Advanced ~~Computing~~ **Analysis** Tools

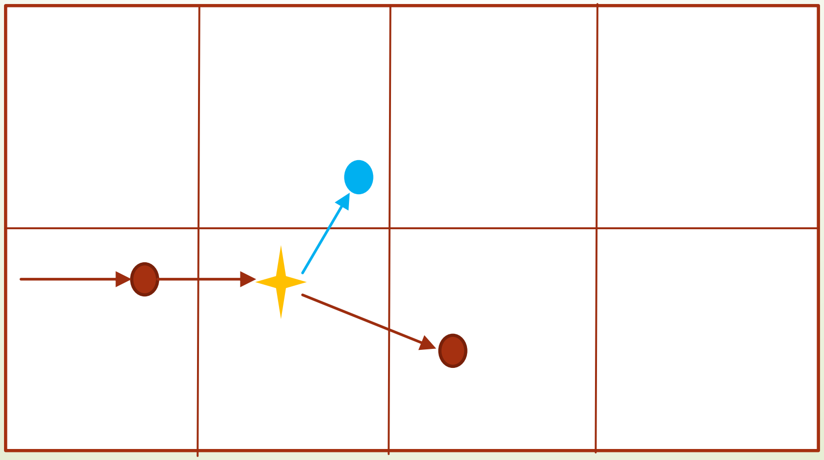
Ian Lam

Introduction

- Estimating detector response can be difficult.
- On paper, you can calculate (using physics principles) what you expect but what you actually measure might be slightly different.
- Detector effects are difficult to take into account analytically – geometry, material properties, etc.
- The brute-force method: run simulations, also commonly called Monte Carlo.

Introduction

- GEANT4 is a particle physics simulation engine.
- Build your detector with the various materials and simulate!
- Very roughly speaking, GEANT4 divides the detector into smaller individual boxes and evaluates the physics in each box as the particle makes its way across the detector.



Introduction

- GEANT4 has a steep learning curve.
- Individual experiments design a wrapper for GEANT so that end-users can run simulations without the hassle of GEANT. In SNO+, the wrapper is called RAT (Reactor Analysis Tools).
- Lets see this in action!
- Show how you can do meaningful but quick analysis to do rough checks – don't have to worry about writing scripts.
  - of course, if you see the analysis being taken further/require more methodical analysis, write the script!

Some physics

- Taken from: Glenn F. Knoll, "Radiation Detection and Measurement". (amazing book!)
- Scintillators basically convert particle energy/momentum into measureable visible light.
- There are various chemicals that can function as scintillators. Due to their molecular makeup, they convert energy to visible light with different efficiencies.
- This conversion to visible light is known as 'light yield'.
- Can also depend on particle type – more in a bit.

Some physics

- Birk's formula quantifies this light yield:
  - $$\frac{dL}{dx} = \frac{S\frac{dE}{dx}}{1+B\frac{dE}{dx}}$$
  - dL/dx = fluorescent energy emitted per unit path length
  - dE/dx = specific energy loss of the charged particle
  - B = an adjustable parameter to fit experimental data for a particular scintillator.
  - S = normalization constant

Some physics

- If we look at the formula, can see that the light yield (dL/dx) depends on the amount of energy the particle deposits per unit path length (dE/dx)
- "…high ionization density along the track of the particle leads to quenching from damaged molecules and a lowering of scintillation efficiency."
- So, one can expect that an alpha particle will generate less scintillation light than an electron/beta with the same energy since alphas are more ionizing. We can check this!

RAT time

- generate 20 3MeV electrons at a point in the center of the detector
- do the same for 20 3MeV alphas
- plot ‘nhits’ (number of triggered PMTs) for both.

RAT time

- generate 20 3MeV electrons at a point in the center of the detector
- do the same for 20 3MeV alphas
- plot ‘nhits’ (number of triggered PMTs) for both.
- Notice that the distribution is not smooth; choppy and gaps. Can still fit but the fit won’t be good.
- But already we can see the alphas have lower average nhit than electrons.

RAT time

- To solve the problem, just generate more events !
- Notice that it took some time to generate 50 events. I generated 1000 events ahead of time, to save time.
- plot ‘nhits’ (number of triggered PMTs). Fit a Gaussian. Note the mean.
- plot energy (reconstructed quantity). Again, fit a Gaussian and note the mean.

Quick Analysis

1. Calculate the average simulated light yield for both alphas and electrons by dividing ‘nhit’ (mean of Gaussian for nhit) of each with the true energy (i.e. 3MeV).
2. How many times larger is the electron light yield compared to alpha?
3. How does the average of the *reconstructed* energy compare to the *true* energy for both?

Quick Analysis

1. alpha: (89.3/3.0)[nhits/MeV] = 29.8 nhits/MeV

   electron: (1317.4/3.0)[nhits/MeV] = 439.1 nhits/MeV

2. 439.1/29.8 = 14.7 times

   According to Knoll, electrons and protons defer by about factor 10 at the energies we are looking at so this is about the right order of magnitude. (we expect the difference to be greater since alphas are larger and more ionizing than protons)

3. alpha: 0.19 MeV ; electron: 2.97 MeV

   Notice the difference in the reconstructed energy.

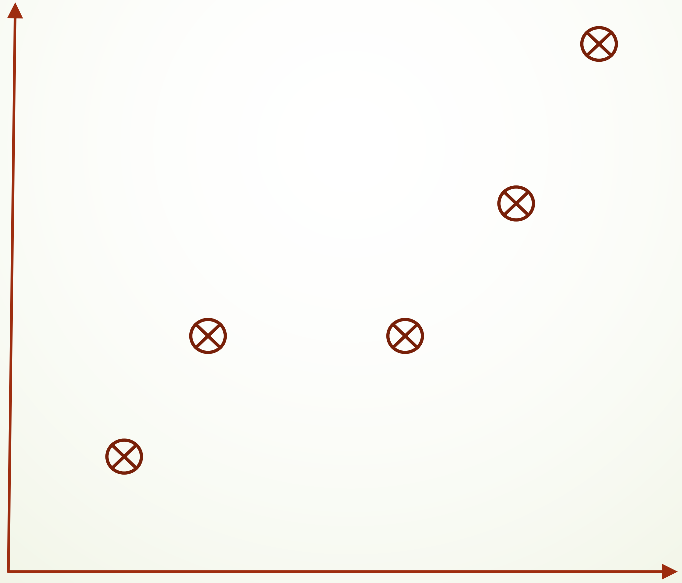
Analysis

- With this relatively quick simulation and analysis, we have checked that our simulated detector is behaving correctly for a specific case, as expected by Birk's Law.
- We have also checked the expected light yield for the detector – the detector response.
- Of course, this is just a simulation. Next step is to calibrate the detector (a challenge on its own!). But the simulations gives us a ballpark of what we should expect in real life.

Closing Remarks

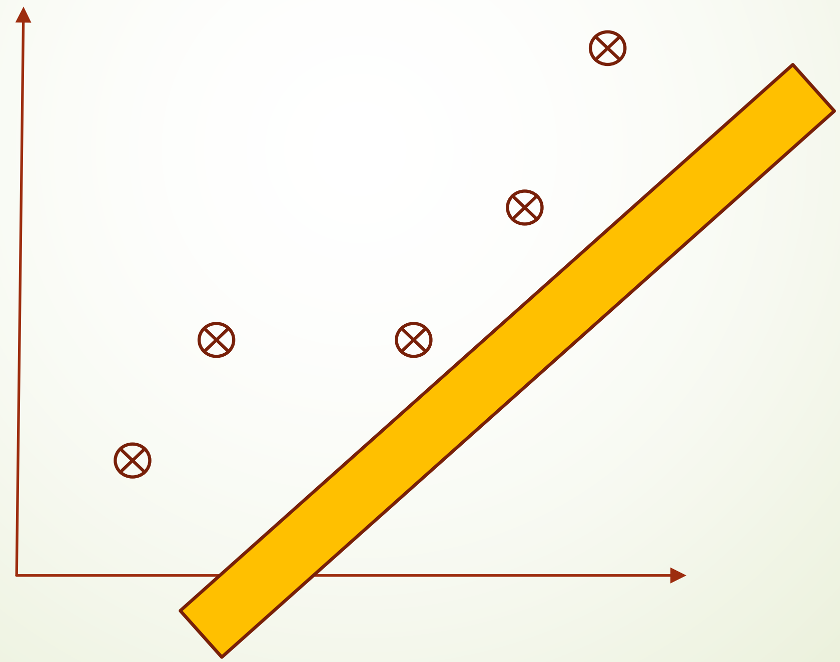
- Though this exercise seems simple, it serves as a quick check to make sure the simulations and the algorithms used to reconstruct the events look ok.
- As time goes on, your experiment's analysis software development might become more complicated.
- Simple checks like these are done routinely to make sure no bugs creep in.

Fitting – what happens?



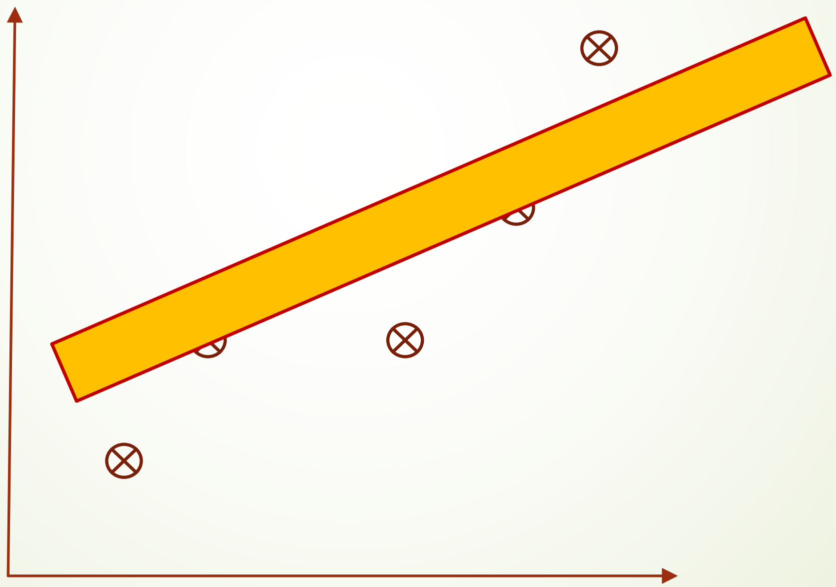
Fitting – what happens?

y = mx + c ; m = slope, c= y-intercept



Fitting – what happens?

y = mx + c ; m = slope, c= y-intercept



Fitting – what happens?

- You're basically adjusting the slope and y-intercept, trying to find the line that passes through the most points, and make the best fit line as close as possible to the points that don't fall on it – *minimizing* the distance to best fit.
- ROOT is basically doing the same thing when you call fit, although it is minimizing a different function.
- 'Parameter estimation'

Fitting – Chi^2 and log-likelihood

$$\chi^2(\theta) = \sum_{i=1}^{N} \frac{(y_i - \lambda(x_i;\theta))^2}{\sigma_i^2}.$$

y = data point; λ = value of function given a parameter; σ = error in the data point

for binned data, y = number of stuff in bin, σ = √y – can see how it can cause issues

$$\log L(\theta) = \sum_{i=1}^{N} n_i \log \nu_i(\theta),$$

n = number of stuff in bin, v = expected number of stuff, given parameter – depends on function used in fit.

Fitting

- So, ROOT basically tries various values of the parameters, eventually finding the best minimum value of either χ^2 or log-likelihood
  - 'steps through parameter space before converging to a minimum'
- Uses minimizers: TMinuit is the main minimizer package of ROOT.
- Minimizers: balance between speed and precision
- MIGRAD: a minimizer
- SIMPLEX: a basic minimizer – might see these in other analysis software like Matlab.
- HESSE: calculates the Hessian matrix (matrix containing second order derivatives)
- MINOS: another minimizer, better at estimating errors in fit but slower.
- input of one can be passed to another

Fitting

```
root [3]  FCN=61.9026 FROM MIGRAD    STATUS=CONVERGED      67 CALLS          68 TOTAL
                     EDM=1.04918e-10    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                    STEP          FIRST
  NO.   NAME       VALUE            ERROR          SIZE       DERIVATIVE
   1  Constant      2.35475e+01   9.13067e-01   4.06215e-03  -1.07189e-05
   2  Mean          1.31740e+03   1.07221e+00   5.84858e-03   5.41116e-08
   3  Sigma         3.38865e+01   7.61263e-01   3.33224e-05  -1.61304e-03
                               ERR DEF= 0.5
```

Main:
FCN: minimum of function (log-likelihood or χ^2)
Status converged: found the minimum

Misc:
calls: number of steps
EDM: ‘estimated distance to minimum’
Step size: size of a single step taken for that parameter
First derivative: used to evaluate where to step
ERR Def: ‘error definition’. used to determine the error.

Resources

- Glenn F. Knoll, “Radiation Detection and Measurement”.
- Glen Cowan, “Statistical Data Analysis”. (was told this was ‘the’ statistical reference book for CERN experiments)
- Harrison Prosper, “Practical Statistics”. https://indico.cern.ch/event/358542/
- Glen Cowan, TRISEP 2016 lectures. https://indico.triumf.ca/conferenceTimeTable.py?confId=2115