PDAML Geant4 Project

Ben Wynne (bwynne@cern.ch)

1 Aim

In the last few weeks you've seen examples of simple detector design, the reconstruction of particle properties from simulated data outputs, and validation of that data by comparing results with the truth. For this project the goal is to connect these ideas together and design your own experiment.

You should complete the following steps, shown with approximate marking weighting in percent:

Step 1 - Geant4 simulation (20%)

Create a simple particle detector simulation in Geant4. You may use code provided in the workshops as a starting point for your simulation, but using these examples unmodified is not sufficient. Ideally you should include two different detector structures, and each structure could contain multiple Geant4 solids. For example, one detector structure could be a sampling calorimeter like the one in GeantExample2.

Step 2 - Data reconstruction (20%)

Your detector should be capable of measuring the energy or momentum of particles over a range of two orders of magnitude (e.g. 200 MeV to 20 GeV), or more. It should also have some way to distinguish between two different types of particle (e.g. electron and photon). For each input particle you should produce an energy or momentum measurement, and a way of indicating which type it is. Write Python code to perform these measurements, using the output from your simulation.

Step 3 - Data validation (20%)

The results of your reconstruction can be tested by comparing them to the truth output of your simulation. Calculate the resolution for your energy/momentum measurements, and the efficiency ratio (correct:total) for your particle classification. Examine whether these results vary over a range of input particle energies, and show what happens. Depending on the classification task you attempt your efficiency may vary significantly, but you should aim to exceed 70%.

Step 4 - Improvement and discussion (40%)

Attempt to improve the quality of your result: try to obtain a narrower resolution, a more efficient classification, or both. You may achieve this by improving your detector, or your reconstruction approach, or both. Describe and discuss the process of improvement, showing "before and after" results, and include interesting failed attempts. Simply repeating your approach on a larger scale for an incremental improvement may receive fewer marks than a creative but unsuccessful idea.

Please submit your work as a Jupyter notebook containing all your code and commentary to LEARN by **10am on Friday 22nd March**. Include sufficient diagrams or visualisations to describe your simulation, and submit your full simulation code and raw data.

2 Report format

The project report has to be in the Jupyter notebook format. It is expected that you are comfortable with writing, running and annotating code within a Jupyter notebook.

Annotation and commentary

It is important that all code is annotated and that you provide brief commentary at each step to explain your approach. We expect well-documented Jupyter notebooks, not an unordered collection of code snippets. For example, explain how your reconstruction algorithm for particle data should work. You can also include any interesting failed approaches if you provide reasonable explanation.

The submission is not in the form of a written report so do not provide pages of background material. Only provide substantial text for the description of your simulation, and the improvement step. Aim to present your work clearly so that the markers can easily follow your reasoning and can reproduce each of your steps through your analysis.

To add commentary above or below a code snippet, create a new cell and add your text in markdown format. Do not add large amounts of discussion as a code comment in the same cell as the code. To change the new cell into markdown select this option from the drop down menu on the bar above the main window (the default is code).

Submission steps

Before marking we will open your notebook and run all the cells. It is important your code is fully functional before it is submitted, and you may receive fewer marks if it is not. When you are ready to submit your report perform the following steps:

- In Jupyter run Kernel >Restart Kernel and Run All Cells to ensure that all your analysis is reproducible and all output can be regenerated
- Run Kernel >Restart Kernel and Clear All Outputs
- Save the notebook
- Close Jupyter
- Tar and zip your project folder (i.e. `tar cvfz geant-UUN.tar.gz geant-UUN/`)
- Submit this file through LEARN

Please include all simulation code and raw data. If your compressed project folder exceeds 40 MB please contact Dr. Christos Leonidopoulos ahead of the submission deadline for instructions.

3 Use of example code

We do not expect you to write your Geant4 simulation from scratch: you may base it on an existing example. Naturally you may use example code from the workshops, and you may also be interested in the full Geant4 example library: http://geant4-userdoc.web.cern.ch/geant4-userdoc/Doxygen/examples_doc/html/ Anything that you use in this way should be acknowledged in the project notebook. **You are expected to modify some Geant4 simulation code yourself**, please use code comments to identify what your own contributions were.

Similarly, in the reconstruction and analysis steps you may use Python code that you have already written in the workshops. You are expected to research potential algorithms to use, but any significant use of code written by someone else must be acknowledged.

4 Machine Learning approaches

You are encouraged to apply Machine Learning to the task of reconstructing your data. This step is deliberately written to include a classification study and a regression study, and either or both could be attempted using Machine Learning.

You might use a conventional algorithm in the first place, and then compare this to Machine Learning in your “improvement” step. Bear in mind that there’s no guarantee that Machine Learning will actually improve your result, but this would still be an interesting project. If your Machine Learning model takes a large amount of time to train, please make a note of this and provide it in a trained form.

5 Time management

There is the potential for this project to spiral rapidly into something very complex. Any additional data produced by your simulation will need to be incorporated in your data format, loaded correctly by your analysis, and processed in your reconstruction algorithms. Any change to your reconstruction algorithms should be sanity-checked and the output validated. **You are strongly encouraged to start with a simple, but functional, project:** note that 60% of the available project marks are for this. Since your detector should contain two major structures it will already be twice as complex as one of the workshop examples. You can then get more ambitious in the “improvement” step.

Remember that Neural Network training can be very slow, as can Geant4 simulation. Please allocate sufficient time to perform any calculations that you need. Remote access to the CPLab machines is available.

6 Simulation truth

You should make a clear distinction in your analysis between simulated outputs of your detector, and truth data. Geant4 can output information that could not physically be gathered in a real experiment, such as the type of process that created a particle. In the “reconstruction” step of your project you should use only measurable properties of particles. Reconstructed results can then be calibrated and validated by comparing them with the truth. Neural Networks can use truth information as labels for the training and validation datasets.

The workshop examples show the kind of measurements you can make with a real experiment: the coordinates of charged particles hitting silicon wafers, or the energy deposited in scintillating material like liquid Argon. While there are many other detector technologies that you could use, all are ultimately ways of finding the position of a deposit of energy or charge. Even direct recording of a vector property like momentum is (to my knowledge) impossible, as it must be inferred from position information.

If you do use unphysical information in your reconstruction algorithms you may receive fewer marks. This is a judgement call that you should be able to make based on your knowledge of physics. If in doubt, ask us! Allow time to change your code if need be.

7 Enough doom and gloom

I’d really like to see some novel ideas! As I said at the start, an interesting failure may well get more marks than an unambitious success. The project is structured so that you have room to demonstrate something simple and practical (please do this first!), and then to get creative.