



Segundo Taller de Cómputo de la Colaboración MexNICA

Dra. I. Maldonado¹, Dra. I. Domínguez¹, Dra. Ma. E. Tejeda-Yeomans^{2,3}

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

July 1, 2020

Outline

Instalación MpdRoot

Simulación de eventos RunMC.C

UrQMD

PHSD

Propagación en el detector

Práctica

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

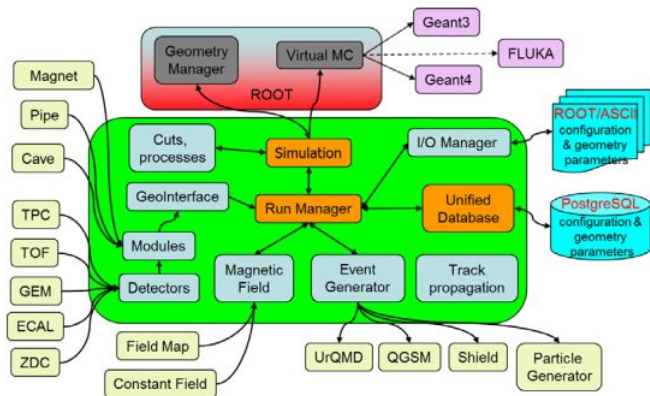
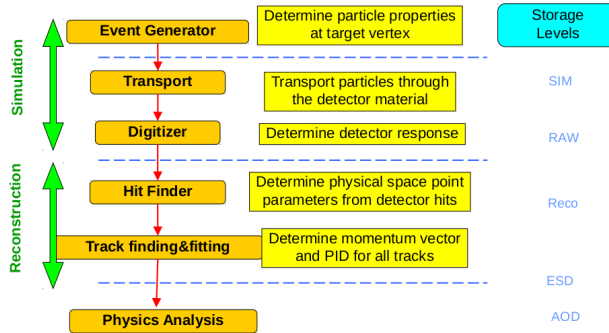


Diagrama de Flujo de MpdRoot

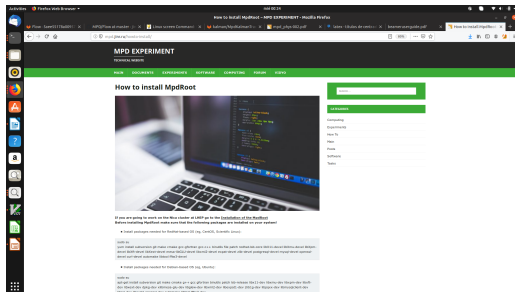


Instalación

Se instalarán los siguientes paquetes:

- ▶ **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=~ /Software/NICA/`
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

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

```
mkdir ~/Software/Taller2  
cd ~/Software/Taller2  
cp ~/Software/NICA/mpdroot/macro/mpd/runMC.C .
```


- ▶ Inicializar el gestor de simulación

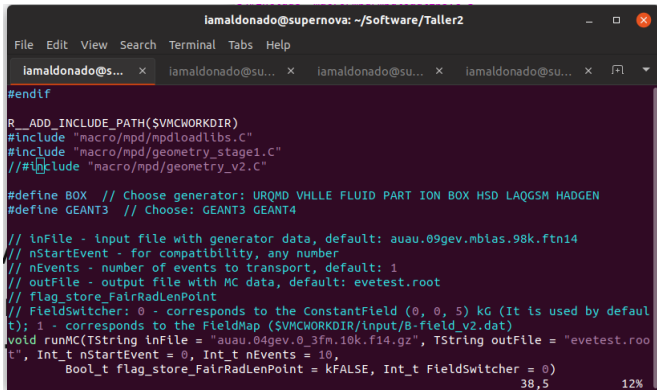
```
FairRunSim* fRun = new FairRunSim();
```

- ▶ Definir la geometría

```
geometry_stage1(fRun);
```

- ▶ Gestor de Generador primario

```
FairPrimaryGenerator* primGen = new FairPrimaryGenerator();  
fRun->SetGenerator(primGen);
```

A terminal window titled 'iamaldonado@supernova: ~/Software/Taller2' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help) and several tabs. The code is a C++ header file for a simulation. It includes macros for adding include paths and defining generators. It also defines constants for the number of events and the output file name. The code is as follows:

```
#endif

R__ADD_INCLUDE_PATH($VMCWORKDIR)
#include "macro/mpd/mpdloadlibs.C"
#include "macro/mpd/geometry_stage1.C"
// #include "macro/mpd/geometry_v2.C"

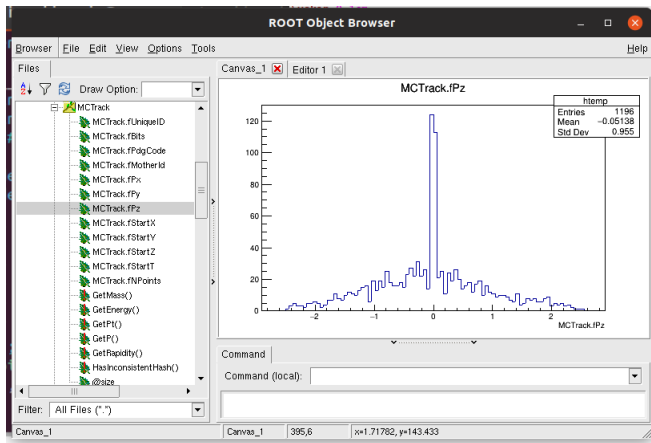
#define BOX // Choose generator: URQMD VHLLE FLUID PART ION BOX HSD LAQGSM HADGEN
#define GEANT3 // Choose: GEANT3 GEANT4

// inFile - input file with generator data, default: auau.09gev.mbias.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 ($VMCWORKDIR/input/B-field_v2.dat)
void runMC(TString inFile = "auau.04gev.0_3fm.10k.f14.gz", TString outFile = "evetest.root", Int_t nStartEvent = 0, Int_t nEvents = 10, Bool_t flag_store_FairRadLenPoint = kFALSE, Int_t FieldSwitcher = 0)
    38,5 12%
```

realizamos la simulación y leemos el archivo de salida:

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

```
root evetest.root
```



Ultrarelativistic Quantum Molecular Dynamics (UrQMD)

- ▶ git mexnica: <https://github.com/MexNICA/urqmd>
passwd: 2020mexnica
- ▶ git clone <https://github.com/MexNICA/urqmd.git> [o download zip file](#)

```
lamaldonado@supernova:~/Software/Taller2$ git clone https://github.com/MexNICA/urqmd.git
Cloning into 'urqmd'...
Username for 'https://github.com': MexNICA
Password for 'https://MexNICA@github.com':
```

- ▶ Accedemos al directorio y compilamos:

```
cd urqmd
make
```

UrQMD - inputfile

Editamos el archivo que define los parámetros de la simulación

inputfile:

```
pro 197 79 // Au
tar 197 79 // Au
nev 10 // # eventos
imp -15.0 // parámetro de impacto (MB)
ecm 11 // Energía CM
tim 200 200 // Tiempo de la colisión
eos 0 // Ecuación de estado
#f13 // Archivos de salida
#f14
#f15
cto 41 1 // Modelo hidrodinámico
```

Y corremos la simulación con:

```
./runqmd.bash
```

UrQMD to ROOT

- ▶ git mexnica: <https://github.com/MexNICA/urqmd-doc>
passwd: 2020mexnica
- ▶ git clone <https://github.com/MexNICA/urqmd-doc> o [download zip](#)
- ▶ Accedemos al directorio y corremos el archivo

```
cd urqmd-doc  
root urqmdtoroot.C
```
- ▶ Genera el archivo de salida con la información de las partículas en un árbol de root.

```
root LeerEntradas.C
```
- ▶ Generar histogramas

Parton-Hadron-String Dynamics (PHSD)

- ▶ Intel Fortran Compiler (ifort): <https://software.intel.com/en-us/fortran-compilers/>
- ▶ /intel/bin
- ▶ source compilervars.sh -arch intel64 -platform linux
- ▶ git mexnica: <https://github.com/MexNICA/phsd>
passwd: 2020mexnica
- ▶ git clone <https://github.com/MexNICA/phsd.git> o [download zip](#)

cd phsd
make
- ▶ Modificar inputfile
- ▶ Corremos el macro:

./phsd

PHSD to ROOT

- ▶ `git clone https://github.com/MexNICA/phsd-doc.git` [o download zip](#)
- ▶ `$ cd phsd-doc`
- ▶ copiar el archivo generado con `phsd:phsd.dat`
- ▶ `$ root ConvertToRoot.C`
- ▶ `$ root`
- ▶ `.L Sim3.C`
- ▶ Generar histogramas

UrQMD en el MPD

- ▶ Crear un directorio de trabajo y copiar los archivos de mpdroot al directorio
 - `cp /Software/NICA/mpdroot/macro/mpd/runMC.C work`
 - `cp /Software/NICA/mpdroot/macro/mpd/mpdloadlibs.C work`
 - `cp /Software/NICA/mpdroot/macro/mpd/geometry_stage1.C work`
- ▶ Copiar el archivo generado con UrQMD
 - `cp /Software/UrQMD/urqmd/test.f14 work`

runMC.C con eventos de UrQMD

Editar runMC.C:

- ▶ Incluir macros locales:

```
#include "mpdloadlibs.C"
```

```
#include "geometry_stage1.C"
```

- ▶ Definir el MC

```
#define URQMD
```

- ▶ Definir archivo de entrada, de salida y número de eventos

```
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)
```

- ▶ `$root runMC.C`

Manos a la obra

Edita los macros y ponlos en práctica