem->Load("libVMC");
   gSystem->Load("libXMLIO");
   gSystem->Load("libPhysics");
   gSystem->Load("libPWG2flowCommon");
   ```

---

[1] `http://alisoft.cern.ch/viewvc/trunk/PWG2/FLOW/?root=AliRoot` .

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 4.1), and second the so called Q-cumulant method (see section 4.4). `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 trackcut 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,AliFlowEventSimpl`
   `// 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);`

```
} // end of for(Int_t i=0;i<nEvts;i++)
```

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();
```

1.2   What is in the output file?

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

1.3   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.

1.4   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 2

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 3

The Program

Here we describe the program.

Chapter 4

Methods

Here we put an intro for the various methods.

4.1   The Monte-Carlo Truth

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

4.2   Scalar Product Method

Here we describe the scalar product method and how it is implemented.

4.3   Generating Function Cumulant Method

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

4.4 Q-vector Cumulant Method

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

4.5   Lee-Yang Zero Method

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

4.6 Lee-Yang Zero Event Plane Method

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

4.7   Fitting the Q-vector Distribution

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

Chapter 5

Summary

This sums it all up.

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.