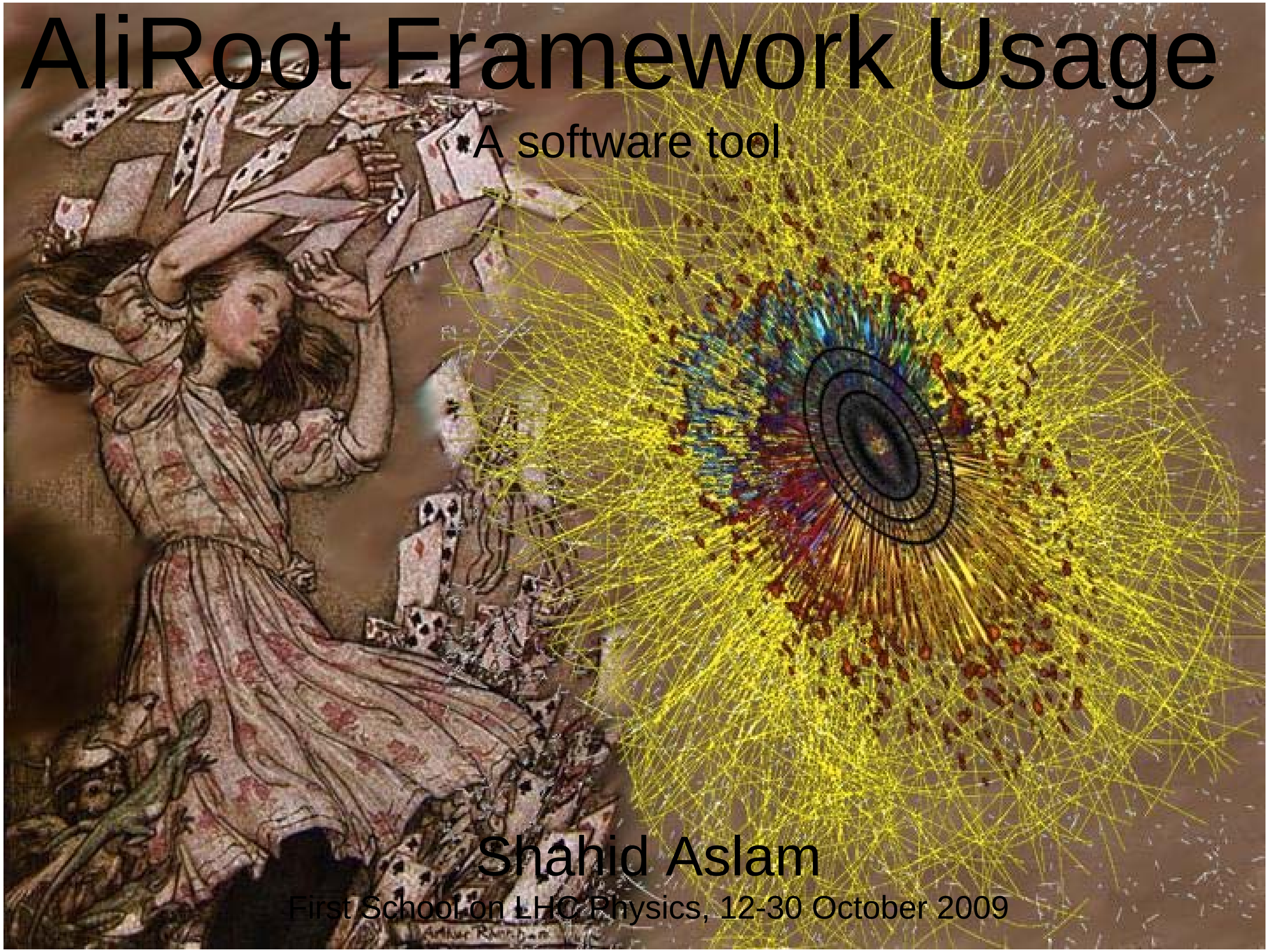
AliRoot Framework Usage

A software tool



Shahid Aslam

First School on LHC Physics, 12-30 October 2009

New state of matter at ALICE

- Collisions in the LHC will generate temperatures more than 100 000 times hotter than the heart of the Sun.
- Physicists hope that under these conditions, the protons and neutrons will 'melt', freeing the quarks from their bonds with the gluons.
- This should create a state of matter called **quark-gluon plasma**, which probably existed just after the Big Bang when the Universe was still extremely hot.
- The ALICE collaboration plans to study the quark-gluon plasma as it expands and cools, observing how it progressively gives rise to the particles that constitute the matter of our Universe today.

Executables available in Aliroot Framwork

- aliroot, aliengui, alieve, alimdc, alirawdump,, dateStream, deroot, dHLTdumpraw, mchview, monitorCheck, monitorGDC

Directory Structure

AliRoot

ACORDE			BCM		build	
EVGEN	FASTSIM	FMD	GRP	HBTAN	HERWIG	
	Makefile	man	MICROCERI	MONITOR	MUON	
PYTHIA8	RALICE	RAW	share	SHUTTLE	STAT	
THydjet	TIsajet	TOF	TPC	TPHIC	TRD	
doc	DPMJET	EMCAL	ESDCheck	etc	Euclid	
HMPID	html	include	ISAJET	ITS	JETAN	
picts	PMD	PWG0	PWG1	PWG2	PWG3	
T0	TDPMjet	TEPEMGEN	test	TFluka	THbtp	
ZDC		s				
CORRFW	data				macros	
HIJING	HLT	EVE	lib	ALIROOT	ANALYSIS	
	PHOS	LHAPDF	LHC		bin	
STEER	STRUCT	PWG4	PYTHIA6			
TTherminator	VZERO	THerwig	THijing			

Simulation Steps 1

Time

Steps	**File Name**	**Tree Name**	**Behind the Scene**
Mc Particle	Kinematics.root	TreeK	Different particles
Tracking Particles	Hits.root	TreeH	Transport through detector
Conversion Hits->Sdigits	Sdigits.root	TreeS	Pads with its associated charge.No noise and electronics response function applied.
Sdigits->Digits	Digits.root	TreeD	Calibration/adding pedestal and divide by gain.**Trigger (local, regional, global)**
Ready for Reconstruction			

Simulation steps 2

Some other output files from simulation:

- HLT.digits.root
- Trigger.root
- galice.root

Tracking (behind the scene)

- **Usage of numerical Algorithm :**

Geant : HELIX method and RKUTA method.

What to choose:

HELIX is fast.

RKUTA is good approx. for MUON

If Magnetic field is required : upto 10T

```
AliMagFMaps* field = new AliMagFMaps ("Maps ","Maps", 1, 1., 10., AliMagFMaps::k5kG);

 gAlice->SetField(field);
```

--- ---

Special Cuts to Optimize Tracking Speed :

Parameters that can be changed for each detector as needed:

```
*  ---  GAM   ELEC  NHAD  CHAD  MUON  EBREM MUHAB EDEL  MUDEL MUPA
*  ---  ANNI BREM COMP DCAY DRAY HADR LOSS MULS PAIR PHOT RAYL
```

For Example : Gas Mixture medium of tracking chamber of muon for realistic energy loss in active gas parameter LOSS=1

all above parameters are set in one file.

Starting AliRoot Session

- To start up AliRoot after installation just type
- $ aliroot
- this starts AliRoot in interactive mode. When the **root [0]** prompt appears, you can start an AliRoot interactive session by creating the standard menu
- root[0] .x menu.C
- The macro menu.C invokes the AliRun::Init() method. In turn this calls the configuration file Config.C where the configuration of the detector and several simulation parameters can be specified.
- When the initialisation is finished the **root** prompt apears again and the Root control bar appears.

Simulation

- The response function of ALICE detector can be simulated within AliRoot for different physics cases.
- So AliRoot is interfaced with different Monte Carlo event generators, like Pythia, Herwig and Hijing.
- Response functions of the ALICE sub detectors are simulated from the hit information in the active volumes provided by the transport codes.
- Finally ALICE raw-data files are generated.

Reconstruction

- AliRoot reconstruction framework performs the clusterization, particle trajectory reconstruction and particle identification from the raw-data of the ALICE detector.
- reconstruction allows for detector alignment, calibration corrections and detector quality assurance.
- Reconstruction output consists of ALICE ESD (event summary data) files.
- Metadata information of reconstructed events in the ESD file (like for instance the muon multiplicity) are stored in the Tag database of AliRoot.

Analysis

- The analysis framework of AliRoot allows for building a train of different analysis tasks that will select specific events and will perform specific physics analysis.
- The results of the analysis will be stored in a small size files called AOD (Analysis Object Data).
- The Config.C file is fundamental for the analysis. In this file you have to write all the things you want to take into account.

Settings inside Config.C

What Is Config.C.mht

How to get info about C++ Function to be used in Config.C File

- **AliMUONSt2GeometryBuilderV2( )**
- **http://aliceinfo.cern.ch/static/aliroot-new/html/roothtml/MUON_Index.html**
- **MUON Station2 detailed geometry construction class**
- Data Members **private:** AliMUON***fMUON**
- **MUON Station2 coarse geometry construction class. Author: SANJOY PAL ,Prof. SUKALYAN CHATTOPADHAYAY Dr.SHAKEEL AHMAD**

Config.c 1

```
•  /// Brief Configuration macro for MUON spectormeter simulation
•  //http://aliceinfo.cern.ch/Offline/Activities/Simulation/configC.html
•  /// Remember to define the directory and event generator option:
•  ///
•  /// gAlice->SetConfigFunction("Config('$HOME','box');");

•  void Config(char directory[100]="", char option[6]="param", const char*
   digitstore="AliMUONDigitStoreV2S")
•  {
•    //================================================================
•    // Config file for MUON test
•    //================================================================
•    //  Libraries required by geant321
•    gSystem->Load("liblhapdf.so");      // Parton density functions
•    gSystem->Load("libpythia6.so");     // Pythia
•    gSystem->Load("libgeant321.so");
•    gSystem->Load("libEG");
•    gSystem->Load("libEGPythia6");
•    gSystem->Load("libAliPythia6.so");  // ALICE specific implementations

•    new TGeant3TGeo("C++ Interface to Geant3");
```

Config.c   2

```
 //  Create the output file
Text_t filename[100];
sprintf(filename,"%sgalice.root",directory);
cout << ">>> Output file is " << filename << endl;
cout << ">>> Config.C: Creating Run Loader ..."<<endl;
AliRunLoader* rl=0x0;
rl = AliRunLoader::Open ( filename, AliConfig::GetDefaultEventFolderName(), "recreate");
if (rl == 0x0)
  { gAlice->Fatal("Config.C","Can not instatiate the Run Loader");
    return; }
rl->SetCompressionLevel(2);
rl->SetNumberOfEventsPerFile(100);
gAlice->SetRunLoader(rl);
//==========================================
// For having more debuging messages
//AliLog::SetModuleDebugLevel("MUON", 1);
```

Config.c 3

```
// Set External decayer (Pythia is used)
 TVirtualMCDecayer *decayer = new AliDecayerPythia();
 decayer->SetForceDecay(kAll);
 decayer->Init();
 gMC->SetExternalDecayer(decayer);
 //============================================================
 // ******* GEANT STEERING parameters FOR ALICE SIMULATION *******
//Switch on/off the physics processes (global)
 // Please consult the file data/ galice.cuts for detector
 // specific settings
 gMC->SetProcess("DCAY",1);
 gMC->SetProcess("PAIR",1);
 gMC->SetProcess("COMP",1);
 gMC->SetProcess("PHOT",1);
 gMC->SetProcess("PFIS",0);
 gMC->SetProcess("DRAY",0);
 gMC->SetProcess("ANNI",1);
 gMC->SetProcess("BREM",1);
 gMC->SetProcess("MUNU",1);
 gMC->SetProcess("CKOV",1);
 gMC->SetProcess("HADR",1);
 gMC->SetProcess("LOSS",2);
 gMC->SetProcess("MULS",1);
```

Config.c   4

```
gMC->SetProcess("RAYL",1);
 Float_t cut = 1.e-3;                    // 1MeV cut by default
 Float_t tofmax = 1.e10;
 gMC->SetCut("CUTGAM", cut);
 gMC->SetCut("CUTELE", cut);
 gMC->SetCut("CUTNEU", cut);
 gMC->SetCut("CUTHAD", cut);
 gMC->SetCut("CUTMUO", cut);
 gMC->SetCut("BCUTE",  cut);
 gMC->SetCut("BCUTM",  cut);
 gMC->SetCut("DCUTE",  cut);
 gMC->SetCut("DCUTM",  cut);
 gMC->SetCut("PPCUTM", cut);
 gMC->SetCut("TOFMAX", tofmax);

 //========================================
```

Config.c 5

```
// Examples of generators. Only option param is sistematically tested
  if (!strcmp (option ,"box")) {
    AliGenBox * gener = new AliGenBox(1);
    gener ->SetMomentumRange(20.,20.1);
    gener ->SetPhiRange(0., 360.);
   gener->SetThetaRange(171.000,178.001);
    gener->SetPart(kMuonMinus);                    // Muons
    gener->SetOrigin(0.,0., 0.);  //vertex position
    gener->SetSigma(0.0, 0.0, 0.0);        //Sigma in (X,Y,Z) (cm) on IP position
  }
  if (!strcmp(option,"gun")) {
    AliGenFixed *gener = new AliGenFixed(1);
    gener->SetMomentum(10);
    gener->SetPhiRange(0.);
    gener->SetThetaRange(0.);
    gener->SetOrigin(30,30,-1200);//vertex position
    gener->SetPart(kMuonMinus);        //GEANT particle type  13 is muons
  }
```

Config.c 6

```
•  if (!strcmp (option ,"scan")) {
•      AliGenScan *gener = new AliGenScan(-1);
•      gener ->SetMomentumRange(10,10);
•      gener ->SetPhiRange(0, 0);
•      gener ->SetThetaRange(-180, -180);
•      //vertex position
•      //gener ->SetSigma(1,1,0);    //Sigma in (X,Y,Z) (cm) on IP position
•      gener ->SetPart(kRootino);
•      gener ->SetRange(100, -300., 300., 100, -300., 300., 1, 2000, 2000);
•    }
```

Config.c 7

```
•   if (!strcmp (option ,"param")) {
•       AliGenParam *gener = new AliGenParam(1, AliGenMUONlib::kUpsilon);
•       gener->SetMomentumRange(0,999);
•       gener->SetPtRange(0,100.);
•       gener->SetPhiRange(0., 360.);
•       gener->SetCutOnChild(1);
•       gener->SetChildPhiRange(0.,360.);
•       gener->SetChildThetaRange(171.0,178.0);
•       gener->SetOrigin(0,0,0);          //vertex position
•       gener->SetSigma(0,0,0);           //Sigma in (X,Y,Z) (cm) on IP position
•       gener->SetForceDecay(kDiMuon);
•       gener->SetTrackingFlag(1);
•       gener->Init();
•     }
•      if (!strcmp (option ,"hijing")) { //Hijing generator from ConfigPPR in macros
•
•       AliGenHijing *gener = new AliGenHijing(-1);
•       // centre of mass energy
•       gener->SetEnergyCMS(5500.);
```

Config.c 8

```
      // reference frame
          gener->SetReferenceFrame("CMS");
      // projectile
          gener->SetProjectile("A", 208, 82);
          gener->SetTarget    ("A", 208, 82);
      // tell hijing to keep the full parent child chain
          gener->KeepFullEvent();
      // enable jet quenching
          gener->SetJetQuenching(1);
      // enable shadowing
          gener->SetShadowing(1);
      // neutral pion and heavy particle decays switched off
          gener->SetDecaysOff(1);
      // Don't track spectators
          gener->SetSpectators(0);
      // kinematic selection
          gener->SetSelectAll(0);
      // impact parameter range
          gener->SetImpactParameterRange(0., 5.);
     // 0. - 5. fm corresponds to ~10% most central
          gener->Init();
    }
```

Config.c 9

```
if (!strcmp(option,"muoncocktail")) { // Muon cocktail for PbPb
    AliGenMUONCocktail * gener = new AliGenMUONCocktail();
    gener->SetPtRange(1.,100.);       // Transverse momentum range
    gener->SetPhiRange(0.,360.);    // Azimuthal angle range
    gener->SetYRange(-4.0,-2.5);
    gener->SetMuonPtCut(0.5);
    gener->SetMuonThetaCut(171.,178.);
    gener->SetMuonMultiplicity(2);
    gener->SetImpactParameterRange(0.,5.); // 10% most centra PbPb collisions
    gener->SetVertexSmear(kPerTrack);
    gener->SetOrigin(0,0,0);        // Vertex position
    gener->SetSigma(0,0,0.0);       // Sigma in (X,Y,Z) (cm) on IP position
    gener->Init();
  }
// Field (L3 0.5 T)
  AliMagFMaps* field = new AliMagFMaps("Maps","Maps", 1, 1., 10., AliMagFMaps::k5kG);
  gAlice->SetField(field);
```

Config.c 10

- AliMUONResponseTriggerV1
- // MUON->SetTriggerResponseV1(0);
- // activate 4/4 trigger coincidence (0=default (coinc 3/4), 1=coinc 4/4)
- // MUON->SetTriggerCoinc44(0);
- // activate trigger chamber efficiency by cells (0=default, 1=trigger efficiency according to AliMUONTriggerEfficiencyCells
- // MUON->SetTriggerEffCells(0);

- // Use SetDigitStoreClassName() to change the digitStore implementation used by (s)digitizer
- MUON->SetDigitStoreClassName(digitstore);
-
- cout << "MUON DigitStore is " << MUON->DigitStoreClassName().Data() << endl;
-
- // Noise-only digits in tracker/trigger (0=no noise, 1=default (noise in tracker), 2=noise in tracker and trigger):
- //MUON->SetDigitizerWithNoise(kFALSE);

- // Use fast raw data decoder
- // MUON->SetFastDecoder(kTRUE);

Config.c 11

- // // If SetAlign, the detection elements transformations
- // are taken from the input file and not from the code
- // MUON->SetAlign("transform.dat");
- // To generate and read scaler trigger events in rawdata
- // MUON->SetTriggerScalerEvent();
- // To switch off the tail effect
- // MUON->SetTailEffect(kFALSE);

- // If you want to play with builders, first reset the geometry builder,
- // and then add yours.
- // MUON->ResetGeometryBuilder();
- // MUON->AddGeometryBuilder(new AliMUONSt1GeometryBuilderV2(MUON));
- // MUON->AddGeometryBuilder(new AliMUONSt2GeometryBuilderV2(MUON));
- // MUON->AddGeometryBuilder(new AliMUONSlatGeometryBuilder(MUON));
- // MUON->AddGeometryBuilder(new AliMUONTriggerGeometryBuilder(MUON));
- }

Config.c 12

- //=================== Alice BODY parameters ======================
- AliBODY *BODY = new AliBODY("BODY","Alice envelop");
- //=================== ABSO parameters ============================
- AliABSO *ABSO = new AliABSOv3("ABSO", "Muon Absorber");
- //=================== DIPO parameters ============================
- AliDIPO *DIPO = new AliDIPOv3("DIPO", "Dipole version 2");
- //================== HALL parameters ============================
- AliHALL *HALL = new AliHALLv3("HALL", "Alice Hall");
- //=================== PIPE parameters ============================
- AliPIPE *PIPE = new AliPIPEv3("PIPE", "Beam Pipe");
- //=================== SHIL parameters ============================
- AliSHIL *SHIL = new AliSHILv3("SHIL", "Shielding Version 2");
-

Config.c 13

- //=================== MUON Subsystem =========================
- AliMUON *MUON = new AliMUONv1("MUON", "default");
- // The 3 switches below are to be used for the trigger code
- // their default value is set in AliMUON.h
- // activate trigger cluster-size (0=default, 1=cluster-size according to AliMUONResponseTriggerV1
- // MUON->SetTriggerResponseV1(0);
- // activate 4/4 trigger coincidence (0=default (coinc 3/4), 1=coinc 4/4)
- // MUON->SetTriggerCoinc44(0);
- // activate trigger chamber efficiency by cells (0=default, 1=trigger efficiency according to AliMUONTriggerEfficiencyCells
- // MUON->SetTriggerEffCells(0);
- // Use SetDigitStoreClassName() to change the digitStore implementation used by (s)digitizer
- MUON->SetDigitStoreClassName(digitstore);
-
- cout << "MUON DigitStore is " << MUON->DigitStoreClassName().Data() << endl;
-
-

Config.c 14

- // Noise-only digits in tracker/trigger
- // (0=no noise, 1=default (noise in tracker), 2=noise in tracker and trigger):
- //MUON->SetDigitizerWithNoise(kFALSE);
- // Use fast raw data decoder
- // MUON->SetFastDecoder(kTRUE);
- // If SetAlign, the detection elements transformations
- // are taken from the input file and not from the code
- // MUON->SetAlign("transform.dat");
- // To generate and read scaler trigger events in rawdata
- // MUON->SetTriggerScalerEvent();
- // To switch off the tail effect
- // MUON->SetTailEffect(kFALSE);
- // If you want to play with builders, first reset the geometry builder,
- // and then add yours.
- // MUON->ResetGeometryBuilder();
- // MUON->AddGeometryBuilder(new AliMUONSt1GeometryBuilderV2(MUON));
- // MUON->AddGeometryBuilder(new AliMUONSt2GeometryBuilderV2(MUON));
- // MUON->AddGeometryBuilder(new AliMUONSlatGeometryBuilder(MUON));
- // MUON->AddGeometryBuilder(new AliMUONTriggerGeometryBuilder(MUON));
- }

MUON spectrometer

- The offline software for simulation, reconstruction and analysis of the muon spectrometer is shared by five modules of AliRoot: MUON, EVGEN, EVE, HLT and PWG3.

MUON

- This module contains most of the software that has been developed in AliRoot for the muon spectrometer. It contains the code which is needed to perform the main goals of AliRoot: simulation and reconstruction. The present document will mainly focus on this development.

EVGEN

- This module contains the parameterized event generators of AliRoot and the virtual classes for interfacing AliRoot with the Monte Carlo event generators.
- EVGEN module contains a library class for MUON physics with the main parameterizations of (di)muon production from quarkonia decays, from open heavy flavour semi leptonic and from light hadron weak decays.
- AliGenMUONCocktail is focused on muon production in heavy ion collisions at LHC energies.

HLT

- The High Level Trigger (HLT) is an online processing stage whose primary function is to select interesting events from unwanted background events.
- The HLT applies pattern recognition to the online raw data in the form of fast online event reconstruction and makes a trigger decision based on the reconstructed information.
- The ALICE muon spectrometer's HLT implementation (dHLT) is found under HLT/MUON and is compiled to the library libAliHLTMUON.so.

EVE

- The Event Visualization Environment (EVE) is a general framework for the visualisation of the detector geometry and of the event-related data:
- digits, reconstructed points, tracks and hits from the simulations.
- It supports the 3D visualization in several modes, including the rendering of objects with OpenGL

PWG3

- The PWG3 module contains the analysis code prepared for the study of the topics addressed by the PhysicsWorking Group 3 of ALICE, i.e. the study of quarkonia and open heavy flavours.
- Since several PWG3 physics analyses are
- based on measurements performed with the forward muon spectrometer.
- the related code has been developed in two submodules of PWG3: PWG3/muon and PWG3/muondep

How to use Config.C using Shell Script

```
#!/bin/sh
# $Id: sim.sh 22703 2007-12-03 10:16:38Z ivana $
CURDIR=`pwd`
OUTDIR=sim_out
rm -fr $OUTDIR
mkdir $OUTDIR
cp $ALICE_ROOT/MUON/.rootrc $ALICE_ROOT/MUON/rootlogon.C $OUTDIR
#cp .rootrc rootlogon.C $OUTDIR
cd $OUTDIR
# Minimum number of events to have enough stat. for invariant mass fit
# 10000 is ok, 20000 is really fine
NEVENTS=20
SEED=1234567
echo "Running simulation  ..."
aliroot -b >& testSim.out << EOF
AliSimulation MuonSim("/home/alice01/666/MUON/Config.C");
MuonSim.SetSeed($SEED);
MuonSim.SetMakeTrigger("MUON");
MuonSim.Run($NEVENTS);
.q
EOF
more  testSim.out | grep 'RunSimulation: Execution time:'  > testTime.out
more  testSim.out | grep 'RunSDigitization: Execution time:'  >> testTime.out
more  testSim.out | grep 'RunDigitization: Execution time:'  >> testTime.out
echo "Finished"
echo "... see results in sim_out"
cd $CURDIR
```

How to use Shell Script for Reconstruction

```
•    #!/bin/sh
•    # $Id: reco.sh 22703 2007-12-03 10:16:38Z ivana $
•    CURDIR=`pwd`
•    OUTDIR=sim_out
•    # Minimum number of events to have enough stat. for invariant mass fit
•    # 10000 is ok, 20000 is really fine
•    NEVENTS=20
•    SEED=1234567
•    echo "Running reconstruction  ..."
•    aliroot -b >& testReco.out << EOF
•    man->SetDefaultStorage("local://$ALICE_ROOT");
•    gRandom->SetSeed($SEED);
•    AliMagFMaps* field = new AliMagFMaps("Maps","Maps", 1, 1., 10., AliMagFMaps::k5kG);
•    AliTracker::SetFieldMap(field, kFALSE);
•    AliReconstruction MuonRec("galice.root");
•    MuonRec.SetRunVertexFinder(kFALSE);
•    MuonRec.SetRunLocalReconstruction("MUON");
•    MuonRec.SetRunTracking("MUON");
•    MuonRec.SetFillESD("MUON");
•    MuonRec.SetLoadAlignData("MUON");
•    MuonRec.SetNumberOfEventsPerFile($NEVENTS);
•    MuonRec.Run();
•    .q
•    EOF
•    more  testReco.out | grep 'RunLocalReconstruction: Execution time for MUON'  >> testTime.out
•    more  testReco.out | grep 'Execution time for filling ESD ' >> testTime.out
•    echo "Finished"
•    echo "... see results in sim_out"
```

How to use Shell Script for Drawing Graphs

```
#!/bin/sh
# $Id: graphs.sh 22703 2009-12-03 10:16:38Z ivana $
CURDIR=`pwd`
OUTDIR=sim_out
#rm -fr $OUTDIR
#mkdir $OUTDIR
#cp $ALICE_ROOT/MUON/.rootrc $ALICE_ROOT/MUON/rootlogon.C $OUTDIR
cp .rootrc rootlogon.C $OUTDIR
cd $OUTDIR
# Minimum number of events to have enough stat. for invariant mass fit
# 10000 is ok, 20000 is really fine
NEVENTS=20
SEED=1234567
echo "Running Trigger efficiency  ..."
aliroot -b >& testTriggerResults.out << EOF
.L MUONTriggerEfficiency.C+
MUONTriggerEfficiency("galice.root", "galice.root",0);
.q
EOF
echo "Running efficiency  ..."
aliroot -b >& testEfficiency.out << EOF
.L MUONefficiency.C+
// no argument assumes Upsilon but MUONefficiency(443) handles Jpsi
MUONefficiency("galice.root");
.q
EOF
aliroot -b >& testResults.out << EOF
// no argument assumes Upsilon but MUONplotefficiency(443) handles Jpsi
.L MUONplotefficiency.C+
MUONplotefficiency();
.q
EOF
```

Thanks

Lab Exercise

- ssh –p 3145 alice01@
- mkdir [yourname]
- cd [yourname]
- cp –r ..src/MUON .
- Save script in with name sim.sh in directory MUON/.
- Save script in with name reco.sh in directory MUON/.
- Save script in with name graphs.sh MUON/.
- cd MUON
- vi sim.sh

Links

- http://aliceinfo.cern.ch/Offline/Activities/EventGeneratorDoc.html
- http://aliceinfo.cern.ch/Offline/Activities/Simulation/configC.html

Extra Slides

System commands

- To unzip any file
- tar -zxvf [filename].tar.gz

Invariant Analysis of generated Upsilon or J/psi

Mchview

- AliMUONPainterContourMaker
  - AddSegment R:0.0237s C:0.0200s (12302 slices)
  - AddSegments R:0.3454s C:0.3100s (28488 slices)
  - ConvertEdgePadsToContour R:0.5978s C:0.5700s (214 slices)
  - ConvertSegmentsToContour R:3.8848s C:4.0600s (3126 slices)
  - FindLocalManuContour R:0.1724s C:0.2800s (16828 slices)
  - FindMotifPosition R:0.0701s C:0.0900s (33656 slices)
  - FindPoint R:1.2782s C:1.3000s (632246 slices)
  - GenerateManuContour
    - R:6.1061s C:6.1000s (16828 slices)
    - Generation of local contour R:0.6309s C:0.6000s (214 slices)
  - InsertSegment R:3.1610s C:3.3200s (319849 slices)
  - MergeContours R:13.9378s C:13.9600s (2912 slices)
  - RemoveInsideSegments R:0.9782s C:0.9100s (2912 slices)
  - Simplify R:0.0146s C:0.0200s (3726 slices)
  - SplitSegments R:9.1520s C:9.0500s (11938 slices)
- AliMUONPainterHelper
  - GeneratePadStore R:1.9333s C:1.9400s (1 slices)
  - GetContour R:5.9725s C:5.7000s (59624 slices)
  - RegisterContour R:0.6014s C:0.7400s (19740 slices)

mchview

- **
- Welcome to mchview
- mchview Version 0.93 ($Id: AliMUONMchViewApplication.h 24336 2008-03-06 16:44:11Z laphecet $)

Cross section table

- Geometry (simulation process) gphysi-dat
- http://morsch.web.cern.ch/morsch/AliGenerator.html
- http://aliceinfo.cern.ch/static/aliroot-new/html/roothtml/ClassIndex.html
- http://aliceinfo.cern.ch/Offline/AliRoot/Releases.html