Jlab MeV Mott Polarimeter GEANT4 Simulation User's Guide

Martin McHugh
The George Washington University
mjmchugh@jlab.org

2015-07-29

Abstract

The purpose of this writing is to guide users of the GEANT4 Simulation of the Jlab MeV Mott Polarimeter after my departure. This serves as the technical guide compared to the documentation of the methods and results found in the tech note. Particularly, I discuss the classes in the simulation, the user commands I've added and the output rootfile structure.

MottG4 Classes

This section describes briefly the importance of each class, what files contain the pertinent information and some of the dependencies each has.

- `Mott.cc`: This is simply the `main()` function of the MottG4 simulation since it is after all `C++`. If you're doing work on this, you've likely changed enough to know exactly what it does.
- `MottDetectorConstruction`: Contains all the geometry definitions of the apparatus. Defines material properties of the various pieces. Defines which volumes are sensitive detectors.
- `MottDetectorMessenger`: Allows the user to modify certain parameters pertaining to the target and the step length in the simulation. Gives these inputs to the `MottDetectorConstruction` class.
- `MottDumpParameterisation`: In `MottDetectorConstruction`, the dump, made of beryllium backed by copper, is parameterized in steps of the cylindrical coordinates (z, r, ϕ). This is due to a depricated simulation version in which the dump was a sensitive detector. It was used to analyzed power deposition to determine the necessary cooling. The volume is still built in `MottDetectorConstruction` but it is no longer a sensitive detector.
- `MottEventAction`: Class which determines, at the end of each event, whether a sensitive volume has been hit and then writes it to the `ROOT`file.
- `MottEventActionMessenger`: Allows the user to determine what events the `MottEventAction` interactively.

- `MottPhysicsList`: Standard physics list for e^- (i.e. multiple scattering, ionization, bremsstrahlung). Only non-typical piece is optical photon physics, however, the simulation currently does not track optical photons in the scintillators because it was shown to be superfluous and computationally expensive.
- `MottPhysicsListMessenger`: Allows the user to activate optical photon tracking easily. By Default this is off for the above mentioned reasons.
- `MottPrimaryGeneratorAction`: The true heart of the simulation. This class contains all of the methods which interface with the theoretical tables we use. Based on user input, it calculates the kinematics and dynamics of either one or two elastic Mott scatterings inside of the target foil and then generates electrons which are thrown from a number of angular distributions. If you're going to make changes to how we do things, you're probably going to end up here.
- `MottPrimaryGeneratorMessenger`: Allows the user to interactively specify which angular distribution of scattered electrons to throw and whether or not the electrons are to have one or two scatterings calculated.
- `MottRunAction`: Determines the output `ROOT`file structure.
- `MottRunActionMessenger`: Allows the user to change output `ROOT`file destinations and names etc.
- `MottSteppingAction`: If optical photons are turned on by `MottPhysicsListMessenger`, This serves as the place where detections in photo-cathodes are calculated. The hits are then stored in the appropriate `MottTrackerSD`.
- `MottSteppingVerbose`: Allows for different levels of output per step. Useful for debugging and not much else.
- `MottTrackerHit`: Records hits in a `MottTrackerSD`.
- `MottTrackerSD`: Records the detector response in the sensitive detector volumes indicated in the `MottDetectorConstruction` class. There are 8 instances currently, representing the 4 dE and 4 E detectors.

Interactive and Macro User Commands

This section details the various commands that the user can run in either interactive or batch (macro driven) modes to access the different functionality of the simulation. Briefly, they are:

- `/Target/SetTargetLength`: Sets the target thickness along the beam-axis. Command requires a double and a unit (length). Simulation default is 1 μm.
- `/Target/TargetIn`: Puts the target-center at the nominal physical position. This command is not necessary for most simulations as the target foil already begins in it's physical location unless moved.

- `/Target/TargetOut`: Moves the target-center to the location (0, 1 m , -3.8978 mm), well outside of the scattering chamber. Useful for seeing what happens in the dump.
- `/Target/SetTargetMaterial`: Choose Silver or Gold for the target material. Command requires the user to enter either "Au" or "Ag."
- `/Stepping/stepMax`: Sets the maximum step length of a particle track if it does not encounter a new volume or undergo some process. The default value is 1/2 the length of the world volume. Decreasing the step size is not recommended for simulations with large numbers of events as it can increase computing time dramatically.
- `/PhysicsList/SwitchOpticalPhotons`: Turn on (1) or off (0, DEFAULT) Optical photon creation and propogation in the scintillators.
- `/Beam/SetBeamEnergy`: Set the nominal kinetic energy of the beam. Due to the use of data tables for Mott scattering in the primary generator the value selected should be one of the following, (3 MeV, 5 MeV, 6 Mev, 8 MeV).
- `/Beam/SetEnergySpread`: Sets the standard deviation of the energy about the central value. Command requires a double and a unit (energy). Simulation default is 25 keV.
- `/Beam/SetBeamDiameter`: Sets the FWHM value for the circular beam profile. Command requires a double and a unit (length). Simulation default is 1.0 mm.
- `/PrimaryGenerator/EventType`: Allows the user to specify what angular distribution they want to throw from. All event types are defined in `MottPrimaryGeneratorAction::GeneratePrimaries()` method. Options are:
  - 0 - Throw from upstream into the target
  - 1 - Throw single scattered e- at the detectors (default)
  - 2 - Throw double scattered e- at the detectors
  - 3 - Throw single scattered e- at the detectors [Rejection Sampling]
  - 4 - Throw double scattered e- at the detectors [Rejection Sampling]
  - 5 - Throw single scattered e- isotropically in specified angular range
- `/PrimaryGenerator/SetThetaMin`: Defines the minimum scattering angle. Must be a double with a unit (angle) between 0 and π radians.
- `/PrimaryGenerator/SetThetaMax`: Defines the maximum scattering angle. Must be a double with a unit (angle) between 0 and π radians.
- `/PrimaryGenerator/SetPhiMin`: Defines the minimum azimuthal angle. Must be a double with a unit (angle) between 0 and 2π radians.
- `/PrimaryGenerator/SetPhiMax`: Defines the maximum azimuthal angle. Must be a double with a unit (angle) between 0 and 2π radians.
- `/Analysis/RootFileName`: Takes a string of the form `/path/to/rootfile_name.root` and writes the output `ROOT`file to that location.

- `/EventAction/StoreAllEvents`: User inputs 0 - (simulation default) to store only those events which hit one of the eight detector scintillators, or 1 - to store all events regardless of detector hits.

Simulation Outputs

The simulation has two primary outputs. The first, to command line, can be piped or manipulated in the typical fashion. This typically only includes standard GEANT4 information and errors and warnings. The second and more useful output is a `ROOT`file which includes all of the physics information of interest and detector responses for each event. The output `ROOT`file has the following leaves:

- `Up_E`: Energy [MeV] deposited in the UP E detector.
- `Up_dE`: Energy [MeV] deposited in the UP ΔE detector.
- `Down_E`: Energy [MeV] deposited in the DOWN E detector.
- `Down_dE`: Energy [MeV] deposited in the DOWN ΔE detector.
- `Left_E`: Energy [MeV] deposited in the LEFT E detector.
- `Left_dE`: Energy [MeV] deposited in the LEFT ΔE detector.
- `Right_E`: Energy [MeV] deposited in the RIGHT E detector.
- `Right_dE`: Energy [MeV] deposited in the RIGHT ΔE detector.
- `Up_E_PMT`: Photo-electrons detected in the UP E detector PMT.
- `Up_dE_PMT`: Photo-electrons detected in the UP ΔE detector PMT.
- `Down_E_PMT`: Photo-electrons detected in the DOWN E detector PMT.
- `Down_dE_PMT`: Photo-electrons detected in the DOWN ΔE detector PMT.
- `Left_E_PMT`: Photo-electrons detected in the LEFT E detector PMT.
- `Left_dE_PMT`: Photo-electrons detected in the LEFT ΔE detector PMT.
- `Right_E_PMT`: Photo-electrons detected in the RIGHT E detector PMT.
- `Right_dE_PMT`: Photo-electrons detected in the RIGHT ΔE detector PMT.
- `Event_ID`: Simulation event number.
- `PrimaryVertexKEprime`: Kinetic energy [MeV] of the primary electron as generated.
- `PrimaryVertexX`: Position [mm] on the x-axis of the primary scattering.
- `PrimaryVertexY`: Position [mm] on the y-axis of the primary scattering.

- `PrimaryVertexZ`: Position [mm] on the z-axis of the primary scattering.
- `PrimaryVertexTheta`: Scattering angle [deg] of the primary scattering.
- `PrimaryVertexPhi`: Azimuthal angle [deg] of the primary scattering.
- `PrimaryVertexPX`: Polarization along the x-axis of the post-scattering electron.
- `PrimaryVertexPY`: Polarization along the y-axis of the post-scattering electron.
- `PrimaryVertexPZ`: Polarization along the z-axis of the post-scattering electron.
- `PrimaryCrossSection`: Differential cross-section (cm^2/sr) of the primary scattering.
- `PrimarySherman`: Sherman function of the primary scattering.
- `PrimarySpinT`: Spin T function of the primary scattering.
- `PrimarySpinU`: Spin U function of the primary scattering.
- `SecondaryVertexKEprime`: Kinetic energy [MeV] of the secondary electron as generated.
- `SecondaryVertexX`: Position [mm] on the x-axis of the secondary scattering.
- `SecondaryVertexY`: Position [mm] on the y-axis of the secondary scattering.
- `SecondaryVertexZ`: Position [mm] on the z-axis of the secondary scattering.
- `SecondaryVertexTheta`: Scattering angle [deg] of the secondary scattering.
- `SecondaryVertexPhi`: Azimuthal angle [deg] of the secondary scattering.
- `SecondaryVertexPX`: Polarization along the x-axis of the post-scattering electron.
- `SecondaryVertexPY`: Polarization along the y-axis of the post-scattering electron.
- `SecondaryVertexPZ`: Polarization along the z-axis of the post-scattering electron.
- `SecondaryCrossSection`: Differential cross-section (cm^2/sr) of the secondary scattering.
- `SecondarySherman`: Sherman function of the secondary scattering.
- `SecondarySpinT`: Spin T function of the secondary scattering.
- `SecondarySpinU`: Spin U function of the secondary scattering.
- `Theta`: Scattering angle [deg] at which the electron is thrown relative to the z-axis.
- `Phi`: Azimuthal angle [deg] at which the electron is thrown relative to the x-axis.