Computing Methods for Physics – 5 September 2022

Your exam material (code files, plots, datafiles, etc.) must be submitted via google classroom by 13:15 as a single `zip` file.

C++ evaluation will be based on: correct syntax, proper return types, proper arguments of functions, data members and class interfaces, setters/getters, unnecessary void functions, comments throughout the code, separation of class implementations and interfaces.

Python evaluation will be based on: correct syntax, avoiding C-style loops, using Python features in general, comments throughout the notebook/scripts, labels, legends and plot styling and clarity in general.

Part 1 – C++

Implement a C++ class `ConicSection` to represent and manipulate quadratic equations in two variables, i.e.,

$$ax^2 + bxy + cy^2 + dx + ey + f = 0\,. \tag{1}$$

The class must provide the following.

- A copy constructor and a constructor that takes as arguments the 6 coefficients of the quadratic equation (as 6 separate numbers, as an array, as a vector, you decide).
- Setters and getters for all coefficients.
- A method to calculate the discriminant ($b^2 - 4ac$) and a data member that stores information about whether the instance represents an ellipse, a parabola, or a hyperbola.
- An overload of the `<<` operator to print information about `ConicSection` instances.
- The derived classes `CanonicalEllipse`, `CanonicalParabola`, and `CanonicalHyperbola` to allow the user to construct instances of

  $$\frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2} = 1$$
  $$y = \alpha x^2 + \beta x + \gamma$$
  $$\frac{x^2}{\alpha^2} - \frac{y^2}{\beta^2} = 1\,,$$

  by providing α, β, and (possibly) γ.
- In the case of `CanonicalParabola`, implement methods to determine the value of the first and second derivatives with respect to x at a given x.

Include a file `app.cpp` that showcases the classes you implemented.

Part 2 – Python

Use a Python notebook or Python scripts to complete the following tasks.

1. Design a class to represent instances of parabolas in the canonical Cartesian form $y = ax^2 + bx + c$.

2. Implement a method to find where the parabola crosses the x axis.

3. Exploit the SciPy package to provide a method to integrate the parabola in x between the values x_1 and x_2 provided by the user.

4. Draw 100 x and y values from the parabola $y = 5x^2 - 3x + 2$, with x between 5 and -5. Add noise to the y values by using a Gaussian distribution centered in 0, with standard deviation 15. Store the x, y, and $y + y_{err}$ values to file.

5. Use a Monte Carlo package of your choice to fit the x vs $y + y_{err}$ data with a parabola. Plot the posterior distributions for a, b and c, and comment your results.