Introduction to Geant4 with Remoll

Cameron Clarke

Cameron.Clarke@stonybrook.edu

Department of Physics and Astronomy
Stony Brook University

May 27, 2021

Overview

Stony Brook University

G4 is Your Virtual Laboratory (and Friend)

- Why do we want to perform simulations?

Basic Building Blocks of G4

- How do we set up a simulation, what is Remoll?
- Key Ingredients
- What happens inside a simulation (and why should I care)?

Utilizing G4

- How do we run a simulation?
- How do we visualize a simulation?
- How do we analyze a simulation?

Why do we want to perform simulations?

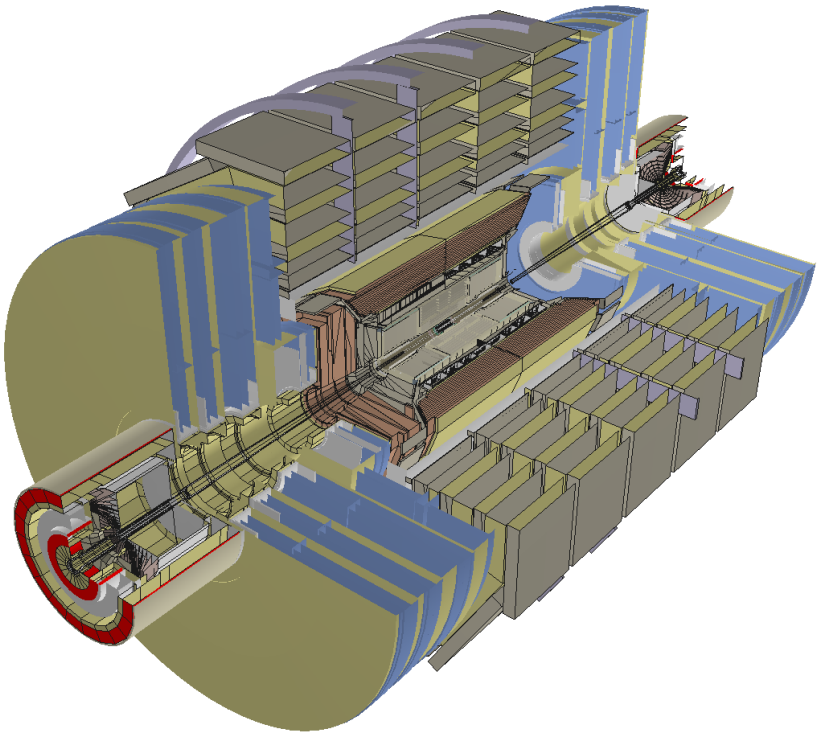
Stony Brook University

Geant4 lets you construct a virtual copy of a physical world and simulate whichever physics processes into whatever geometries you want.

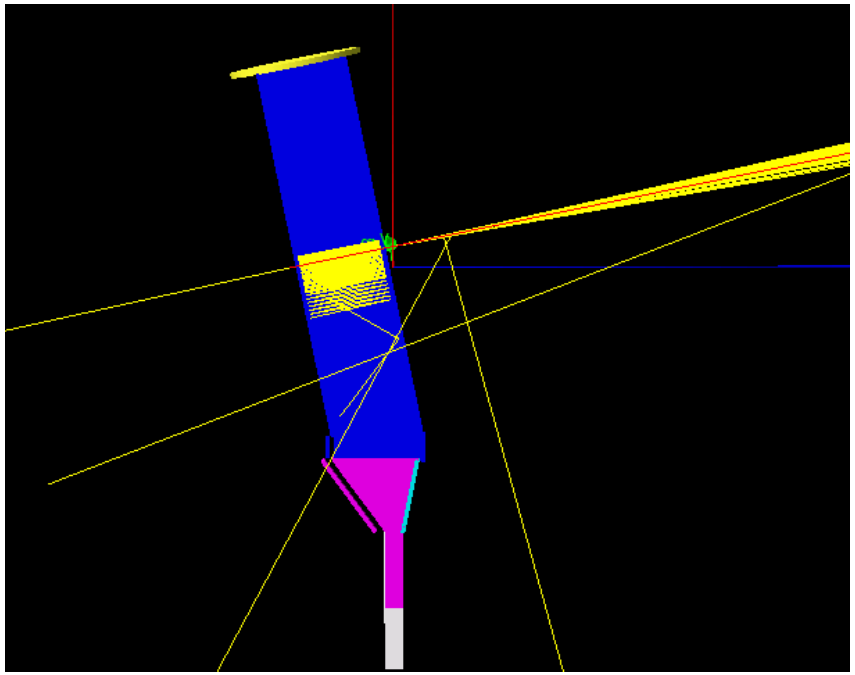
We can utilize this power in many ways:

- **For "real world" predictions (carefully and in collaboration)**
  - Most basically, to Monte Carlo-estimate what signals are visible under what initial and boundary conditions, without having to do explicit calculations or measurements.
  - To build a full-scale replica of an experiment and predict what it will measure.
  - To extract physics observables from measured distributions convoluted with theory predictions.
- **As a step in the scientific method (your virtual lab/friend)**
  - To test the impact of changes in an experiment's design on relevant observables.
  - To do extensive, omniscient optimizations, consequence free.
  - To use benchmarking from expensive real-world data to extrapolate beyond what practical limitations allow.

Why do we want to perform simulations? (cont.)



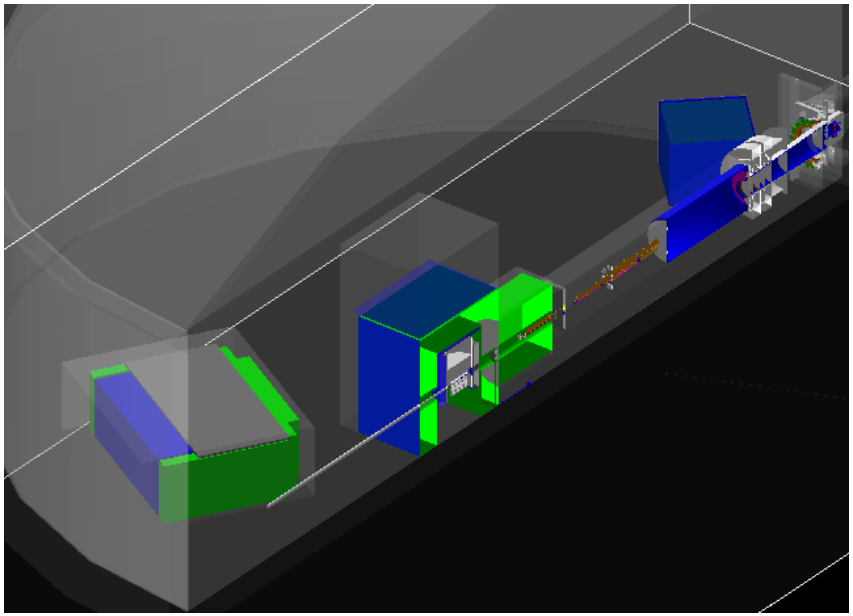
(a) The entire CMS detector at CERN



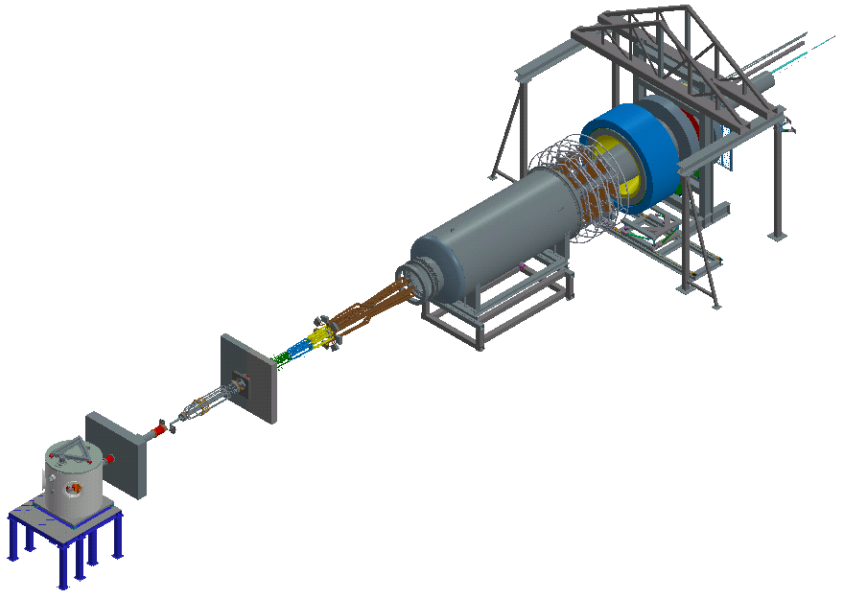
(b) MOLLER light guide optimization

Figure 1: Two examples of what you can do with Geant4, from simple to complex scenarios.

Why do we want to perform simulations? (cont.)



(a) MOLLER as it stands in `remoll`



(b) MOLLER in CAD

Figure 2: MOLLER requires lots of testing, tweaking, optimizing, and calibrating to achieve it's few % error bar goals with minimum cost and maximum safety.

Consider the virtual simulation world as a consequences free (other than your time and computing resources) laboratory.

How do we set up a simulation

Stony Brook University

Geant4 Basics:

- Ubiquitously used in nuclear and particle physics, as well as in adjacent fields and industry.
- Simulates particle interactions with matter (in addition to other things)
- Toolkit based in C++, allowing for user optimization, and can interface with `ROOT`
- Available, already installed at JLab and on various institutions' computing resources

Introduction and documentation:
https://geant4.web.cern.ch/support/user_documentation

What is Remoll?

Stony Brook University

Hypothetically, someone asks you to "run a simulation" in Geant4:

- Geant4 is a toolkit, meaning the user must specify which components to utilize, as well as how and when.
- What are the bare minimum components you need?
- Has someone else already implemented these in a way you can use?

Yes, `remoll` exists (a **re**implementation of the older **MOLL**ER simulation)

OK, so let's look at the `remoll` source code and see what is in there

What is Remoll? (cont.)

Stony Brook University

```
src/HepMCG4PythiaInterface.cc
src/HepMCG4Interface.cc
src/HepMCG4AsciiMessenger.cc
src/HepMCG4AsciiInterface.cc
src/remollGenFlat.cc
src/remollGenAl.cc
src/remollGen12CElastic.cc
src/remollFileReader.cc
src/remollEvent.cc
src/remollEventAction.cc
src/remollActionInitialization.cc
src/radDamage.cc
src/HepMCG4PythiaMessenger.cc
src/remollHEPEvtMessenger.cc
src/remollHEPEvtInterface.cc
src/remollGenPion.cc
src/remollGenpInelastic.cc
src/remollGenMoller.cc
src/remollGenLUND.cc
src/remollGenHyperon.cc
src/remollGenericDetectorSum.cc
src/remollGenericDetectorHit.cc
src/remollGenericDetector.cc
src/remollVertex.cc
src/remollTrackReconstruct.cc
src/remollTrackingAction.cc
src/remollTextFile.cc
src/remollSystemOfUnits.cc
src/remollSteppingAction.cc
src/remollRun.cc
src/remollRint.cc
src/remollPhysicsList.cc
src/remollGenBeam.cc
src/remollVEventGen.cc
src/remollRunAction.cc
src/remollGenpElastic.cc
src/remollGenExternal.cc
src/remollDetectorConstruction.cc
src/remollBeamTarget.cc
src/remollMultScatt.cc
src/remollMagneticField.cc
src/remollIO.cc
src/remollGlobalField.cc
src/remollSearchPath.cc
src/remollRunData.cc
src/remollPrimaryGeneratorAction.cc
src/remollParallelConstruction.cc
```

```
include/remollGenPion.hh
include/remollGenpElastic.hh
include/remollGenMoller.hh
include/remollGenLUND.hh
include/remollGenHyperon.hh
include/remollGenFlat.hh
include/remollGenericDetectorSum.hh
include/remollGenericDetectorHit.hh
include/remollGenericDetector.hh
include/remollGenAl.hh
include/remollGen12CElastic.hh
include/remollFileReader.hh
include/remollEvent.hh
include/remollEventAction.hh
include/remollActionInitialization.hh
include/radDamage.hh
include/HepMCG4PythiaMessenger.hh
include/HepMCG4PythiaInterface.hh
include/HepMCG4Interface.hh
include/HepMCG4AsciiMessenger.hh
include/HepMCG4AsciiInterface.hh
include/remollVertex.hh
include/remollUserTrackInformation.hh
include/remollTrackReconstruct.hh
include/remollTrackingAction.hh
include/remollTextFile.hh
include/remollSystemOfUnits.hh
include/remollSteppingAction.hh
include/remollRun.hh
include/remollRunAction.hh
include/remollRint.hh
include/remollPrimaryGeneratorAction.hh
include/remollPhysicsList.hh
include/remollPhysicalConstants.hh
include/remollLinkDef.hh
include/remollHEPEvtMessenger.hh
include/remollHEPEvtInterface.hh
include/remollglobs.hh
include/remollGenpInelastic.hh
include/remollGenBeam.hh
include/remollVEventGen.hh
include/remolltypes.hh
include/remollSearchPath.hh
include/remollParallelConstruction.hh
include/remollRunData.hh
include/remollMultScatt.hh
include/remollMagneticField.hh
include/remollIO.hh
include/remollGlobalField.hh
include/remollGenExternal.hh
include/remollDetectorConstruction.hh
include/remollBeamTarget.hh
```

```
README.mc
beampipe
detector
mollerMother_Clamshell_Optimized.gdml
misc
matrices.xml
issues
mollerMother_noShlds.gdml
huts
hall
generators
electronics
mollerParallel.gdml
mollerMother_parametrized.gdml
materials.xml
tracking
solids
mollerMother.gdml
donut
meshes
hybrid
positions.xml
pion
showermax
target
upstream
```

```
# store tracks
#/tracking/storeTrajectory 1

# This must be called before initialize
/remoll/geometry/setfile geometry/mollerMother.gdml
# Parallel world geometry is optional - detector 28
# (the primary detector array's idealize vacuum detector)
# is included in this parallel world now.
/remoll/parallel/setfile geometry/mollerParallel.gdml

#/remoll/physlist/register QGSP_BERT_HP
/remoll/physlist/parallel/enable
# if optical physics is turned on it will only
# work if parallel physics is not turned on.
#/remoll/physlist/optical/enable

# This must be explicitly called
/run/initialize
/remoll/printgeometry true
/control/execute macros/load_magnetic_fieldmaps.mac

# Raster and initial angle stuff
/remoll/oldras true
/remoll/rasx 5 mm
/remoll/rasy 5 mm

#/remoll/evgen/set beam
# To hit Ring 5 open in septant 4
#/remoll/evgen/beam/x -1300 mm
#/remoll/evgen/beam/rasx -30 mm
#/remoll/evgen/beam/origin -987.5 0 -50 mm
#/remoll/evgen/beam/direction (-0.0523,0,0.9986)
# Mainz test energy
#/remoll/beamene 855 MeV

/remoll/evgen/set moller
#/remoll/evgen/thcommin 30.0 deg
#/remoll/evgen/thcommax 150.0 deg

/remoll/evgen/beamPolarization +L
/remoll/field/equationtype 2
/remoll/field/steppertype 2
/remoll/field/print

/remoll/beamene 11 GeV
/remoll/beamcurr 85 microampere

# Make interactions with W, Cu, and Pb
# realistic rather than pure absorbers
/control/execute macros/kryptonite.mac

/process/list

# Specify random number seed
/remoll/seed 123456

/remoll/filename remollout.root

#/tracking/verbose 2
/remoll/target/print
/run/beamOn 100
```

Figure 3: All of the `remoll` source, header, and geometry files, and an example simulation macro file.

What is Remoll? (cont.)

Stony Brook University

Yikes! ... how to even get started?

- Step back
- Consider the key ingredients
- All the Geant4/`remoll` code must boil down

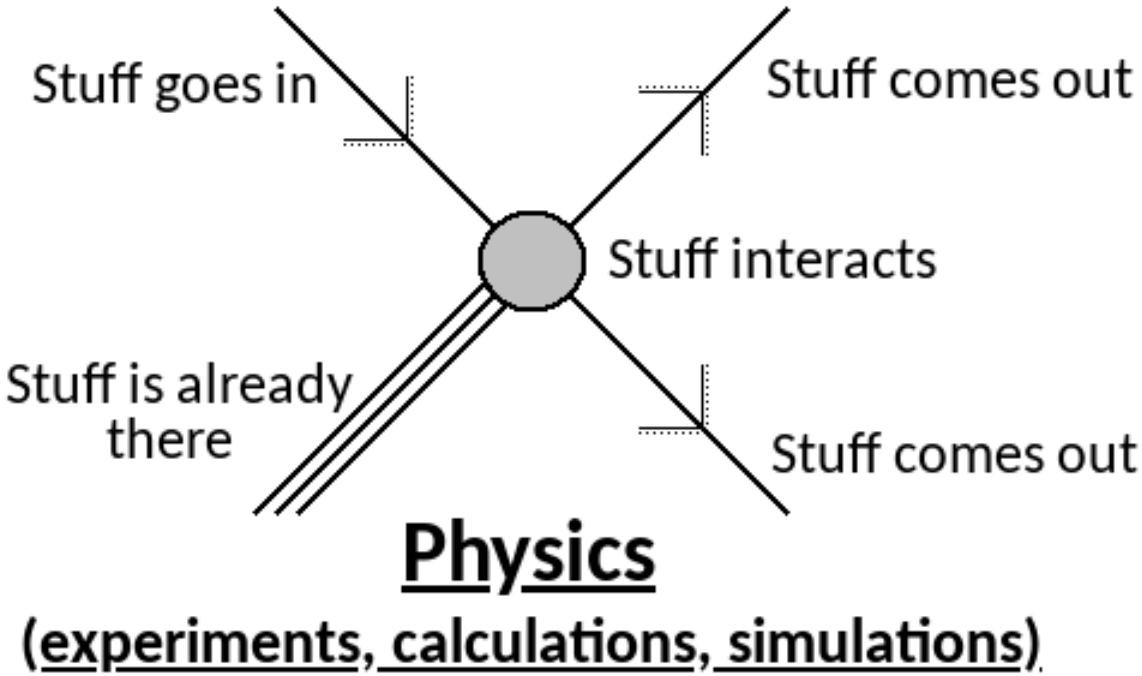


Figure 4: Why we are all here.

What is Remoll? (cont.)

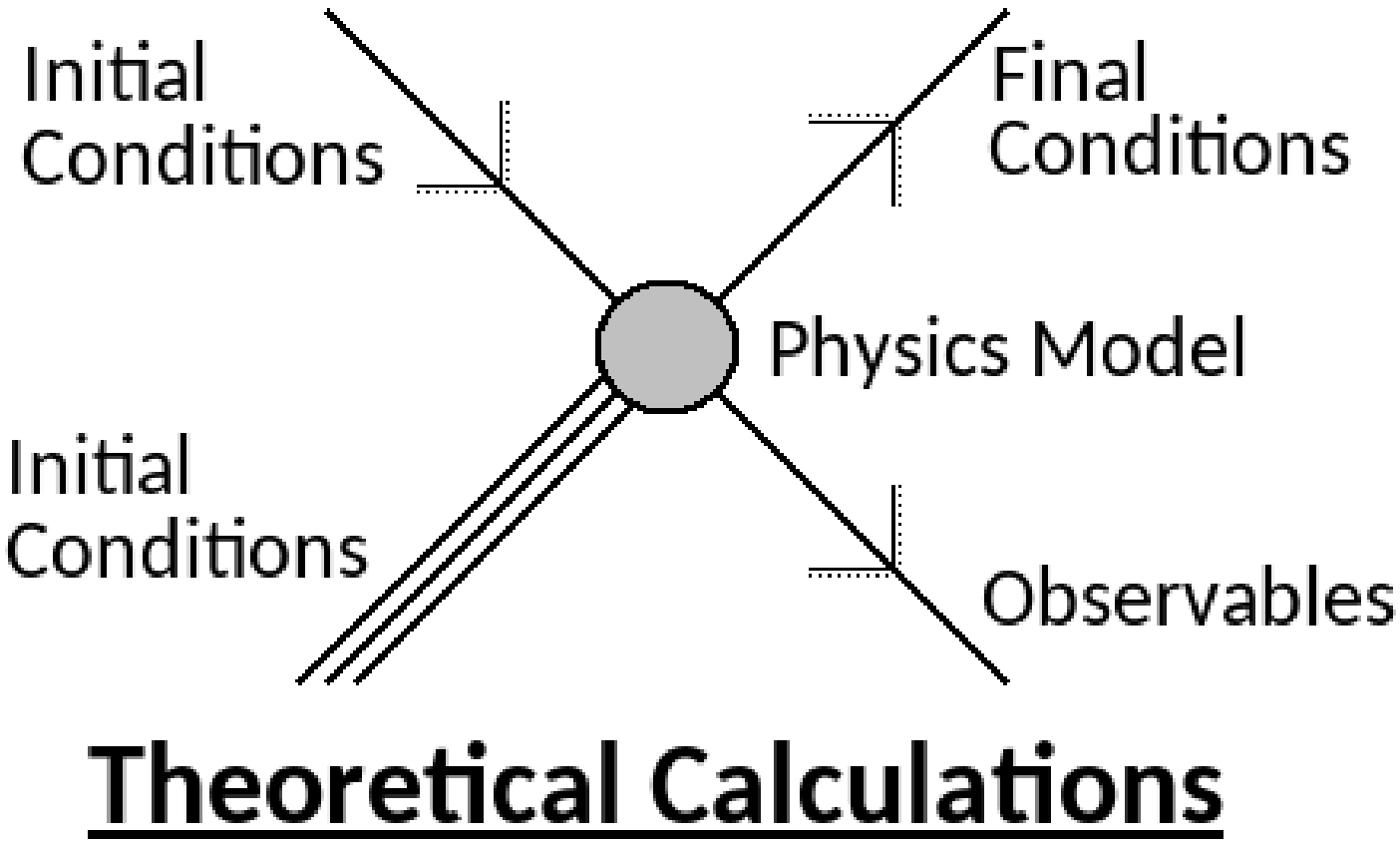


Figure 5: The key ingredients for theoretical predictions.

What is Remoll? (cont.)

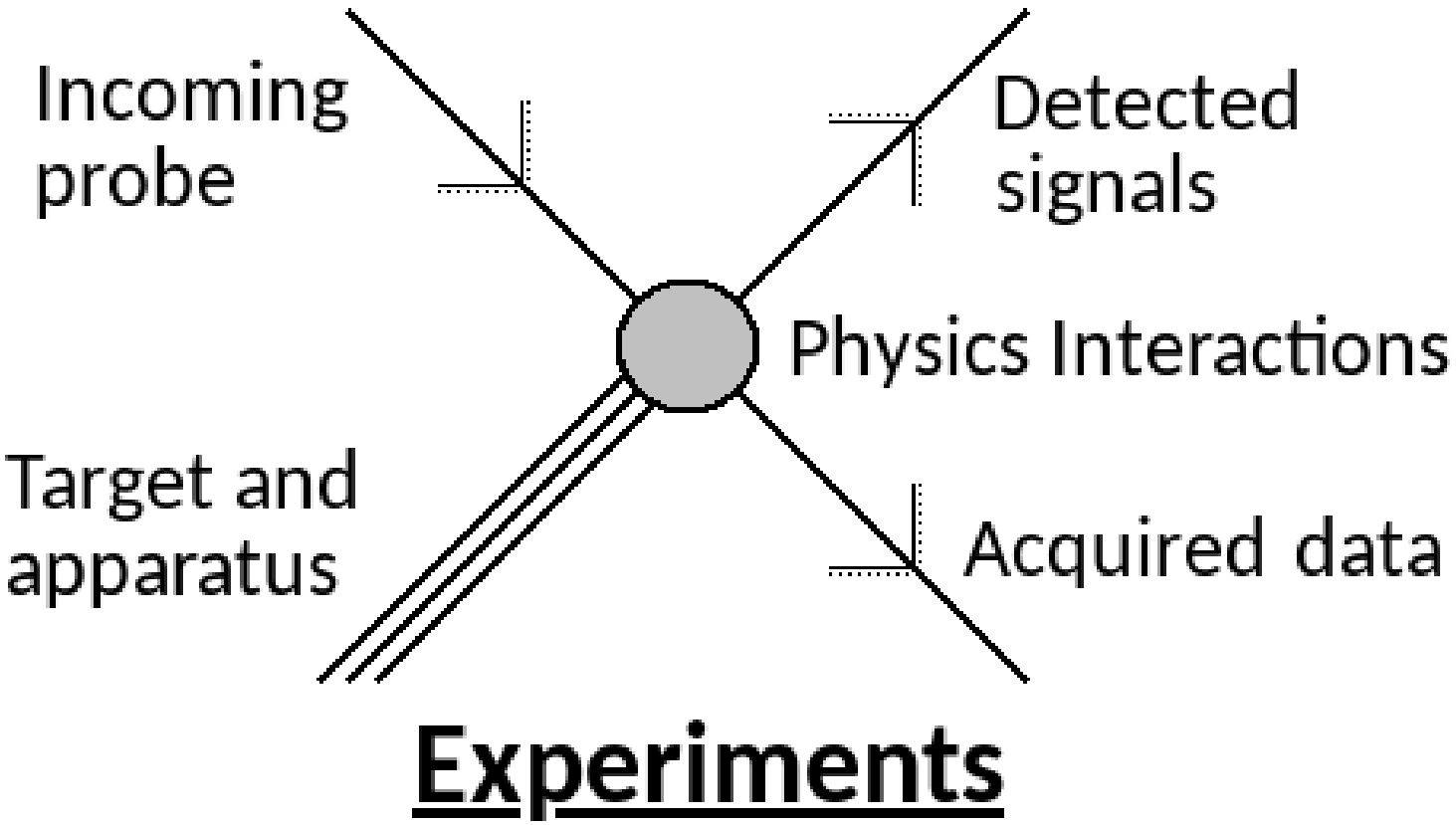


Figure 6: The key ingredients for measuring in an experiment.

Key Ingredients

Stony Brook University

The following 4 key ingredients are the backbone of a Geant4 simulation:

1. **Event generators**
   - Choosing what kinds of events, their kinematics, $\frac{d\sigma}{d\Omega}$, A_{PV}, etc.
   - Can be as simple as a plain electron beam, or as complex as desired
2. **Geometry definitions**
   - Geometry placements and material definitions
   - Rules for recording particles' information
3. **Physics lists**
   - Defines what physical processes Geant4 will simulate
   - Includes choices of which models to use (speed/accuracy trade-offs) and optional additional processes (generators *can* replace these)

4* **Some form of output**
   - Geant4 doesn't automatically print outputs of simulated physics
   - It provides some default formats and information recording routines
   - *Optional: must be defined by the user, with lots of options available

Key Ingredients (cont.)

Stony Brook University

OK, great, but how do we utilize these key ingredients in `remoll`?

1. **Event generators**
   - Pick which to use and configure them using commands in macro files
   - Gives you control over initial conditions in a simulation job
2. **Geometry definitions**
   - Define geometries using GDML that `remoll` parses (see later talk)
   - Gives you control over the boundary conditions, in the form of material placements, properties, and particle sensitivity (detectors)
3. **Physics lists**
   - Set and configure these using commands in macro files
   - Gives you control over which physical processes can take place
4. **Some form of output**
   - Define what outputs each simulation job writes to a `ROOT` file
   - Gives you control to analyze what happens in simulations

Start with those steps and you will find that Geant4 and `remoll` are your own personal laboratory (and can even be your friend).

Key Ingredients (cont.)

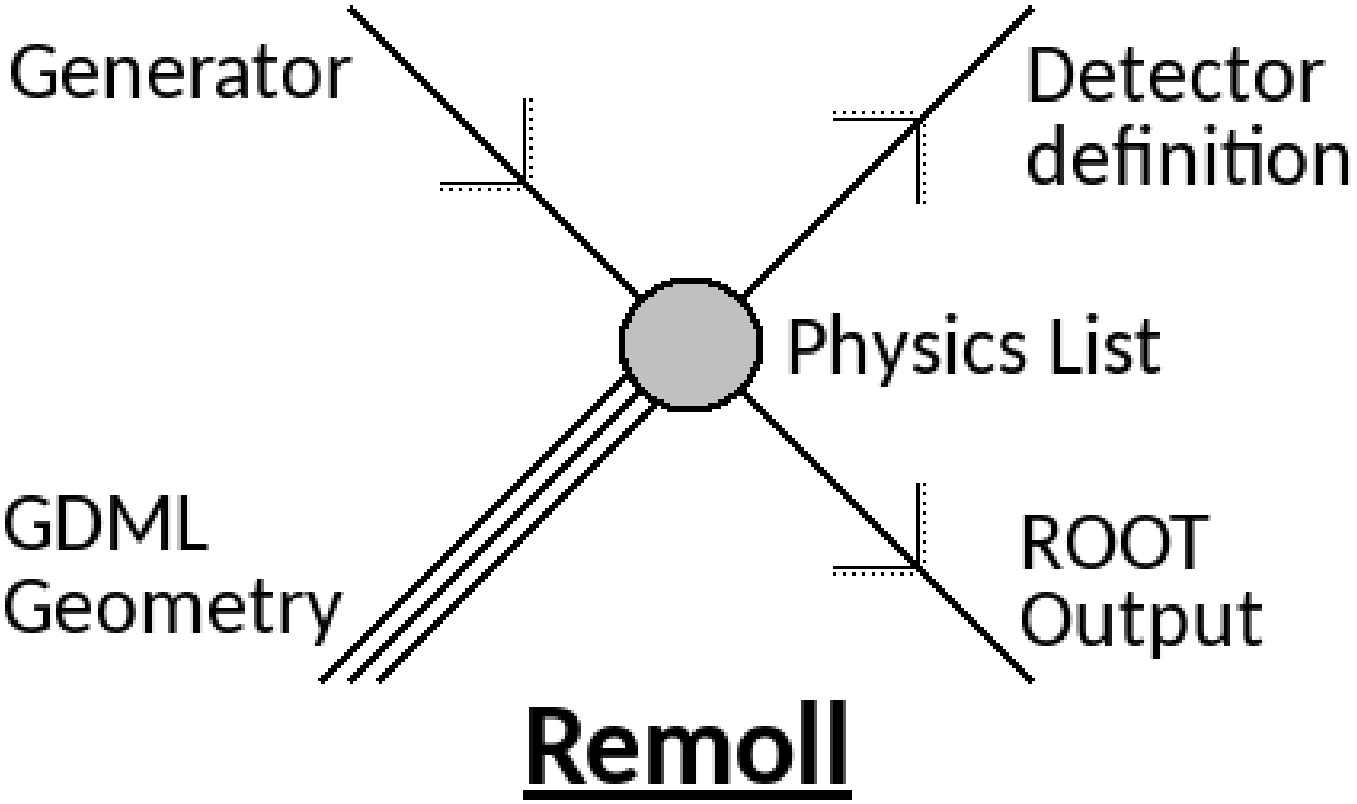


Figure 7: The key ingredients for simulating with remoll.

Wait! Before jumping head first into simulating:

- It is important to understand how Geant4 produces and processes events
- This helps avoid mistakes and grants access to the most useful information

---

When submitting a single simulation job Geant4 does the following:

- The generator, geometry, and physics list are set and stored in memory
- Nested loops perform the computations simulating particles' passage through matter and all of their consequences
- Selected data are stored in `ROOT` outputs

So, let's look at a flow chart of the steps of simulating particles, the classes that govern them, and mistakes to avoid.

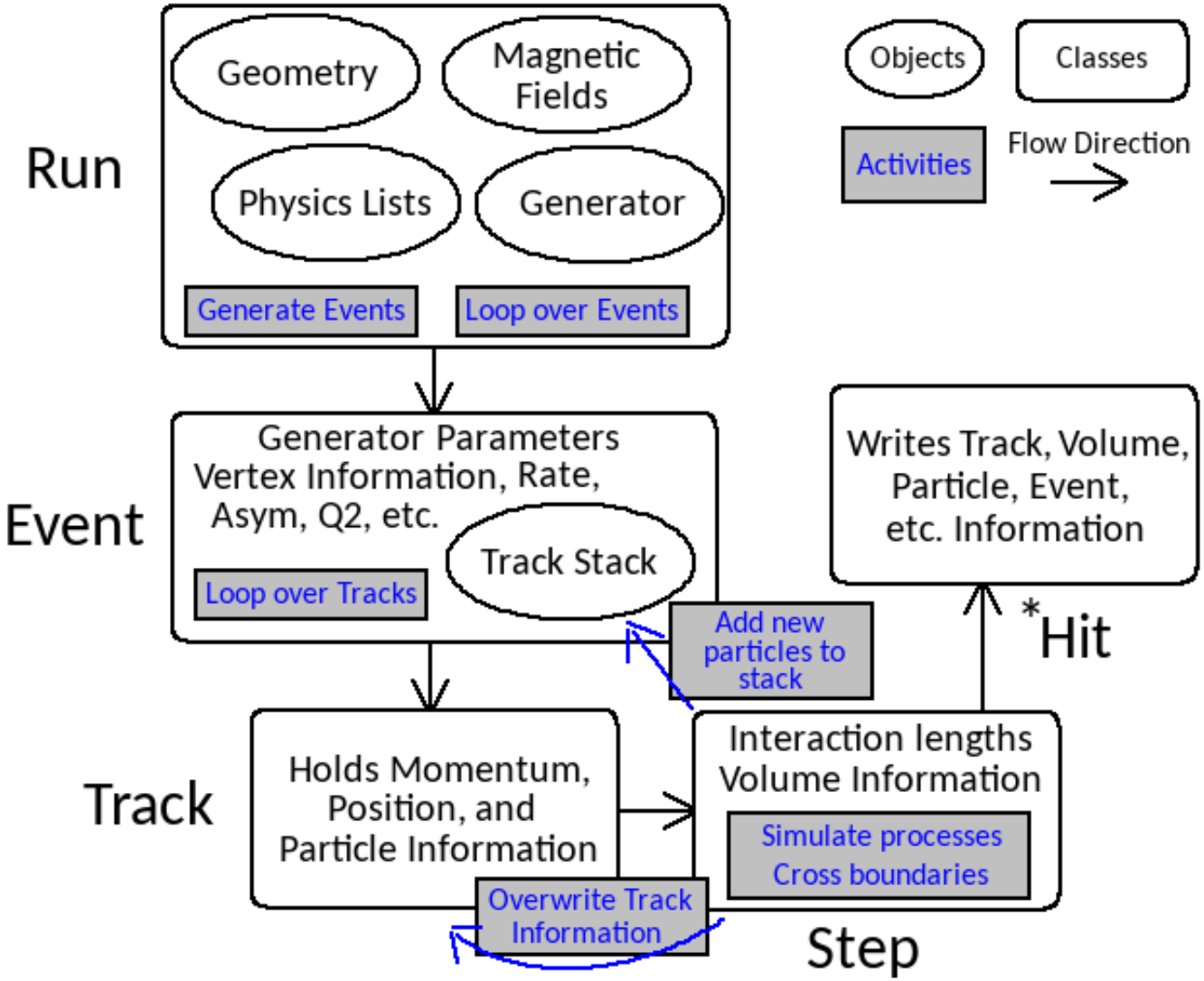
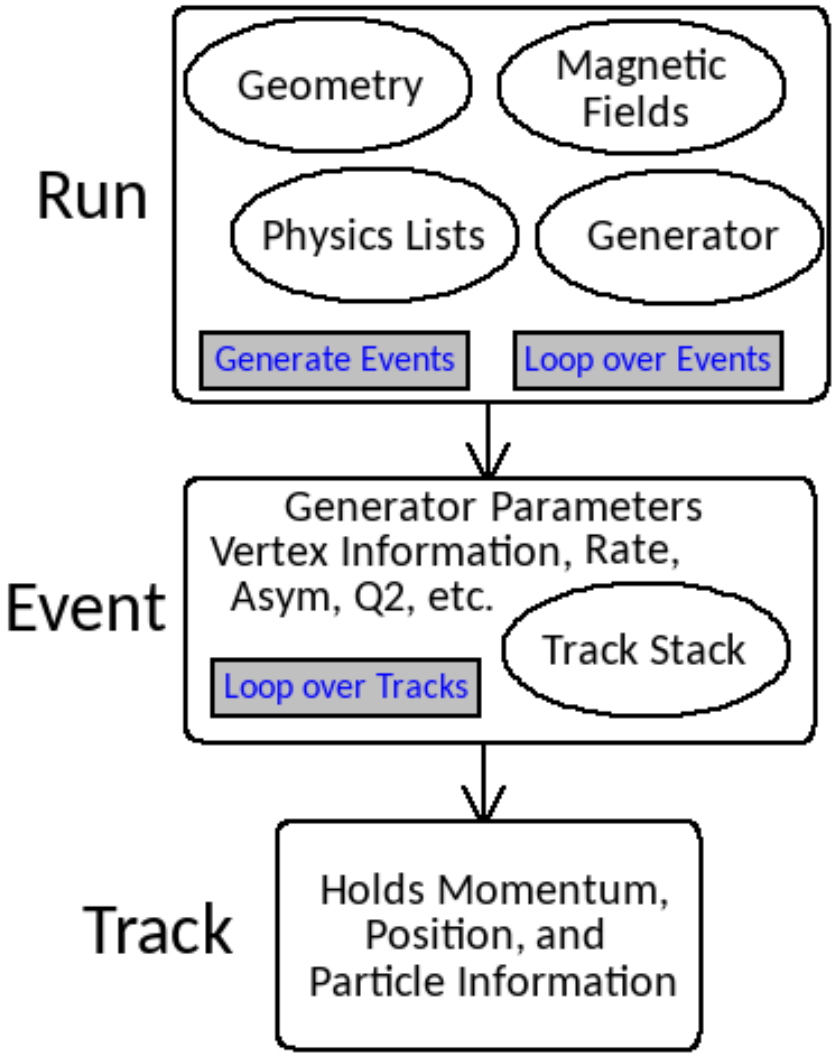


Figure 8: Rough flowchart of Geant4's internal workings (approximate terminology)

Run, Event, and Track Classes



The Run class:

- Loads all the sim components
- If multi-threaded, governs threads
- Calls Generator to produce Events

The Event class:

- Loops over the stacked parent (Generator) and child (Physics List produced) tracks
- Generators' "Rate"/Event assume a Run is the "entire experiment"
- `ROOT` tree entries are per event!

The Track class:

- Simply holds information about a particle being simulated
- Gets updated each new step
- Per event, unique track IDs (`trid`)

Step Class

The Step class - does the most work:

- Physical processes and boundaries
- "Transport" also a step process
  - Showering & B-field interaction are "Transport"
- Updates the Track object
- New particles (secondaries) pushed to stack

Things to be aware of:

- Hit = per Step (unless specified*)
- Trajectory features can be turned on, which store data per step
- Step limiter "physics list" exists, allows specific max step length
  - Useful for high-res trajectories in B-field

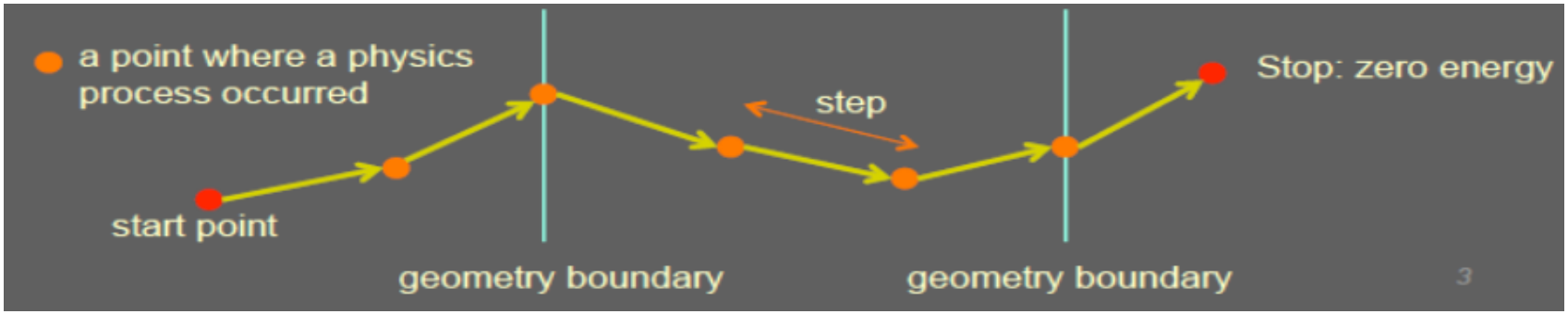


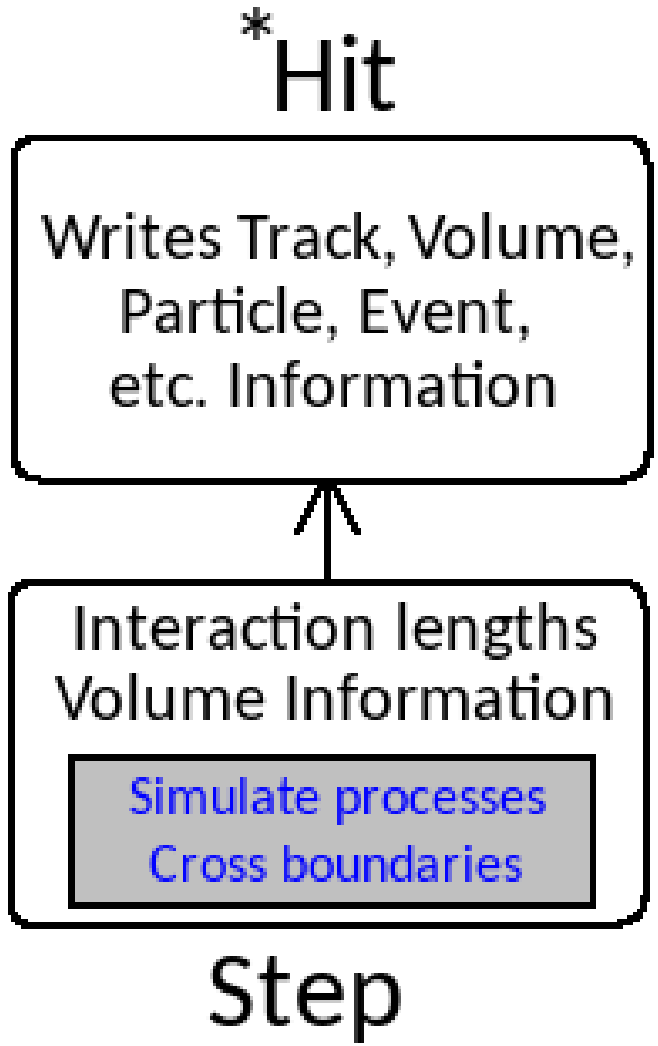
Figure 9: Life cycle of a single track

Hit Class

The Hit class - where choices matter most:

- Stores data when Step is in "sensitive" volumes (= sensitive detector = SensDet)
- *Available sensitivity types include:
  - Not sensitive (if not declared a SensDet)
  - Only primary particles (default SensDet)
  - Include secondaries
  - Include low energy neutrals
  - Include optical photons
  - Only store boundary crossings
  - [Your new condition here]
- Sensitivities are assigned within geometry
  - Can be changed easily in macro
  - Parallel world geom ideal for "boundary"
- Sum class exists too = integral over Hits

Double check functionality meets your needs



How do we run a simulation?

Stony Brook University

To run `remoll`:

- Run the `remoll` executable
- Give it a macro file as a command line argument
- Let it run, it's that easy

How do we visualize a simulation?

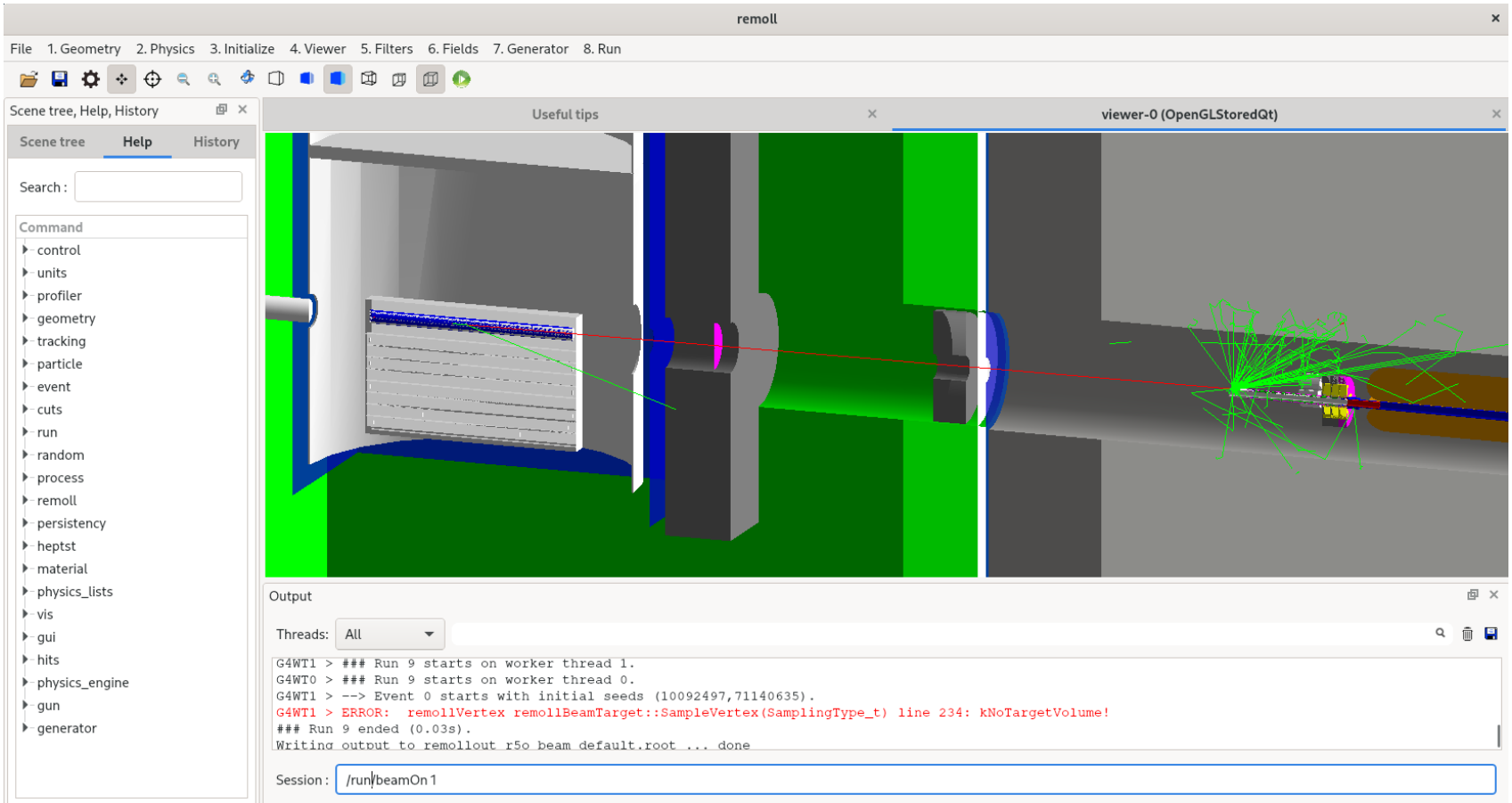


Figure 10: `remoll` interactive GUI example. A beam-generator event electron (red track) scattering through the target (blue tube), hitting a collimator (white/red) and showering gammas (green tracks) into shielding blocks (grey).

How do we visualize a simulation? (cont.)

Stony Brook University

To visualize `remoll` geometry and view simulated events:

1. Run the `remoll` executable
   - No macro file as argument this time
   - A blank GUI with a command line will open
2. Execute: `/control/execute [path to visualization macro]`
   - Use the mouse to interact with the geometry viewer
3. Execute: `/run/beamOn/ [Number of events to simulate]`
   - Over-network, your graphics card, and macro details all matter

How do we analyze a simulation?

Stony Brook University

To analyze simulation outputs we can use `ROOT`:

- Tree "T" contains one entry per each event simulated
- Each entry may contain many hit, part (particle), and others
  - Tagged by track id (trid) and mother track id (mtrid) per Event
- `T->Draw("[selections]","[cuts]","[draw options]")` cuts must be chosen carefully
- Explicitly looping over entries' hits, etc. requires remollTypes.hh structs passed into `T->SetBranchAddress()`
- You can write your own scripts or compiled programs

Conclusion

Stony Brook University

- The `remoll` source code and macros may look intimidating
- But, they combine to give you control over the 4 key ingredients:
  1. Generating useful physics events
  2. Geometry implementation and omniscient sensitive detector particle monitoring
  3. Physical process selection and optimization
  4. Obtaining and analyzing useful data
- Start from those considerations and things should flow naturally
- Ask questions! Join the JLab #moller_simulation SLACK Channel (jlab12gev.slack.com) and moller_simulation email list

Thanks!

Advantages of Multi-Threading

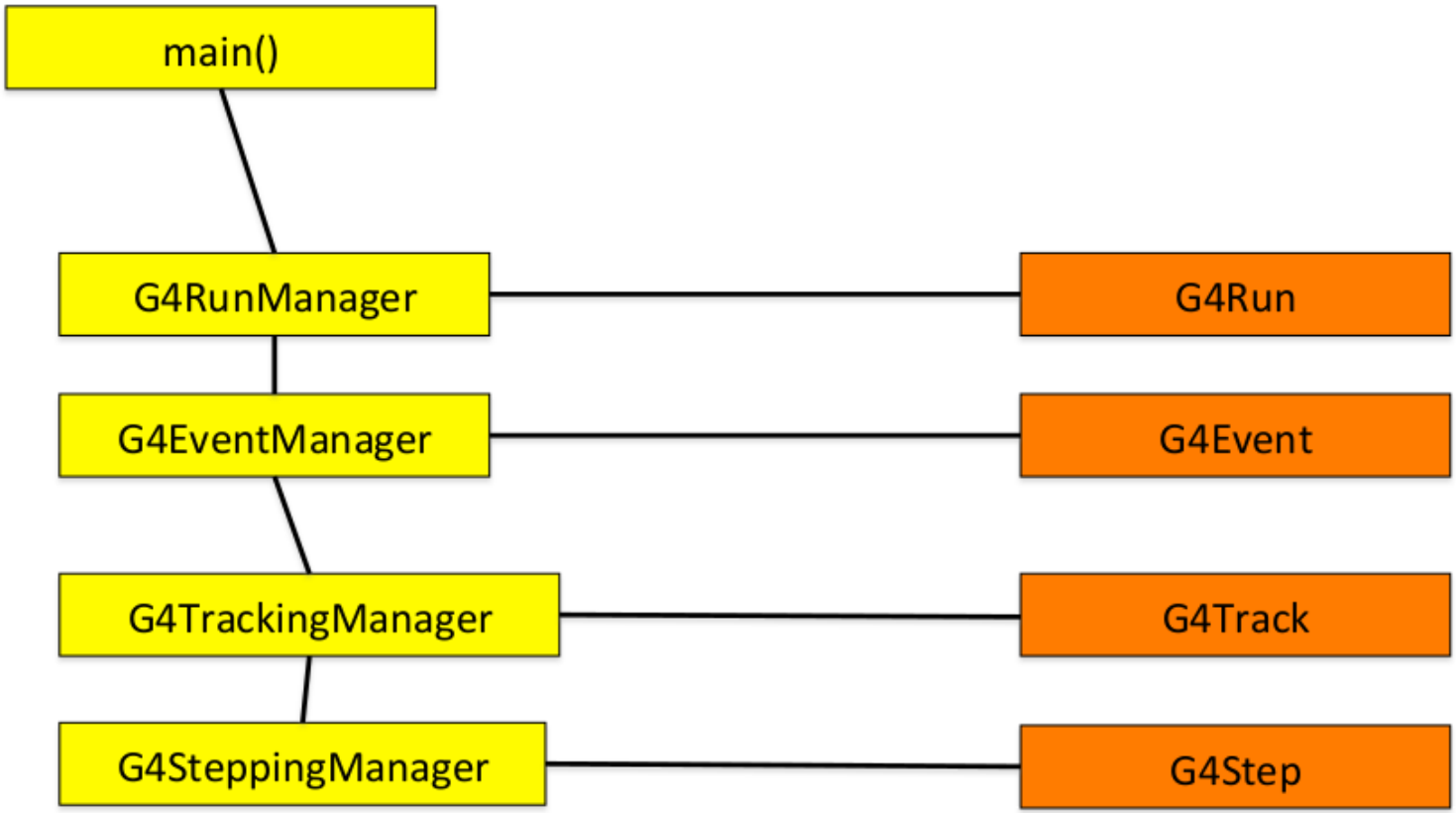


Figure 11: In sequential mode, each job will spend time loading all of the components into memory on each independent core.

Advantages of Multi-Threading (cont.)

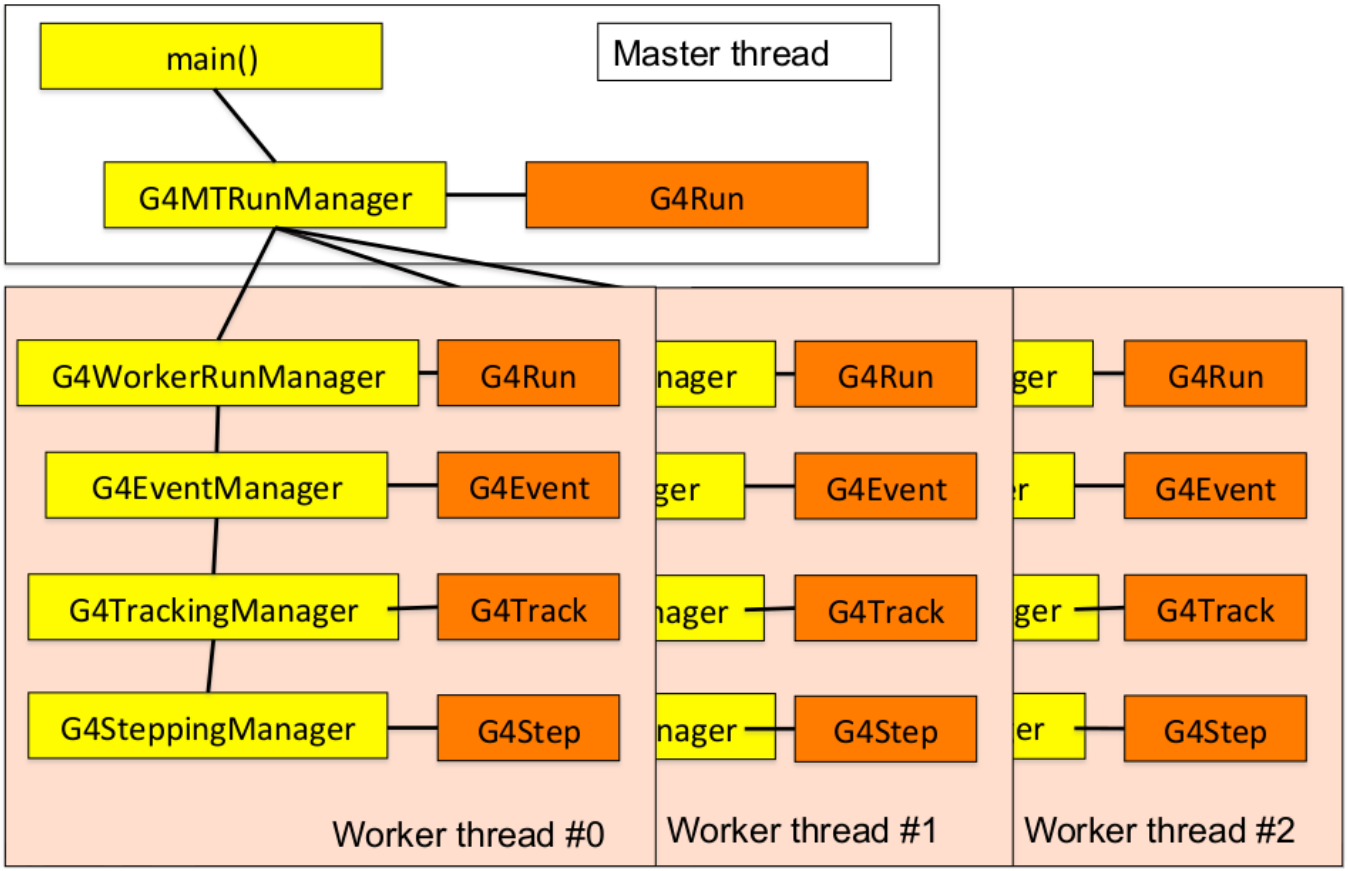


Figure 12: In multi-threading mode, each job will spend time loading all of the components into memory only once, and run independent event loops on each core.

Advantages of Multi-Threading (cont.)

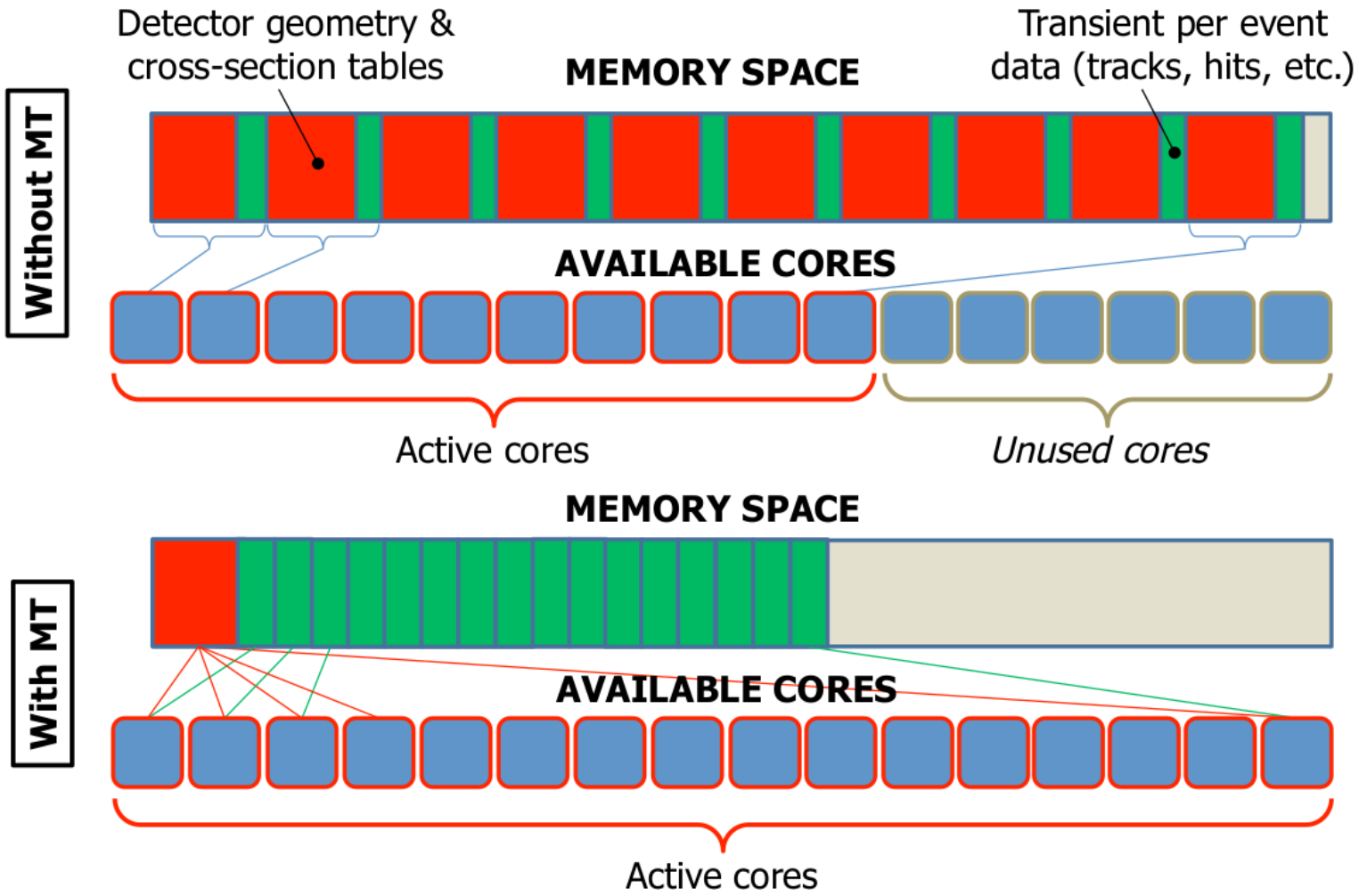


Figure 13: Multi-Threading separates shared, static memory from independent event loops. This saves time per job during setup and allows for optimal memory use in many-core HPC systems.

Building Remoll

Stony Brook University

After obtaining the `remoll` source code (from GitHub)

- `$ module load cmake 3.5.0` (or higher)
- `$ source [/path/to/]geant4.10.00.p00/bin/geant4.sh`
  (Geant4 version $>=$ 4.10.06 recommended)
- `$ source [/path/to/]root-6.0.0-build/bin/thisroot.sh`
  (ROOT version $>=$ 6.10.22 recommended)

Build remoll

- `$ mkdir build`
- `$ cd build`
- `$ cmake ../`
- `$ make`
- `$ make install`
- `$ cd ../`

Initialize Analyses

- `$ mkdir analysis/build`
- `$ cd analysis/build`
- `$ cmake ../`
- `$ make`
- `$ make install`
- `$ cd ../../`

Building Remoll (cont.)

Stony Brook University

You can run simulations from any location provided the following conditions are satisfied:

- The 'remoll' executable is in the PATH,
- The 'libremoll.so' library is in the LD_LIBRARY_PATH,
- The graphical user interface and visualization macros are accessible from the current directory.

If you are in the directory into which you cloned the source code repository, you can run the simulation as 'build/remoll'.

If you have run 'make install' and 'remoll' is now installed in a directory in your PATH, you can run the simulation in batch mode from any directory.

Explicit Macro Descriptions

Stony Brook University

Here is a minimum example macro to get you started:

- /remoll/geometry/setfile geometry/mollerMother.gdml
- /remoll/parallel/setfile geometry/mollerParallel.gdml
- /remoll/physlist/parallel/enable
- /run/initialize
- /control/execute macros/load_magnetic_fieldmaps.mac
- /remoll/oldras false
- /remoll/rasx 5 mm
- /remoll/rasy 5 mm
- /remoll/evgen/set moller
- /remoll/evgen/thcommin 30.0 deg
- /remoll/evgen/thcommax 150.0 deg
- /remoll/beamene 11 GeV
- /remoll/beamcurr 65 microampere
- /control/execute macros/kryptonite.mac
- /run/beamOn 100

Explicit Macro Descriptions (cont.)

Stony Brook University

Descriptions:

- /remoll/geometry/setfile geometry/mollerMother.gdml
  - This sets the Physical world geometry file
  - `./geometry/` can be a symbolic link
- /remoll/parallel/setfile geometry/mollerParallel.gdml
  - This sets the Parallel world geometry file
  - Intended to hold idealized detector volumes
  - These are always boundary only volumes
  - Permits overlap with the physical world volumes
- /remoll/physlist/register QGSP_BERT_HP
  - Explicitly sets which physics list to run with
  - The "_HP" part of this indicates a list with a high-energy physics (HP/HEP) optimization
  - Default physics list (when none declared here) = QGSP_BERT
- /remoll/physlist/parallel/enable
  - The Parallel world implementation is technically a physics list
  - Must be set to active to utilize (may become default active soon)

Explicit Macro Descriptions (cont.)

Stony Brook University

- /remoll/physlist/steplimiter/enable
  - Physics list for declaring a maximum transport step length
- /remoll/geometry/userlimits/usermaxallowedstep logicUpstream 10*cm
- /remoll/geometry/userlimits/usermaxallowedstep logicDownstream 10*cm
  - Specifically declare a steplimiter max for a specific logical volume
- /run/initialize
  - This formally loads the physics lists and geometries
  - As a result, all of those must be chosen before executing this
  - But also, this must be executed before modifying their contents
- /remoll/SD/disable_all
  - Turns off sensitivity for all sensitive detectors (to save time/disk)
- /remoll/SD/enable 28
  - Turns on sensitivity for one (number 28 here, the main-det plane)
- /remoll/SD/detect 28 3
  - Set detector 28 to sensitive detector type 3
- /remoll/SD/enable_range 50000-59999
  - Enable the detectors from 50000 to 59999

Explicit Macro Descriptions (cont.)

Stony Brook University

- /remoll/tracking/set 0
  - This command determines particles to kill (not even simulate)
    - -> 0 : Track primary electrons only
    - -> 1 : Track primary electrons and optical photons only
    - -> 2 : Track all particles except optical photons
    - -> 3 : Track all particles (Default Case)
- /tracking/storeTrajectory 1
  - Activates the built in "G4TrajectoryContainer" class
  - Allows storing trajectories in the `part` branch
  - Currently hardcoded to only store primaries trajectories (for envelopes)
  - Alternatively could get tracking information with parallel world dets
  - Trajectory points come from steps, and steps bypass
- /remoll/printgeometry true
  - Prints geometry info to the screen/output text file
- /control/execute macros/load_magnetic_fieldmaps.mac
  - Executes a separate macro containing default magnetic field maps:
    - -> /remoll/addfield map_directory/V2U.1a.50cm.parallel.txt
    - -> /remoll/addfield map_directory/V2DSg.9.75cm.parallel.txt

Explicit Macro Descriptions (cont.)

- /remoll/oldras false
  - The "Old Raster" is a gaussian spread, with $\vec{P}/\vec{X}$ angle correlation
  - Default: oldras = "true" then it has MOLLER angle correlation
  - If "false" then it is a flat raster, from the raster coils' in Hall A
- /remoll/rasx 5 mm
- /remoll/rasy 5 mm
  - Widths of a raster strafe on the z-midpoint face of the target
- /remoll/evgen/set moller
  - Choses the generator to use (beam, moller, epelastic, etc.)
- /remoll/evgen/thcommin 30.0 deg
- /remoll/evgen/thcommax 150.0 deg
  - Sets the min and max angles for the set generator
- /remoll/beamene 11 GeV
  - Sets the incoming beam energy for the generator
- /remoll/beamcurr 65 microampere
  - Multiplies all output "rate" data. Defaults to 1uA (uA has units of 1)

Explicit Macro Descriptions (cont.)

Stony Brook University

- /control/execute macros/kryptonite.mac
  - Volumes with materials declared as "kryptonite" kill all particles
  - Serves as a shortcut to bypass expensive showering in collimators, etc.
  - Calls another macro which contains the kryptonite setup commands:
    -> /remoll/kryptonite/enable
    -> /remoll/kryptonite/add VacuumKryptonite
    -> /remoll/kryptonite/add Tungsten
    -> /remoll/kryptonite/add Copper
    -> /remoll/kryptonite/add Lead
    -> /remoll/kryptonite/add CW95
    -> /remoll/kryptonite/list
- /process/list
  - Shows which processes are available in the physics list
- /remoll/seed 123456
  - Declares an explicit seed
  - Default behavior is to generate a random number seed for each job
  - Manually setting a seed allows for exactly rerunning simulations

Explicit Macro Descriptions (cont.)

Stony Brook University

- /remoll/filename remollout.root
  - Specify the output ROOT file name
- /remoll/target/print
  - Print out target volume information
  - The target is handled differently than normal geometry
  - Dependent on generator, target volume may apply radiative corrections
- /run/beamOn 100
  - Perform 100 events
  - The "rate" variable assumes these N events are the "entire experiment"
  - When adding together outputs from M parallel jobs, divide "rate" by M

Some Examples of Analyzing Events

Stony Brook University

- `reroot` contains remollTypes.hh struct definitions already
- CMakeLists.txt can be edited to compile your own analysis scripts
- `T->Draw()` will cut on Entry and Hit (or Part, etc.), but mixing Hit and Part (etc.) cuts at the same time will cause confusion

`remoll` is a Git project, hosted on GitHub. This comes with some standard best-practices:

- Generate a personal fork of the JeffersonLab/remoll repository
- Create your own branch on your personal fork for development work
- Branch off of the "develop" branch, if you can
- Create a Pull Request (PR) on the GitHub website between your fork branch and the JeffersonLab "develop" branch
- Never commit directly into the main "master" branch or "develop" $\rightarrow$ always do a PR
- When downloading updates to your local copy (on your computer) from the remote (the GitHub website copy), always fetch and fastforward, to avoid unnecessary conflicts or merge commits

Example series of commands when polishing off development work:

- `git remote show origin`
- `git branch -a`
- `git fetch`
- `git pull --ff-only`
- `git status` (can pass as an argument the folder of file of interest to simplify outputs)
- `get log -N` (N = number of recent commit logs to read)
- `git diff [some local file that has changes]`
- `git blame [some local file you want change history of]`
- `git add [some local file that has changes you want to push]`
- `git commit --author=[your github username]`
- `git push -u [origin name] [your branch name]` (the origin names - typically "origin" or "remote" - are set in .git/config)

Some Examples of Simulation Workflows

Stony Brook University

Recent examples:

- Envelopes (and step limiter+trajectories vs. parallel world options for precision vs. speed/size)
- Optical photons and geometry optimization (doing it as external parameter scan loops or internal rasterized parameter spread, phase space brute force vs. bayesian update techniques)
- Factorizing a complex problem (splash back from shielding or detectors, light guide optical properties beam-test benchmarking at Mainz with tube setup splitting cherenkov and scintillation vs. simulations with full detector, light guide backgrounds with rate-weighted flux)
- Radiation shielding optimization and dose estimates