Managing FLUKA Simulation Output Files for FOOT

Exercises

G.B. Y.D. S.M.

Possible Basic Exercises using SHOE

1. Make a plot of the multiplicity per event of tracks produced anywhere in the detector
2. Make a plot of the multiplicity per event of tracks produced by the primary in the target
3. Make the previous plot only for those particle which exit the target going in the forward region and are produced with E>50 MeV/u
4. Make a plot of the energy distribution of fragments produced in target for a few different Z and/or A
5. Make a plot of the energy released per event in the TW
6. Make a plot of the energy released per event in the CA and for a selected crystal of your choice

Slightly Increasing Difficulty:

7. Compare the distribution of energy released by p and ^{4}He in the 1st layer of MSD
8. Select particles produced in the target which arrive at TW and make a plot of the energy that they have lost in the path from target to TW

Processing and macro template

In /shoe/build/Reconstruction/level/0

../../bin/DecodeMC –in 12C_C_200.root –out 12C_C_200decMC.root –exp CNAO2020 –run 1

Basic macros:

ReadShoeMC.C ReadShoeMC.h ReadShoeMCFunc.C

main program

Functions called by the main

Available functions

Booking()	to book Histos (divided in directories)
MC()	to analyse tracks
StartCounter()	to analyse SC hits
BeamMonitor()	to analyse BM
Vertex()	to analyse VT
Msd()	to analyse MSD hits
TofWall()	to analyse TW hits
Calo()	to analyse CA hits

Usage of macro

In /shoe/build/Reconstruction/level/0

File processed by DecodeMC

root –b ‘ReadShoeMC.C+(“CNAO2020”,1,”12C_C_200decMC.root”,0)’

Campaign name

Run

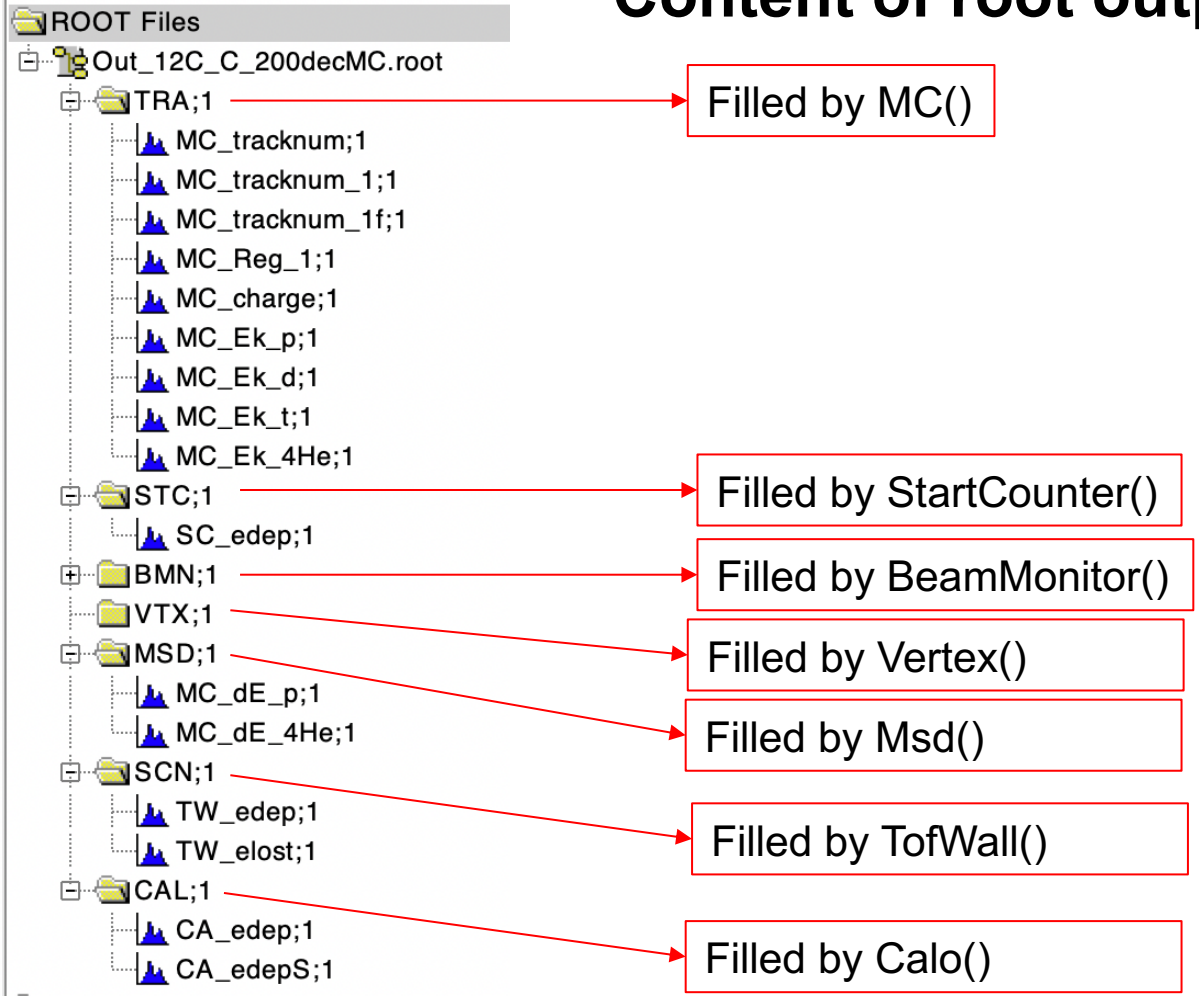
0 to process all events

Produces:

Out_12C_C_200decMC.root

root file with histos

Content of root output



Solution of exercise no. 1 a)

Book the histogram in Booking():

```
int Booking(TFile* file_out, float timecut){

  file_out->cd();
  TH1D *h;
  TH2D *h2;
  char tmp_char[200];

  if(IncludeMC){
    file_out->mkdir("TRA");
    file_out->cd("TRA");
    h = new TH1D("MC_tracknum","Multiplicity of MC particles/event;No. of particles;Events",31,-0.5,30.5);

    gDirectory->cd("..");
    file_out->cd("..");
```

Solution of exercise no. 1 b)

In your main:

```
TTree *tree = 0;
tree = (TTree*)f->Get("tree");

TAMCntuEve *mcNtuEve; // MC tree
mcNtuEve = new TAMCntuEve();  // Get MC Tree
tree->SetBranchAddress(TAMCntuEve::GetBranchName(), &mcNtuEve);
nentries = tree->GetEntries();

for (evnum = 0; evnum < nentries; ++evnum) { // Loop on the entries

  tree->GetEntry(evnum);
  status=MC(mcNtuEve);   // Analyse Track Structure
```

Solution of exercise no. 1 c)

Inside MC():

```
//MC analysis of Tracks
int MC(TAMCntuEve* mcNtuEve){

  int  Nmctrack = mcNtuEve->GetTracksN(); // Get No. of particles (TRn)

  myfill("TRA/MC_tracknum", Nmctrack); // Fills histo with no. of particles per event

  return 0;
}
```

Then you can fill your histogram with Nmctrack

MC is of course declared in ReadShoeMC.h as:

```
//MC Track Analysis
int MC(TAMCntuEve* mcNtuEve);
```

Solution of exercise no. 2

Inside MC():

Check in your case

```
//MC analysis of Tracks
int MC(TAMCntuEve* mcNtuEve){

  Int_t  Nmctrack = mcNtuEve->GetTracksN(); // Get No. of particles (TRn)

  Int_t track1 = 0;

  for( Int_t iTrack = 0; iTrack < mcNtuEve->GetTracksN(); ++iTrack ) { // Loop on particles
    TAMCeveTrack* track = mcNtuEve->GetTrack(iTrack); // Gets the track (particle)
    Int_t Mid = track->GetMotherID();               // Get TRpaid-1
    Int_t Reg = track->GetRegion();                 // Get TRreg

    if ( Mid == 0 && Reg==50) { // Selects particles created by primari in target (50 is for CNAO2020 Camp.)
      track1++;
    }
  }

  myfill("TRA/MC_tracknum_1", track1); // Fills histo with no. of particles per event produced by primary in target

  return 0;
}
```

Then, after the loop on the no. of track, you can fill your histogram with track1

In principle, considering the rule chosen for writing the event on file, there should be no real reason to make a selection on region number…

Solution of exercise no. 3

Inside MC():

```
//MC analysis of Tracks
int MC(TAMCntuEve* mcNtuEve){

  int  Nmctrack = mcNtuEve->GetTracksN(); // Get No. of particles (TRn)

  Int_t trackfw = 0;

  for( Int_t iTrack = 0; iTrack < mcNtuEve->GetTracksN(); ++iTrack ) { // Loop on particles
    TAMCeveTrack* track = mcNtuEve->GetTrack(iTrack); // Gets the track (particle)
    Int_t Anumb = track->GetBaryon();               // Get A (TRbar)
    Int_t Mid = track->GetMotherID();               // Get TRpaid-1
    Int_t Reg = track->GetRegion();                 // Get TRreg
    Double_t Mass = track->GetMass();               // Get TRmass
    TVector3 FinalPos = track->GetFinalPos();       // Get TRfx, TRfy, TRfz
    TVector3 InitP = track->GetInitP();             // Get TRipx, TRipy, TRipz

    // build Kinetic Energy per nucleon (GeV/u)
    Double_t Ekin = (pow(pow(InitP(0),2) + pow(InitP(1),2) + pow(InitP(2),2) + pow(Mass,2),0.5) - Mass)/(double)Anumb;

    if ( Mid == 0 && Reg==50) { // Selects particles created by primari in target (50 is for CNAO2020 Camp.)
      if(FinalPos(2) > 0.3 && InitP(2)>0. && Ekin>0.05) trackfw++; // Selects fast (>50 MeV/u) forward particles
    }
  }
```

Then, after the loop on the no. of track, you can fill your histogram with trackfw

Same considerations as before about the selection on region number

Solution of exercise no. 4 a)

Inside Booking():

```
if(IncludeMC){
  file_out->mkdir("TRA");
  file_out->cd("TRA");

  h = new TH1D("MC_Ek_p","Energy/nucleon of p;E [MeV/u];Events",200,0.,400.);
  h = new TH1D("MC_Ek_d","Energy/nucleon of d;E [MeV/u];Events",200,0.,400.);
  h = new TH1D("MC_Ek_t","Energy/nucleon of t;E [MeV/u];Events",200,0.,400.);
  h = new TH1D("MC_Ek_4He","Energy/nucleon of 4He;E [MeV/u];Events",200,0.,400.);
```

Solution of exercise no. 4 b)

Inside MC():

```
//MC analysis of Tracks
int MC(TAMCntuEve* mcNtuEve){

  int  Nmctrack = mcNtuEve->GetTracksN(); // Get No. of particles (TRn)

  for( Int_t iTrack = 0; iTrack < mcNtuEve->GetTracksN(); ++iTrack ) { // Loop on particles
    TAMCeveTrack* track = mcNtuEve->GetTrack(iTrack); // Gets the track (particle)
    Int_t Charge = track->GetCharge();              // Get Charge (TRcha)
    Int_t Anumb = track->GetBaryon();               // Get A (TRbar)
    Int_t Fid = track->GetFlukaID();                // Get TRfid
    Int_t Mid = track->GetMotherID();               // Get TRpaid-1
    Int_t Reg = track->GetRegion();                 // Get TRreg
    Double_t Mass = track->GetMass();               // Get TRmass
    TVector3 InitP = track->GetInitP();             // Get TRipx, TRipy, TRipz

    // build Kinetic Energy per nucleon (GeV/u)
    Double_t Ekin = (pow(pow(InitP(0),2) + pow(InitP(1),2) + pow(InitP(2),2) + pow(Mass,2),0.5) - Mass)/(double)Anumb;

    if ( Mid == 0 && Reg==50) { // Selects particles created by primari in target (50 is for CNAO2020 Camp.)
      if (Charge==1 && Anumb==1) {
        myfill("TRA/MC_Ek_p", Ekin*1000.); //Fills Selected energy distributions
      } else if (Charge==1 && Anumb==2) {
        myfill("TRA/MC_Ek_d", Ekin*1000.); //Fills Selected energy distributions
      } else if (Charge==1 && Anumb==3) {
        myfill("TRA/MC_Ek_t", Ekin*1000.); //Fills Selected energy distributions
      } else if (Charge==2 && Anumb==4) {
        myfill("TRA/MC_Ek_4He", Ekin*1000.); //Fills Selected energy distributions
      }
    }
  }
```

Solution of exercise no. 4 c)

myfill is a simple class to fill histos in a given directory of the output root tree

```
//some method to fill histos
template <class t>
void myfill(const char *graphname, t x){
  if(gDirectory->Get(graphname)!=nullptr){
    ((TH1D*)(gDirectory->Get(graphname)))->Fill(x);
  }else{
    cout<<"ERROR!!!: "<<graphname<<" cannot be filled because it does not exist in gROOT, check Booking!"<<endl;
  }
return;
}
```

Solution of exercise no. 5

In Booking():

```
    file_out->mkdir("SCN");
    file_out->cd("SCN");

    h = new TH1D("TW_edep","Energy deposition in the TW;E [MeV]; Events",200,0.,200.);
```

In TofWall():

```
//MC TW analysis
int TofWall(TAMCntuHit *twMChits, TAMCntuEve *mcNtuEve){

  Int_t ntwMCHits  = twMChits->GetHitsN();

  Double_t edepTW = 0.;

  for (int i=0; i<ntwMCHits; i++) { // Loop on hits
    TAMChit* twhit=twMChits->GetHit(i); // Gets the hit
    edepTW += twhit->GetDeltaE(); // builds energy deposition in SC (here is already in MeV!)
  }

  myfill("SCN/TW_edep",edepTW);
```

Solution of exercise no. 6

In Booking():

```
file_out->mkdir("CAL");
file_out->cd("CAL");

h = new TH1D("CA_edep","Energy deposition in Calo;E [MeV]; Events",240,0.,2400.);
h = new TH1D("CA_edepS","Energy deposition in one Crystal;E [MeV]; Events",240,0.,2400.);
```

In Calo():

```
//MC CA analysis
int Calo(TAMCntuHit *caMChits, TAMCntuEve *mcNtuEve){

  Int_t ncaMCHits  = caMChits->GetHitsN();
  Double_t edepCA = 0.;
  Double_t edepCAs = 0.;

  for (int i=0; i<ncaMCHits; i++) { // Loop on hits
    TAMCHit* cahit=caMChits->GetHit(i); // Gets the hit
    edepCA += cahit->GetDeltaE(); // builds energy deposition in Calo
    if(cahit->GetCrystalId()==0) edepCAs += cahit->GetDeltaE(); // builds energy deposition in Calo in Crystal no. 0
  }

  if (edepCA>0.) myfill("CAL/CA_edep",edepCA*1000.);
  if (edepCAs>0.) myfill("CAL/CA_edepS",edepCAs*1000.);
```

Solution of exercise no. 7 a)

In Booking():

```
file_out->mkdir("MSD");
file_out->cd("MSD");

h = new TH1D("MC_dE_p","Enegy released by p;E [MeV/u];Events",200,0.,2.);
h = new TH1D("MC_dE_4He","Enegy released by 4He;E [MeV/u];Events",200,0.,2.);
```

Solution of exercise no. 7 b)

In Msd():

```
//MC MSD analysis
int Msd(TAMCntuHit *msMChits, TAMCntuEve *mcNtuEve){

  Int_t nmsMCHits  = msMChits->GetHitsN();
  Double_t edepMSDp = 0.;
  Double_t edepMSD4He = 0.;

  for (int i=0; i<nmsMCHits; i++) { // Loop on hits
    TAMChit* mshit=msMChits->GetHit(i); // Gets the hit
    TAMCeveTrack* mctrack=mcNtuEve->GetTrack(mshit->GetTrackIdx()-1); //retrievs TrackID
    if (mctrack->GetFlukaID()==1 && mshit->GetLayer()==0) { // selects protons generating the hit
      edepMSDp += mshit->GetDeltaE(); // builds energy deposition by p in MSD
    } else if (mctrack->GetFlukaID()==-6 && mshit->GetLayer()==0) { // selects 4He generating the hit
      edepMSD4He += mshit->GetDeltaE(); // builds energy deposition byu 4He in MSD
    }
  }

  if(edepMSDp>0.)  myfill("MSD/MC_dE_p",edepMSDp*1000.);
  if(edepMSD4He>0.)  myfill("MSD/MC_dE_4He",edepMSD4He*1000.);
```

Solution of exercise 8

In Booking() add in SCN directory:

```
h = new TH1D("TW_elost","Energy loss from target to TW;E [MeV]; Events",200,0.,20.);
```

In TofWall():

```
//MC TW analysis
int TofWall(TAMCntuHit *twMChits, TAMCntuEve *mcNtuEve){

  Int_t ntwMCHits  = twMChits->GetHitsN();

  for (int i=0; i<ntwMCHits; i++) { // Loop on hits
    TAMChit* twhit=twMChits->GetHit(i); // Gets the hit

    if(twhit->GetLayer()==1) { // selects hits on the first layer
      TAMCeveTrack* mctrack=mcNtuEve->GetTrack(twhit->GetTrackIdx()-1); //retrievs TrackID
      if (mctrack->GetRegion()==50) { // selects if track was produced in target (50 is for CNAO2020)
        TVector3 Mom_i = mctrack->GetInitP(); // retrieves starting P of track
        Double32_t Mass = mctrack->GetMass(); // retrieves Mass of track
        Double_t Ekin_i = pow(pow(Mom_i(0),2) + pow(Mom_i(1),2) + pow(Mom_i(2),2) + pow(Mass,2),0.5) - Mass; // Kinetic energy at origin
        TVector3 Mom_f = twhit->GetInMomentum(); // gets the momentum at TW arrival
        Double_t Ekin_f = pow(pow(Mom_f(0),2) + pow(Mom_f(1),2) + pow(Mom_f(2),2) + pow(Mass,2),0.5) - Mass; // Kinetic energy at TW
        Double_t Elost = Ekin_i - Ekin_f; // Energy lost along the whole path from production to TW
        myfill("SCN/TW_elost",Elost*1000.);
      }
    }
  }
```

All the material for this hands-on session is available on:

https://drive.google.com/drive/folders/1-KKcNBQyWIc-QMA7Titk_aZOY0LwxXzd