A Customizable GeantV Calorimeter Application

Ryan Schmitz -- University of Minnesota

Advised by Sofia Vallecorsa, Andrei Gheata

(honorable mention to Mihaly Novak)

Status: Early June

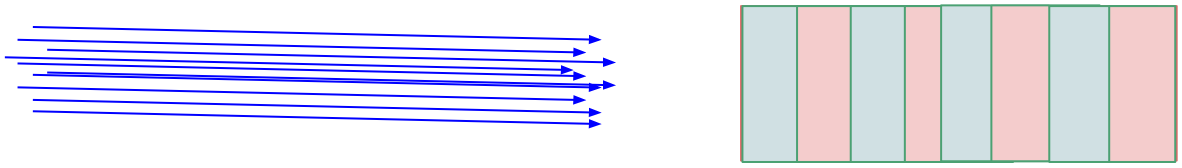
A physics simulation toolkit must support two fundamental user-defined inputs:

Incident Particles

- Type
- Energy
- Direction

Detector

- Materials
- Geometry



Particles
(e-, e+, p, etc.)

Detector
(Calorimeter, Tracker,
CMS/Atlas geometry, etc.)

Status: Early June

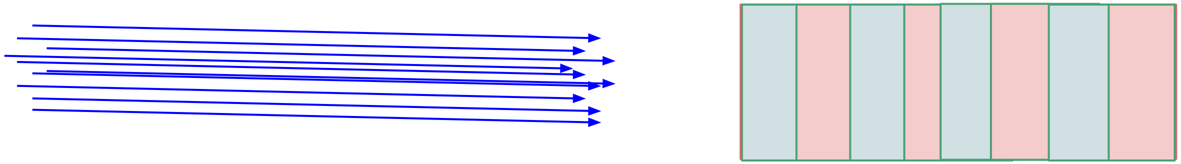
A physics simulation toolkit must support two fundamental user-defined inputs:

Incident Particles

- Type
- Energy
- Direction

Detector

- Materials
- Geometry



Particles
(e-, e+, p, etc.)

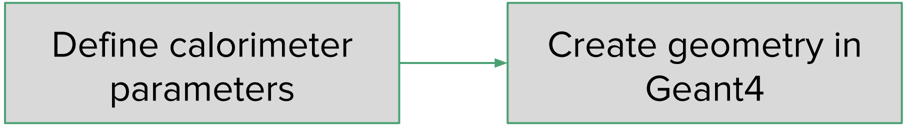
Detector
(Calorimeter, Tracker,
CMS/Atlas geometry, etc.)

GeantV can’t do this by itself!
(circa June 2017)

June Workflow: Defining and testing a calorimeter

Define calorimeter parameters

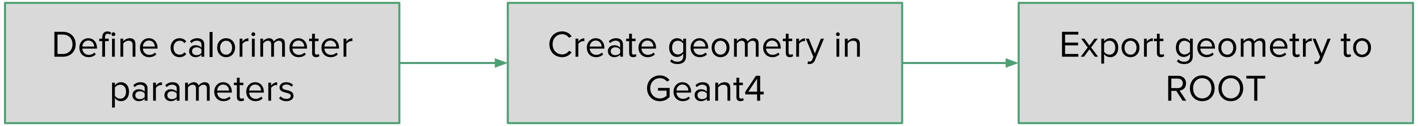
June Workflow: Defining and testing a calorimeter



Define calorimeter parameters

Create geometry in Geant4

June Workflow: Defining and testing a calorimeter

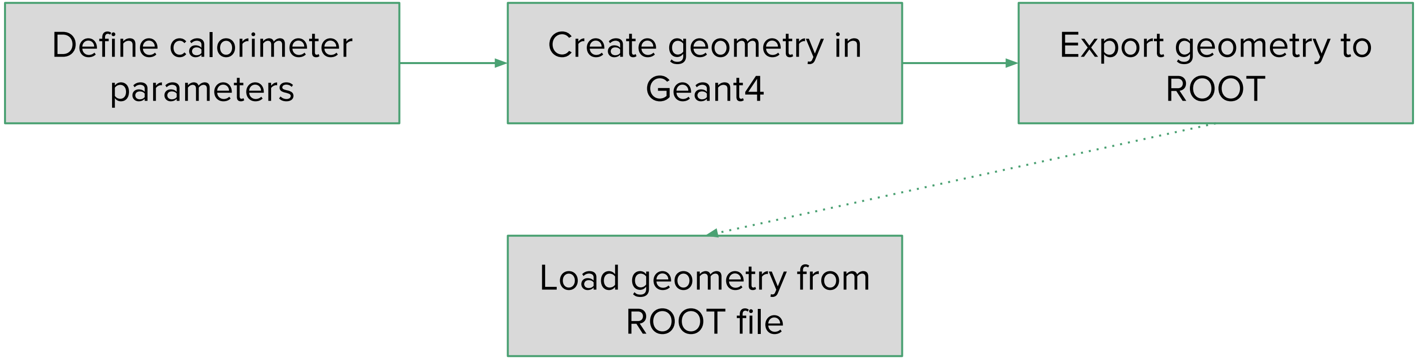


Define calorimeter parameters

Create geometry in Geant4

Export geometry to ROOT

June Workflow: Defining and testing a calorimeter



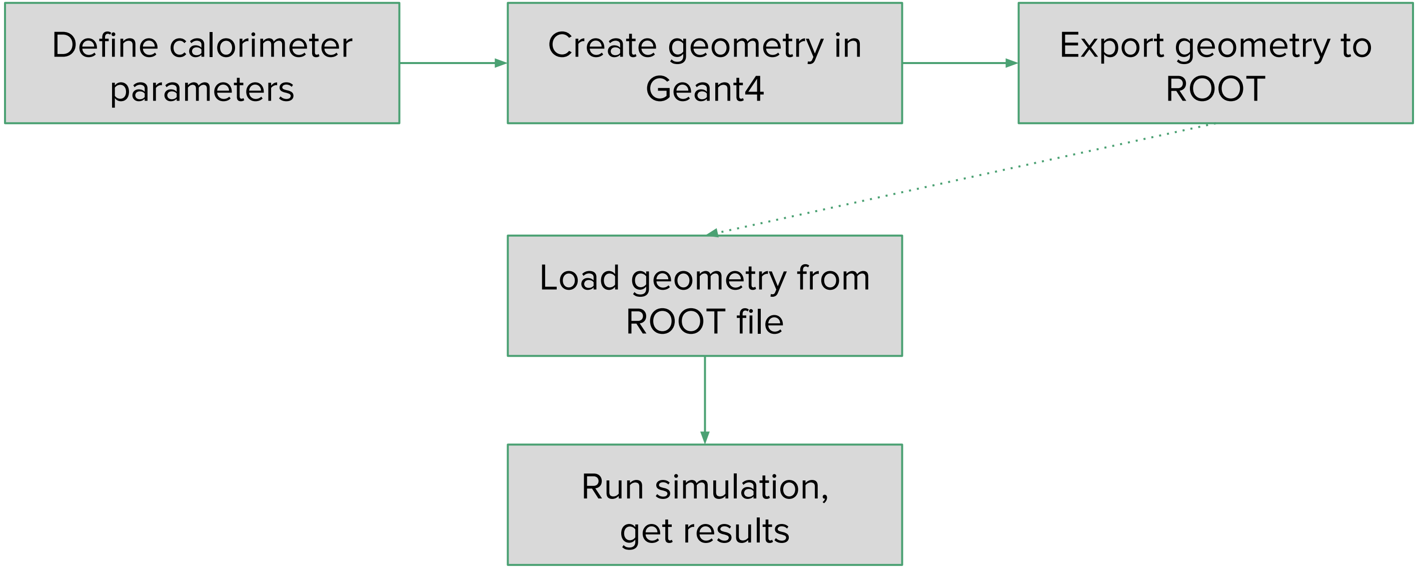
Define calorimeter parameters

Create geometry in Geant4

Export geometry to ROOT

Load geometry from ROOT file

June Workflow: Defining and testing a calorimeter



Define calorimeter parameters

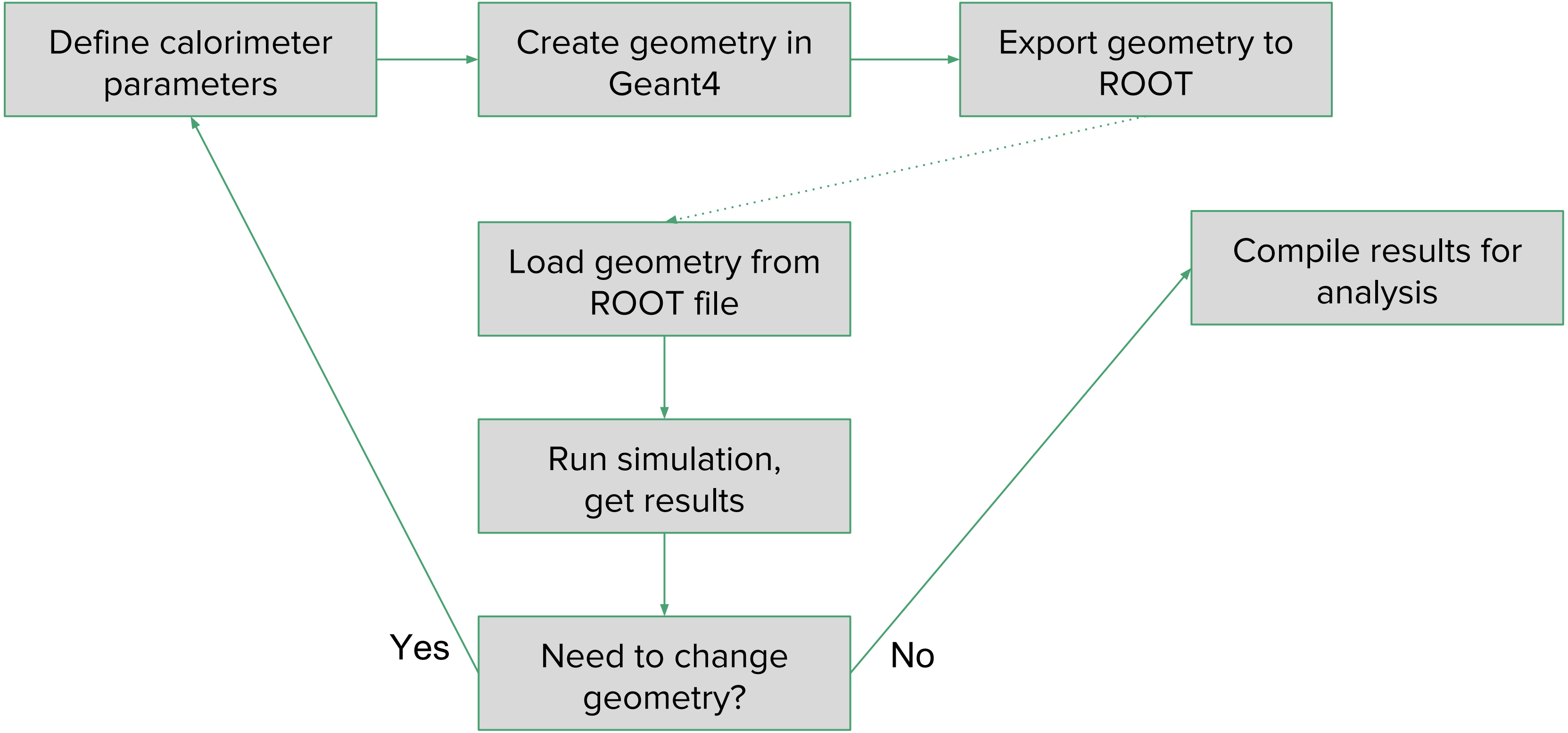
Create geometry in Geant4

Export geometry to ROOT

Load geometry from ROOT file

Run simulation, get results

June Workflow: Defining and testing a calorimeter



Define calorimeter parameters

Create geometry in Geant4

Export geometry to ROOT

Load geometry from ROOT file

Run simulation, get results

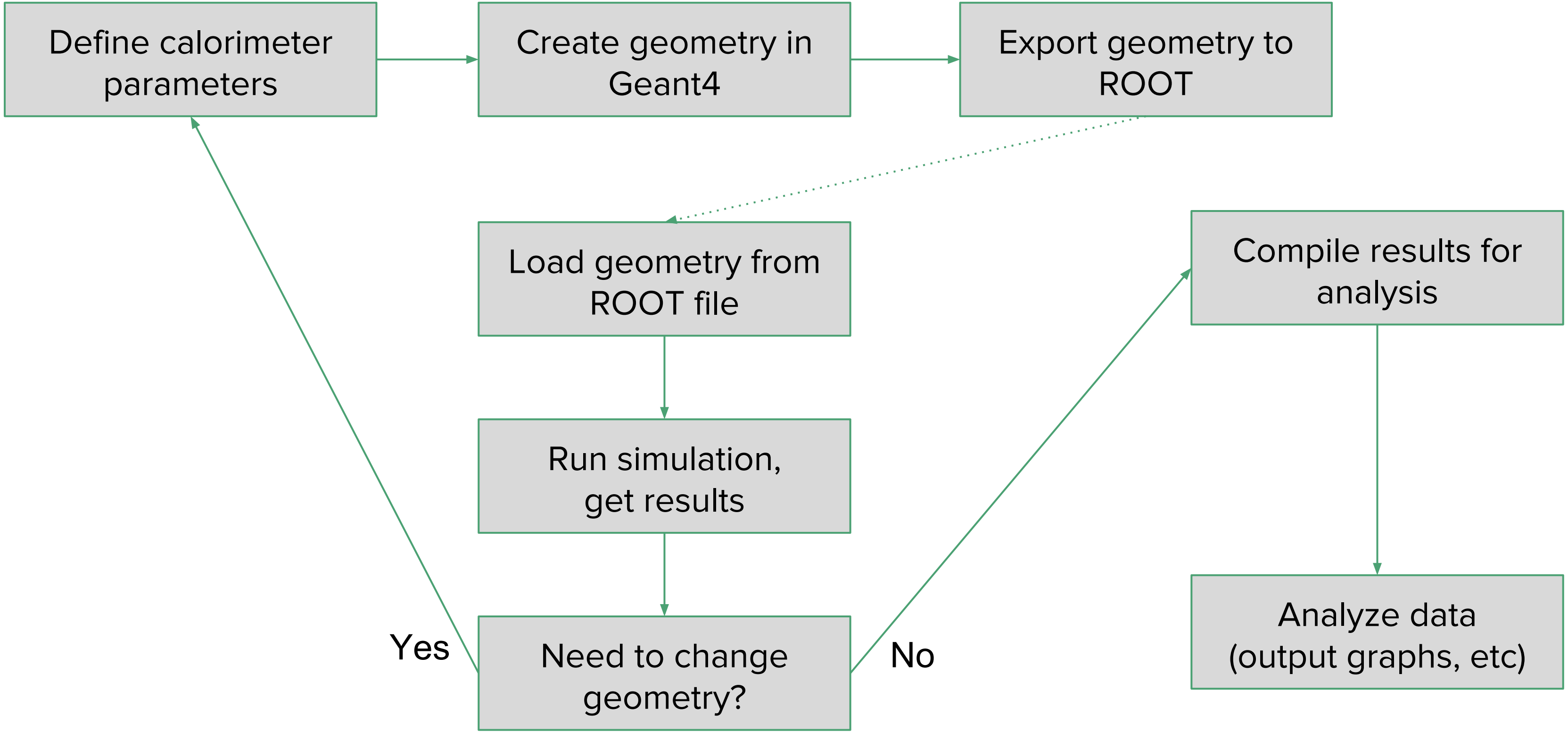
Need to change geometry?

Yes

No

Compile results for analysis

June Workflow: Defining and testing a calorimeter



Define calorimeter parameters

Create geometry in Geant4

Export geometry to ROOT

Load geometry from ROOT file

Run simulation, get results

Need to change geometry?

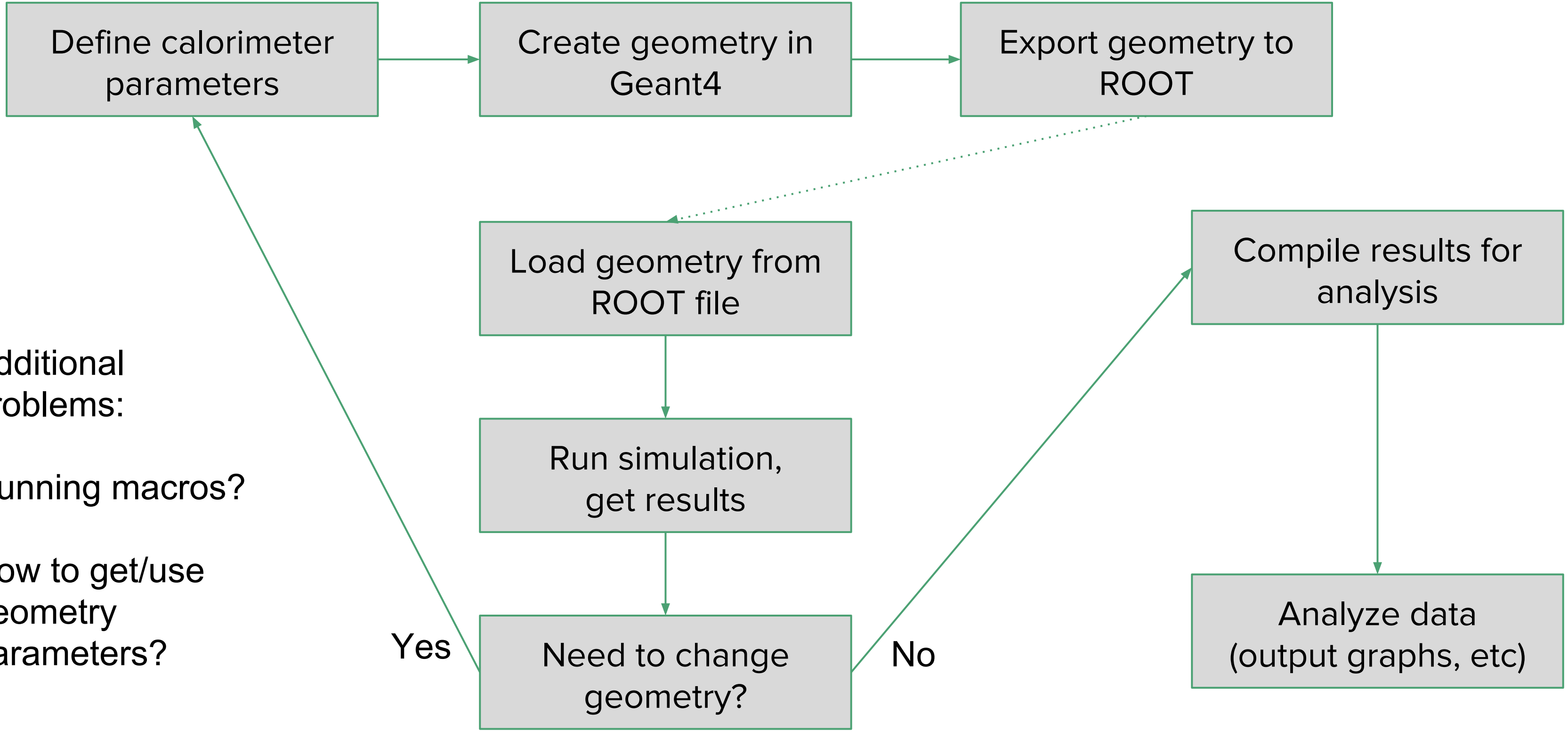
Yes

No

Compile results for analysis

Analyze data (output graphs, etc)

June Workflow: Defining and testing a calorimeter


Define calorimeter parameters
Create geometry in Geant4
Export geometry to ROOT
Load geometry from ROOT file
Run simulation, get results
Need to change geometry?
Yes
No
Compile results for analysis
Analyze data (output graphs, etc)

Additional problems:

Running macros?

How to get/use geometry parameters?

Project Goals:

- Provide an implementation of a **user-defined detector**, including geometry and materials

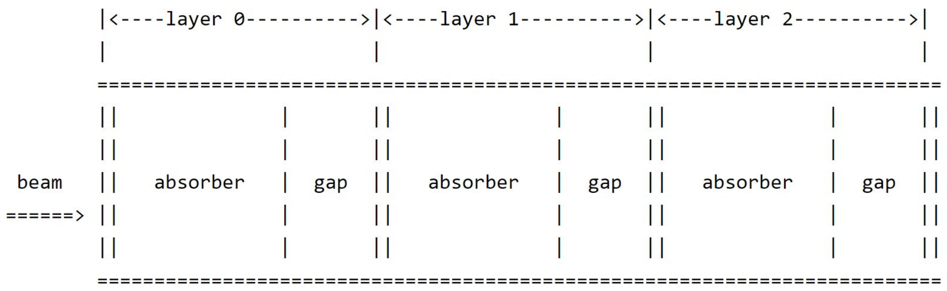
- Allow for this detector to be **fully customizable** without need to recompile

- Use “**real physics**” (as opposed to tabulated physics)

- Implement **new data structures** which allow for outputs to be associated with individual primaries

GeantV Calorimeter: User Detector Construction

Goal: Develop an example detector construction which can be easily grasped by users familiar with Geant4



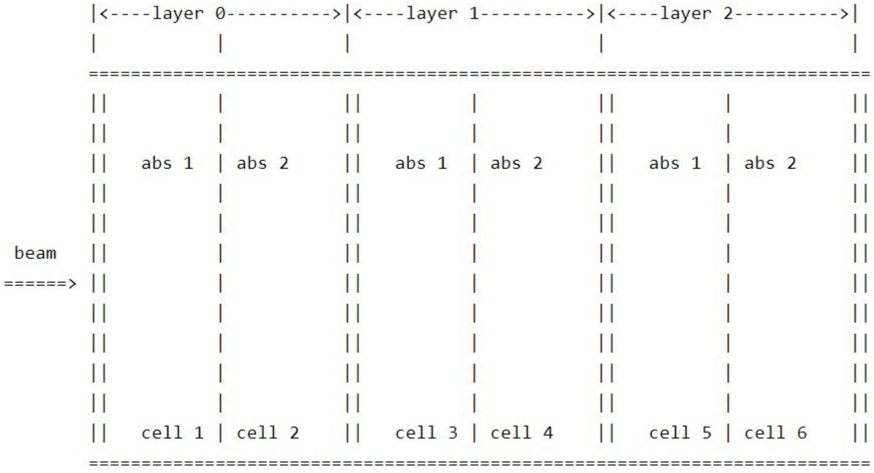
```
        |<----layer 0---------->|<----layer 1---------->|<----layer 2---------->|
        |                       |                       |                       |
        ==========================================================================
        ||              |       ||              |       ||              |       ||
        ||              |       ||              |       ||              |       ||
 beam   ||   absorber   |  gap  ||   absorber   |  gap  ||   absorber   |  gap  ||
======> ||              |       ||              |       ||              |       ||
        ||              |       ||              |       ||              |       ||
        ==========================================================================
```

- Detector construction style inspired by Geant4 TestEm3
  - Includes a CreateMaterials() and CreateGeometry() method, following Geant4 convention
- VecGeom used as basis for geometry creation and placement
  - UnplacedBox = G4Box
  - LogicalVolume = G4LogicalVolume
  - world->PlaceDaughter() = G4PVPlacement
- Inherits from base class, so other detector types can be created

GeantV Calorimeter: User Detector Construction

Detector Features:

- Customizable inputs to geometry
  - Number of layers
  - Number of absorbers per layer
  - Absorber properties (thickness, material)
  - World size, Calo YZ cross-section, etc.
- Use of NIST and/or custom materials
- Detector Regions and Production Cuts
- Get/Set methods to receive detector properties as inputs and pass detector information to other classes (e.g. detThickness -> initial particle gun position)



```
        |<----layer 0----------->|<----layer 1----------->|<----layer 2----------->|
        |           |            |                        |                        |
        =============================================================================
        ||          |            ||          |            ||          |            ||
        ||          |            ||          |            ||          |            ||
        ||   abs 1  | abs 2      ||   abs 1  | abs 2      ||   abs 1  | abs 2      ||
        ||          |            ||          |            ||          |            ||
        ||          |            ||          |            ||          |            ||
 beam   ||          |            ||          |            ||          |            ||
======> ||          |            ||          |            ||          |            ||
        ||          |            ||          |            ||          |            ||
        ||          |            ||          |            ||          |            ||
        ||          |            ||          |            ||          |            ||
        ||          |            ||          |            ||          |            ||
        ||   cell 1 | cell 2     ||   cell 3 | cell 4     ||   cell 5 | cell 6     ||
        =============================================================================
```

Current GeantV Calorimeter Workflow

Create geometry in GeantV

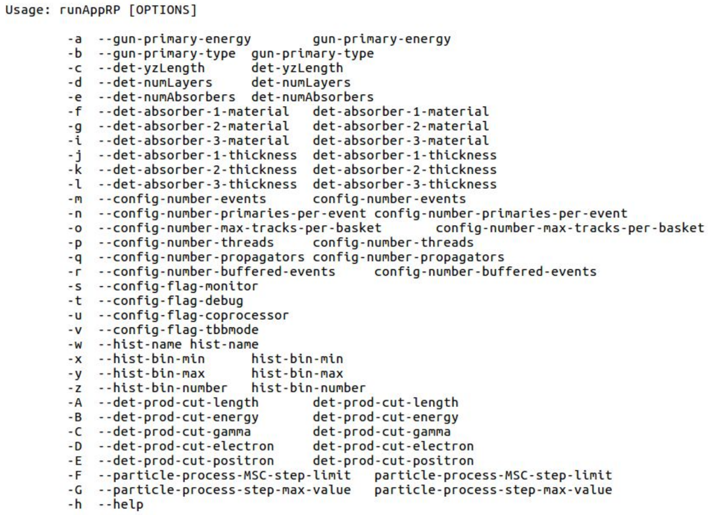
GeantV Calorimeter: Detector Inputs (Messenger)

Goal: Create interface which mirrors a messenger from Geant4, allowing users to set parameters from a macro

Since we don’t have an equivalent to a Geant4 messenger right now, this is handled manually:

- A user can run the application executable directly from terminal, using arguments to set parameters

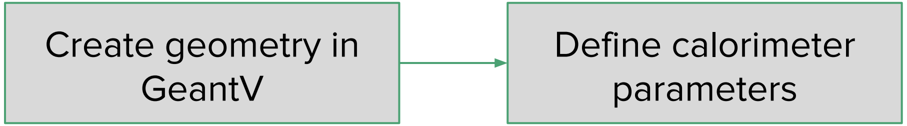
- A user can run an executable macro in which all of the listed arguments may be set beforehand



```
Usage: runAppRP [OPTIONS]

        -a  --gun-primary-energy        gun-primary-energy
        -b  --gun-primary-type  gun-primary-type
        -c  --det-yzLength      det-yzLength
        -d  --det-numLayers     det-numLayers
        -e  --det-numAbsorbers  det-numAbsorbers
        -f  --det-absorber-1-material   det-absorber-1-material
        -g  --det-absorber-2-material   det-absorber-2-material
        -i  --det-absorber-3-material   det-absorber-3-material
        -j  --det-absorber-1-thickness  det-absorber-1-thickness
        -k  --det-absorber-2-thickness  det-absorber-2-thickness
        -l  --det-absorber-3-thickness  det-absorber-3-thickness
        -m  --config-number-events      config-number-events
        -n  --config-number-primaries-per-event config-number-primaries-per-event
        -o  --config-number-max-tracks-per-basket       config-number-max-tracks-per-basket
        -p  --config-number-threads     config-number-threads
        -q  --config-number-propagators config-number-propagators
        -r  --config-number-buffered-events     config-number-buffered-events
        -s  --config-flag-monitor
        -t  --config-flag-debug
        -u  --config-flag-coprocessor
        -v  --config-flag-tbbmode
        -w  --hist-name hist-name
        -x  --hist-bin-min      hist-bin-min
        -y  --hist-bin-max      hist-bin-max
        -z  --hist-bin-number   hist-bin-number
        -A  --det-prod-cut-length       det-prod-cut-length
        -B  --det-prod-cut-energy       det-prod-cut-energy
        -C  --det-prod-cut-gamma        det-prod-cut-gamma
        -D  --det-prod-cut-electron     det-prod-cut-electron
        -E  --det-prod-cut-positron     det-prod-cut-positron
        -F  --particle-process-MSC-step-limit   particle-process-MSC-step-limit
        -G  --particle-process-step-max-value   particle-process-step-max-value
        -h  --help
```

Current GeantV Calorimeter Workflow



Create geometry in GeantV

Define calorimeter parameters

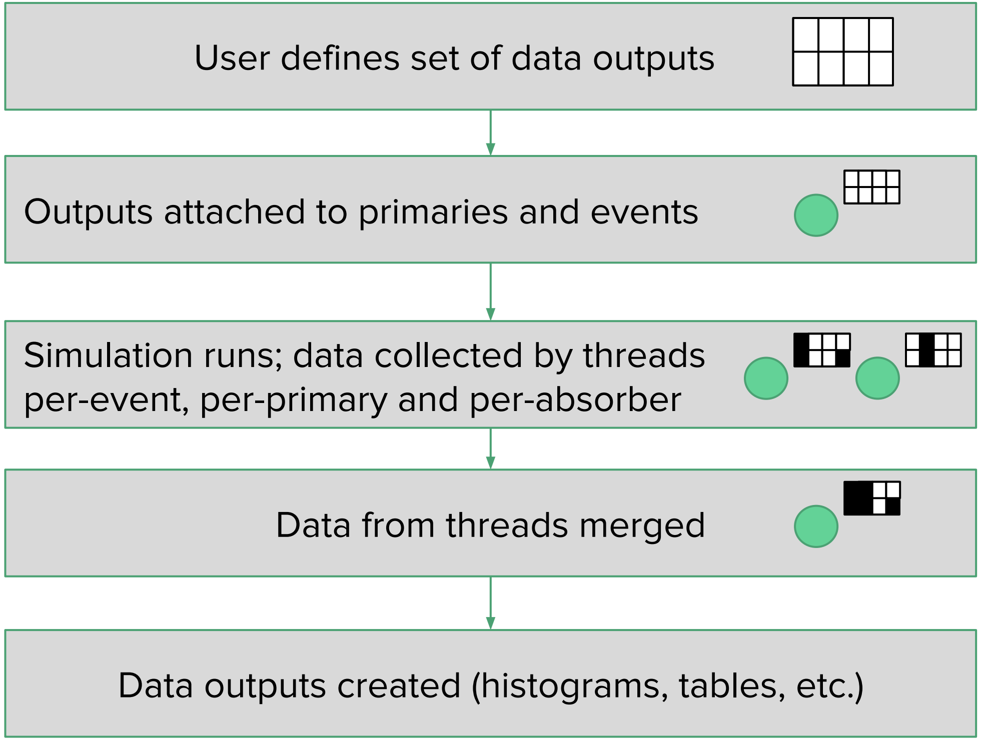
GeantV Calorimeter: Real-Physics Application

Goal: Create application which uses real physics and data structures which assign data to each primary

Derived an example application structure created by Mihaly (TestEm5)

- Adapted this for calorimeter application to allow data assignment to variable number of logical volumes

- Also created “legacy” application which outputs data in the old style (mean outputs only)


User defines set of data outputs

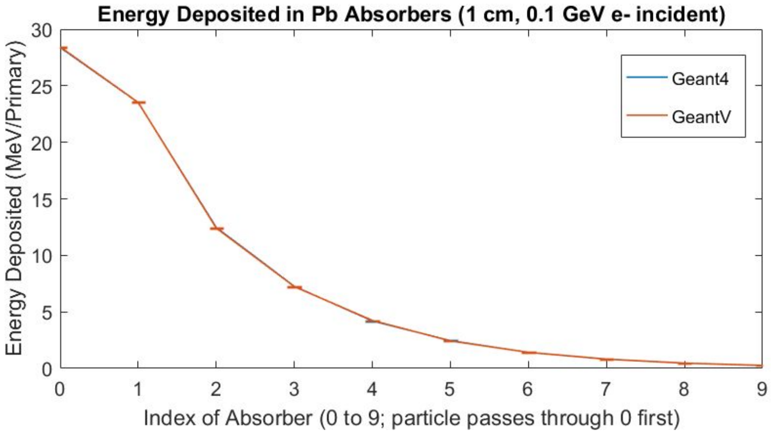
Outputs attached to primaries and events

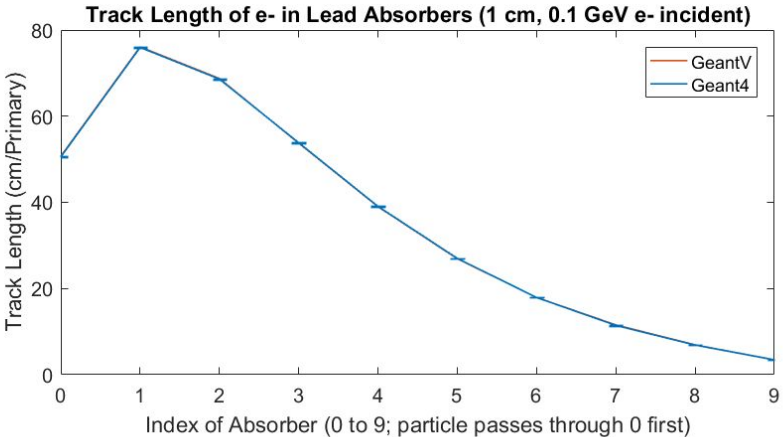
Simulation runs; data collected by threads per-event, per-primary and per-absorber

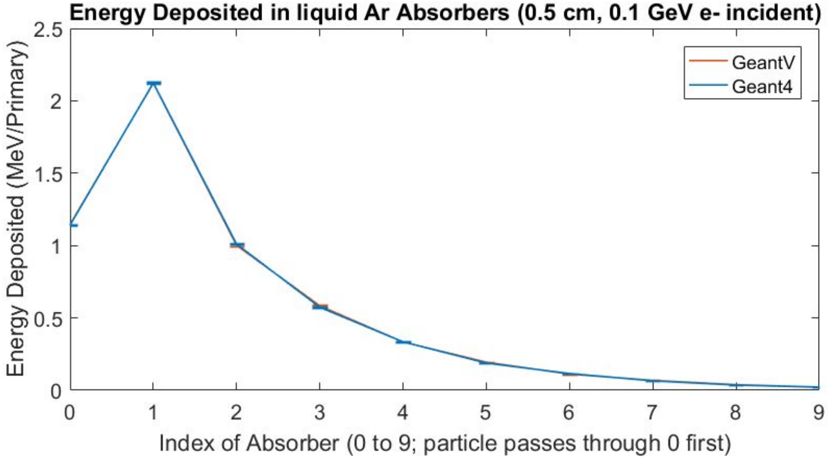
Data from threads merged

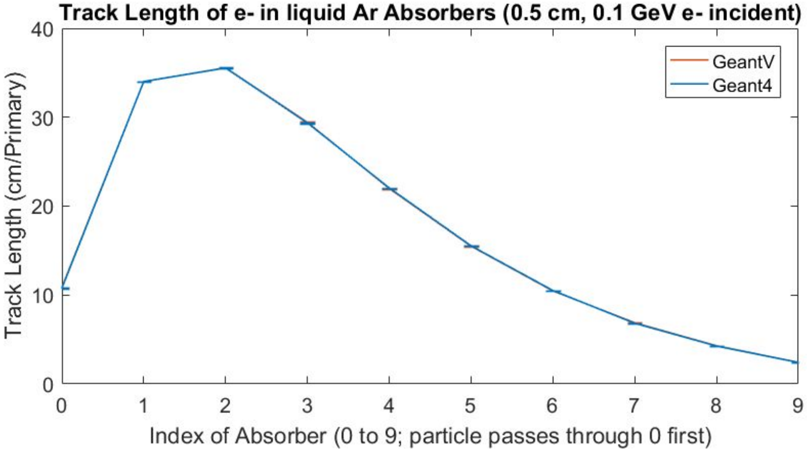
Data outputs created (histograms, tables, etc.)

Output comparison to Geant4 (TestEm3) -- Perfect match

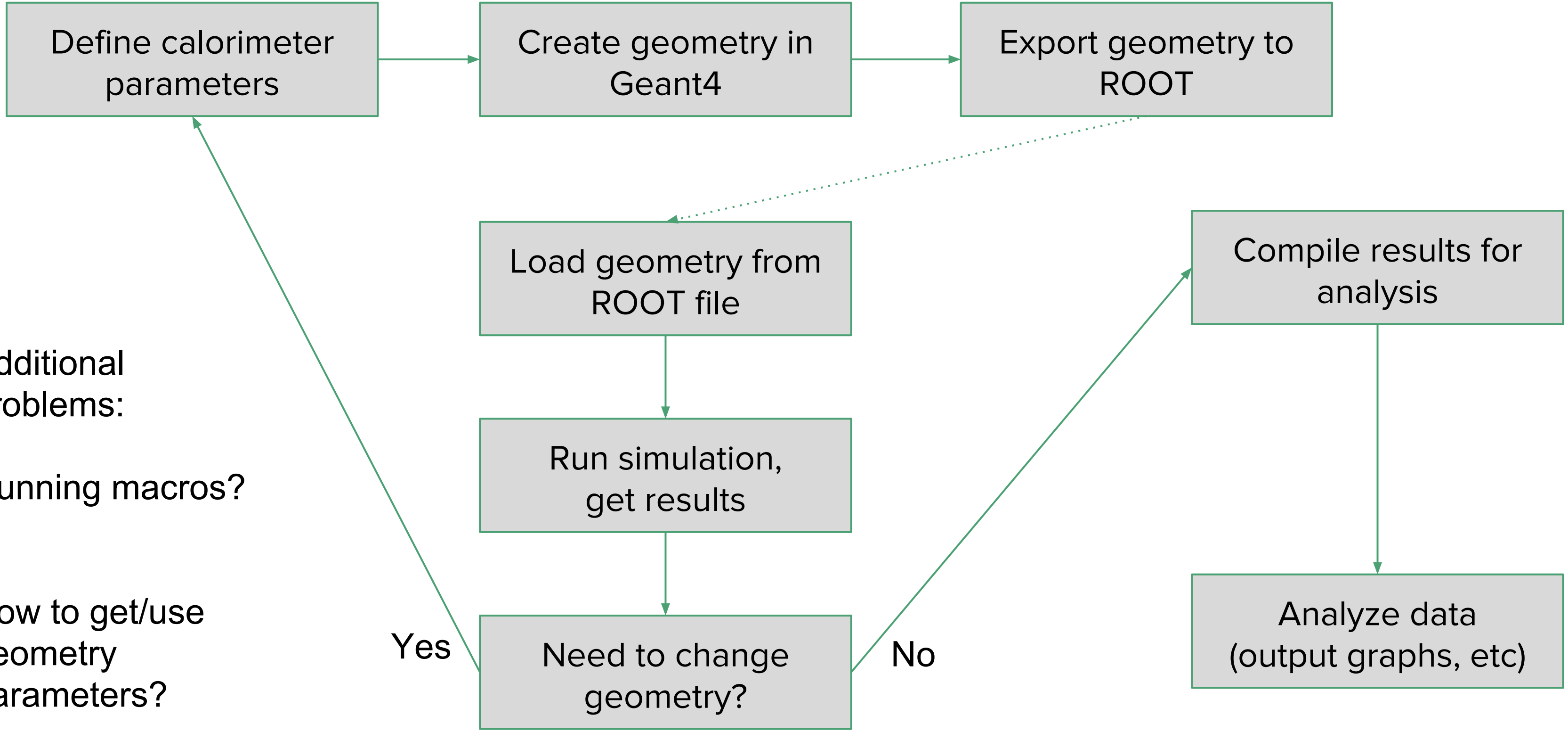

Energy Deposited in Pb Absorbers (1 cm, 0.1 GeV e- incident)


Track Length of e- in Lead Absorbers (1 cm, 0.1 GeV e- incident)


Energy Deposited in liquid Ar Absorbers (0.5 cm, 0.1 GeV e- incident)


Track Length of e- in liquid Ar Absorbers (0.5 cm, 0.1 GeV e- incident)

June Workflow: Defining and testing a calorimeter



Define calorimeter parameters

Create geometry in Geant4

Export geometry to ROOT

Load geometry from ROOT file

Run simulation, get results

Need to change geometry?

Compile results for analysis

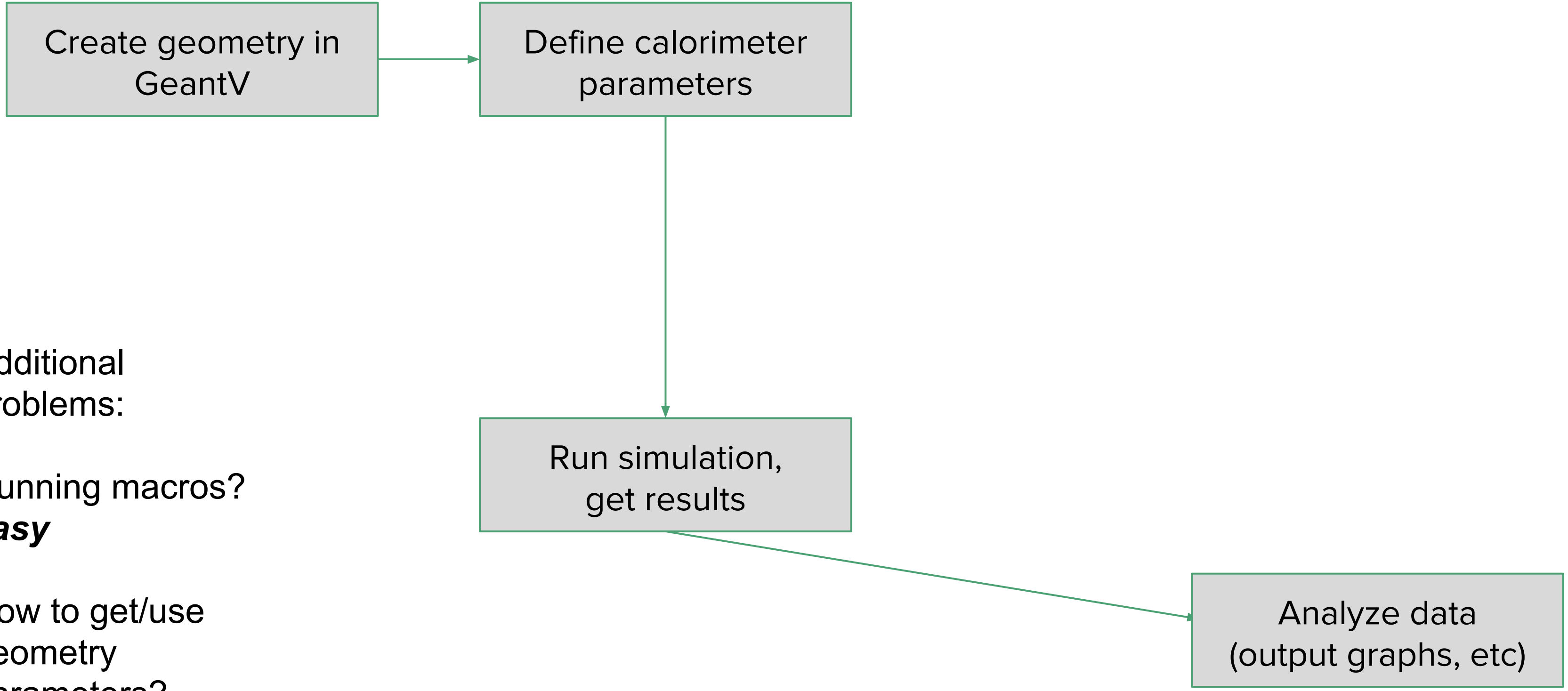
Analyze data (output graphs, etc)

Additional problems:

Running macros?

How to get/use geometry parameters?

Current Workflow: Defining and testing a calorimeter



Create geometry in GeantV

Define calorimeter parameters

Run simulation, get results

Analyze data (output graphs, etc)

Additional problems:

Running macros?
easy

How to get/use geometry parameters?
Use get methods

Summary

With an Alpha release on the horizon, GeantV had an urgent need for the ability to define its own detectors

I have developed a fully customizable, self-contained calorimeter application using a user-defined detector, real physics, and new data structures

This example, with its simplified workflow and user-level syntax similar to Geant4, provides an important tool for teaching new users how to create detectors and applications in GeantV