# Geometry HandOn

# Exercice

1. Run reconstruction with DecodeMC executable for 12C_200 experiment with ST+BM+VT on over 1kEvents
2. Retrieve first BM track and extrapole to the target for the first events
   - Write ExtrapoleBMtoTG.C macro and type root ExtrapoleBMtoTG.C+
3. Retrieve VTX vertex and compute distance to the BM track
   - Write and execute TrackDistanceBMtoVT.C macro (root TrackDistanceBMtoVT.C+)
4. Redoing 2 & 3 with the help of methods:
   - TAGgeoTrafo::FromTGLocalToBMLocal(TVector3 apoi)
5. Run DisplayMcFOOT.C with TG+VTX on and play with the different options

# Answer 1

- Command line:

```
. . .
EnableTree:          y
EnableHisto:         y
EnableTracking:      y

EnableSaveHits:      n
EnableRootObject:    n
EnableTofZmc:        n
EnableTofCalBar:     n
. . .
IncludeDI:           n
IncludeST:           y
IncludeBM:           y
IncludeTG:           y
IncludeVT:           y
IncludeIT:           n
IncludeMSD:          n
IncludeTW:           n
IncludeCA:           n
. . .
```

Set global parameters

- DecodeMC **-in** *12C_C_200_1.root* **-out** *12C_C_200_L0Out.root* **-nev** *1000* **-exp** *12C_200* **-run** *1*

# Answer 2 (i)

- Macro:

```
void ExtrapoleBMtoTG(TString nameFile = "12C_C_200_L0Out.root", TString expName = "12C_200",
                     Int_t runNumber = 1)
{
  GlobalPar::Instance(expName);
  GlobalPar::GetPar()->Print();

  TAGroot tagr;

  TAGcampaignManager* campManager = new TAGcampaignManager(expName);
  campManager->FromFile();

  TAGgeoTrafo* geoTrafo = new TAGgeoTrafo();
  TString parFileName = campManager->GetCurGeoFile(TAGgeoTrafo::GetBaseName(), runNumber);
  geoTrafo->FromFile(parFileName);

  TAGparaDsc* parGeoBm = new TAGparaDsc(TABMparGeo::GetDefParaName(), new TABMparGeo());
  TABMparGeo* pGeoMap = (TABMparGeo*)parGeoBm->Object();
  parFileName = campManager->GetCurGeoFile(TABMparGeo::GetBaseName(), runNumber);
  pGeoMap->FromFile(parFileName);


  TABMntuTrack *vtTrack = new TABMntuTrack();
  TAGdataDsc* vtTrck    = new TAGdataDsc("vtTrck", vtTrack);
  TAGactTreeReader* vtActReader  = new TAGactTreeReader("vtActEvtReader");
  vtActReader->SetupBranch(vtTrck, TABMntuTrack::GetBranchName());

  vtActReader->Open(nameFile);
  tagr.AddRequiredItem(vtActReader);

  tagr.BeginEventLoop();

  Int_t ev = 0;
```

Read global parameters

Initialise global tranformation

Read BM geometry

Set BM track branch

# Answer 2 (ii)

- Macro:

```
. . .
 while (tagr.NextEvent() ){

     TABMtrack* track = vtTrack->GetTrack(0);
     if (!track) continue;

     TVector3 A0 = track->Intersection(pGeoMap->GetMylar2().Z());
     A0.Print();

     Float_t posZtg = geoTrafo->FromTGLocalToGlobal(TVector3(0,0,0)).Z();

     posZtg = geoTrafo->FromGlobalToBMLocal(TVector3(0, 0, posZtg)).Z();

     TVector3 A1 = track->Intersection(posZtg);
     A1.Print();

     TVector3 A2 = geoTrafo->FromBMLocalToGlobal(A1);
     A2.Print();

     ev++;

     if (ev == 1)
       break;
  }
  tagr.EndEventLoop();
}
```

Retrieve first BM track

Track position at the output of BM

Target position in global framework

Target position in BM framework

Track position at target in BM framework

Track position at target in global framework

# Answer 3

- Macro:

```
void TrackDistanceBMtoVT(TString nameFile = "12C_C_200_L0Out.root", TString expName = "12C_200",
                         Int_t runNumber = 1)
. . .
 while (tagr.NextEvent()) {

     TABMtrack* track = bmTrack->GetTrack(0);
     if (!track) continue;

     TAVTvertex* vertex = vtVtx->GetVertex(0);
     if (!vertex) continue;

     Float_t posZtg = geoTrafo->FromTGLocalToGlobal(TVector3(0,0,0)).Z();

     posZtg = geoTrafo->FromGlobalToBMLocal(TVector3(0, 0, posZtg)).Z();

     TVector3 bmLoc = track->Intersection(posZtg);

     TVector3 bmGlo = geoTrafo->FromBMLocalToGlobal(bmLoc);


     TVector3 vtLoc = vertex->GetPosition();

     TVector3 vtGlo = geoTrafo->FromVTLocalToGlobal(vtLoc);

     TVector3 diff = vtGlo-bmGlo;

     fpHisDist->Fill(diff[0]);
  }

  tagr.EndEventLoop();

  fpHisDist->Draw();
}
```

Retrieve first BM track

Retrieve first VTX vertex

Target position in global framework

Target position in BM framework

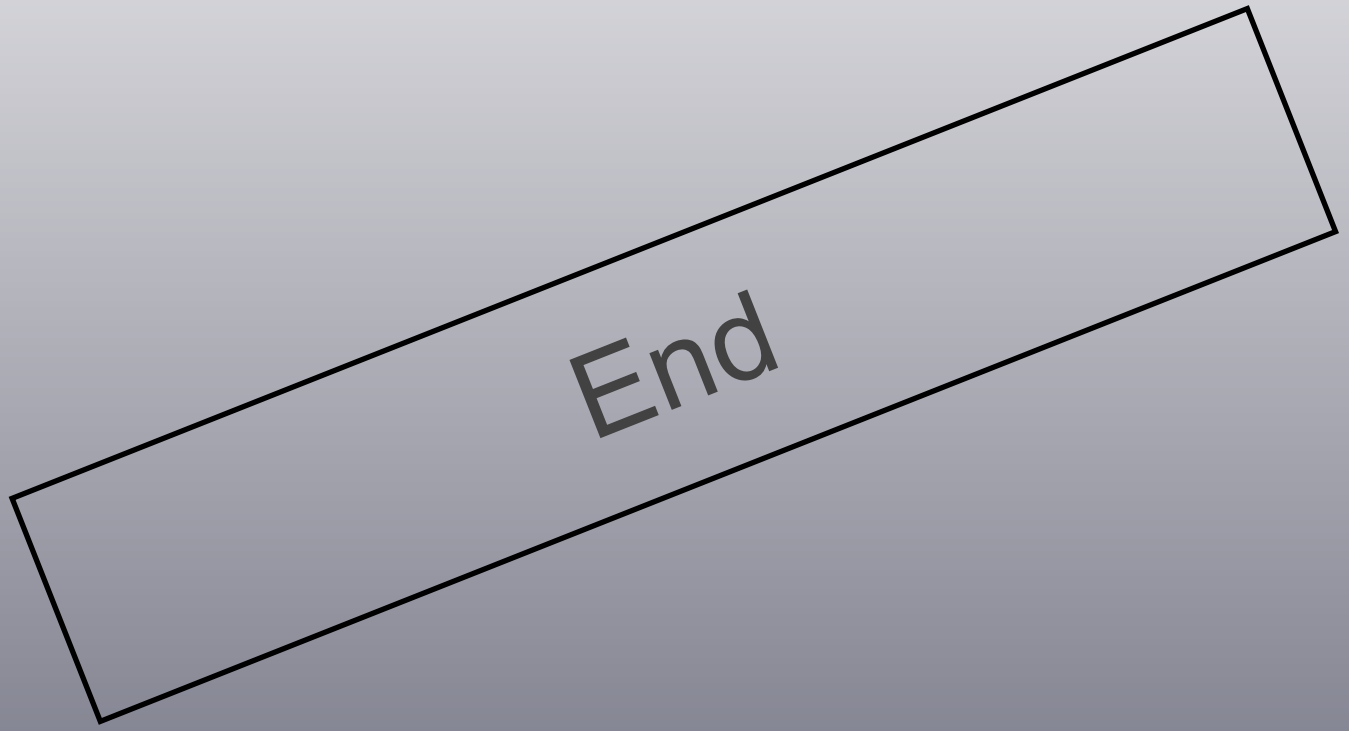
Track position at target in BM framework

Track position at target in global framework

Vertex position in VTX framework

Vertex position in global framework

Position difference


End