

Exercises for Statistical Methods in Particle Physics

<http://www.physi.uni-heidelberg.de/~nberger/teaching/ws13/statistics/statistics.php>

Dr. Niklaus Berger (nberger@physi.uni-heidelberg.de)

Dr. Oleg Brandt (obrandt@kip.uni-heidelberg.de)

Exercise 5: A tracking detector

11. November 2013

Hand-in solutions by 14:00, 18. November 2013

Please send your solutions to obrandt@kip.uni-heidelberg.de by 18.11.2013, 14:00. Make sure that you use *SMIPP:Exercise05* as subject line. Please put each macro into one separate .C or .py file, which can easily be tested. This time, in preparation for the next exercise, please write your *compiled* programme in native (C++) root code. The code can be compiled either using a Makefile, or via `root -l my_code.C+` (note the + at the end). If plots are requested, please include print statements to produce pdf files in your code, and provide the plots separately. Please add comments to your source code explaining the steps. Test macros and programs before sending them off...

In this exercise, we will simulate a simple particle tracking detector in 2D. This can in principle be generalised to 3D in a straight forward way, however, then the devil is hiding in the details... This exercise is composed with the inner tracking detector of the ATLAS detector in mind. Its *barrel* part consists of three detectors, from the beampipe to the outside: the pixel detector, the SemiConductor Tracker (SCT), and the Transition Radiation Tracker, which we ignore for this exercise. For an illustration of the real setup in 3D refer to Fig. 1. A technical drawing of the cross-section of the inner detector is shown in Fig. 2. The pixel detector consists of three layers with a pitch of $50 \mu\text{m}$ in the transverse detector plane, while the SCT has a pitch of $80 \mu\text{m}$. For simplicity, we will ignore the fact that each SCT module is actually comprised of two sub-modules.

1 Setting up the geometry for a tracking detector

In our 2D world, we consider the transverse plane of the inner detector. For simplicity, we will simulate only a $\pm 0.9^\circ$ wedge of the detector. Particles are produced at the interaction point in the centre of the detector at $x = 0$, $y = 0$, and propagate towards positive x . They are equally distributed within our considered wedge, i.e. approximating a cosine with its 0-th order Taylor expansion, their slopes s_0 are equidistributed between $[-0.1, 0.1]$. Prepare a setup where there are 3 infinitely thin layers of the pixel detector at distances of $x = 50, 100, 150$ mm from the interaction point, and 4 layers of the SCT at $x = 300, 400, 500, 600$ mm. As a first test, generate randomly one track and propagate it from layer to layer through the detector setup. Store the true positions at which it traverses each of the detector layers, and store them in a `TGraph` object (use the `SetPoint()` method). Also generate a detector response to this propagated track (a *hit*) at each intersection and store it in a `TGraphErrors`. For this part of the exercise we do not consider any resolution effect, i.e. the hit positions are known with infinite precision, and the error of the hit can be set to 0 for now. After traversing all layers, draw the detected hits (`TGraphErrors`) with option "A*") and the true intersection points (`TGraph`) with option "L", then fit the hits with a straight line (create a `TF1` object with a linear polynomial "[0]+[1]*x"

as the formula and call `TGraphErrors::Fit(TF1*)`. Draw the result for one randomly generated track (`ex_5.1.pdf`). Make sure they are what you expect for a perfect detector. Compare your fit results to the true generated track. For this, fit parameters of the TF1 can be retrieved via `TF1::GetParameter(int index)`.

2 Detector with finite resolution

In a real detector, hit positions are always measured with a finite resolution. Take your code from above and simulate the pixel and the SCT detectors. Note that in our 2D model they are both strip detector (albeit with different pitches). To model the resolution, assume digital binary readout, i.e. the detector cannot further resolve the position of the hit than the strip that has fired, so assume all reconstructed hits are in the middle of the strip they fired. For the same track above, adjust the hit results to account for this effect. Do not forget that now the uncertainty on the hit is finite and known, so set the error in y accordingly in your `TGraphError` object¹ using `SetPointError()`. Fit the `TGraphError` as above (`ex_5.2.pdf`).

3 Tracker resolution studies with many tracks

Now skip the drawing part of your code and generate 1000 tracks, and fill the residuals, i.e. $r := y_{\text{hit}} - y_{\text{true}}$ into a histogram (`ex_5.2_res_pix.pdf` / `ex_5.2_res_sct.pdf`). Do this separately residuals in the pixel and SCT detectors. What do you obtain? Interpret the differences in the results between the detectors, and also interpret results generally in terms of Problem 4.1 from the previous problem sheet.

Furthermore, plot the offset and the slope parameters of the track fits into two histograms (`ex_5.2_offset.pdf` / `ex_5.2_slope.pdf`). What do you get?

Check whether your error description is correct by plotting the pulls

$$\text{pull} := \frac{\text{reconstructed} - \text{true}}{\text{error}}$$

of the residuals, as well as of the offset and slope parameters (`ex_5.2_res_pix_pull.pdf` / `ex_5.2_res_sct_pull.pdf` / `ex_5.2_offset_pull.pdf` / `ex_5.2_slope_pull.pdf`) What do you expect for the pull distribution in this case? What do you get? You can access fit errors via `TF1::GetParError(int index)`. How does this change if you completely remove the pixel detector from the setup?

4 Adding multiple Coulomb scattering to the detector model

A real detector will be made of a finite amount of material, leading to multiple scattering of particles in the detector. The precise simulation of multiple scattering is a very difficult problem, but the core of the angular distribution is well approximated by a normal distribution with a σ of

$$\sigma_{MS} = \frac{13.6\text{MeV}}{\beta cp} z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \ln \frac{x}{X_0} \right), \quad (1)$$

where β is the fraction of the speed of light c of the particle speed, p the particle momentum, z the magnitude of its charge and $\frac{x}{X_0}$ the thickness of the detector in radiation lengths. Assume that $\beta = 0.95$, $p = 1\text{GeV}$ and $\frac{x}{X_0} = 2.5\%$ for each detector layer. Again propagate your track, this time also dicing a multiple scattering angle at every detector plane. What does this do to your resolution on the residuals and fit parameters? Also investigate the pulls. Why are they off now?

¹If you do not know what the error should be, find out using a histogram of the differences between the true and the reconstructed hit y positions for many tracks.

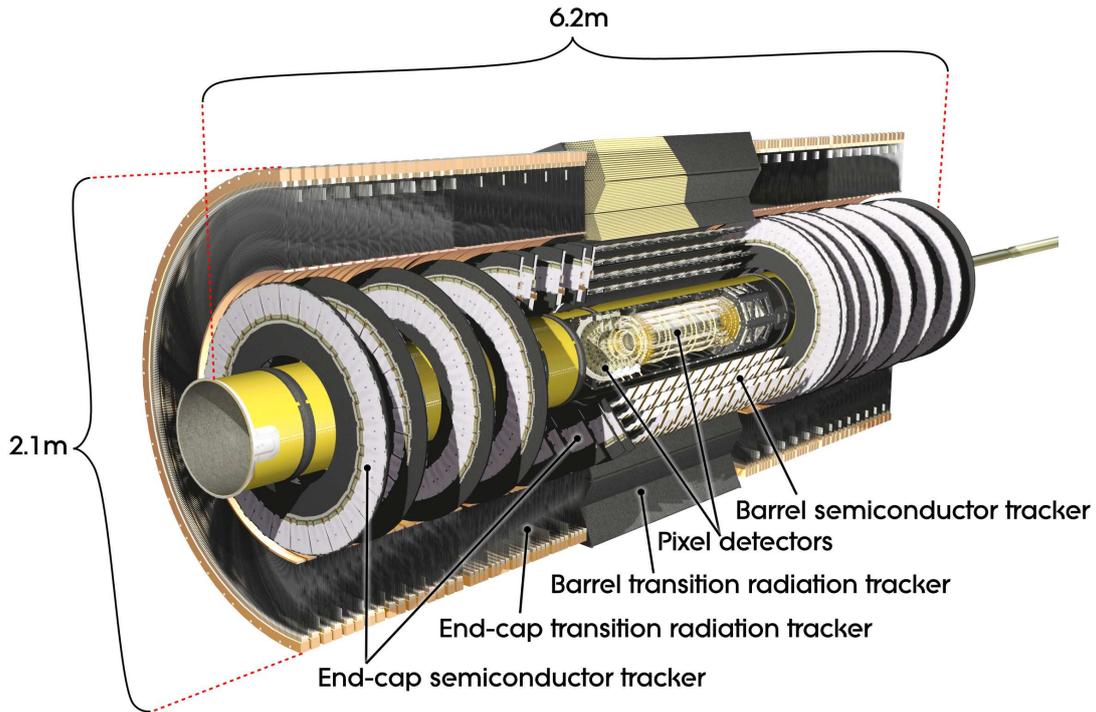


Abbildung 1: The inner tracking detector of the ATLAS experiment. The pixel detector is the innermost one to the beampipe and its barrel part consists of three layers. It is contained within the SCT detector, whose barrel part consists of four layers. Outermost is the Transition Radiation Tracker, which we ignore for this exercise.

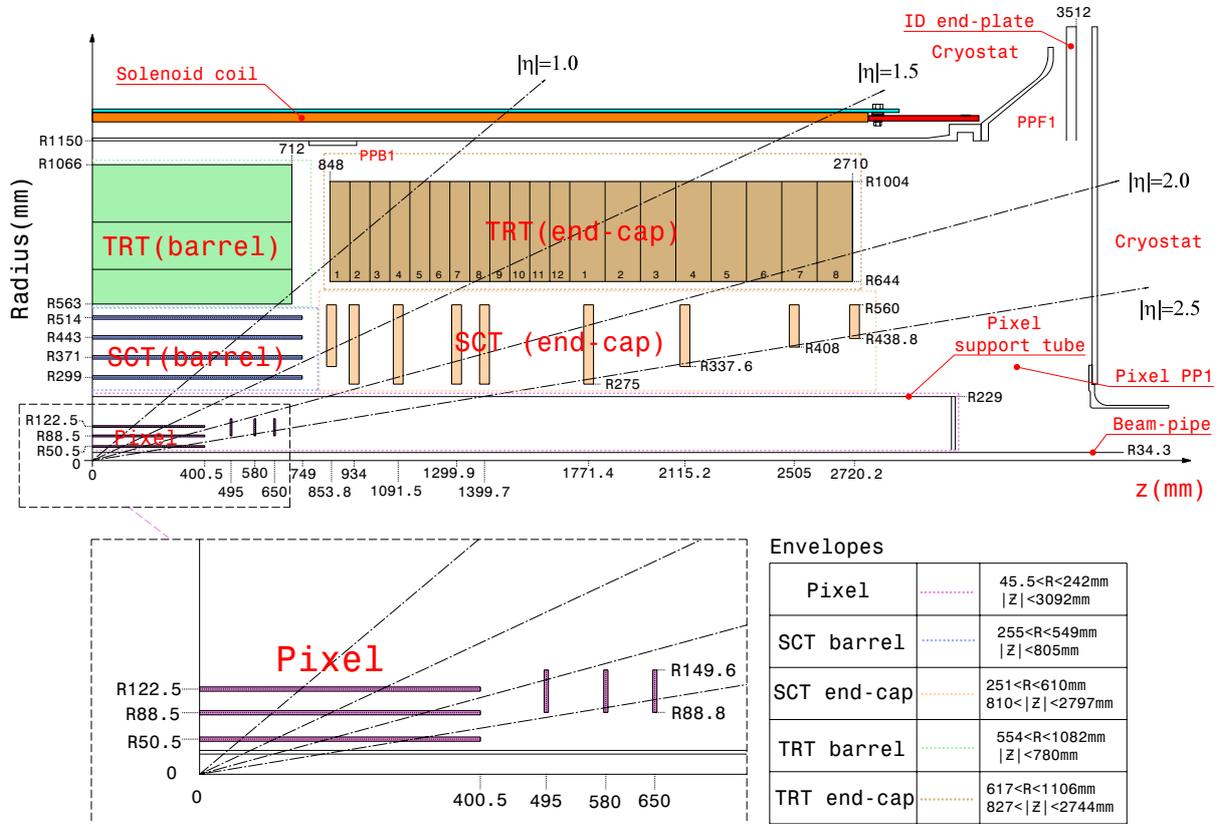


Abbildung 2: The cross section of the inner tracking detector of the ATLAS experiment.