

Project 3: Tinkertoy

Due: Feb 26, 2018, 11:55PM

Your task is to simulate two beads under gravity. One bead must stay on a circular wire and the second bead is connected to the first one by a fixed-length rod. You will implement one constraint that keeps a bead on the circle and another that keeps two beads a fixed distance apart. Your program should simulate the motion of the beads under gravity and should draw the beads as they move.

You should initialize the particle at a position that satisfies constraints. To improve the stability, you should add the feedback term when solving for λ . A linear system ($Ax = b$) can be solved using the Eigen library included. For example, the inverse matrix of `Eigen::Matrix2d A`, is simply `A.inverse()`.

Skeleton code

Run the executable of the skeleton code. You will see a circle, a rod, and two particles on the screen. Starting the simulation by hitting the space bar, you will see these two particles fall due to gravity. The function that requires most of your effort is `Simulator::simulate()` in `simulator.cpp`. Currently, this function only considers gravity and integrates the two particles using Explicit Euler method. Your job is to compute the constraint forces and incorporate them into the system such that the particles satisfy the circle constraint as well as the distance constraint.

Extra points:

1. Upgrade the integrator to midpoint method (1 point).
2. Implement additional types of constraints for the tinkertoy (e.g. a bead should stay on a plane or three beads should always form an equilateral triangle). If the constraint outputs more than a single scalar, you need to modify the indexing scheme for the global matrices and vectors so that the constraint evaluations and derivatives are stored in the correct places (1 point for each additional type of constraint, up to 3 points).
3. Use mouse click to select the closest particle to the cursor and use mouse drag to apply force on that particle. The magnitude and the direction of the force should be controlled by the mouse action (1 point).
4. Allow the user to add particles and constraints interactively using the mouse and/or the keyboard. This feature requires you to implement the entire tinkertoy system in a general way because you cannot fix the size of your global structures in advance. The data structures need to be extensible on the fly as the user adds more and more particles and constraints (2 points).