Second School on Dark Matter and Neutrino Detection

July 8-19, 2024

São Paulo, Brazil

ICTP-SAIFR/IFT-UNESP

LAr experiment design using Monte Carlo simulations

Franciole Marinho[1], Laura Paulucci[2]

[1]Instituto Tecnológico de Aeronáutica, Brazil, franciole@ita.br

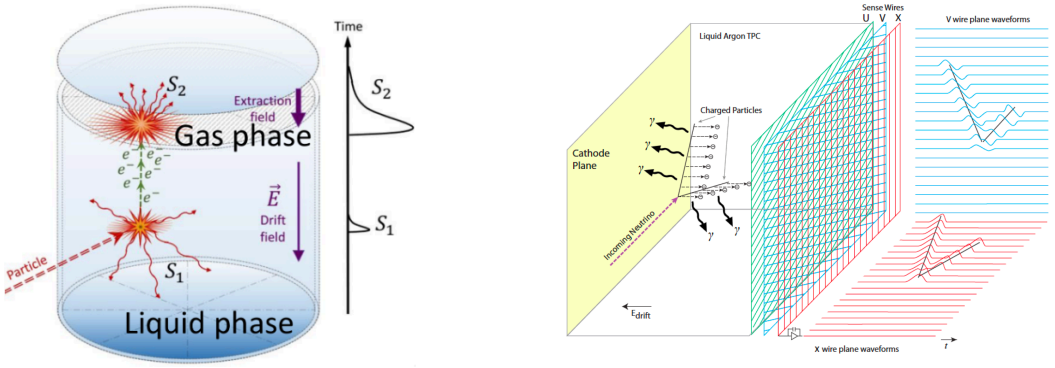
[2]Universidade Federal do ABC, Brazil, laura.paulucci@ufabc.edu.br

1. Brief introduction

Noble liquids are often employed as active media for neutrino interaction detection and dark matter (DM) searches as their physical and optical properties allow for the use of large cryostat volumes and the collection of both the scintillation light and free ionization charge. Scintillation light is produced in large amounts with energy deposited in the media by the passage of the charged particles coming from the final state of the primary interaction event. Scintillation light allows the detection of electron and nuclear recoils on low energy events in DM, and in CEvNS, solar and supernovae neutrino searches. It can provide fast signals for triggering of non-beam events in time projection chambers and also potentially independent calorimetric measurements on neutrino interactions in a broad range of energies from a few MeV up to the GeV scale. In LAr time projection chambers (LArTPCs) an applied electric field is used to collect free electric charges in segmented anode planes such that the 3D reconstruction of the trajectories of the produced particles in the neutrino interactions can be achieved with great resolution and their identification performed.

Figure 1 shows the main characteristics of two typical types of LArTPCs. On the dual phase type (left) two types of luminous signals are detected (S_1 & S_2) where S_1 corresponds to the detection of the ~128 nm VUV light emitted from the decays of the Ar_2 excimers produced on the passage of a charged particle through the liquid media while S_2 is the light obtained from the extraction of the avalanche electric charge which is initiated with the ionization electrons drifted from the interaction point up to the gas phase. On the single phase type (right) the same scintillation and free electron charges occur and are collected with the purpose to determine the interaction time t_0, particles trajectories, deposited energy and identification.

Figure 2 shows some examples of performance for noble liquid experiments. For instance, the top panel indicates an experiment's nuclear/electronic recoil discrimination capability. The bottom panel shows two examples of reconstructed events in a single phase LArTPC illustrating its trajectories with millimetric spatial resolution.



http://deap3600.ca/darkside-20k/ B. Abi *et al* 2020 *JINST* **15** T08008

Figure 1: Illustration of the functioning principle of dual phase (left) and single phase (right) LArTPCs.

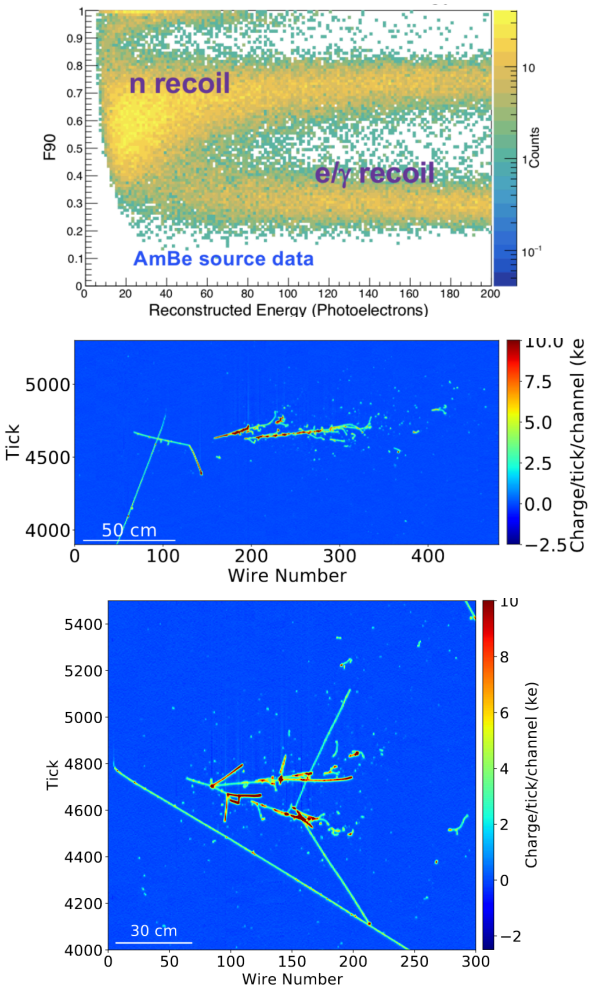


Figure 2: Recoil type discrimination (top) and LArTPC reconstructed events (bottom).

Provided the versatility of this type of detector technology and as they are increasingly becoming important tools in the particle physics field, it can be useful to develop expertise on how to implement simulation tools to describe their functioning and characterize their expected performance.

In this hands-on activity we will explore the development of Monte Carlo simulations with focus on liquid argon experiments using the Geant4 toolkit. After a short introduction to the basic elements needed to run the simulation (description of the geometry, physical and optical properties of the materials, particles interactions) participants will observe particle's propagation in the simulated media and design analyses producing results based on physical interpretation with simulated data.

2. Practical information

- In this lab you will receive a set of specific application/codes to use in the activities
- For each task there will be .pdf files with support material that you can access.
- Depending on the task at hand you will run/develop a Geant4 application, a ROOT standalone VUV sensor response simulation or perform scintillation light analysis.

Part 1: Geant4 LAr simulation

In this first part of the activities you are going to use a simple and already compiled Geant4 application to change parameters and/or implement additional code for including new features to the simulation.

After the first part of the introductory presentation on general MC simulation features you should be able to follow the steps described below.

Task:

1) A first look at the code.

- Open a terminal in you Virtual Machine environment and type "cd MC_LAr/"
- Check the following files in g4workshop_example/src/
  DetectorConstruction.cc, PhysicsList.cc, PrimaryGeneratorAction.cc

Can you guess what you should obtain/see when running the code?

- Go back to the terminal and on "g4workshop_build/" do "./g4workshop"
  - To change viewpoint type: "/vis/viewer/set/viewpointVector 10 5 10"
  - You should get on terminal a description of:
    Materials composition and characteristics
    Physics processes per particle type
    Optical, electromagnetic and hadronic and list of interactions (occurred only)

Does the outcome make sense?

Now let's move onto the optical processes simulation details.

2) Shooting optical photons

- Edit g4workshop_example/src/PrimaryGeneratorAction.cc
    - Make “opticalphoton” as primary particle
        G4ParticleTable::GetParticleTable()->FindParticle("opticalphoton");
    - Energy at ~eV range (VUV light)
        fParticleGun->SetParticleEnergy(9.68*eV);
    - Set polarization to
        G4ThreeVector polar = Polarisation(dir_vec);
        fParticleGun->SetParticlePolarization(polar);

- Optical properties of materials need to be included as well
  edit g4workshop_example/src/DetectorConstruction.cc and on *DefineMaterials()* function include:

```
const G4int nEntries = 5;
G4double energy[nEntries] = { 1.0*eV, 3.*eV, 6.*eV, 9.*eV, 12.*eV };
G4double n_lAr[nEntries] =  { 1.5, 1.5, 1.5, 1.5, 1.5};
```

Also on *DefineMaterials()* function include:

```
G4MaterialPropertiesTable* lAr_pt = new G4MaterialPropertiesTable();
lAr_pt->AddProperty("RINDEX", energy, n_lAr, nEntries);
env_mat->SetMaterialPropertiesTable(lAr_pt);
```

Can you guess what the event visualization should look like next time you run the application?

- Go back to terminal on “g4workshop_build/”
  compile the newest changes by doing “make -j4” and do “./g4workshop”

NOTE: If your laptop does not have four cores, you will need to change it in the virtual machine configurations. Please refer to the file explaining how to use the virtual machine made available.

Based on what you know about LAr properties, can you think of any desirable improvements towards a more realistic behavior for the optical photon trajectories?

Would implementing Rayleigh scattering make any difference?

```
G4double lR_lAr[nEntries] = {1.0*m,  1.0*m,  1.0*m,  1.0*m, 1.0*m};
 lAr_pt->AddProperty("RAYLEIGH", energy, lR_lAr, nEntries);
```

What happens when you do it?

3) Shooting optical photons onto something (wavelength shifters)

- Edit g4workshop_example/include/DetectorConstruction.hh
  insert TPB as a class member material

```
G4Material*   fTPB;
G4VPhysicalVolume*  tpbPhys;
G4LogicalVolume*     tpbLogic;
G4Box*          tpbSolid;
```

- Edit g4workshop_example/src/DetectorConstruction.cc.
  On *DefineMaterials()* function include the following:

```
//tpb material instance
G4Material* tpb_mat = new G4Material("tpb_mat",1.079*g/cm3,2);
G4Element* H = new G4Element ("Hydrogen","H",1.,1.01*g/mole);
G4Element* C = new G4Element ("Carbon","C",6.,12.01*g/mole);
tpb_mat->AddElement (C, 28);
tpb_mat->AddElement (H, 22);

//TPB optical info
const int tpb_entries = 10;

G4double tpb_e_energy[tpb_entries] = {3.125*eV, 3*eV, 2.875*eV, 2.75*eV,
2.625*eV, 2.5*eV, 2.375*eV, 2.25*eV, 2.125*eV, 2*eV};
G4double tpb_e[tpb_entries] = {0.120479, 0.718092, 1.00175, 0.640985, 0.389969,
0.157005, 0.0509217, 0.0080711, 0.000595335, 2.04258e-05};
G4double tpb_a_energy[tpb_entries] = {10.9729*eV, 7.07431*eV, 5.22486*eV,
4.2614*eV, 3.71444*eV, 3.36*eV, 3.16*eV, 2.96*eV, 2.9064*eV, 2.901*eV};
G4double tpb_l[tpb_entries] = {0.0594703*um, 0.0594703*um, 0.0573079*um,
0.106937*um, 0.0323873*um, 0.0608373*um, 0.581187*um, 13.2291*um, 87.4751*um,
950.347*um};

G4MaterialPropertiesTable* tpb_pt = new G4MaterialPropertiesTable();
tpb_pt->AddProperty("RINDEX", energy, n_lAr, nEntries);
tpb_pt->AddProperty("WLSABSLENGTH", tpb_a_energy, tpb_l, tpb_entries);
tpb_pt->AddProperty("WLSCOMPONENT", tpb_e_energy, tpb_e, tpb_entries);
tpb_pt->AddConstProperty("WLSTIMECONSTANT", 5*ns);
tpb_mat->SetMaterialPropertiesTable(tpb_pt);

fTPB = tpb_mat;
```

- Now on *ConstructLine()*:
  Reduce LAr volume size to 10cm
  Create a thin PTB layer volume (2cm x 2cm x 2μm)

```
tpbSolid = new G4Box("tpbSolid",1.0*cm,1.0*cm,1*um);
tpbLogic = new G4LogicalVolume(tpbSolid,fTPB,"tpbSolid");
tpbPhys = new G4PVPlacement(0,G4ThreeVector(0,0,0),"tpbSolid", tpbLogic,
fPhysiWorld, false, 0);
```

- Now change g4workshop_example/src/PrimaryGeneratorAction.cc accordingly

```
z0=1.0*cm;
G4ThreeVector dir_vec (0.,0.,-1.);
```

- Go back to the terminal on “g4workshop_build/” compile and run again

Check out the outcome of the simulation. Is there any additional information that could be interesting to access?

Part 2: ARAPUCA light trap simulation

Now let’s consider a hypothetical experiment instrumented with a certain sensor used to detect VUV scintillation light and try to estimate its electronic output signal taking into account the time characteristics of: LAr scintillation; WLS conversions; collection of trapped photons and single photo-electron pulse shape.

The ARAPUCA concept has been proposed as a simple solution for increasing the effective collection area of SiPMs through the shifting and trapping of scintillation light in noble liquids, with potential of improving timing and calorimetry resolution in neutrino and dark matter search experiments using time projection chambers. It is expected to achieve a single photon detection efficiency larger than 1%. The original design consists of a box made of highly reflective internal surface material and with an acceptance window for photons composed of a dichroic filter with two wavelength shifters deposited on its surfaces. The first shifter (PTP) converts liquid argon scintillation VUV light (127 nm) to a photon of wavelength smaller than the dichroic cutoff (~400 nm), so the surface is highly transparent to it. When passing through the dichroic filter, it reaches the second shifter (TPB) which allows the photon to be shifted to the visible region (400-560 nm) and be detected by the SiPM nested inside it. When it enters the box, the photon will likely reflect a few times, including on the dichroic filter surface, before being detected.

Figure 3 shows a 3D model of a detector indicating its acceptance window, reflective cavity and a silicon photomultiplier (SiPM). The acceptance window consists of a dichroic filter with one layer of ~ 7μm PTP deposited on top and one layer of ~ 3μm TPB deposited on the bottom of the filter as illustrated in figure 3. The widths of the PTP and TPB are chosen to ensure wavelength shifting. The incoming VUV photons are absorbed by the PTP material which in turn emits photons isotropically. As a consequence, half of these emitted photons pass through the dichroic material, reach the TPB and are absorbed. The TPB emits photons, which therefore, are trapped within the cavity.

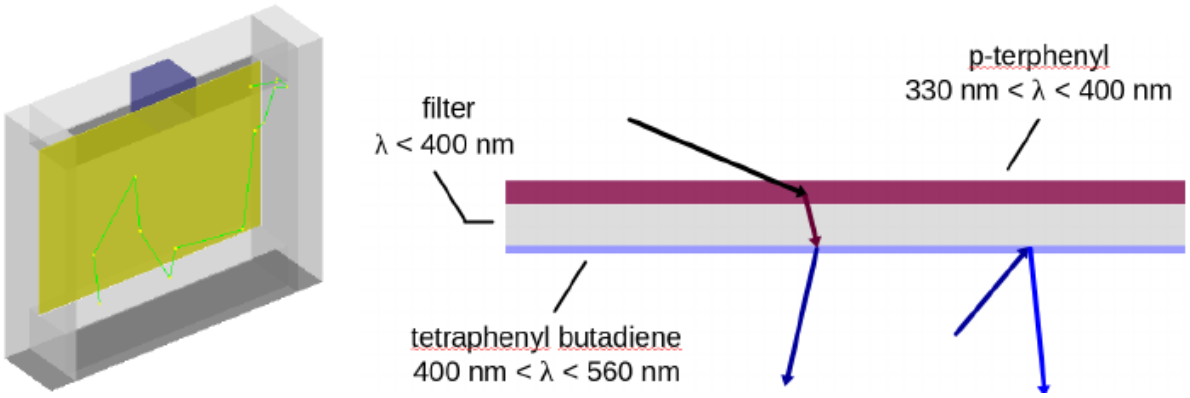


Figure 3: ARAPUCA device geometry(left): acceptance window(yellow), SiPM sensor(blue) and reflective cavity(gray). Acceptance window concept(right): PTP(magenta), dichroic filter(gray), TPB(blue).

Another aspect is the time characteristics of the ARAPUCA. There are three major delay sources that determine the acquisition time of the device:

$$t_{acquisition} = t_{PTP} + t_{TPB} + t_{collection},$$

where t_{PTP} and t_{TPB} are the emission decay time of the wavelength shifters and $t_{collection}$ is the time it takes for the photon accepted in the cavity to arrive on the SiPM after successive reflections. All these times are random quantities in an event-by-event basis which are used to determine the distribution for $t_{aquisition}$ assuming a mixture of exponential distributions for each of the emission times and their respective measured values. A first estimate of the most probable acquisition time was ~ 4 ns of a long tailed distribution.

Goal: Can arapucas identify types of particles from detected scintillation light?

The scintillation process characteristics depends on the particle type interacting in LAr on the balance between the fast and slow emission fractions

$$L(t) = \frac{A_S}{\tau_S} exp(-t/\tau_S) + \frac{A_f}{\tau_f} exp(-t/\tau_f)$$

with τ_f = 5.2ns and τ_s = 1.4s. For alpha particles and neutrons, $A_f/(A_s+A_f)$ = 0.71, while for electrons and muons $A_f/(A_s+A_f)$ = 0.23.

Suppose now that data is taken in a small LAr volume with just a few cm for light to travel from the generation point to the sensor.

Task:

1) Using the ROOT code provided in “MC_LAr/sensor_sim”, plot:

– LAr scintillation light time profile (em, nucl)

– Wavelength shifters emission profile

What would be the time distribution of photons arriving on the SiPM (em, nucl)?

2) Estimate the arapuca waveform using:

– time distribution of photons arriving on the SiPM

– single pe shape provided (1tick = 6.66ns)

and assuming:

– 500 detected photons

– 4 ADC noise level

EXTRA: Are you able to evaluate the nuclear/electron recoil discrimination capability of this device according to the number of PE?

Reminder: ROOT commands

● TF1 functions

– Draw(): plots the function in defined range

– GetRandom(): samples value assuming function shape as distribution

● TH1D histograms

– TH1D* h = new TH1D(“h”,“title”,nbins,xmin,xmax);

– “Draw” and “GetRandom” also work

– Fill(xvalue) and SetBinContent(bin,weight) to set values

– GetBinContent(bin) to access weight value

Part 3: PMT Light Analysis

The practice consists in the analysis of the light signal events for PMTs in a liquid argon experiment. The data is *simulated* based on the characteristics of the readout hardware and electronics. You will be given access to a collection of histograms for muons crossing a cryostat of size 5m x 4m x 4m and PMTs distributed behind two anodes, which are along the long walls, with a central opaque cathode.

Files are available in the folder “MC_LAr/pmt_data/” in the virtual machine or available for download at the school’s webpage.

Go to folder /opt/light

Task #1:

1) Open the file *pmtsim_muon.root* in ROOT. Open a browser (e.g. new TBrowser()) and check that there is one signal recorded for every PMT in the file. Take some time to understand what kind of information you can extract from these images.

2) Let’s identify the baseline and subtract it. Use the file *baseline_subtract.C* for this operation and choose an event number. For this, enter ROOT and type:

.L baseline_subtract.C

ana(“event number here”)

Check the result again in the ROOT browser (a new file was created with a name *my_event.root*).

3) Find what you believe to be a single PE peak in any of the histograms and obtain the integral of this peak. You can zoom in and use the command *Integral()*. Try to do that with more than one peak and get an average value. Take note.

Task #2:

1) For a given event, integrate the whole signal seen by each PMT and divide the result by the integral of a single pe. You can check the file *baseline_subtract.C* for inspiration on how to do that in a more automatic way. What does this value mean?

2) Open the file *map_pmt.dat* and fill in the last column with the value you obtained for each PMT. There are already four columns in the file, which are: PMT channel, z, y, and x of PMT in the cryostat. This will allow you to construct a map of the light seen by the PMTs in the event you are analyzing.

3) Create a 4D map in ROOT. For this, follow these steps:

Create a tree for reading the file: TTree* tree = new TTree();

Read the tree: tree->ReadFile("map_pmt.dat","ch/D:z/D:y/D:x/D:photons/D");

Change the style of your marker to a filled circle: tree->SetMarkerStyle(20);

Increase the size of your marker: tree->SetMarkerSize(1.5);

And now draw: tree->Draw("x:y:z:photons","","COLZ");

This will use (x,y,z) to position your marker and use the number of photons to give the color scale.

Can you say something about the track of the particle?

Challenge #1: Determine the arrival time of the first photon on the PMT. Choose one PMT and discuss with colleagues what would be a good criteria for identifying the arrival time. Try to determine that for a few PMTs and compare with the timestamp value, which is shown on the horizontal axis (as t – timestamp). Discuss your results.

Challenge #2: Now open file *pmtsim_muon_tpb.root*. This includes the TPB emission time. Discuss how the precision on the determination of the arrival time of the first photon on the PMT changes.