

Санкт-Петербургский государственный политехнический университет  
Институт прикладной математики и механики  
**Кафедра «Теоретическая механика»**

## **О т ч е т**

по дисциплине «Компьютерные технологии в механике»

Реализация метода динамики частиц с использованием языка  
программирования C#.

Выполнила  
студентка гр.53604/2

Е.Л.Светличная

Преподаватель

А.А.Ле-Захаров

Санкт-Петербург

2015

## Постановка задачи

Рассматривается модель плотной сферы, заполненной частицами, расположенными на очень близких расстояниях, под действием парного потенциала Леннард-Джонса.

**Потенциал Леннард-Джонса**— простая модель парного взаимодействия неполярных молекул, описывающая зависимость энергии взаимодействия двух частиц от расстояния между ними. Определяется формулой:

$$U(r) = D \left[ \left( \frac{a}{r} \right)^{12} - 2 \left( \frac{a}{r} \right)^6 \right], \text{ где}$$

- $r$  — расстояние между частицами,
- $D$  — энергия связи,
- $a$  — длина связи.

Предполагается, что частицы разлетятся как в модели большого взрыва.

**Большой взрыв**— общепринятая космологическая модель, описывающая раннее развитие Вселенной, а именно— начало расширения Вселенной, перед которым Вселенная находилась в сингулярном состоянии.

Согласно теории Большого взрыва, дальнейшая эволюция зависит от экспериментально измеримого параметра—средней плотности вещества в современной Вселенной. Если плотность не превосходит некоторого критического значения, Вселенная будет расширяться вечно, если же плотность больше критической, то процесс расширения когда-нибудь остановится и начнётся обратная фаза сжатия, возвращающая к исходному сингулярному состоянию. Современные экспериментальные данные относительно величины средней плотности ещё недостаточно надёжны, чтобы сделать однозначный выбор между двумя вариантами будущего Вселенной.

**Цель работы:** Попытаться смоделировать такой эксперимент и оценить полученный результат.

### **Ход работы.**

Для реализации поставленной цели составлена библиотека классов ParticlesSimulation на языке C# с использованием основ ООП

Были разработаны следующие классы:

1. Constants.cs — класс, содержащий основные константы для всех вычислений — массы, заряды и диаметры частиц; время и шаг интегрирования; константы, задающие потенциал;
2. Creator.cs — класс, предназначенный для создания системы Space: заполнение её частицами Particle с заданными массами, зарядами, координатами и скоростями;
3. IntegratorLJ.cs — класс позволяющий рассчитывать поведение системы Space, используя потенциал Леннарда-Джонса;
4. Integrator.cs — класс, позволяющий рассчитывать поведение системы Space;
5. OutputHelper.cs — класс, позволяющий выводить данные системы Space в текстовый файл;
6. OutputHelperA3R.cs — производный класс OutputHelper.cs, позволяющий выводить данные системы Space в формате, пригодном для визуализации в программе A3R.
7. Particle.cs — класс, описывающий частицу;
8. Space.cs — класс, описывающий систему;
9. Vector3D.cs — класс, описывающий трехмерные вектора;

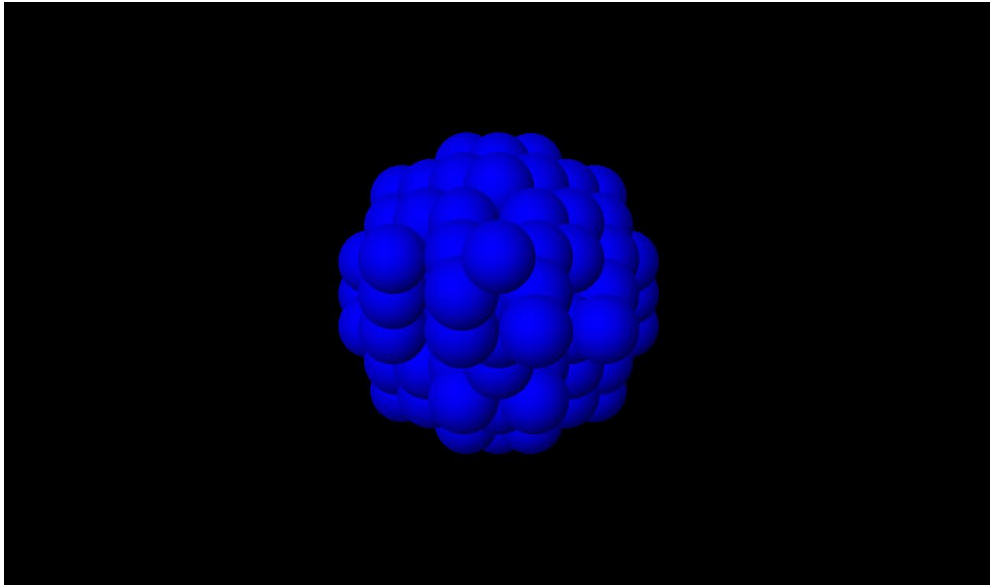


Рис1 Модель в начальном состоянии

### Результаты и выводы

В задаче рассматривается 365 частиц, радиус каждой из которых равен корню из 20. Результаты решения визуально продемонстрированы в программе АЗР. На Рис 2-5 проиллюстрированы состояния системы в различные моменты времени. На начальном этапе при больших  $r$  частицы притягиваются, а на малых расстояниях молекулы отталкиваются из-за обменного взаимодействия, чему соответствует используемый потенциал взаимодействия Леннард-Джонса.

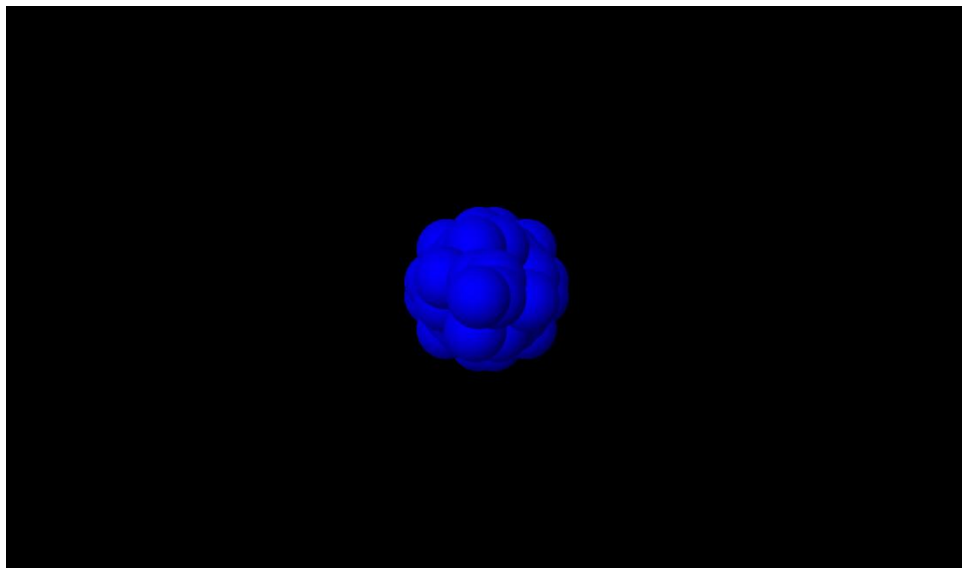


Рис 2. Притягивание частиц

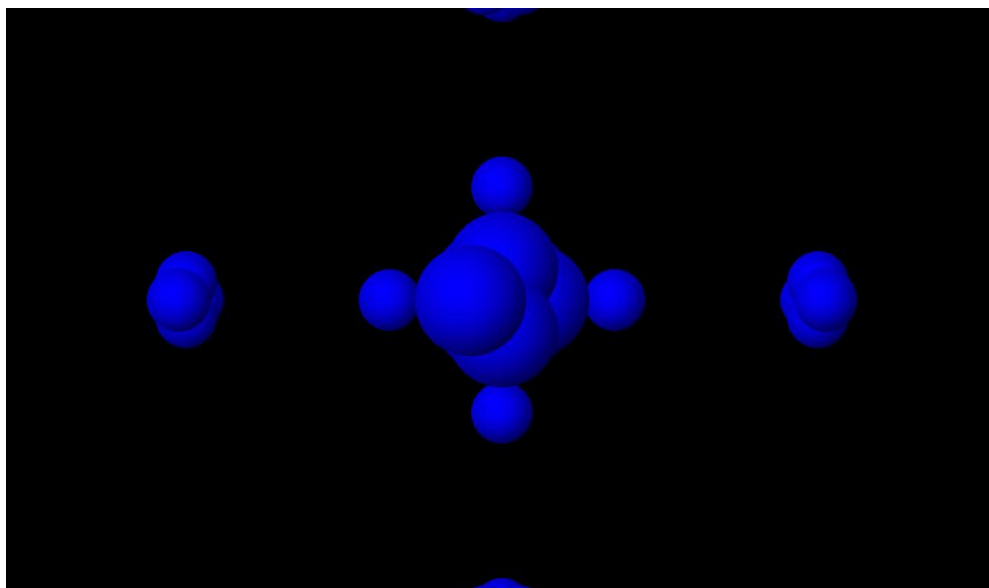


Рис3 Состояние системы на расчетном шаге равном 22

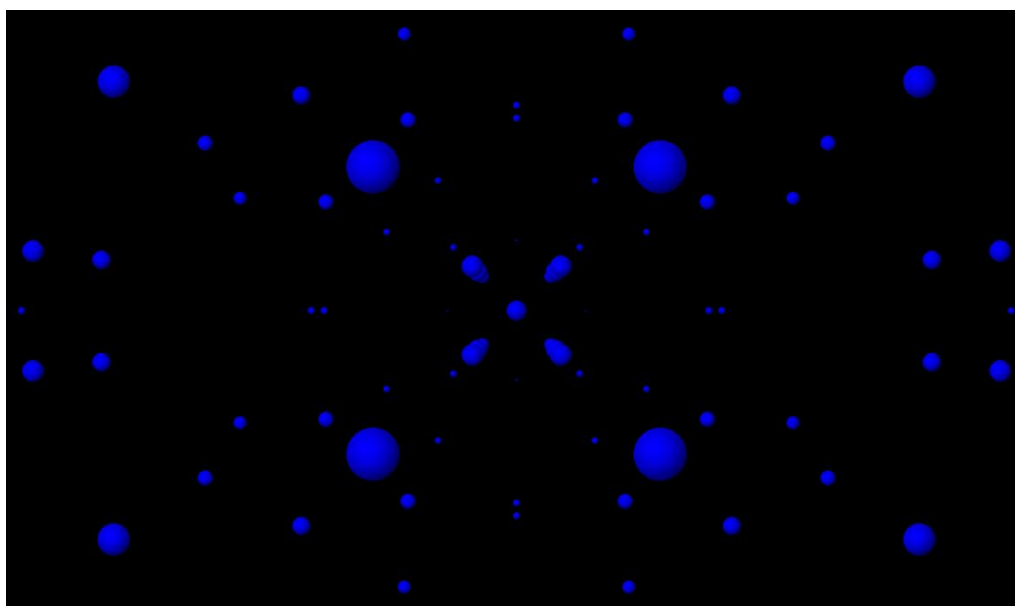


Рис4 Большой взрыв

Полученные результаты демонстрируют модель большого взрыва. Их анализ позволяет сделать предположение о том, что вселенная будет расширяться вечно.

Теория о том, что через некоторое время произойдет схлопывание вселенной, не подтвердилось.

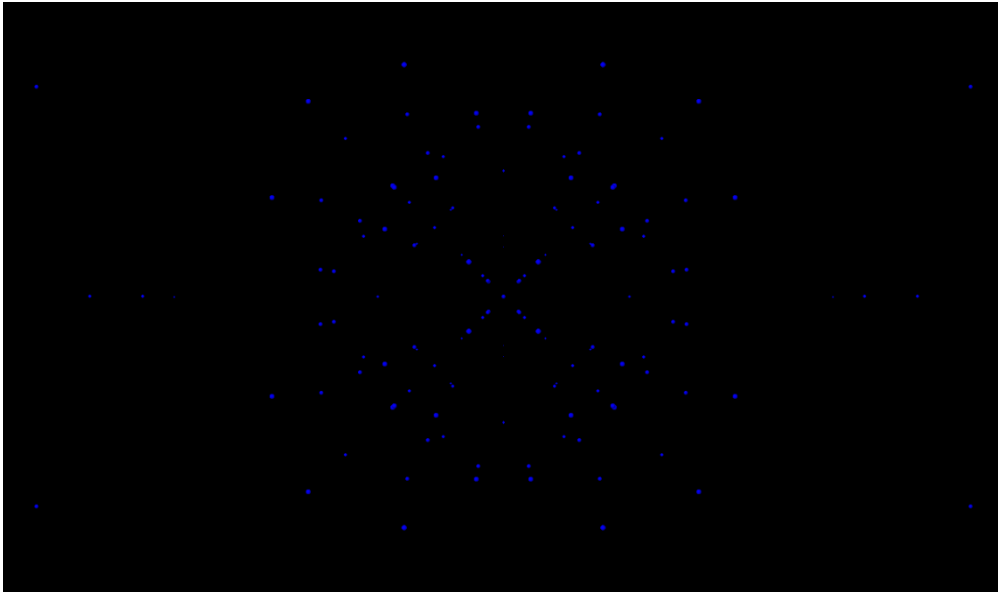


Рис5 Состояние вселенной. Шаг расчета = 691

В результате работы над поставленной задачей написана программа на языке C#, в которой методом динамики частиц произведены расчеты модели «большого взрыва».

Также получен опыт реализации численных вычислений на языке C#, изучены основы и преимущества объектно-ориентированного подхода к написанию решения задач.

# Приложение

Creator.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ParticlesSimulation
{
    public class Creator
    {
        public Space Create()
        {
            Space model = new Space();
            for (int i = -100; i < 100; ++i)
            {
                for (int j = -100; j < 100; ++j)
                {
                    for (int k = -100; k < 100; ++k)
                    {
                        if ((i * i + j * j + k * k) < 20)
                        {
                            Particle p = new Particle(i, j, k);
                            model.Add(p);
                        }
                    }
                }
            }
            return model;
        }
    }
}
```

Constants.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ParticlesSimulation
{
    public static class Constants
    {
        public static readonly double
```

MinMass = 0.001, DefaultMass = 1, DefaultD = 1, MinDistance = 0.0001, dt = 0.01,  
MaxTime = 40000.0, K=1.0, DefaultQ = 1.0;

```
}  
}
```

#### IntegratorLJ.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
  
namespace ParticlesSimulation  
{  
    public class IntegratorLJ : IntegratorBase  
    {  
        public IntegratorLJ(Space s) : base(s) { }  
  
        public override void RecalcParticleForce(Particle pi, Particle pj)  
        {  
            double a = (pi.D / 2 + pj.D / 2);  
            double r = (pi.R - pj.R).Norm();  
            double modfij = (12.0/a)*(Math.Pow(a/r,12.0)-Math.Pow(a/r,6.0));  
            pi.F = pi.F + (modfij / (pi.R - pj.R).Norm()) * (pj.R - pi.R);  
        }  
    }  
}
```