Московский государственный технический университет им. Н.Э.Баумана

Кафедра ФН3  
«Теоретическая механика»

Пример титульного листа

ВЫПОЛНЕНИЕ ДОМАШНЕГО ЗАДАНИЯ №1

за 3 семестр

по курсу

«Информационные технологии»

Вариант №\_\_\_

СтудентгруппыФН3–32Б: Малютин С.А.

*дата, подпись*

Преподаватель: Закалужский А.М.

*дата, подпись*

Москва 2024

Оглавление

[1. Постановка задачи 3](#_Toc212419916)

[1.1. Формулировка задачи 3](#_Toc212419917)

[1.2. Цели и задачи 3](#_Toc212419918)

[2. Математическая модель 4](#_Toc212419919)

[2.1. Физическая постановка задачи 4](#_Toc212419920)

[2.2. Физические законы, лежащие в основе модели 5](#_Toc212419921)

[2.3. Описание силы в модели 5](#_Toc212419922)

[2.4. Переход к системе дифференциальных уравнений 6](#_Toc212419923)

[2.5. Численное интегрирование 6](#_Toc212419924)

[2.6. Энергетические характеристики и момент импульса 7](#_Toc212419925)

[3. Исходные данные 7](#_Toc212419926)

[4. Алгоритм решения задачи 9](#_Toc212419927)

[5. Блок-схема решения задачи 10](#_Toc212419928)

[6. Листинг программы 11](#_Toc212419929)

[6.1. Структура программы 11](#_Toc212419930)

[Программа состоит из следующих основных файлов: 11](#_Toc212419931)

[6.2. Основной класс программы 11](#_Toc212419932)

[6.3. Таблица результатов 12](#_Toc212419933)

[6.4. Основной алгоритм моделирования 12](#_Toc212419934)

[6.5. Графический интерфейс 14](#_Toc212419935)

[7. Таблица результатов расчётов 15](#_Toc212419936)

# **Постановка задачи**

## **Формулировка задачи**

Работа заключается в реализации программы, моделирующей плоское движение материальной точки в поле центральной силы притяжения, направленной к неподвижному центру (началу координат). Программа должна обеспечивать численное интегрирование уравнений движения, визуализацию результатов расчёта и анализ энергетических характеристик системы. Сила имеет постоянную по модулю величину F и всегда направлена к центру.

## **Цели и задачи**

**Цель работы:** создание программного комплекса для исследования движения материальной точки под действием центральной силы и анализа сохранения энергетических характеристик системы.

**Основные задачи:**

1. Описать математическую модель движения материальной точки под действием центральной силы притяжения. Рассмотреть методы для ее моделирования.
2. Создать графический интерфейс для ввода начальных параметров и отображения результатов.
3. Обеспечить расчёт и отображение кинематических и энергетических характеристик системы.
4. Реализовать защиту от вычислительных особенностей(например, деление на 0 или некорректный ввод начальных данных).

**1.3 Требования выполнению задания**

* реализация на языке C++ с использованием библиотеки Qt6;
* наличие графического интерфейса;
* реализация ввода начальных условий:
  + масса точки m (кг);
  + модуль силы притяжения F (Н);
  + начальные координаты , (м);
  + начальные скорости , ();
  + время моделирования T (с);
  + шаг интегрирования h (с);
* контроль корректности вводимых данных;
* вывод расчётных данных в табличной форме:
* время моделирования t (с);
* координаты точки x, y (м);
* составляющие скорости , ();
* расстояние до центра притяжения (м);
* модуль скорости ();
* потенциальная энергия (Дж);
* кинетическая энергия (Дж);
* Полная механическая энергия (Дж);
* Момент импульса ();

# **Математическая модель**

## **Физическая постановка задачи**

Мы рассматриваем материальную точку массой m, которая движется в плоскости (x, y) под действием сил. Основные допущения модели:

* центр притяжения является материальной точкой и неподвижен в начале координат (0, 0);
* исследуемая точка считается материальной – её размерами можно пренебречь;
* сила притяжения имеет постоянный модуль и всегда направлена к центру.

В данном случае рассматривается движение материальной точки, потому что это идеализация тела, размерами которого можно пренебречь по сравнению с расстояниями в задаче. Это позволяет не учитывать вращение и деформацию тела.

## **Физические законы, лежащие в основе модели**

Основой модели является **второй закон Ньютона**:

|  |  |
| --- | --- |
| , | (1) |

где:

* – **результирующая сила**, действующая на точку;
* – ускорение точки;
* – радиус-вектор точки.

Этот закон связывает причину (силы) и следствие (ускорение → изменение движения).

## **Описание силы в модели**

Сила притяжения задаётся как сила постоянного модуля , всегда направленная к центру:

|  |  |
| --- | --- |
|  | (2) |

где:

* Знак "минус" указывает, что сила направлена против радиус-вектора (к центру),
* Вектор  – единичный вектор в направлении от центра к точке,
* – расстояние до центра.

**Особенность модели:** в отличие от гравитационной силы (), здесь сила постоянна по модулю. Такое упрощение сохраняет центральный характер силы, но делает уравнения более устойчивыми при численном интегрировании.

## **Переход к системе дифференциальных уравнений**

Второй закон Ньютона — это уравнение второго порядка. Чтобы решить его численно, его приводят к системе уравнений первого порядка, вводя скорость как отдельную переменную:

|  |  |
| --- | --- |
|  | (3) |
| (4) |

Ускорение выражается через силы:

|  |  |
| --- | --- |
|  | (5) |
| (6) |

Таким образом, получается полная система уравнений, состоящая из (3), (4), (5), (6).

## **Численное интегрирование**

Аналитическое решение системы обыкновенных дифференциальных уравнений возможно только для простых случаев. В данной модели:

* Общее аналитическое решение неизвестно или очень сложное
* Постоянная сила нарушает интегрируемость системы по сравнению с гравитационным случаем
* Численные методы позволяют аппроксимировать траекторию шаг за шагом

Для решения данной задачи выбран метод Эйлера как простейший и явный. Он заключается в том, что на каждом шаге с шагом требуется рассчитывать:

|  |  |
| --- | --- |
|  | (7) |
| (8) |
| (9) |
| (10) |
| (11) |
| (12) |
| (13) |
| (14) |

где  – шаг интегрирования.

При  возникает особенность – деление на ноль. Реализована следующая защита: если , то расчет прерывается, где  м – пороговое значение столкновения с центром.

## **Энергетические характеристики и момент импульса**

Для анализа энергетических характеристик системы также вычисляются для каждого шага с шагом :

* Кинетическая энергия:

|  |  |
| --- | --- |
|  | (15) |

* Потенциальная энергия:

|  |  |
| --- | --- |
|  | (16) |

* Полная энергия:

|  |  |
| --- | --- |
|  | (17) |

* Момент импульса:

|  |  |
| --- | --- |
|  | (18) |

# Исходные данные

Для моделирования плоского движения тела в данной задаче необходимы следующие изначальные данные, которые пользователь вводит после запуска программы:

* + масса точки m (кг);
  + модуль силы притяжения F (Н);
  + начальные координаты , (м);
  + начальные скорости , ();
  + время моделирования T (с);
  + шаг интегрирования (с);

Согласно варианту №23, исходные данные имеют следующие значения:

Таблица 1 – Исходные данные варианта №

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| m | F |  |  |  |  | T |  |
|  |  |  |  |  |  |  |  |

**3.4.1 Начальные кинематические параметры**

* Расстояние до центра:  м
* Модуль скорости:  м/с

**3.4.2 Начальные энергетические параметры**

* Потенциальная энергия:  Дж
* Кинетическая энергия:  Дж
* Полная энергия:  Дж

**3.4.3 Начальный момент импульса**

* Момент импульса:  кг·м²/с

# **Алгоритм решения задачи**

Алгоритм решения задачи моделирования движения материальной точки в поле центральной силы состоит из следующих основных этапов:

1. Ввод исходных данных. Вводятся данные, указанные разделе 3. Проверка введенных значений на корректность: T > 0, > 0, , числа правильно записаны.
2. Расчет производных данных для начального момента времени по формулам 7–9, 15–18.
3. Вычисление всех характеристик движения тела для следующего момента времени на основе характеристик прошлого шага по формулам 7–14.
4. Расчёт энергетических характеристик и момента импульса для текущего момента времени по формулам 15–18.
5. Вывод результатов в табличной форме. Выводятся значения следующих параметров:

* время моделирования t (с);
* координаты точки x, y (м);
* составляющие скорости , ();
* расстояние до центра притяжения (м);
* модуль скорости ();
* потенциальная энергия (Дж);
* кинетическая энергия (Дж);
* Полная механическая энергия (Дж);
* Момент импульса ();

1. Повторное выполнение пунктов 3–5, пока не достигнуто конечное время, либо не произошло столкновение с центром.

# **Блок-схема решения задачи**

m, F, , , , , T,

Данные корректны

Расчёт: , , , , ,

Вывод начальных данных в таблицу

t < T

Расчет , , , , , , , , , по формулам 7–18

 м

Занесение результатов в таблицу

# **Листинг программы**

## Структура программы

## Программа состоит из следующих основных файлов:

* widget.h - объявление класса Widget и его методов;
* widget.ui - описание графического интерфейса;
* widget.cpp - реализация логики программы;
* main.cpp - точка входа в приложение;
* CMakeLists.txt - конфигурация сборки проекта.

## Основной класс программы

Класс Widget (описан в widget.h) – основной класс приложения, управляющий графическим интерфейсом и вычислительными процессами.

Основные элементы класса:

01: public:

02: Widget(QWidget \*parent = nullptr); // конструктор

03: ~Widget(); // деструктор

04: QStandardItemModel \*TableModel; // модель таблицы результатов

05: public slots:

06: void Simulation(); // метод моделирования

06: private:

07: Ui::Widget \*ui; // указатель на интерфейс

## Таблица результатов

Таблица результатов реализована с помощью модели таблицы QStandardItemModel и является атрибутом класса Widget. Её создание выполняется в конструкторе класса Widget:

01: TableModel = new QStandardItemModel(0, 11);

02: TableModel->setHeaderData(0, Qt::Horizontal, "Время");

03: TableModel->setHeaderData(1, Qt::Horizontal, "x");

04: TableModel->setHeaderData(2, Qt::Horizontal, "y");

05: TableModel->setHeaderData(3, Qt::Horizontal, "Vx");

06: TableModel->setHeaderData(4, Qt::Horizontal, "Vy");

07: TableModel->setHeaderData(5, Qt::Horizontal, "r");

08: TableModel->setHeaderData(6, Qt::Horizontal, "V");

09: TableModel->setHeaderData(7, Qt::Horizontal, "U");

10: TableModel->setHeaderData(8, Qt::Horizontal, "K");

11: TableModel->setHeaderData(9, Qt::Horizontal, "E");

12: TableModel->setHeaderData(10, Qt::Horizontal, "L");

13: ui->Table->setModel(TableModel);

14: ui->Table->horizontalHeader()->setSectionResizeMode(QHeaderView::Stretch); //”растягиваание” столбцов по всей ширине таблицы

Фрагмент создаёт табличную модель для отображения результатов расчёта с заданными заголовками.

## Основной алгоритм моделирования

Simulation (описана в widget.cpp) – функция, выполняющая численное моделирования движения материальной точки.

Проверка корректности введённых числовых данных при помощи регулярных выражений:

01: int AllNumberIsRight = 1; //сохраняет состояние правильности введённых данных

02: QList<QLineEdit\*> ListOfLineEdit = {ui->EditWeight, ui->EditForce,

03: ui->EditX0, ui->EditY0, ui->EditVx0,

04: ui->EditVy0, ui->EditTime, ui->EditStepOfIntegration};

05: QRegularExpression RXnumber("^-?\\d\*\\.?\\d+$");//регулярное выражение действительного числа

06: QRegularExpressionMatch match;

07: while(itLineEdit != ListOfLineEdit.end())

08: {

09: match = RXnumber.match((\*\*itLineEdit).text());

10: if(!match.hasMatch())

11: {

12: AllNumberIsRight = 0;

13: QMessageBox::warning(this, "Ошибка", "Введено не число");

14: break;

15: }

16: itLineEdit++;

17: }

Инициализация начальных данных и расчет производных характеристик:

01: QHash<QString, qreal> data({{"m", ui->EditWeight->text().toDouble()},

02: {"F", ui->EditForce->text().toDouble()},

03: {"x", ui->EditX0->text().toDouble()},

04: {"y", ui->EditY0->text().toDouble()},

05: {"Vx", ui->EditVx0->text().toDouble()},

06: {"Vy", ui->EditVy0->text().toDouble()},

07: {"T", ui->EditTime->text().toDouble()},

08: {"h", ui->EditStepOfIntegration->text().toDouble()}});

09: data.insert("r", sqrt(pow(data["x"], 2) + pow(data["y"], 2)));

10: data.insert("V", sqrt(pow(data["Vx"], 2) + pow(data["Vy"], 2)));

11: data.insert("U", -data["F"] \* data["r"]);

12: data.insert("K", 0.5 \* data["m"] \* pow(data["V"], 2));

13: data.insert("E", data["U"] + data["K"]);

14: data.insert("L", data["m"] \* (data["x"]\*data["Vy"] - data["y"]\*data["Vx"]));

QHash используется по причине наличия ключей, что упрощает работу с данными.

Цикл численного интегрирования

01: while(data["t"] < data["T"])

02: { //расчет необходимых данных

03: data["ax"] = ((-1)\*data["F"]/data["m"])\*(data["x"]/data["r"]); //ускорения рассчитываются для момента n шага, так как оно необходимо для расчета шага n+1

04: data["ay"] = ((-1)\*data["F"]/data["m"])\*(data["y"]/data["r"]);//координаты, скорости, расстояние до центра, энергетические характеристики и момент импульса рассчитываются для n+1 шага

05: data["x"] = data["x"] + data["h"]\*data["Vx"];

06: data["y"] = data["y"] + data["h"]\*data["Vy"];

07: data["Vx"] = data["Vx"] + data["h"]\*data["ax"];

08: data["Vy"] = data["Vy"] + data["h"]\*data["ay"];

09: data["V"] = pow(pow(data["Vx"], 2) + pow(data["Vy"], 2), 0.5);

10: data["t"] = data["t"] + data["h"];

11: data["r"] = pow(pow(data["x"], 2) + pow(data["y"], 2), 0.5);

12: //защита от столкновения с центром

13: if(data["r"] < pow(10, -6))

14: {

15: qDebug() << "Столкновение с центром";

16: break;

17: }

18: data["U"] = (-1)\*data["F"]\*data["r"];

19: data["K"] = 0.5\*data["m"]\*pow(data["V"], 2);

20: data["E"] = data["K"] + data["U"];

21: data["L"] = data["m"]\*(data["x"]\*data["Vy"] - data["y"]\*data["Vx"]);

22: items.append(new QStandardItem(QString::number(data["t"], 'g', 4)));

23: //занесение в массив items (QList<QString>) остальных необходимых для вывода в таблицу данных по аналогии со строчкой 22 в строчках 23-32

..

33: TableModel->appendRow(items); //занесение результатов в таблицу

34: items.clear(); //очищение массива items для заполнения в следующей итерации цикла

}

## Графический интерфейс

Интерфейс создан отчасти с помощью QtDesigner.

Группа "Ввод начальных данных" содержит поля для ввода:

* масса m;
* модуль центральной силы F;
* начальные условия (, , , );
* параметры интегрирования (T, h);
* кнопки управления ("Смоделировать", "Закрыть").

Также справа есть таблица для отображения расчетных данных**.** Таблица содержит 11 столбцов со следующими заголовками:

Таблица 2 – Заголовки таблицы результатов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Время | *x* | *y* |  |  | *r* |  |  |  |  |  |

# Таблица результатов расчётов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Время | *x* | *y* |  |  | *r* |  |  |  |  |  |
| 0 | 1.8 | 0 | 0 | 2 | 1.8 | 2 | -6.48 | 1.8 | -4.68 | 3.24 |
| 1 | 0.1 | 1.36 | -3 | -0.10 | 1.36 | 3 | -4.91 | 4.06 | -0.85 | 3.66 |
| 2 | -2.1 | -0.21 | -0.33 | -2.34 | 2.02 | 2.36 | -7.26 | 2.51 | -4.75 | 4.16 |
| 3 | -0.66 | -1.79 | 2.79 | -0.15 | 1.91 | 2.79 | -6.86 | 3.5 | -3.36 | 4.58 |
| 4 | 1.88 | -0.22 | 1.19 | 2.9 | 1.89 | 3.14 | -6.8 | 4.43 | -2.38 | 5.14 |
| 5 | 1.31 | 2.18 | -2.17 | 1.15 | 2.54 | 2.45 | -9.15 | 2.71 | -6.44 | 5.61 |
| 6 | -1.28 | 1.54 | -2.25 | -2.61 | 2 | 3.45 | -7.2 | 5.35 | -1.85 | 6.11 |
| 7 | -1.88 | -1.68 | 1.3 | -2.81 | 2.52 | 3.1 | -9.08 | 4.32 | -4.76 | 6.72 |
| 8 | 0.36 | -2.9 | 2.67 | 0.79 | 2.92 | 2.78 | -10.5 | 3.49 | -7.02 | 7.22 |
| 9 | 2.25 | -0.45 | 0.39 | 3.77 | 2.3 | 3.79 | -8.26 | 6.46 | -1.8 | 7.79 |
| 10 | 0.93 | 2.82 | -2.71 | 1.86 | 2.97 | 3.29 | -10.7 | 4.87 | -5.82 | 8.44 |
| 11 | -1.85 | 2.82 | -2.24 | -1.98 | 3.37 | 2.99 | -12.1 | 4.02 | -8.11 | 8.98 |
| 12 | -2.68 | -0.36 | 1.08 | -3.84 | 2.7 | 3.99 | -9.71 | 7.15 | -2.57 | 9.58 |