

# Particle Track Finding

Final Project for “Functional Parallel Programming”, Fall 2022

## 1. Team members

Brendan Cunnie (bec2148)

Lukas Arnold (la2789)

## 2. Project Description

Inspired by particle detectors at the Large Hadron Collider, our project is dedicated to a computational problem in particle physics: track finding.

Particle collisions create an immense amount of collision products – particles – that are captured by the detectors surrounding the collision point. The particle tracks are identified as pixels by the detectors; in order to identify the particle, the possible tracks have to be reconstructed. Due to the immense amount of particles, parallelization of track finding seems to be an appropriate answer to the computational load.

In figure 1, a particle detector is shown where the pixels are represented by the dots; the track findings' task will be to “connect the dots” to output the correct track and identify the particle. In our example, we will emulate a particle detector that has sparser resolution and more tracks, i.e. a particle detector in which particle tracks are harder to identify.

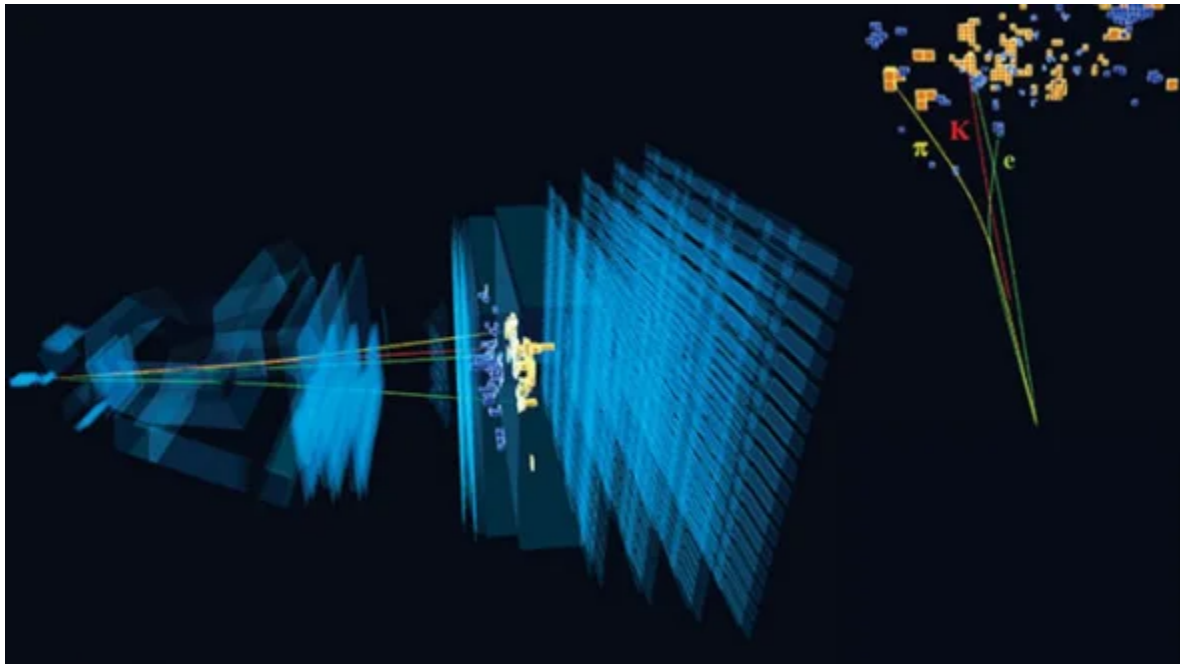


Figure 1: Example Particle Detector. From: Scientific American, March 2021

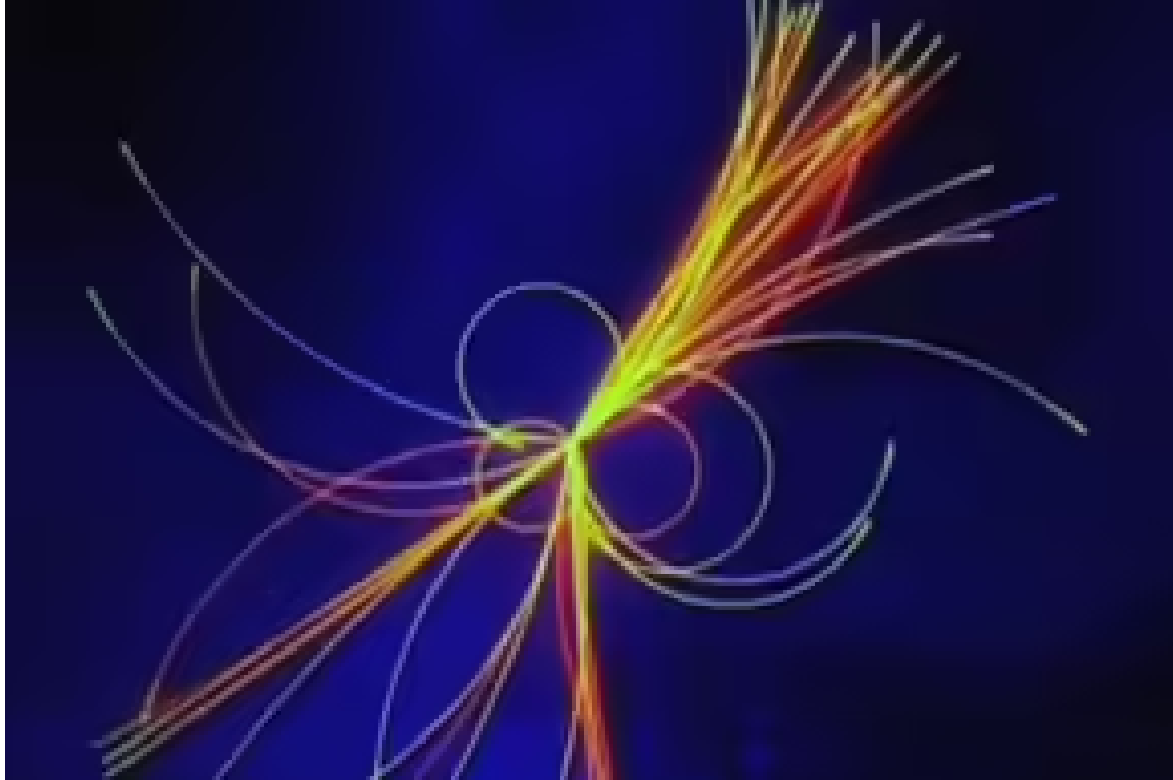


Figure 2: Image with particle paths identified. From Science, August 2017

### 3.Environment

For our purposes, we will emulate a simple particle detector. As the complexity of the LHC is not appropriate for the project, the setup is significantly simplified: a homogeneous magnetic field is applied to the particles, the particle detectors are assumed to have ideal efficiency, and the dimension of the particle detector is defined by us.

### 4.Development plan

Our project will contain the following tasks:

1. Building the detector simulation that emulates a collision and outputs:
  - a. the pixel activated by the collision
  - b. the ground truth: the tracks and the particle types.
2. Writing the particle track finder code serially and verify its output
3. Writing the particle track finder code in a parallel manner and verify its output
4. Prepare the submission/presentation