



Tercer Taller de Cómputo de la Colaboración MexNICA

Dra. Lucina Gabriela Espinoza Beltrán¹, Dra. I. Maldonado², Dra. I. Domínguez², Dra. Ma. E. Tejeda-Yeomans^{3,4}

¹ Benemerita Universidad Autónoma de Puebla, ² Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Sinaloa, ³ Facultad de Ciencias - CUICBAS, Universidad de Colima, ⁴ Departamento de Física, Universidad de Sonora

Outline

- 1 Instalación MPDroot
- 2 Simulación básica de eventos: RunMC.C
- 3 Práctica 1: Introduccion a MPDroot
- 4 Transporte en el detector
- 5 Práctica 2: transporte
- 6 Análisis de observables
- 7 Práctica 3: análisis

MPDroot-Software for MultiPurpose Detector

|

El marco de trabajo MPDroot se ha desarrollado para apoyar a las tareas:

- Desempeño del detector
- Simulación de eventos
- Desarrollo de algoritmos de reconstrucción
- Y análisis de los datos producidos en el MPD

Se basa en el entorno ROOT.

Esquema de los módulos MpdRoot en la simulación

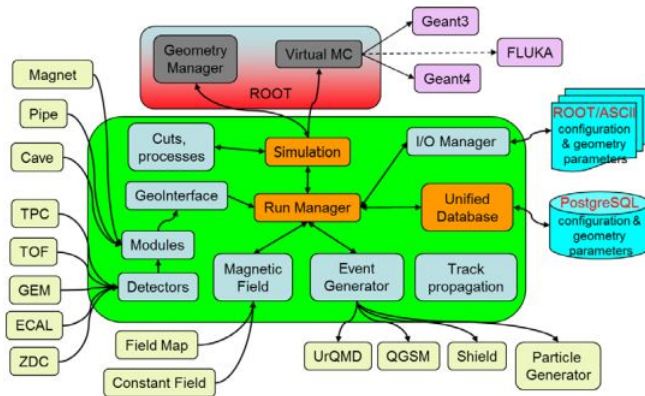
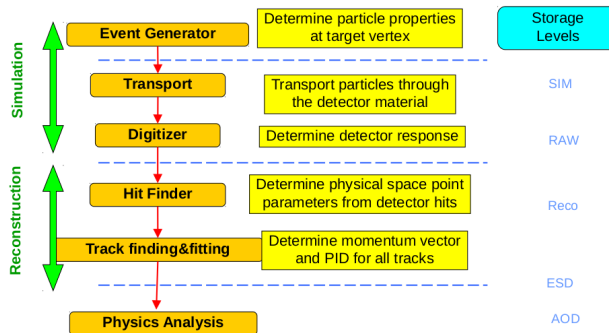


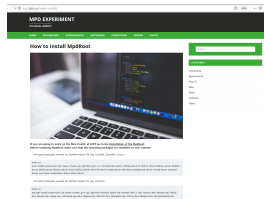
Diagrama de Flujo de MpdRoot



1

- **fairsoft:** Incluye paquetes para la generación de partículas y paquetes para el transporte de las mismas.
- **fairroot:** Software para la simulación, reconstrucción y análisis de datos especialmente de los experimentos de FAIR.
- **mpdroot:** Software que permite simular, reconstruir y analizar datos del experimento MPD de NICA.

Para ello se tienen que seguir con las instrucciones en la página del MPD:



<http://mpd.jinr.ru/howto-install/>

Sugerencia: Instala en un directorio local y define la ruta al directorio export `INSTALLATION_PATH=~ /NICA/MPDroot/` y no olvides usar esta ruta en el directorio `SetEnv.sh`

Generadores de eventos

Los generadores de eventos son paquetes de software que simulan la producción de partículas en colisiones de altas energías, conocidos como "generadores Monte Carlo". Los generadores Monte Carlo soportados por MPDRoot son:

- **Ultrarelativistic Quantum Molecular Dynamics (UrQMD)**
- **Quark Gluon String Model (QGSM, LAQGSM)**
- Shield
- **Parton Hadron String Dynamics (PHSD, HSD)**
- Pluto
- Hybrid UrQMD
- **EPOS**
- 3 Fluid Dynamics (for baryon stopping)

Simulación básica de eventos: RunMC.C

Para llevar a cabo la simulación, se requiere usar y modificar el archivo runMC.C
Sugerencia: crear un directorio y copiarlo

```
mkdir ~/MPDroot/Taller3  
cd ~/MPDroot/Taller3  
cp ~/NICA/MPDroot/mpdroot/macro/mpd/runMC.C .
```

```
void runMC(TString inFile = "", TString outFile = "",
Int_t nStartEvent = 0, Int_t nEvents = 10,
Bool_t flag_store_FairRadLenPoint = kFALSE, Int_t FieldSwitcher = 0)
```

- Inicializar el gestor de simulación
`FairRunSim* fRun = new FairRunSim();`
- Definir la geometría
`geometry_stage1(fRun);`
- Gestor de Generador primario
`FairPrimaryGenerator* primGen = new FairPrimaryGenerator();`
`fRun→SetGenerator(primGen);`
- Smear of the Beam
`primGen→SetBeam(0.0,0.0,0.0,0.0);`
- Psi Mc event plane angle
`urqmdGen→SetEventPlane(min * TMath::DegToRad(), max * TMath::DegToRad());`

```
GNU nano 4.8                                runMC.C
#endif

#_ADD_INCLUDE_PATH($WORKDIR)
#include "mpdloadlibs.C"
#include "geometry_stage1.C"
// #include "macro/mpd/geometry_v2.C"

#define BOX // Choose generator: URQMD VHLLE FLUID PART ION BOX HSD LAQSM HADGEN
#define GEANT3 // Choose: GEANT3 GEANT4

// inFile - input file with generator data, default: auau.09gev.nbias.98k.ftn14
// nStartEvent - for compatibility, any number
// nEvents - number of events to transport, default: 1
// outFile - output file with MC data, default: evetest.root
// flag_store_FairRadLenPoint
// FieldSwitcher: 0 - corresponds to the ConstantField (0, 0, 5) kG (It is used by default); 1 - corresponds to the FieldMap ($WORKDIR/Input/8-FieldMap)
void runMC(TString inFile = "test.f14", TString outFile = "evetest.root", Int_t nStartEvent = 0, Int_t nEvents = 10,
          Bool_t flag_store_FairRadLenPoint = kFALSE, Int_t FieldSwitcher = 0)
{
    TStopwatch tmer;
    tmer.Start();
    gDebug = 0;

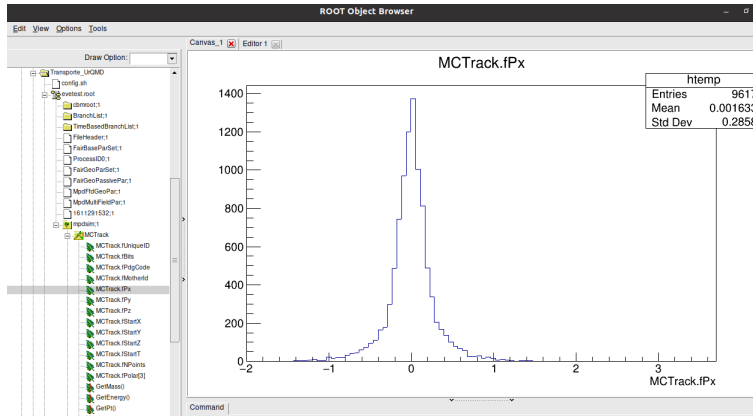
    FairRunSin* fRun = new FairRunSin();
    // Choose the Geant Navigation System
    #ifdef GEANT3
        fRun->SetName("TGeant3");
    #else
        fRun->SetName("TGeant4");
    #endif

    geometry_stage1(fRun); // load mpd geometry
    // geometry_v2(fRun); // load mpd geometry
}
```

realizamos la simulación y leemos el archivo de salida:

```
root runMC.C test.f14 evetest.root 0 10
```

```
root evetest.root
```



14/22