

## Billiard Ball-to-ball collision simulation

This project focuses on optimizing a billiard ball-to-ball collision model supported in the open-source Python library "pooltool" [1]. The optimization targets the `collide_balls(..)` function [2], which implements an iterative scheme to compute ball impulses after collision [3].

The project will contain the following steps:

- 1) Understanding an algorithm and establishing test scenarios (numbers & positions of the balls on the billiard table, algorithm iterations, etc.) for the `collide_balls(..)` function using the Pooltool SDK.
- 2) Implementing the naive `collide_balls(..)` function in C/C++. This function could be implemented in a shared library to interface the pooltool Python simulation through C-to-Python wrapper (e.g., see [4]), and baseline benchmarking.
- 3) Applying optimizations learned through the ASL course to achieve an optimal version of the `collide_balls(..)` function.

[1] Pooltool <https://github.com/ekiefl/pooltool>

[2]

[https://github.com/ekiefl/pooltool/blob/f5bd153a453d1f4d4d3d98ee13267e4e50b33886/pooltool/physics/resolve/ball\\_ball/frictional\\_mathavan/\\_init\\_.py#L76](https://github.com/ekiefl/pooltool/blob/f5bd153a453d1f4d4d3d98ee13267e4e50b33886/pooltool/physics/resolve/ball_ball/frictional_mathavan/_init_.py#L76)

[3] Mathavan et. al.

[https://billiards.colostate.edu/physics\\_articles/Mathavan\\_Sports\\_2014.pdf](https://billiards.colostate.edu/physics_articles/Mathavan_Sports_2014.pdf)

[4] <https://asiffer.github.io/posts/numpy/>