User guide for MULTIPASS example

Michael Hubbard
University of Surrey
Github: thehubbard

April 2, 2020, revision 1.0

1 What is MULTIPASS?

MULTIPASS is an acronym for **MUL**tiple **TI**me constant **P**sd scintill**AT**ion emi**SS**ion and is an example code that extends Geant4 such that scintillation emission with more than two decay time constants can be modelled. It is based off of the code for simulating the EJ-276 detector system in Michael Hubbard's PhD thesis.

2 Code design and layout

Layout of the files follows the standard of other examples provided my Geant4. Headers are found in the `include` directory with corresponding source files located in `src`

```
 [User:  MULTIPASS ]$   ls --group-directories-first
__DOCS__  ChangeLog.md    MULTIPASS.cc     ScanningTable.mac  STsubloop_X.mac
include   CMakeLists.txt  photon.mac       singlescint.mac    STsubloop_Y.mac
random    defaults.mac    README.txt       singlescint.rndm   subloop_xdim.mac
src       gui.mac         reviewEvent.mac  slabLoop.mac       vis.mac
```

The equivalent to the `main` is `MULTIPASS.cc`. There are a variety of macros that have been created these are for simulations or visualisations. Inside the `include` directory is the `cxxopts.hpp` header, this is an external lightweight C++ option parser library. The key files in this simulation are the `G4OpticalPhysicsMod` and `G4ScintillationMod`, which are modifications of the original scintillation and optical physics files that now allow for multiple time constants to be simulated. An overview for the rest of the files can be found in Table. 1

3 Compiling

The `MULTIPASS` source code has been build and tested again Gean4.10.04.p01 other versions may work. For details on versions of ROOT and cmake, review the source code.

Create a build directory for example `build_MULTIPASS`.

```
[User:  ~ ]$   mkdir build_MULTIPASS
[User:  ~ ]$   cd build_MULTIPASS
[User:  build_MULTIPASS ]$
```

Then inside this directory, compile using the standard cmake method.

```
[User:  build_MULTIPASS ]$   cmake -DGeant4_DIR=path/to/geant4/install/directory
  path/to/source
```

then use `make` top produce the executable file.

File	Description
G4OpticalPhysicsMod	Modified Optical physics processes.
G4ScintillationMod	Multi time constant scintillation code.
MULTIPASSActionInitialization	Action initialisations.
MULTIPASSComptonHit	Collector for gamma interactions.
MULTIPASSDetectorConstruction	Material definitions and basic setup.
MULTIPASSDetectorMessenger	Control options for the detector.
MULTIPASSEventAction	Data collected is written to file.
MULTIPASSEventMessenger	Control of data collection.
MULTIPASSHistoManager	Data collection setup.
MULTIPASSMainVolume	Building the scintillator and PMT setup.
MULTIPASSOpticalHit	Optical photon data collection.
MULTIPASSPhysicsList	Physics processes setup.
MULTIPASSPMTHit	Counter for detected photons that reach PMT.
MULTIPASSPMTSD	Stores data in PMThit and Opticalhit collections.
MULTIPASSPrimaryGeneratorAction	Particle generator.
MULTIPASSPrimaryGeneratorMessenger	Controller for particle generator.
MULTIPASSRunAction	Starts the run and creates the output file.
MULTIPASSScintHit	Energy deposited in scintillator data collector.
MULTIPASSScintSD	Sensitive detector stores energy deposited and gamma interaction processes to relevant collections.
MULTIPASSStackingAction	Scintillation and Cherenkov counters.
MULTIPASSSteppingAction	Optical photon detection code.
MULTIPASSSteppingMessenger	Stepping action controller.
MULTIPASSSteppingVerbose	Stepping verbose output leftover from LXe example.
MULTIPASSTrackingAction	From LXe example stores track information.
MULTIPASSTrajectory	Chooses when is drawn when visualisation is enabled.
MULTIPASSUserEventInformation	Event information variables.
MULTIPASSUserTrackInformation	Track information variables.
ChangeLog.md	Change log from EJ-276 example.
CMakeLists.txt	For compiling the code.
gui.mac	Visualisation setup.
README.txt	Acronym readme file.
reviewEvent.mac	Example of how to review event from stored information.
ScanningTable.mac	For replicating scanning table experiment.
singlescint.mac	Another implementation of reviewEvent.mac with different verbose output.
singlescint.rndm	Random event seed file.
slabLoop.mac	Leftover example from EJ-200 detector for increasing volume. This no longer works but shows how to use loops.
STsubloop_X.mac	Components needed for ScanningTable.mac
STsubloop_Y.mac	Components needed for ScanningTable.mac
subloop_xdim.mac	Components needed for slabLoop.mac
vis.mac	Visualisation setup component.

Table 1: Overview of files included in the MULTIPASS example.

```
[User: build_MULTIPASS ]$  make
```

If you amend the code just make should be required to compile with the latest updates. However, if new `.hh, .cc, .mac` files are added the `cmake` command will have to be run again. New files in the `include` and `src` will not need to be added to the `CMakeLists.txt`, any files such as macros outside these directories need to be added to the `CMakeLists.txt` file.

4 Use

`MULTIPASS` has multiple options for operation to see a full extent run the file with the help command.

```
[User: build_MULTIPASS ]$  ./MULTIPASS --help
 - a simulation software for MULTIPASS detector
Usage:
  ./MULTIPASS [OPTION...]

  -m, --macro arg    run macro
  -r, --runmode arg  Run mode [Full, Gamma, Optics] (default: Optics)
  -h, --help         Print help
```

The macro flag allows a macro file to be passed such that the simulation is started in batch mode with no visualisations enabled. There are several different run modes an overview is provided in Table. 2, the default mode is `Optics`.

Mode	Description
Full	Fires a gamma-ray into the scintillator. Scintillation process is enabled and optical photons are generated when interactions occur.
Gamma	Fires a gamma-ray into the scintillator. Scintillation and optical photons are disabled.
Optics	

Table 2: Different run modes available in the MULTIPASS example.

When running the program several options are accessible by messengers. Some can only be used in certain run modes. An overview of some of the important commands is found below:

`/MULTIPASS/detector/dimensions <x> <y> <z> <unit>`
Change scintillator slab dimensions.

`/MULTIPASS/detector/housingThickness <thickness> <unit`
Change scintillator housing thickness.

`/MULTIPASS/detector/housingThickness <radius> <unit>`
Change PMT radius

`/MULTIPASS/detector/reflectivity <reflectivity>`
Change percentage reflectivity accepts values from 0 to 1.

`/MULTIPASS/detector/MainScintYield <scintillationYield>`
Change scintillation yield in Full mode. Unit is per MeV.

`/MULTIPASS/gun/particle <particle>`
Choose between Gamma or Neutron. In Optics mode this selects the time constants of the particle type for simulation.

`/MULTIPASS/gun/particleEnergy <energy> <unit>`
Choose energy of the particle.

`/MULTIPASS/gun/scintYield <scintillationYield>`
Change scintillation yield in Optics mode. The total number of photons simulated in this mode is equal to the particle energy multiplied by the scintillation yield. Unit is per MeV.

`/MULTIPASS/gun/InteractionOrigin <location>`
Location of optical photon generation in Optics mode. Default is Center. Available: Center[1], Back, Top, Topback, Front, Topfront, Side, Sideback, Sidefront, Cornerback, Cornerfront, Edge.

`/MULTIPASS/gun/offset <offset>`
Offset from nearest face(s) in Optics mode.

`/MULTIPASS/gun/wavelength <wavelength>`
Wavelength in nm of generated optical photons in Optics mode.

`/MULTIPASS/gun/ScanTableX <position>`
X positional interaction location centre in Full and Gamma mode. (0,0) is bottom left where, top is defined as face with PMT.

`/MULTIPASS/gun/ScanTableY <position>`
Y positional interaction location centre in Full and Gamma mode. (0,0) is bottom left where, top is defined as face with PMT.

In all modes the standard `/run/beamOn` starts the simulation. Each run creates a data file output. For the Gamma and Full modes the particles interact with a spot size with a 1 cm diameter on the face of the detector. The x and y position marks the centre of the spot, this was chosen to replicate the scanning table experimental setup.

5 Data output

Depending on run mode the data output is different. In the `HistoManager` the data output it currently selected as `.root`, editing this file allows for other data formats to be selected. The output file name starts with the run mode and includes the number of events in the run. If the optics mode is used the emission location in terms of top right etc is stored in the name. For gamma and full modes the x-y positional information is stored in the file name.

5.1 Full

Energy deposited in scintillator is recorded as both a ROOT histogram and in an ntuple. The total number of optical photons detected is also stored in a histogram. Along with separate histograms for number of reflections and pathlength of optical photons. Two 2D histograms are also filled. One has the information relating to the emission direction of the detected photons, stored in terms of the azimuthal and polar angles. An additional 2D histogram produces pseudo pulse shapes. The x-axis is time in ns, with 1 ns intervals. The y-axis corresponds to the event number in the run.

In the ntuple along with energy deposited, the total number of reflections, number of optical absorptions, number of boundary absorptions and the information regarding scintillation emission and Cherenkov photons is stored.

5.2 Gamma

Only energy deposited in scintillator is recorded. This is stored in a ROOT histogram and an ntuple.

[1] American English spelling used as in programming mode Michael's brain switches to this. Mainly because of the word color in the programming languages!

5.3 Optics

The same as full mode data output, however, energy deposited is not recorded.