N-bodies simulation

Szymon Bugaj

October 5, 2016

Abstract

A system of N material points under the force of gravity. Symulation and visualisation.

Contents

1 Theoretical introduction

The project is devoted to symulation and visualisation of the problem of N-bodies.

Newtonian gravitational force between two material points is given as:

$$\vec{F_G} = G\frac{m_1 m_2}{\|r\|^2}\frac{\vec{r}}{\|r\|}$$

In the classical physics, the equation linking acceleration, mass and force is given as follow:

$$\vec{F} = \vec{a}m$$

The relationship between the position, velocity and acceleration is given by equations:

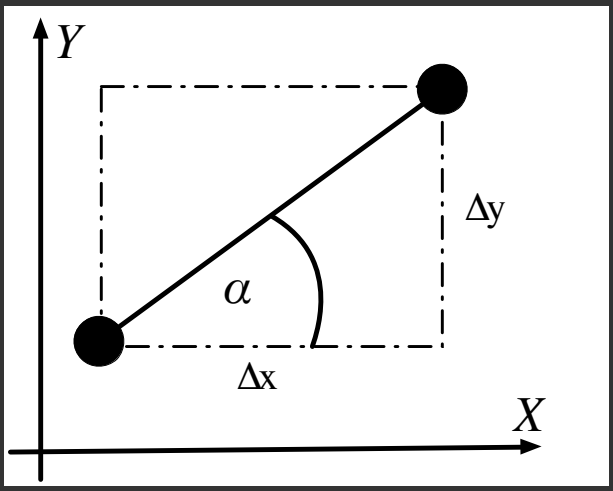
$$v(t) = \frac{dp(t)}{dt}$$
$$a(t) = \frac{dv(t)}{dt}$$

Given the position of all N particles and the speed for a given torque, using the above equations, we can calculate the location for any point at any other time.

2 Resolution of gravity into components parallel to axis

In the case of 2D should be considered rectangle with sides parallel to the axes X and Y, where the opposite tops are considered two of the body.

.Resolution of gravity into components



$$a_x(t) = sin\alpha * a$$
$$a_y(t) = cos\alpha * a$$

For the 3D case should be considered in an analogous manner Box (body should be at the vertices of the opposite walls at opposite corners).

3 Numerical model

We are interesed in position of each of a body in proceding discrete time moments. We assume, that during a smallest time unit speed, acceleration of a body is constant. All variables are updated in order:

- 1) speed
- 2) position
- 3) acceleration

What must be stressed, in consequence in calculation of update of speed is taken previous value of acceleration.

```
void NBodiesSystem::step( time_type delta_t ) {
    p_prev = p_curr;
    v_prev = v_curr;

    /*
        UPDATE v SPEED and p POSITION
    */

    for (int d = 0; d < D; ++d)
        for (int i = 0; i < N; ++i) {
            v_curr.setVal(d, i,
                v_prev.getVal(d, i)  +
                a.getVal(d, i) * delta_t);
            p_curr.setVal(d, i,
                p_prev.getVal(d, i)  +
                    (v_prev.getVal(d, i) +
                    v_curr.getVal(d, i))
                        * 0.5 * delta_t);
        }

    /*
        UPDATE a ACCELERATION
        For each two bodies i,j where i != j;
    */

    for (int d = 0; d < D; ++d)
        for (int i = 0; i < N; ++i)
            a.setVal(d, i, 0.0f);

    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < N; ++j) {
            if ( i == j ) continue;

            /*
                delta X, delta Y, delta Z
            */
            position_type* r_axis = new position_type[D];

            position_type r_squared = 0;
            for (int d = 0; d < D; ++d) {
                r_axis[d] = (p_curr.getVal(d, i) -
                    p_curr.getVal(d, j));
                r_squared += r_axis[d] * r_axis[d];
            }

            position_type a_scalar =
                G * m.getVal(0, j) /
                pow(r_squared + efactor, 1.5);

            for (int d = 0; d < D; ++d) {
                /*
```

```
52                      If both objects positions are
53                      the same there is division
54                      by zero; what to do then?
55                      I just set acceleration to 0;
56                  */
57                  if (r_axis[d]) {
58                      a.setVal(d, i,
59                          a.getVal(d,i) -
60                              a_scalar *
61                              (r_axis[d]/sqrt(r_squared)));
62                  }
63              }
64
65              delete [] r_axis;
66          }
67      }
68  }
```