

User's guide for MINOS simulations

V1 Oct 2013 L.Audirac, CEA Saclay ; V2 Sept 2015 A.Corsi, CEA Saclay

The main file

MINOS_simulation: MINOS_sim.cc and executable MINOS_sim

The main files allow constructing the different classes and the visualization window.

The different folders

The folder src/ contains the source files of the classes.

The folder include/ contains the header files of the classes.

The folder result/ contains the result files of the simulation.

The folder Inputs/ contains the files used to define some characteristics of the setup and read by the libraries.

The folder lib/ contains some libraries used by the simulation.

The Inputs files

ConfigFileBeam.txt: definition of the characteristics of the beam which is sent in the setup in terms of energy, momentum, position, charge and mass number.

ConfigFileSetup.txt: definition of the dimensions of the elements constituting the setup

ConfigFileRings.txt: definition of all the parameters used in the second step of the simulations for the drift of ionization electrons

The libraries

For each library one has the header file .hh, the source file .cxx and the GNUmakefile.

The library used to define the beam

libExN03BeamIn.so: this library allows to determine the characteristics of the beam by reading the ConfigFileBeam.txt file.

ExN03BeamIn.cxx: source file defining the function ReadConfigFile() which allows to read ConfigFileBeam.txt

ExN03BeamIn.hh: header file defining the beam parameters

MeanX	central position of the beam in X (mm)
SigmaX	standard deviation of the position in X (mm)
MeanY	central position of the beam in Y (mm)
SigmaY	standard deviation of the position in Y (mm)
Z0	Z position of emission of the beam (mm)
MeanMomentumX	mean momentum along X
SigmaMomentumX	standard deviation in momentum along X
MeanMomentumY	mean momentum along Y
SigmaMomentumY	standard deviation in momentum along Y
MomentumZ0	momentum along Z
MeanEnergy	mean total kinetic energy (MeV)
SigmaEnergy	standard deviation in energy (MeV)
Z	charge
A	mass

The library used to define the setup

libExN03Setup.so: this library allows to determine the characteristics of the setup by reading the ConfigFileSetup.txt file.

ExN03Setup.cxx: source file defining the function ReadConfigFile() which allows to read ConfiFileSetup.txt

ExN03Setup.hh: header file defining the setup parameters

TargetRadius	target radius in mm
TargetLength	target length in mm
ChamberInnerRadius	chamber inner radius in mm
ChamberThickness	chamber thickness in mm
ChamberLength	chamber length in mm
InnerRohacellThickness	inner Rohacell thickness in mm
KaptonThickness	Kapton thickness in mm

OuterRohacellThickness	outer Rohacell radius in mm
TPCRadiusExt	TPC outer radius in mm
WindowThickness	Mylar window thickness in mm

The output library

libExN03datai.so: this library allows to define observables which will be saved by the simulation; for example energy loss as a function of the position in the different materials, total energy loss in the different materials, vertex information in terms of energy, momentum, position, charge and mass of particles created, detection in the TPC.

ExN03Datai.cxx: source file defining the function ClearEvent() which allows to clear all the observables at the beginning of a new event simulated

ExN03DataoRings.hh: header file defining the observables

Z, A	charge and mass of particles
energy0	kinetic energy of each particle at vertex (MeV)
theta0, phi0	θ and ϕ angle of each particle at vertex (deg)
x0, y0, z0	vertex coordinates (mm) of each particle
detection	Boolean of energy deposited by each particle in the TPC
Et_tar	energy in eV deposited by each particle in the target
Et_win	energy in eV deposited by each particle in the Mylar window
Et_ch	energy in eV deposited by each particle in the chamber
Et_InnerRohacell	energy in eV deposited by each particle in the inner Rohacell
Et_Kapton	energy in eV deposited by each particle in the Kapton
Et_OuterRohacell	energy in eV deposited by each particle in the outer Rohacell
Et_tpc	energy in eV deposited by each particle in the TPC
Et_tpc_tot	total energy in eV deposited by all particles in the TPC
x_tar, y_tar, z_tar, e_tar	position (mm) and energy (eV) deposited step by step in the target
x_win, y_win, z_win, e_win	position (mm) and energy (eV) deposited step by step in the Mylar window
x_ch, y_ch, z_ch, e_ch	position (mm) and energy (eV) deposited step by step in the

	chamber
x_InRoh, y_InRoh, z_InRoh, e_InRoh	position (mm) and energy (eV) deposited step by step in the inner Rohacell
x_Kap, y_Kap, z_Kap, e_Kap	position (mm) and energy (eV) deposited step by step in the Kapton
x_OutRoh, y_OutRoh, z_OutRoh, e_OutRoh	position (mm) and energy (eV) deposited step by step in the outer Rohacell
x_tpc, y_tpc, z_tpc, e_tpc	position (mm) and energy (eV) deposited step by step in the TPC
MomentumXAfterTPC	momentum of particle in the X direction
MomentumYAfterTPC	momentum of particle in the Y direction
MomentumZAfterTPC	momentum of particle in the Z direction

The drift library

libExN03DataoRings.so: this library is used for the second step of the simulations to achieve the drift of ionization electrons. It defines a concentric matrix of several rings divided in several pads. Three main aspects:

- Allows to read the output file of the simulation
- Allows to define parameters used for the drift of electrons by reading the ConfigFileRings.txt file
- Allows to define observables saved by the program of drift, especially number of electrons deposited on the different pads of the Micromegas as a function of the position of creation

ExN03DataoRings.cxx: source file defining the two following functions:

- The function ClearEvent() allows to clear all the observables at the beginning of a new event treated
- The function ReadConfigurationFile() allows to read the parameters given in ConfigFileRings.txt

ExN03DataoRings.hh: header file defining the observables/parameters/functions

dataai	class ExN03Dataai
x_pad	X coordinate of the pads (mm)

y_pad	Y coordinate of the pads (mm)
t_pad	time coordinate of the pads (ns)
q_pad	charge deposited on the pads (number of electrons)
dt	difference between time fitted and real time of pads
Et_pad	total charge deposited on the detection matrix
nb_pads	number of pads triggered (number of electrons)
NRings	number of rings of the detection matrix
NSegments	number of pads per ring
sigL	longitudinal diffusion in the gas (mm/v(cm))
sigT	transverse diffusion in the gas (mm/v(cm))
driftV	drift speed of electrons in the gas (mm/ns)
Ionis	mean energy of creation of an electron-ion pair (eV)
TimeBinSize	sampling time of the electronics (ns)
ShapingTime	shaping time of the electronics (ns)
Threshold	detection threshold of the electrons
Gain	mean gain of the Micromegas in the Polya distribution
Theta	θ parameter in the Polya distribution
NoiseRMS	σ_{rms} of the noise (number of electrons)
FirstRing	distance from the center of the first ring
LastRing	distance from the center of the last ring

Description of the classes:

The class ExN03DetectorConstruction

This class allows to define the experimental setup, more precisely by the following way.

The constructor allows to define:

- the dimensions of the materials by calling the class ExN03Setup
- several materials by the function DefineMaterials()
- the material constituting each element of the setup by the functions SetXMaterial("Y") where X will be filled by the material Y

The function Construct() allows to build the setup by first calling the ComputeMinosParameters() function defined in the corresponding header file, only used to define the size of the "world volume" from the size of the TPC. Take care of giving dimensions of the world big enough to contain all the elements of the detector. Elements defined from inner to outer are:

- the target (LH2)
- the Mylar window (entrance window, exit window and tube)
- the chamber (Inox)
- a first Rohacell, Kapton and a second Rohacell for the field cage
- the TPC ("mix" gas for a T2K mix)

Moreover, in the function Construct(), we define the region TPCLog used by ExN03PhysicsList and we define the magnetic field too.

The class ExN03PhysicsList

This class allows to define the standard electromagnetic physics list. In the constructor we define the default cut value for tracking. The function SetCuts() allows to assign this cut value for electrons, positrons and gammas and a minimum energy value for the creation of particles. Moreover, a more accurate value for tracking can be added for the region TPCLog corresponding to the TPC.

The same parameters can be defined and modified in the QGSP_INCL_ABLA physics list in the file /source/physics_lists/lists/include/QGSP_INCL_ABLA.cc

The class ExN03EventAction

This class defines some functions at the beginning (BeginOfEventAction which calls ExN03ROOTuple::RecordBeginOfEvent) and at the end (EndOfEventAction which calls ExN03ROOTuple::RecordEndOfEvent) of each event generated.

The class ExN03RunAction

This class defines functions called at the beginning (BeginOfRunAction which calls ExN03ROOTuple::RecordBeginOfRun) and the end (EndOfRunAction which calls ExN03ROOTuple::RecordEndOfRun) of the simulation.

The class ExN03TrackingAction

This class defines functions called at the beginning (PreUserTrackingAction which calls ExN03ROOTuple::RecordBeginOfTrack) and the end (PostUserTrackingAction which calls ExN03ROOTuple::RecordEndOfTrack) of tracking a particle.

The class ExN03SteppingAction

This class defines the function UserSteppingAction which allows to call the functions ExN03ROOTuple::RecordStepDExxx at each step of the tracking of particles where xxx is the material encountered.

The class ExN03ROOTuple

This class defines user's functions called by the previous classes. The main functions are as follows:

- RecordBeginOfRun: used to define the output file and the output tree containing the observables defined in libExN03Data.so
- RecordEndOfRun: used to write tree and close output file
- RecordBeginOfEvent: clears the observables at each beginning of event
- RecordEndOfEvent: used to fill the output tree at the end of each event
- RecordBeginOfTrack: used to initialize some observables calculated for each tracked particle like the energies loss in materials, the detection in the TPC.
- RecordEndOfTrack: used to save the characteristics of tracked particles (initial conditions of emission, charge, mass, energy loss in materials, detection in the TPC)
- RecordStepDExxx: used to save at each step of tracking the energy loss and the position of particles. For some materials, only the total energy loss in materials is incremented.

The class ExN03PrimaryGeneratorAction

- The function SetDefaultPrimaryParticle defines the default parameters of the particle which is to be sent in the setup in terms of position, momentum, charge, mass, energy. If an input file is needed, it is opened here.
- The function GeneratePrimaries generates randomly the particle(s) sent in the setup.

The class MyMagneticField

This class allows by the constructor to define the magnetic field.

To run a simulation

- 1) Change the paths inside the codes. Compile the libraries in ./lib using the corresponding GNUmakefileXXX

```
cd MINOS_simulation/lib; make -f GNUmakefileXXX
```

Add MINOS_simulation/lib to the LD_LIBRARY_PATH

- 2) Change the paths inside the codes. Generate the Makefile for the main program using the CMakeList.txt file. Simulation is tested with Geant4.10.00.

```
cd MINOS_simulation; cmake .
```

- 3) Run the simulation; root output file is written in ./results

The simulations need the file vis.mac to define the visualization window (can be modified to visualize particles; see inside the file) or run_nothing.mac if only root file is needed as output.

Drift of ionization electrons

The second step of the simulations needed to achieve the drift of electrons and obtain the signal on pads is situated in the folder Drift/. The source file is DriftRings.cxx compiled with GNUmakefileRings to generate the executable DriftRings.

This program simulates a concentric matrix of rings divided in several segments. The characteristics of the gas, the amplification stage, the AGET electronics and the noise are taken into account. The program uses the library libExN03DataorRings.so. Parameters for the drift simulation are defined in ConfigFileRings.txt. Results of drift are in the folder Drift/result/.

More explanations are described in the source file itself of the folder MINOS_beam.

Calculation of the resolution

The program used to calculate the resolution is in Analysis/. Resolution is calculated along the beam axis. The histogram of the difference between reconstructed vertex and real vertex is in the folder result/ and can be fitted by a Gaussian or Lorentz function. Tracks of protons are fitted with parametric lines and with a helicoid (preliminary, without energy loss taken into account along the track) if a magnetic field is applied. Note that this simulation uses the library which defines the beam, so the beam used for the Geant4 simulation has to be considered.

The source file is tpcFit.cxx, compiled with GNUMakefileResol, which generates the executable tpcFit.

More explanations can be found in the source files of MINOS_beam and MINOS_Bfield.