МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

КУРСОВАЯ РАБОТА

по дисциплине «Дифференциальные уравнения»

Тема: Моделирование работы кинескопа

Студент гр. 8382 _____ Нечепуренко Н.А.

Студент гр. 8382 _____ Терехов А.Е.

Преподаватель _____ Павлов Д.А.

Санкт-Петербург

2021

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

Студенты: Нечепуренко Н.А., Терехов А.Е.

Группа 8382

Тема работы: Моделирование работы кинескопа

Исходные данные:

Для моделирования работы кинескопа необходимо решить дифференциальное уравнение формулы Лоренца. При этом необходимо сравнить различные методы численного интегрирования и выбрать подходящий. Для корректного заполнения колбы кинескопа частицами, необходимо вывести алгоритм изменения модуля магнитных полей, при котором будет получен желаемый результат.

Содержание пояснительной записки:

«Содержание», «Введение», «Постановка задачи», «Численные методы решения», «Сравнение методов», «Графическое моделирование», «Заключениие», «Список использованных источников».

Предполагаемый объём пояснительной записки:

Не менее 20 страниц.

Дата выдачи задания: 20.04.2021

Дата сдачи работы: 07.06.2021

Студент гр. 8382	_____	Нечепуренко Н.А.
Студент гр. 8382	_____	Терехов А.Е.
Преподаватель	_____	Павлов Д.А.

АННОТАЦИЯ

В работе рассматривается упрощенная модель кинескопа. Движение частиц в электромагнитном поле описывается с помощью формулы Лоренца. Применив к ней численное интегрирование, можно получить траекторию движения заряженной частицы. В работе происходит визуализация процесса заполнения колбы кинескопа электронами.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В работе рассматривается упрощенная модель кинескопа. Электронная пушка придает частице начальную скорость, а затем с помощью изменения значений модулей магнитных полей частица попадает в нужное место на колбе. Движение частицы в электромагнитном поле описывается с помощью формулы Лоренца. Применив к ней численное интегрирование, можно получить траекторию движения заряженной частицы. В работе рассмотрены три метода численного интегрирования: метод Эйлера, метод Рунге-Кутты 4 порядка и метод Адамса-Башфорта 3 порядка. Была выполнена визуализация результатов.

1. ПОСТАНОВКА ЗАДАЧИ

1.1. Модель кинескопа

Кинескоп состоит из следующих основных частей:

- электронная пушка, предназначенная для формирования электронного луча; в цветных кинескопах и многолучевых осциллографических трубках несколько пушек объединяются в электронно-оптический прожектор
- экран, покрытый люминофором — веществом, светящимся при попадании на него пучка электронов
- отклоняющая система, управляющая лучом таким образом, что он формирует на экране требуемое изображение

Схематически строение кинескопа приведено на рисунке 1.

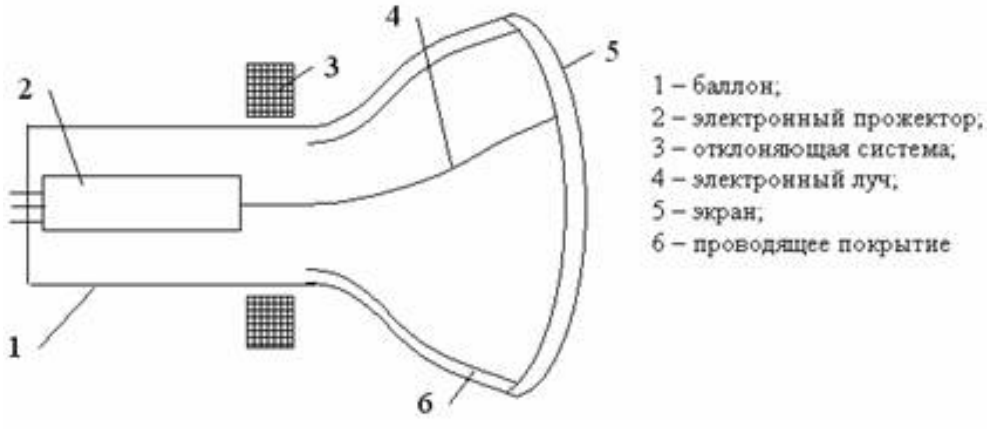

Рисунок 1 – Схема кинескопа

В работе будет рассмотрен кинескоп, способный выдавать черно-белое изображение.

1.2. Уравнение движения частиц в электромагнитном поле

Отклоняющая система кинескопа представлена тремя однородными полями: электрическим и двумя магнитными. Заряженная частица (электрон) движется в электромагнитном поле за счет действующей на нее силе Лоренца.

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B})$$

С учетом 2-го закона Ньютона уравнение выше приводит нас к следующему диффернциальному уравнению:

$$\dot{\vec{v}} = \frac{q}{m}(\vec{E} + \vec{v} \times \vec{B})$$

Эта формула справедлива для однородных полей. Положительно заряженная частица будет ускоряться в том же направлении, что и поле E, но её траектория будет изгибаться перпендикулярно как вектору мгновенной скорости v, так и полю B в соответствии с правилом буравчика. Сила Лоренца — это сила, которую оказывает электромагнитное поле на заряженную частицу, или. другими словами, скорость, с которой передается линейный импульс от электромагнитного поля частице. Магнитное поле не совершает работы, потому что магнитная сила всегда перпендикулярна скорости частицы.

1.3. Допущения и проверяемые инварианты системы

В работе рассматриваются только однородные поля. Также пренебрегаем коэффициентом $\frac{q}{m}$, считая его равным 1. Моделируемую частицу считаем дискретным носителем заряда. Частицы между собой не взаимодействуют, на них не действует иные силы (например сила тяжести), отсутствует сопротивление среды (в колбе вакуум). Можно считать силу их взаимодействия пренебрежимо малой.

Аналитического решения выбранной задачи найти не удалось, поэтому для проверки корректности работы методов и оценки их ошибок воспользуемся следующим инвариантом. Введем два однородных поля: электрическое и магнитное. Для простоты расположим их перпендикулярно друг другу. Тогда известна траектория движения частицы, более того, она циклическая.

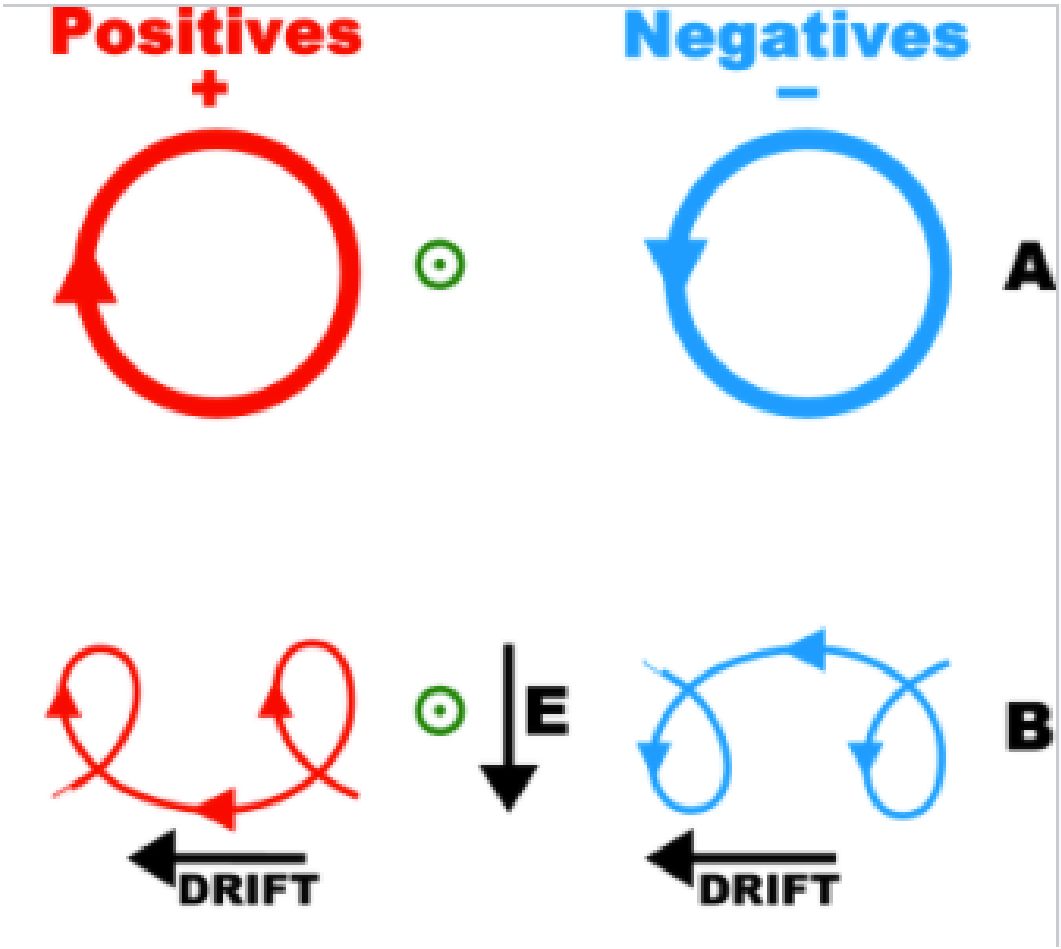

Рисунок 1.1 – Траектория частицы

здесь линии электрического поля направлены вниз, линии магнитного поля направлены от наблюдателя.

Таким образом, если в среде сохраняются постоянные условия, то траектория частицы не должна изменяться. Используя разные численные методы будем подбирать их параметры (шаг), чтобы этот инвариант сохранялся.

2. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ

2.1. Функция правой части

Для решения поставленной задачи была написана программа на языке C++ с использованием фреймворка Qt5.

Для представления функции правой части был разработан интерфейс RhsFunction.

```
class RhsFunction {
public:
    virtual State apply(State state) = 0;
};
```

Все возможные функции правой части должны выглядеть как черный ящик, принимая состояние в текущий момент времени и возвращая состояние в следующий.

Состояние представляет собой структуру из координаты, скорости и заряда (не используется).

```
class State {
public:
    Point3D coordinate;
    Point3D velocity;
    double charge;
    ...
```

Используемая в работе функция правой части от электрических и магнитных полей описана следующим образом:

```
class EMFieldMovingFunction : public RhsFunction {
private:
    std::vector<UniformField *> electricFields;
    std::vector<UniformField *> magneticFields;
...
```

Класс UniformField представляет собой структуру из направления векторов поля и модуля.

```
class UniformField {
public:
    double value;
    Point3D direction;
...
```

Реализация метода apply представлена ниже

```
State EMFieldMovingFunction::apply(State state) {
    if (terminatePredicate(state.coordinate))
        throw 1;
    Point3D dVelocity(0, 0, 0);
    for (auto &electricField : electricFields) {
        dVelocity = dVelocity + electricField->value *
           electricField->direction;
    }

    for (auto &magneticField : magneticFields) {
        dVelocity = dVelocity + magneticField->value * state.
           velocity.cross(magneticField->direction);
    }

    state.coordinate = state.velocity + dVelocity;
    return State(state.coordinate, dVelocity, state.charge);
}
```

Здесь пользуемся принципом суперпозиции полей. Если частица попала на колбу, бросается исключение и вычисления прерываются.

2.2. Базовый класс метода решения

Для взаимозаменяемости методов решения уравнений был написан интерфейс Solver и базовый абстрактный класс AbstractSolver

```
class Solver {
public:
    virtual void solve(State initialState) = 0;
};

class AbstractSolver : public Solver {
protected:
    std::deque<State> previousStates;
    std::shared_ptr<RhsFunction> rhsFunction;
    std::function<void(State)> onUpdateConsumer;
    double h;
    long long maxIterations = 0;
    virtual State step() = 0;
...
```

Контракт интерфейса Solver – любой метод должен генерировать решение из заданного начального состояния.

Базовый класс определяет способ взаимодействия с другими компонентами и набор параметров. Для хранения состояний используется deque, что удобно для многошаговых методов. Каждый конкретный метод решения обязан переопределить шаг метода. Базовый класс отвечает за итерирование по сетке решения и использование функции обратного вызова.

```
void AbstractSolver::solve(State initialState) {
    previousStates.push_back(initialState);
    for (long long iteration = 0; iteration < maxIterations;
       iteration++) {
        onUpdateConsumer(previousStates.back());
        State nextState = step();
```

```
            previousStates.push_back(nextState);
            previousStates.pop_front();
        }
    }
```

2.3. Прямой метод Эйлера

Форма пересчета

$$x_{i+1} = x_i + hf(x_i)$$

Прямой метод Эйлера реализован следующим образом

```
State EulerSolver::step() {
    State currentState = previousStates.back();
    State nextState = currentState + h * rhsFunction->apply(
       currentState);
    return nextState;
}
```

Этот метод скорее необходим для сравнения других, нежели непосредственного использования в программе.

2.4. Метод Рунге-Кутты порядка 4

Формула пересчета

$$x_{i+1} = x_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

где

$$k_1 = f(x_i)$$

$$k_2 = f(x_i + \frac{h}{2}k_1)$$

$$k_3 = f(x_i + \frac{h}{2}k_2)$$

$$k_4 = f(x_i + hk_3)$$

Реализация данного метода приведена ниже

```
State RK4Solver::step() {
    State currentState = previousStates.back();
    State k1 = rhsFunction->apply(currentState);
    State k2 = rhsFunction->apply(currentState + h / 2. * k1)
       ;
    State k3 = rhsFunction->apply(currentState + h / 2. * k2)
       ;
    State k4 = rhsFunction->apply(currentState + h * k3);
    return currentState + h * (1/6. * k1 + 1/3. * k2 + 1/3. *
        k3 + 1/6. * k4);
}
```

2.5. Метод Адамса-Башфорта порядка 3

Формула пересчета

$$x_{i+1} = x_i + h(\frac{23}{12}f_i - \frac{4}{3}f_{i-1} + \frac{5}{12}f_{i-2})$$

где $f_n = f(x_n)$

Реализация метода чуть хитрее, так как он требует трех вычисленный состояний системы. Для их вычисления используется метод Эйлера с тем же шагом.

```
State AB3::step() {
    if (previousStates.size() < 3) {
        prepareStates();
    }
    State currentState = previousStates.back();
    int idx = 0;
    for (auto &state : previousStates) {
```

```
        currentState = currentState + h * coefficients[idx] *
            rhsFunction->apply(state);
        idx++;
    }
    return currentState;
}
void AB3::prepareStates() {
    State currentState = previousStates.back();
    oneStepSolver->solve(currentState);
}
```

3. СРАВНЕНИЕ МЕТОДОВ

3.1. Проверка инварианта траектории

Для сравнения методов введем следующие начальные условия:

Частица движется из $(0, 0, 0)$ со скоростью $(0, 0, 0)$. Электрическое поле $(1, 0, 0)$ с модулем 1. Магнитное поле $(0, 1, 0)$ с модулем -0.5.

Хотим добиться такой точности, чтобы за 24 витка траектории отклонение было меньше 1e-6.

Метод Эйлера с шагом 0.01 за 30000 итераций.

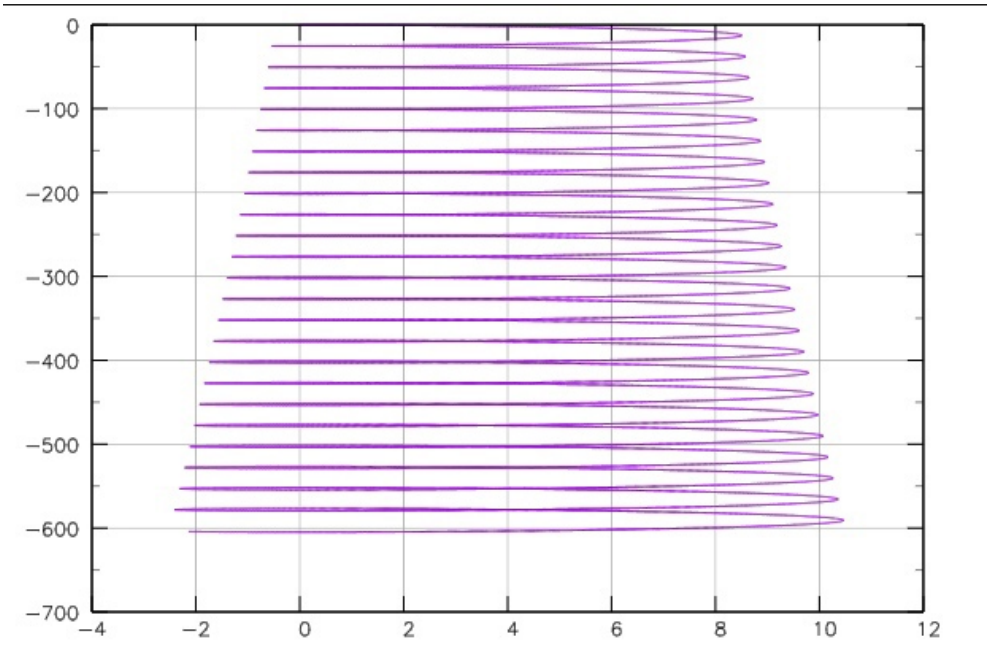

Рисунок 3.1 – Метод Эйлера с шагом 0.01 на 30000 итерациях

Из графика видно, что данный метод с выбранным шагом неустойчив. Размер первого витка (расстояние от минимального до максимального x) – 9.03987, двадцать четвертого 12.87224. Отклонение за 24 витка составило 3.83237.

Уменьшим шаг в 10 раз, увеличим число итераций в 1000.

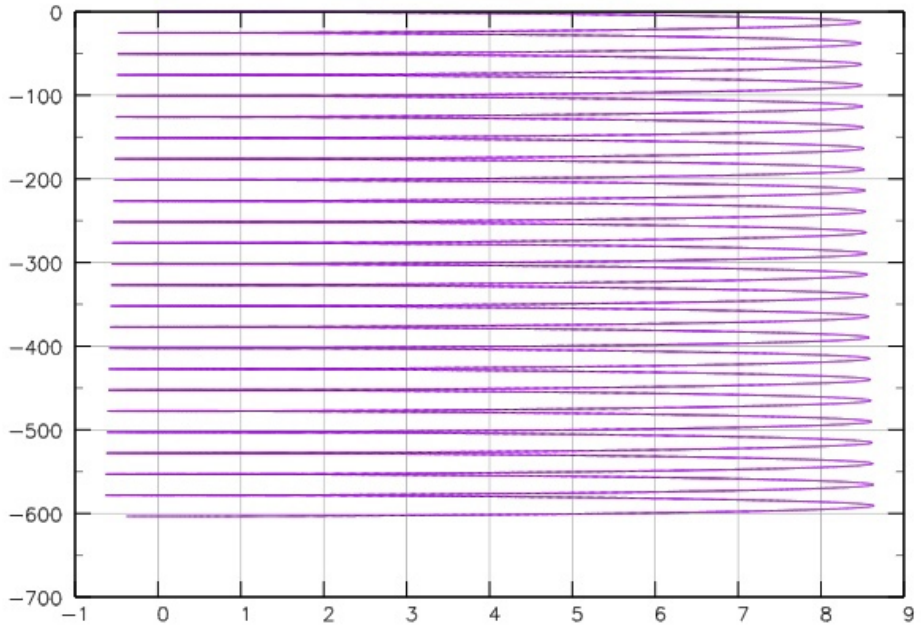

Рисунок 3.2 – Метод Эйлера с шагом 0.001 на 300000 итерациях

Размер первого витка 8.940587, двадцать четвертого – 9.275883. Отклонение за 24 витка составило 0.335296.

Метод Эйлера не подходит для получения решения с заданной точностью.

Протестируем метод Адамса-Башфорта третьего порядка. Возьмем шаг 0.1, 3000 итераций.

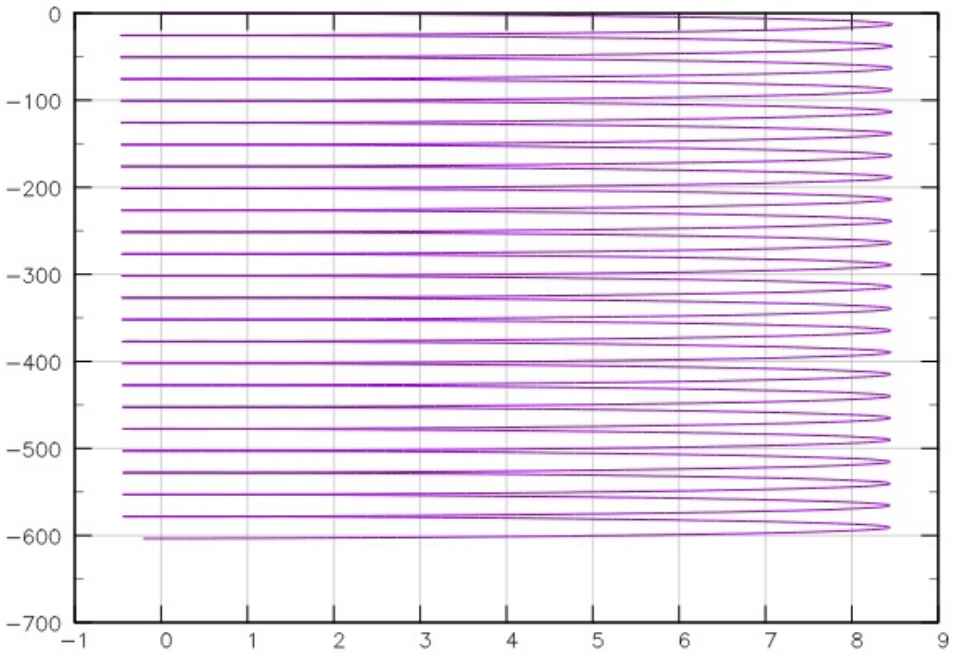

Рисунок 3.3 – Метод Адамса-Башфорта с шагом 0.1 на 3000 итерациях

Размер первого витка 8.942483, двадцать четвертого – 8.885245, модуль отклонения 0.057238.

Уменьшим шаг до 0.01, число итераций увеличим до 30000. График не будет приведен, так как визуально витки не отличимы, рассчеты производятся по координатам точек траектории. Размер первого витка 8.944284, последнего 8.944225, модуль отклонения 6e-5. Будем считать, что метод с шагом 0.001 удовлетворит необходимым требованиям.

Протестируем метод Рунге-Кутты 4 порядка. Выберем шаг 0.1, 3000 итераций.

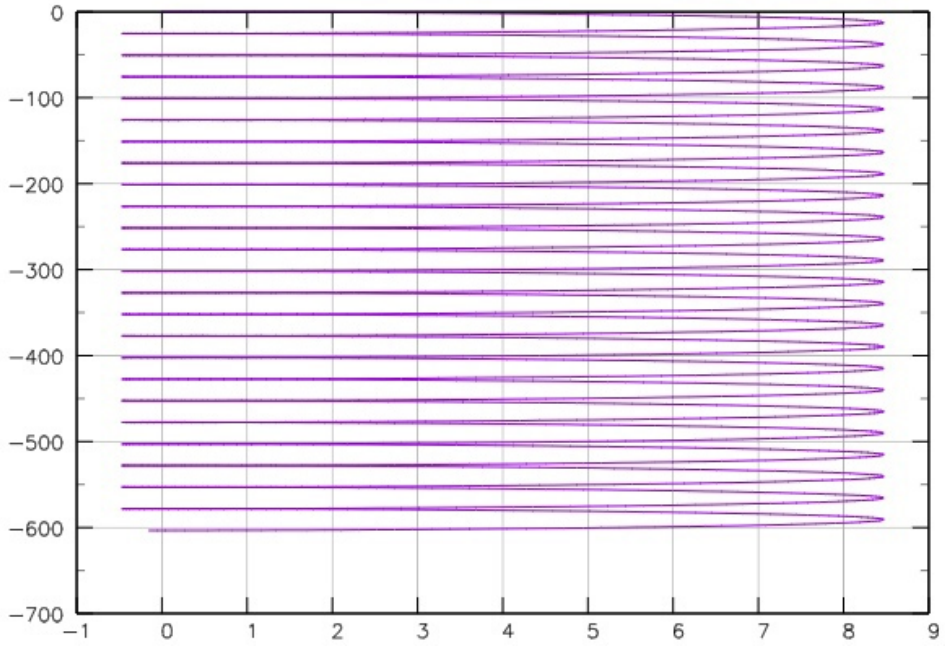

Рисунок 3.4 – Метод Рунге-Кутты с шагом 0.1 на 3000 итерациях

опять визуально не видно отклонения витков, будем пользоваться расчетами.

Размер первого витка 8.942332, последнего 8.944094, модуль отклонения 0.001762.

Уменьшим шаг до 0.01, 30000 итераций. Размер первого витка 8.944265, последнего 8.944266, модуль отклонения 1e-6.

При моделировании будем использовать метод Рунге-Кутты 4 порядка с шагом 0.01.

4. ГРАФИЧЕСКОЕ МОДЕЛИРОВАНИЕ

4.1. Начальные данные

Интерфейс программы выглядит следующим образом

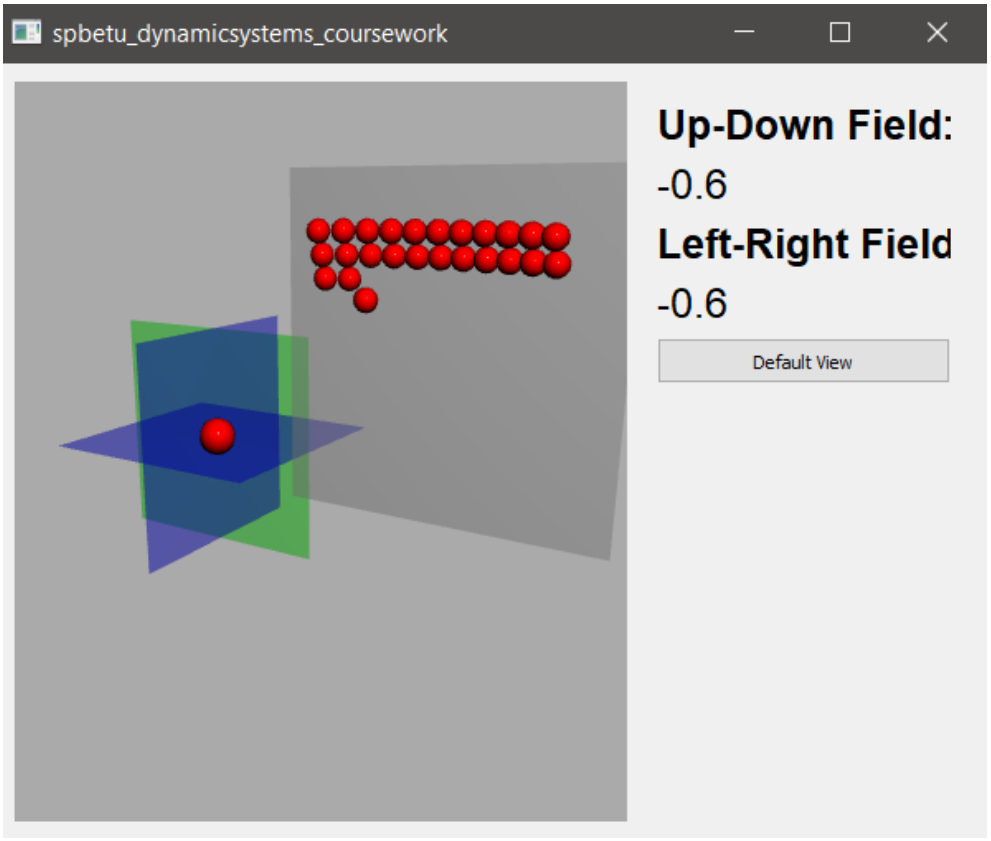

Рисунок 4.1 – Интерфейс программы

Сцена состоит из следующих элементов:

- Три пересеченных плоскости, ортогональные векторам соответствующих полей (синий – магнитные, зеленый – электрическое)
- Серая плоскоть, касательная к сферической части колбы (не удалось изобразить колбу)
- Столбец со значениями модулей соответствующих магнитных полей
- Кнопка «default view»

Частицы начинают двигаться из точки $(0, 0, 0)$. Электронная пушка не

показана. Начальная скорость частицы равна нулю, ее радиус 1.7 у.е.

Электрическое поле не изменяется, его линии сонаправлены с вектором $(1, 0, 0)$, модуль равен 10.

Направления линий магнитных полей не изменяются, они сонаправлены с векторами $(0, 0, 1)$ и $(0, 1, 0)$ соответственно. Их модули изменяются, и эта комбинация своя для каждой частицы. Изначальные значения равны -1. Алгоритм изменения модулей полей будет рассмотрен ниже.

При нажатии на кнопку «default view» камера перемещается в точку $(-20, 0, 0)$, что дает следующий вид на колбу

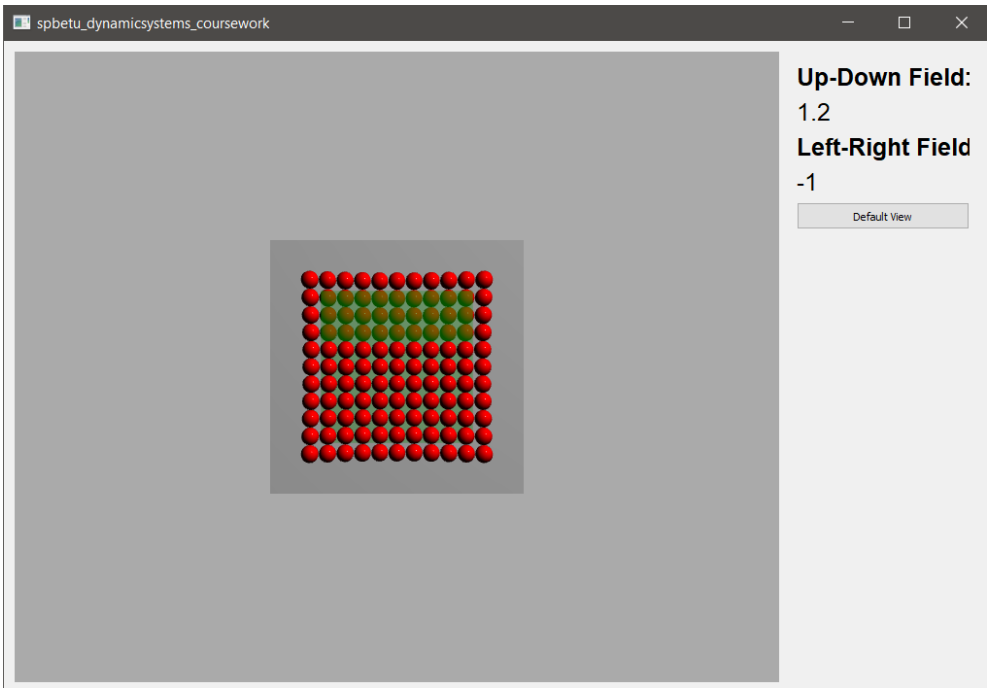

Рисунок 4.2 – Default view

Электроны на колбе, как и ожидалось, формируют часть сферы

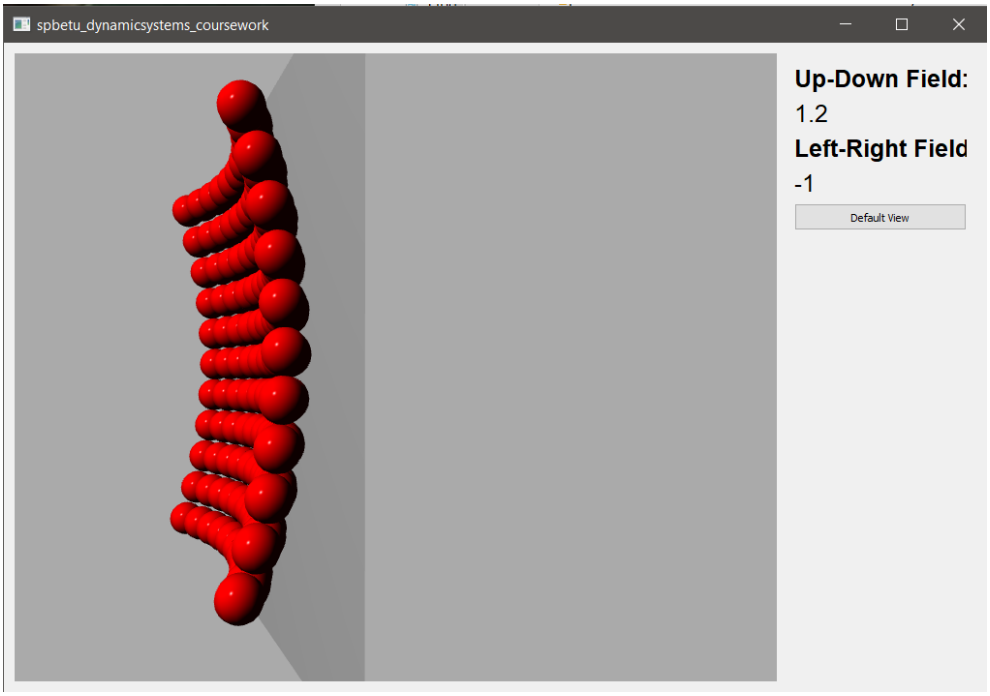

Рисунок 4.3 – Вид на колбу сбоку

4.2. Принцип изменения модулей магнитных полей

За изменение комбинации модулей магнитных полей для направления каждой частицы в отведенное место описан класс MagneticGunManager с методом next.

```
void next() {
    secondField->value += d1;
    iteration++;
    if (iteration >= 11){
        secondField->value = secondFieldValue;
        firstField->value += d2;
        iteration = 0;
    }
```

где $d_1 = 0.2, d_2 = 0.2$.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы была реализована упрощенная модель кинескопа. С учетом допущений (см. раздел 1.3.) для каждого электрона была рассчитана траектория движения согласно формуле Лоренца. Для рассчета траектории было решено дифференциальное уравнение (см. раздел 1.2.). Решение производилось с помощью численных методов: прямого метода Эйлера, метода Рунге-Кутты порядка 4 и метода Адамса-Башфорта порядка 3. В разделе 3 было проведено сравнение методов. В разделе 4 были приведены рисунки с интерфейсом реализованной программы для моделирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сила Лоренца: https://ru.wikipedia.org/wiki/Сила_Лоренца
2. Электричество и магнетизм: https://online.mephi.ru/courses/physics/electricity/data/course/5/5.3.html
3. Как работает телевизор: https://cxem.net/beginner/beginner49.php
4. Метод Адамса: https://ru.wikipedia.org/wiki/Метод_Адамса