

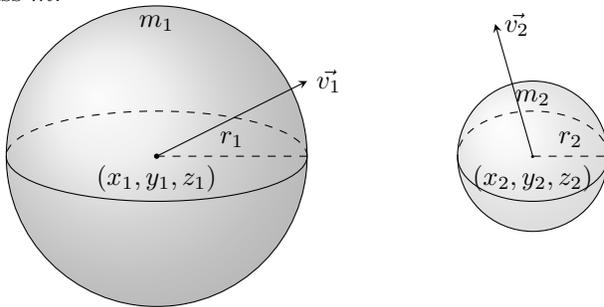
Literate Classical Physics

Tobi R. Lehman

February 3, 2022

1 Introduction

This program simulates a 3D universe subject to the laws of Newtonian mechanics. The basic ontology¹ is a collection of rigid bodies, shaped like spheres, each with a position (x, y, z) , a velocity \vec{v} , radius r , and a mass m .



The collection of all bodies in the simulation is called the Universe. The state of the Universe is the collection of positions and velocities of all the bodies. The state is updated at each time step t . The state update rule is based on the law of Newtonian gravity:

$$\vec{F}_{12} = G \frac{m_1 m_2}{r^2} \hat{r}$$

To understand how this law applies, remember that force is proportional to acceleration, and acceleration is the rate of change of velocity. In vector form: $\vec{F} = m\vec{a}$, and $\vec{a} = \frac{d}{dt}\vec{v}$. Now that we have connected the force law to the state of the Universe, we can guess how to update the state and compute the next moment. We take all pairs of bodies (i, j) and calculate the forces between them \vec{F}_{ij}

¶ This is the overall structure of the program `lcp.c`

- 2 <Header files 3>
- <Constants 4>
- <Struct types 5>
- <The main program 6>

¶ Including standard I/O to get output from the program.

- 3 <Header files 3> ≡
`#include <stdio.h>`

This code is used in chunk 2.

¹An ontology is a scheme defining what exists.

¶ Physics has many constant values, like the speed of light c , the gravitational constant G . In this humble program, there is another constant, dt , the minimum number of seconds used to advance the time loop. For practical reasons, this is much larger than the Planck length.

4 <Constants 4> ≡
const float $dt = 0.0001$;

This code is used in chunk 2.

¶ We need C structs that represent the essential values that define our simple ontology. A *body* has an x , y , and z coordinate, a radius r , and a mass m . It also has a *velocity*

5 <Struct types 5> ≡
struct vec3 {
 float x, y, z ;
};
struct body {
 struct vec3 $position$;
 struct vec3 $velocity$;
 float $mass$;
 float $radius$;
};

This code is used in chunk 2.

¶ Here is the general layout of the *main* function.

6 <The main program 6> ≡
int $main()$
{
 <Set up initial conditions of universe 7>;
 <The main time loop 8>;
}

This code is used in chunk 2.

¶ The initial conditions of the universe are the set of bodies, their positions, masses and velocities. The laws of physics and the inexorable march of time takes over after that. Let's start with a binary star system.

7 <Set up initial conditions of universe 7> ≡
struct body $star1 = \{\{-1, 0, 0\}, \{0, -1, 0\}, 1, 1\}$;
struct body $star2 = \{\{1, 0, 0\}, \{0, 1, 0\}, 1, 1\}$;

This code is used in chunk 6.

¶ The main time loop is where the simulated time flows. Each iteration of the loop adds dt seconds to the current time.

8 <The main time loop 8> ≡
for (**int** $t = 0$; $t < 100$; $t += dt$) {}

This code is used in chunk 6.

¶ Done

2 Rendering

We want the state of the 3D universe to be displayed on a 2D screen.

Index

body: 5, 7.
dt: 4, 8.
main: 6.
mass: 5.
position: 5.
radius: 5.
star1: 7.
star2: 7.
t: 8.
vec3: 5.
velocity: 5.
x: 5.
y: 5.
z: 5.

List of Refinements

- ⟨ Constants 4 ⟩ Used in chunk 2.
- ⟨ Header files 3 ⟩ Used in chunk 2.
- ⟨ Set up initial conditions of universe 7 ⟩ Used in chunk 6.
- ⟨ Struct types 5 ⟩ Used in chunk 2.
- ⟨ The main program 6 ⟩ Used in chunk 2.
- ⟨ The main time loop 8 ⟩ Used in chunk 6.