

# **GEANT4 Simulation of the MeV Mott Polarimeter:**

## **A Tutorial**

**M.J. McHugh  
2016-05-10**

# Overview

- What you need
- Checking out, configuring, and compiling
- Running in interactive mode
- Running in batch mode
- Output Files
- Where do I make changes?
- Assignments
- Resources

# What You Need: ifarm usage

- Computer on the internet
- Access to the JLab ifarm computing clusters
- Access to the GitHub JlabMottGroup repository

There is a working version maintained at:

```
/u/group/mottgrp/MottG4
```

To use this version, you must have:

```
source /site/env/syscshrc.alt
```

```
setenv MOTTG4DIR /u/group/mottgrp/MottG4
```

```
source /site/12gev_phys/production.csh
```

or their bash/zsh equivalents in your startup scripts.

# What You Need: local box Usage

- GEANT4 installed and able to compile and execute the included examples :)
  - <http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/html/>
- cmake version 2.8 or more recent

Note: GEANT4 is rather sensitive about it's running environment. It's not trivial to get and keep running.

# Checking Out & Compiling

- To checkout a version to develop, login to ifarm then:

```
> cd /path/to/directory/ (wherever you want MottG4)
> git clone https://github.com/JLabMottGroup/MottG4.git MottG4
> setenv MOTTG4DIR /path/to/directory/MottG4
```

- If `git clone` didn't work, try this:

```
> git config --global --add http.proxy "http://jprox.jlab.org:8081"
> git clone https://github.com/JLabMottGroup/MottG4.git MottG4
```

- To compile:

```
> cd MottG4/Mott_Polarimeter/
> mkdir build && cd build
> cmake ..
> make
```

# Running Interactively

- **To Run:**

```
> cd build/  
> ./mott
```

Visualization may not work on the farm without proper port forwarding!

- **Try:**

```
> ssh -Y
```

The screenshot shows the mott application interface. On the left is a 'Scene tree' panel with a search bar and a list of expandable categories: control, units, process, geometry, tracking, event, cuts, run, random, particle, Target, Stepping, PhysicsList, Beam, PrimaryGenerator, gun, Analysis, EventAction, material, hits, vis, and gui. A red arrow labeled 'Command Reference' points from the 'Command' header to the right. Below the scene tree is a large empty area labeled 'Command line' with a red arrow pointing to it. On the right is a 'viewer-0 (OpenGLStoredQt)' window showing a 3D visualization of a particle detector component. Below the viewer is an 'Output' window displaying technical data: Initial kinetic energy (IKE): G4BestUnit (G4double), Initial momentum magnitude (IMag): G4BestUnit (G4double), Initial momentum (IMom): G4BestUnit (G4ThreeVector), No. of points (NTP): G4int, PDG Encoding (PDG): G4int, Parent ID (PID): G4int, Particle Name (PN): G4String, G4SmoothTrajectoryPoint: Auxiliary Point Position (Aux): G4BestUnit (G4ThreeVector), Step Position (Pos): G4BestUnit (G4ThreeVector), and Visualization verbosity changed to warnings (3). A red arrow labeled 'Text Output' points to this text. At the bottom right is a 'Session:' input field.

# Running Interactively

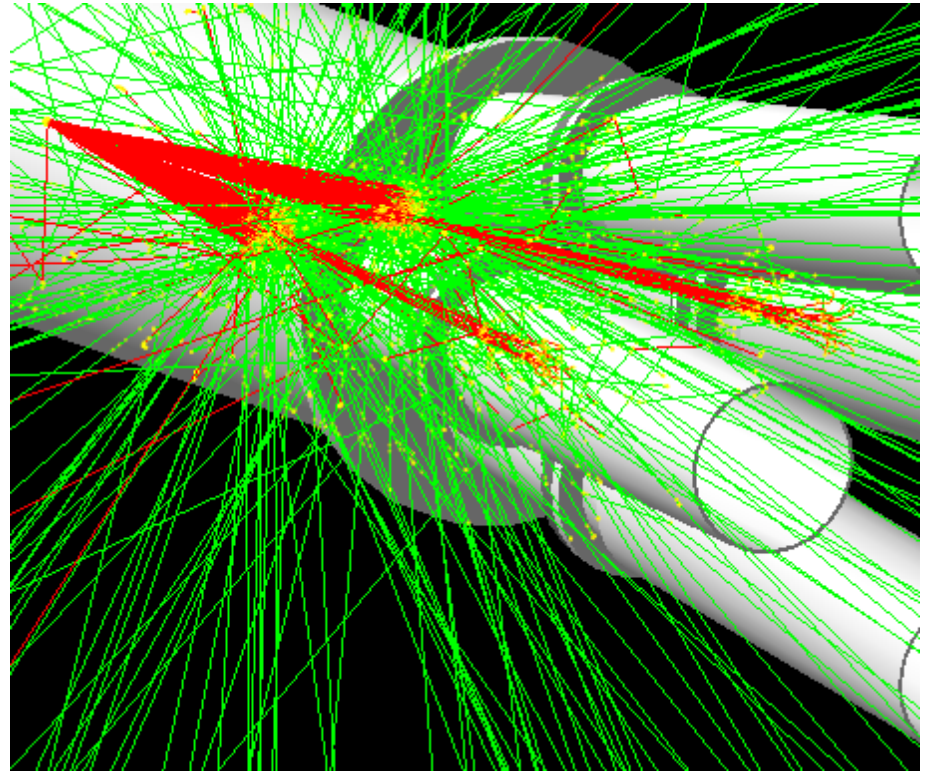
- To make a run we can analyze:

```
> /EventAction/StoreAllEvents 1  
> /run/beamOn 1000
```

- Can manipulate viewer w/ mouse ->

- To look at that run, close the sim, then:

```
> cd ..  
> root -l MottSim.root  
> Mott->Draw("Left_E", "Left_E>0", "");
```



# Running in Batch Mode

- User generated macros determine beam parameters for simulation to run without visualization.

- Examples found in `macros/`

- Make a macro for 4 runs:

```
> cd /path/Mott_Polarimeter/  
> vi macros/myAsym.mac
```

- To run:

```
> cd MottG4/Mott_Polarimeter/  
> ./build/mott macros/myAsym.mac
```

- Macro contents:

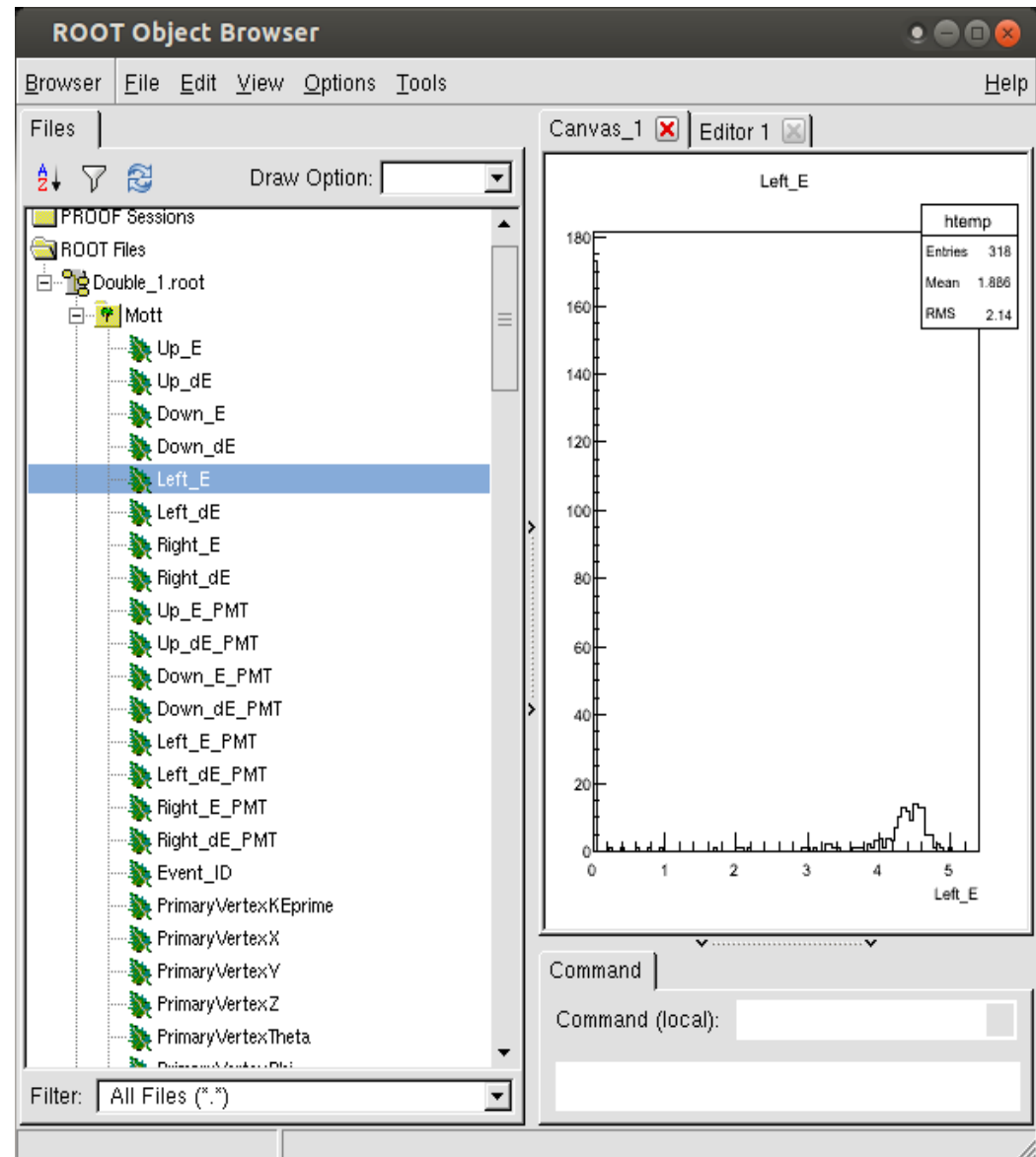
```
/Target/SetTargetLength 0.100 um  
  
/PrimaryGenerator/EventType 3  
/Analysis/RootFileName Single_1.root  
/run/beamOn 1000  
  
/PrimaryGenerator/EventType 4  
/Analysis/RootFileName Double_1.root  
/run/beamOn 1000  
  
/Target/SetTargetLength 0.200 um  
  
/PrimaryGenerator/EventType 3  
/Analysis/RootFileName Single_2.root  
/run/beamOn 1000  
  
/PrimaryGenerator/EventType 4  
/Analysis/RootFileName Double_2.root  
/run/beamOn 1000
```

## Breakout: Look at output rootfiles



# ROOTfile Structure

- Energy deposited in detectors
- PMT response (OFF BY DEFAULT)
- Kinematics:
  - Angle
  - polarization
- Scattering Dynamics:
  - Cross sections
  - Spin functions
- To see the structure:
  - > `root -l Single_1.root`
  - > `Mott->Print()`



# Structure of MottG4: Where do I make changes?

- `Mott.cc`: Boilerplate file, contains `main()`
- `src/MottRunAction.cc`: Creates ROOTfile for output, called at the start of each run.
- `src/MottDetectorConstruction.cc`: Contains the geometry of all objects and assigns sensitive volumes.
- `src/MottPrimaryGeneratorAction.cc`: Loads in Xavier's Mott physics, throws electrons based on this physics and user inputs.
- `src/Mott_____Messenger.cc`: Allows the user to interact with the \_\_\_\_\_ class through command line and macro interfaces.
- Xavier's files are in the directory: `/MottG4/NewMottPhysics/`
- Additional info can be found in the user's guide I wrote:  
[https://wiki.jlab.org/ciswiki/images/4/42/Mott\\_Geant4\\_User\\_Guide.pdf](https://wiki.jlab.org/ciswiki/images/4/42/Mott_Geant4_User_Guide.pdf)

# Correcting the Scattering Angle

- Joe Reported:  $\Theta = 172.71^\circ$
- Lets put that in:
  - > vi src/MottDetectorConstruction.cc
  - Find: G4double ScatteringAngle = 172.6\*deg;
  - Save
  - > cd build/ && make
- Note: After initial cmake, you do not need to re-run unless you add new source files (src/\*.cc or include/\*.hh) you can simply run make

# Pushing Changes to Repository

- Check where you are w.r.t. Github:
  - > `git diff master origin/master`
- If it looks good:
  - > `git push origin master`

# Assignments

- Try everything shown in here:
  - If it doesn't work document your problem and send it to me.
  - Make sure you have all your environments configured properly.
- Check my to-do list on the mott wiki.
- Ask me any questions you can think of!

# Advanced Use: Submitting jobs to the farm

- Be on the farm or know how to submit jobs remotely.
- Make sure that you have scientific computing certificate! If you don't know: read up at <http://scicomp.jlab.org/scicomp/>
- Follow this example:

```
> cd MottG4/Mott_Polarimeter
> perl g4jsub.pl sample.mac --events 20000 --jobs 5 -r "/work/username/directory"
```
- You must use an appropriate `sample.mac` and send the output rootfiles (-r flag) to an appropriate directory.
- Example: `/macros/Single_0.052um.mac`
- The ROOTfiles used in my tech note analysis, which were generated in the above manner, are located in:

```
> /lustre/expphy/volatile/hallc/qweak/mjmchugh/Mott/Round2
```

# Advanced Use: Reproducing Tech Note Analysis

- Be on the farm. Have access to the directory mentioned on the last slide or files generated from the macros included in the package:
- Make sure the `Rate_Calculation.C` and `Rate_Calculation_2.C` scripts are pointing in the right places (you will have to edit them if you do not generate the rootfiles with the macros provided)
- Compile with

```
> g++ -o Rate_Calculation Rate_Calculation.C `root-config --cflags --glibs`
```
- Run with:

```
> ./Rate_Calculation 0.100 10The Probelsm
```
- Similarly for `Rate_Calculation_2.C`
- These calculate the single scattering rate and double scattering rate as described in the tech note.
- Outputs of these files are indicated in the scripts themselves.
- Familiarization is necessary! These WILL need modifications in the future!

# Advanced Assignments

- Reproduce my tech note rates!
  - Hint: Follow the above two slides and look at all of the macros and scripts closely!
- Make a script which analyzes the asymmetry from rejection method generators (3-single and 4-double):
  - Look at appropriate generators in:
    - > `src/MottPrimaryGeneratorAction.cc`
  - Remember: Rootfiles from these generators already show a different number of hits in left and right detectors