**Министерство науки и высшего образования Российской Федерации**



**федеральное государственное бюджетное образовательное учреждение   
высшего образования**

**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ им. А.И. ГЕРЦЕНА»**

Институт компьютерных наук и технологического образования

Кафедра компьютерных технологий и электронного обучения

**Курсовая работа**

**Исследования задач баллистики средствами информационных технологий**

Направление подготовки: «Информатика и вычислительная техника»

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Гончарова

« \_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Автор работы студент 2 группы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.А. Логинова

« \_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Санкт-Петербург

2018

# ОГЛАВЛЕНИЕ

[**ОГЛАВЛЕНИЕ**](#_oqlga5qc1bxi) **2**

[**ВВЕДЕНИЕ**](#_28uah2p39lta) **3**

[**ГЛАВА 1 ОБЗОР НАУЧНО-МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ ПО ПРОБЛЕМЕ И РЕАЛИЗАЦИИ ИССЛЕДОВАНИЯ**](#_7odroc7cwic0) **5**

[1.1 Основные понятия и используемые средства реализации](#_sztnkoungwwq) 5

[1.2 Описание применяемых особенностей языка программирования](#_bnkq09tt9s54) 6

[1.3 Описание применяемых особенностей электронных таблиц](#_dib57mm2e7du) 8

[**ГЛАВА 2 РЕШЕНИЕ И ВИЗУАЛИЗАЦИЯ РЕШЕНИЯ ЗАДАЧИ ИССЛЕДОВАНИЯ**](#_c2muw2yrkplq) **9**

[2.1 Начальные данные и вывод формул](#_qe60i7wobgyp) 9

[2.2 Решение задачи при помощи Pascal ABC](#_uyxrv9qsa6y) 11

[2.3 Визуализация решения задачи при помощи MS Excel 2013](#_q4i1nyx6sipq) 14

[**ЛИТЕРАТУРА**](#_9gv7gd78wfqs) **18**

[**ПРИЛОЖЕНИЕ А**](#_xudf1xglko81) **19**

[**ПРИЛОЖЕНИЕ Б**](#_m4pufdhqwila) **21**

# 

# ВВЕДЕНИЕ

Первые упоминания оружия приходятся еще на времена первобытного общества. Но в то время и речи не шло об оружие, способном стрелять на дальние расстояния. Первые упоминания аналогов современного метательного оружия приходятся на времена ранней античности (VIII век до н.э - IV век до н.э). Среди них такие виды оружия как лук, катапульта, эвтитон - стреломет, полибол.

В наше время, благодаря развитию и механизации технологий производства, а также гонке вооружений, начатой еще в период холодной войны между Советским Союзом и Соединенными Штатами в 1946-1991 годах прошлого столетия, с каждым годом оружие приобретает все бОльше модернизаций, способствующие совершенствованию качества поражений целей. Также развитие в отрасли производства оружия служит увеличению его вариаций и модификаций.

Развитие в технологиях оружие-производства не имели бы места быть без прогресса в области информационных технологий, которые позволили механизировать то, что раньше приходилось выполнять людям. Так, были созданы специальные программы, позволяющие тестировать в виртуальном режиме оружие еще до его непосредственного изготовления. Это позволило сократить число “неудачных” вариаций оружия, которые могли бы привести к катастрофическим результатам.

Безусловно, развитие военной техники и нарастание военной мощи государств не влечет за собой положительных последствий, тем более в том случае, если оружие будет применено против гражданского населения. Стоит также отметить, что оружие в руках неподготовленного человека может привести к печальному исходу. Поэтому исследование в данной работе носит лишь гипотетический характер и не ставит перед собой цели для применения полученные данных на практике.

В работе проведено исследование, которое затрагивает задачи баллистики, их графическое и программное отображение посредством информационных технологий.

**Объект исследования**: задачи баллистики на примере стрельбы по стационарным мишеням из нарезного оружия.

**Целью исследования** является программная и графическая реализация, полученных в ходе проведения ряда теоретических преобразований, вычислений.   
**Гипотеза исследования.** Предполагается, что при изменении дальности объекта (мишени), оружия, из которого производится выстрел, а также при изменении угла расположения дула оружия относительно горизонта, будет меняться точность попадания в мишень.

**Практическая значимость.** Результаты исследования позволят установить связь между изменением начальных параметров, при которых производится выстрел и точностью попадания в мишень. Полученные данные позволят графически отразить закономерность, а также дадут наглядное представление о решении задач баллистики путем реализации их при помощи информационных технологий.

**Задачи исследования:**

1. Построить модель траектории полета снаряда (пули) до мишени при идеальных условиях.
2. Выявить закономерность между изменениями точности выстрела и меняющихся параметрах.
3. Реализовать данные при помощи языка программирования Pascal на базе компилятора PascalABC.
4. Реализовать полученные данные графически при помощи электронных таблиц MS Excel.

# ГЛАВА 1 ОБЗОР НАУЧНО-МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ ПО ПРОБЛЕМЕ И РЕАЛИЗАЦИИ ИССЛЕДОВАНИЯ

## 1.1 Основные понятия и используемые средства реализации

Физика изучает фундаментальные и, вместе с тем, наиболее общие закономерности явлений природы, свойства и строение материи, законы ее движения. Физика изучает количественные закономерности явлений и потому относится к точным наукам. Понятия и законы физики лежат в основе всего естествознания[[1]](#brr971ibnshx). Исследование, представленное в данной курсовой работе, относится к разделу Физики - Механике. Механика делится на три части: кинематику, динамику и статику. В кинематике изучают чисто геометрические формы механических движений материальных объектов без учета условий и причин, вызывающих и изменяющих эти движения. Важным свойством механического движения является его относительность[[2]](#s6a0ur3nln2t). Механическое движение состоит в перемещении тел или их частей относительно друг друга.. При нем происходит изменение взаимного расположения тел, поэтому необходимо указывать относительно какого тела происходит движение[[3]](#lxbhypq78p2t). Систему отсчета образует тело, относительно которого рассматривается движение. Системы отсчета делятся на итерационные и неитерационные. Итерационная система отсчета - та, относительно которой тело движется равномерно и прямолинейно или покоится. Нетерационная система отсчета (СО) связана с ускоренно движущимся телом[[2]](#s6a0ur3nln2t).

Следует также ввести понятие траектории. Траектория это линия, которую описывает при своем движении в пространстве точка[[4]](#dnlnh6gz4iw4). Данная терминология служит основой понимания принципов построения и описания графиков траекторий движущейся точки в пространстве.

Исследование в области изменения положения снаряда(пули) в пространстве с течением времени относится к подразделу, изучаемое баллистикой. Термин “баллистика” определяет науку о движении тела, брошенного в пространстве. Именно на баллистике строятся расчеты в области артиллерии и стрелкового оружия, так что баллистику сейчас можно трактовать как науку о движении снарядом, пуль, а также баллистических ракет, хотя баллистика включает в себя гораздо большее количество подразделов[[5].](#twdi8e2kbnq4)

Задача баллистики, рассматриваемая в данной работе относится к категории задач характеризующимся как “движение тела, брошенного под углом к горизонту”. В ходе исследования сопротивлением воздуха пренебрегаем. Описание законов и формул, используемых при решении данной задачи, подробно раскрыто в Практическом разделе настоящей работы.

Вычисления, производимые в ходе исследования, визуализированы и механизированные при помощи двух средств, относящимся к средствам информационных технологий: электронных таблиц Excel MS-офиса версии 2013 года для построения графиков траектории полета снаряда и языка программирования Pascal ABC, реализованного в компиляторе PascalABC.NET версии 3.4.2, размещенного под свободной лицензией LGPLv3.

## 1.2 Описание применяемых особенностей языка программирования

Язык программирования - это формальная знаковая система (набор команд), которые понимает компьютер. Язык программирования определяет набор лексических, синтаксических и семантических правил, используемых при написании алгоритмов компьютерных программ[[7]](#cs7u46uakfa1). Программа - это последовательность предписаний (команд), записанных на языке, понятном исполнителю[[8]](#9cxq9lrh46h5). Любая программа обрабатывает некоторые данные. Данные могут быть представлены как константы, либо как переменные. Главное отличие переменных от констант в том, что переменные меняются в ходе производимых вычислений, а константы остаются без изменений. Существует различие и в том, как задаются переменные и константы. Константы задаются после названия программы и указания используемых модулей при помощи команды const и оператора приравнивания. После значения каждой константы следует точка с запятой (;), тип данных при этом не указывается. Переменные же задаются в блоке, идущем после const - var, и перечисляются через запятую, если они принадлежат к одному типу данных, который задается после перечисления блока переменных при помощи символов двоеточие(:). Присвоение некоторой переменной или группе переменных типа данных подразумевает под собой отведение каждой отдельной переменной под данным типом данных определенного количества ячеек памяти, для хранения. Поэтому при реализации программы нужно учитывать не только количество выполняемых команд, но и количеств переменных, потому что каждая переменная это дополнительная затрата памяти, а следовательно замедление вычислений, которые при работе с большими значениями данных также замедляются. Так для рационализации проводимых вычислений в программе было использовано несколько типов данных для переменных и констант: real, являющийся вещественным и занимающие 8 байт из расчета на одну переменную либо константу, и boolean, относящийся к логическому типу данных, то есть способному принимать только два значения True (истина - 1) и False (ложь - 0). Переменные и константы данного типа занимают 1 байт из расчета на одну переменную или константу. Одним из основных отличий констант от переменных, как уже упоминалось ранее, является то, что значение констант задаются непосредственно при вводе имени константы в блоке const, через операцию приравнивания, а для задания значения переменной используются либо команда ввода данных с клавиатуры в диалоговом окне - read или readln, если после ввода значения переменной требуется перенести курсор на следующую строку, либо непосредственно в коде программы при помощи оператора присваивания, имеющего вид двоеточия и равно “:=”. Оператор присваивания используется для закрепления за переменной не только числового или буквенного значения, но и для присваивания результата выполнения некоторых операций. Так, сумма двух переменных может быть присвоена третьей переменной. Когда работа затрагивает действия с большим количеством переменных и констант, а также при большом объеме вычислений и ветвлений удобно использовать пользовательские комментарии, которые можно использовать, например, для описания в коде программы кратких характеристик переменных. Введение комментариев в программе данного исследования осуществлялось при помощи символов // (слэш), при этом важно, что комментарии не обрабатываются компьютером при проведении вычислений, а значит, наличие комментариев не влияет на скорость проводимых вычислений, и при этом заметно помогают составителю ориентироваться в коде программы, а также оставлять какие-либо пометки касательно недочетов при доработке программы.

Большинство основных вычислений проводилось при использовании комбинированных циклических итерационных вычислительных процессов. Основной характеристикой данного вычислительного процесса является наличие вложенных циклов с неизвестным количеством повторений, т.е вложенных итерационных циклических вычислительных процессов, которые были реализованы при помощи оператора цикла **repeat** *операторы* **until** *условие* , которое можно понимать как: **повторять** *<операторы>* **до тех пор, пока не выполнится** *<условие>*[[7]](#cs7u46uakfa1). Характеристика комбинированного вычислительного процесса также обусловлена наличием в теле цикла условного оператора выбора **if...then..else**, который способствует организации вывода данных и условий выхода из циклов, при выполнении условий, указанных непосредственно после оператора **if** перед **then,** после которого идет описание команд, представленных к реализации в случае, если условие выполняется. Также, для рационализации проверки был введен оператор **else**, который выполняет действие, если условие **if** не выполняется. Подобная структура упрощает код программы, когда требуется проверить данные, связанные друг с другом.

## 1.3 Описание применяемых особенностей электронных таблиц

Выбор программы из пакета MS Office Excel версии 2013 года был обусловлен значительными обновлениями в данной версии программы, а также многофункциональностью программы по сравнению с аналогами. Excel облегчает задачу ввода в ячейки повторяющихся или изменяющихся стандартным образом значений, называемых прогрессиями[[6]](#qzabk5lkoiwl), что помогает сохранить время при вводе большого объема значений. Для реализации вычислений требовалось ввести формулы. Под формулой в Excel понимается набор чисел и ссылок на числовые ячейки, соединенных знаками математических операций. Для того чтобы задать ссылка на ячейку или на диапазон данных требовалось указать в формуле ее имя либо щелчком (выделение) соответствующей ячейки (диапазона данных). После завершения ввода формулы в таблице отображается результат вычисления. При этом формулу можно увидеть либо в строке формул либо в режиме редактирования формул (клавиша F2)[[6]](#qzabk5lkoiwl). Также, используемые в формуле ячейке выделяются цветной рамкой, которую можно передвигать от одного значения к другому вручную, что заметно упрощает поиск и исправление ошибок, когда требуется произвести многочисленные вычисления. При построении траектории движения снаряда необходимо произвести не только расчеты координат во временной зависимости от времени движения, но и представить их в виде графика. Одним из преимуществ Excel 2013 года является наличие нескольких разнообразных макетов для построения диаграмм, работа с которыми заметно упростилась, в сравнении с предыдущими версиями Office. Для визуализации численных значений была выбрана группа макетов “Точечная диаграмма”, так как именно при помощи данного типа диаграмм строиться график вида параболы, координаты которой получаются в ходе вычислений, представленных в Практической части исследования. Среди макетов группы рациональнее всего использовать диаграмму типа “Точечная диаграмма с гладкими кривыми” из-за того, что среди других вариаций “Точечной диаграммы” ее график отличается плавным отображением, без учета каждой точки, из-за чего график получается гладким, без утолщений и прерываний.

# 

# 

# ГЛАВА 2 РЕШЕНИЕ И ВИЗУАЛИЗАЦИЯ РЕШЕНИЯ ЗАДАЧИ ИССЛЕДОВАНИЯ

## 2.1 Начальные данные и вывод формул

Условие задачи исследования: Найти угол наклона ствола нарезного оружия, при котором возможно попадание по стационарной мишени, расположенной на расстоянии от стрелка и возвышении, и имеющей размеры. Сопротивлением воздуха при решении пренебречь.

Решение: Выберем декартову систему координат x0y с началом отсчета в точке O вылета патрона из ствола. Вдоль оси Ox движение равномерное прямолинейное, как как на патрон действует только сила тяжести, направленная вертикально вниз. Вдоль оси Oy снаряд движется равнозамедленно до верхней точки траектории, а вниз - равноускоренно с ускорением свободного падения, направленного вертикально вниз в любой точке траектории. С учетом этого уравнения движения снаряда координаты имеют вид:

(1)

(2)

Горизонтальная составляющая вектора скорости - это проекция вектора скорости на ось поэтому:

(3)

Подставим (3) в (1) и получим:

(4)

Движение проекции тела на ось будет сначала равнозамедленным. После того, как тело достигнет вершины траектории, проекции скорости станет отрицательной, т.е одного знака с проекцией ускорения, вследствии начнется равноускоренное движение тела вниз. Проекция скорости на ось изменяется со временем по закону:

(5)

Подставим (5) в (2) и получим:

(6)

Для точки падения снаряда на землю:

Подставляем значения в (6) и уравнения координаты принимает вид:

Найдем время полета снаряда:

(7)

Найдем дальность полета :

Подставим в данное уравнение (3) и (7) и преобразуем:

(8)

Данные формулы нужны для решения задачи при помощи средств информационных технологий. При помощи формул (4) и (6) строится траектория полета, через которую будет находится угол попадания в мишень. Важно понимать, что для попадания в цель необходимо, чтобы мишень располагалась на линии траектории полета снаряда, и не располагалась дальше дальности полета снаряда.

Для последующих вычислений потребуется ввести размеры мишени, расстояние до нее, а также тип оружия и снарядов к нему. Возьмем в качестве оружия, из которого будет проводиться выстрел 7,62-мм снайперскую винтовку Драгунова, при стрельбе из которой используются патроны 7,62-54 мм R. Мишень размерами 20см на 10см будет находится от стрелка на расстоянии 500 м. при высоте 100 м. В данной задаче нужно будет построить траекторию полета патрона, так что возьмем реальную начальную скорость патрона равную 860 м/с, и диаметр равный 7,92 мм. [[9]](#y2fvkdkuv5jb).

## 

## 2.2 Решение задачи при помощи Pascal ABC

Полный код программы см. в Приложении А, блок-схема в Приложении Б.

При решении задачи при помощи языка программирования Pascal ABC были использованы следующие идентификаторы:

| Имя | Смысл | Тип |
| --- | --- | --- |
| S | Дальность полета пули | real |
| x2 | Дальность полета в заданный момент времени | real |
| y2 | Высота полета в заданный момент времени | real |
| h | Угол в градусах | real |
| a | Угол в радианах | real |
| diam | Диаметр пули | real |
| t | Время | real |
| d | Высота мишени | real |
| v | Скорость полета пули | real |
| x1 | Расстояние до мишени | real |
| y1 | Высота до нижней границы мишени | real |
| vist | Длина мишени | real |
| e | Шаг изменения угла в градусах | real |
| p, k1, k2, k3 | Переменные проверки | boolean |
| g | Ускорение свободного падения | const |

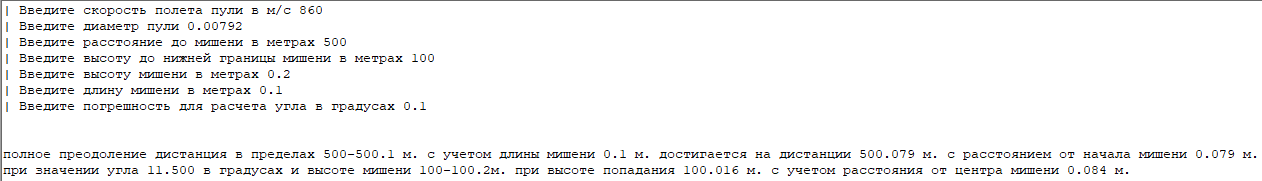
Таблица 1.

Ускорение свободного падения является константой и принимает значение 9,80665 м/с.

Ввод данных осуществляется при помощи команды write, которая выводит сообщение, сообщающее, какой именно параметр должен ввести пользователь. Данные принимаются с клавиатуры при помощи команды readln:

  
Рисунок 2.3.1.

Пользователь видит все выводимые сообщения и значения, которые он вводит в окне вывода в виде:

  
Рисунок 2.3.2.

После ввода пользователем данных идет присвоение начальных значений переменных проверки и угла в градусах:

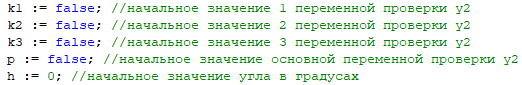


Рисунок 2.3.3.

Далее следует начало внешнего цикла, в котором идет изменение значения угла в градусах, обнуление переменных, используемых во внутреннем цикле, а также считается максимальная дальность полета пули, и если мишень лежит до либо на точке максимума, осуществляется вход в цикл.

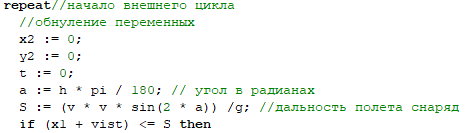


Рисунок 2.3.4.

Выход из внешнего цикла осуществляется в случае, если значение переменной изменилось в ходе вычислений внутреннего цикла, либо, если значение угла превысило 90 градусов.



Рисунок 2.3.5.

Во вложенном, внутреннем цикле, производится вычисление траектории полета снаряда с шагом изменения времени 0,0001. Здесь же происходит проверка: попала ли пуль в цель или нет. В зависимости от разницы между расположением пули и мишени меняются значения переменных проверок, при помощи которых потом осуществляется вывод сообщения результата. Выход из цикла осуществляется двумя путями: в первом случае, если пуля попала в цель при помощи оператора break, в другом случае, если пуля достигает нуля по высоте полета, т.к это может произойти только в случае, если снаряд преодолел всю траекторию.

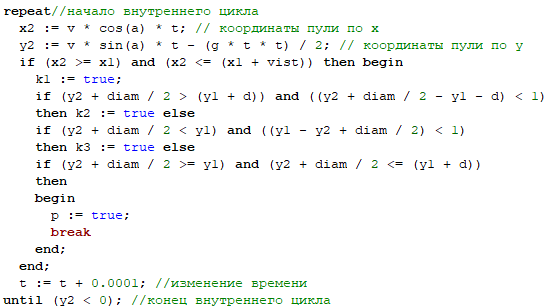


Рисунок 2.3.6.

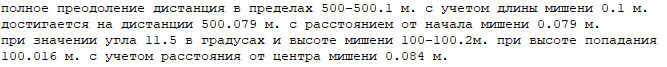
После завершения циклов следует проверка результатов и вывод сообщений о том, было ли попадание или нет. 

Рисунок 2.3.7.

Стоит отметить, что при решении задачи в Pascal существует возможность отказаться от ширины мишени или диаметра пули и задать их нулями, если требуется попасть в конкретную точку на мишени.

## 2.3 Визуализация решения задачи при помощи MS Excel 2013

Для визуализации решения задачи в MS Excel 2013 было построены две таблицы. Первая со значениями угла наклона, начальной скорости и расстояния до мишени. Значения угла было взято из решения задачи при помощи языка программирования, так как перебор значений угла вручную при помощи Excel занимает слишком много времени и не является удобным. Поэтому построение графиков стоит рассматривать только как визуализацию и примерную проверку полученных в ходе вычислений данных, а не как еще один способ решения.

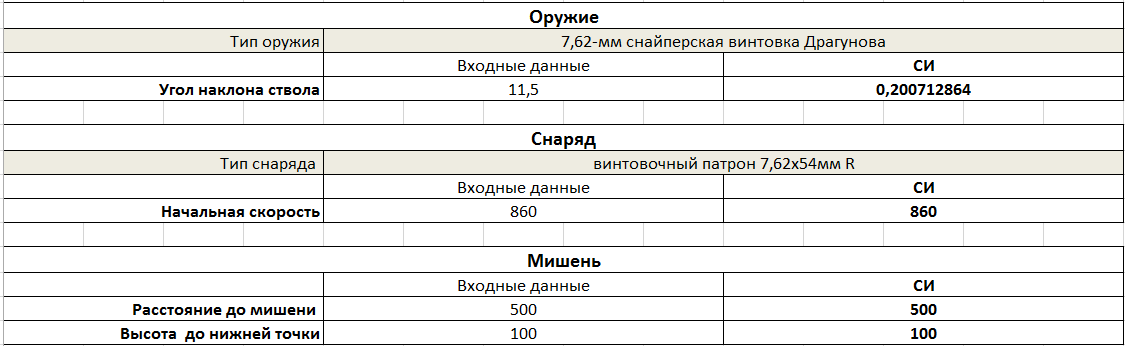


Рисунок 2.2.1.

Вторая таблица со значениями координат траектории полета пули и временем:

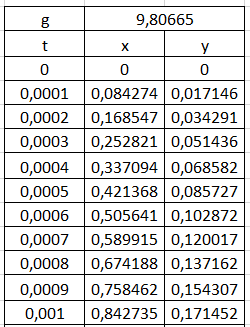


Рисунок 2.2.2. Фрагмент таблицы значений

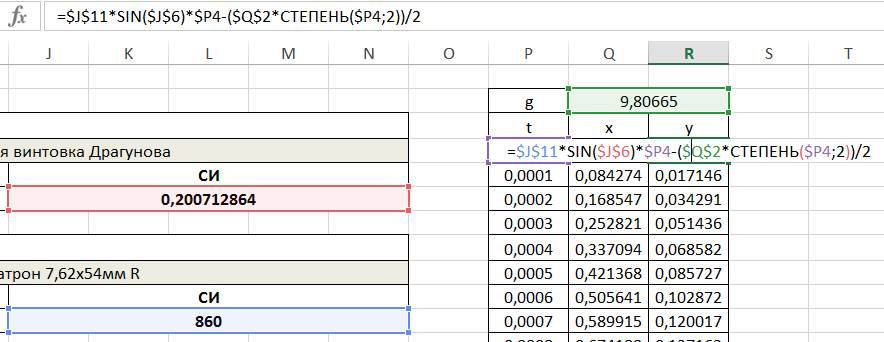
При составлении второй таблицы в ячейки вводились формулы, после чего заполнялись при помощи маркера заполнения до 0 по y.

Рисунок 2.2.3.

Далее было построено два графика. Первый является частью траектории полета пули с точкой - мишенью.

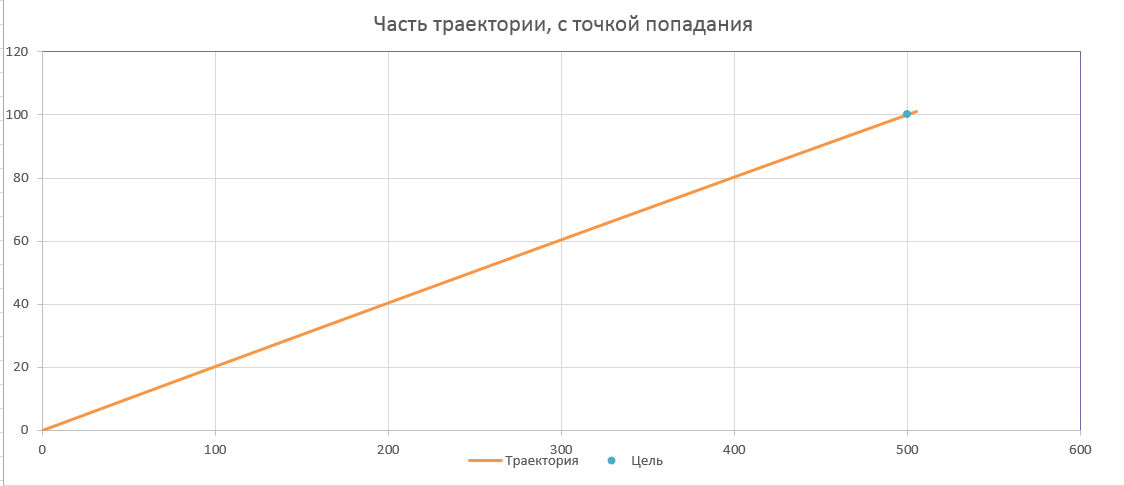


Рисунок 2.2.4.

Второй график представляет собой полную траекторию полета снаряда - параболу - с отмеченной на ней все той же точкой, мишенью.

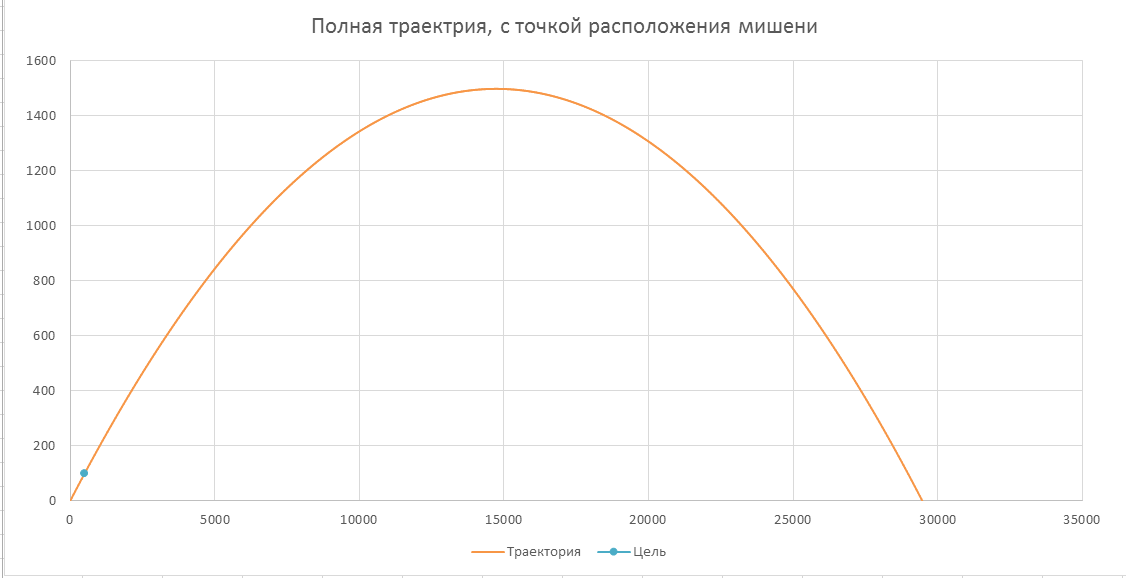


Рисунок 2.2.5. ЗАКЛЮЧЕНИЕ

В ходе проведения исследования была написана программа, которая при помощи перебора и сравнений, выдавала угол, при котором происходит попадание в мишень с заданными размерами. Важно отметить, что в исследовании также было использовано построение графика траектории полета пули, основанное на данных, полученных в ходе вычислений программы. Также стоит упомянуть, что у каждого из способов есть свои преимущества и недостатки. Так, преимуществом решения задачи посредством языка программирования является точный результат: в текстовом сообщение выдается произошло попадание или нет, и если да, то на каком расстоянии от мишени, в то время как при решении задачи при помощи электронных таблиц нельзя учесть размеры мишени и снаряда, и сказать произошло ли попадание точно нельзя, потому что траектория и цель представляют собой набор точек произвольного размера, а при работе с малоразмерными мишенями и снарядами это может вызвать ошибку - по графику попадание будет, хотя на деле нет. Преимуществом языка программирования перед Excel служит и то, что в данном способе нахождение угла является автоматизированным и угол не нужно подбирать самим, исследуя график, меняя область построения графика. При этом важно, что при работе с Excel строится наглядная траектория и можно проследить, как происходит движение снаряда, где будет точка максимума и на какое максимальное расстояние долетит снаряд, если пренебречь мишенью. Для вывода тех же данных на Pascal, потребовался бы ввод дополнительных переменных, а также расчет формул, что сделало бы программу менее быстрой и оптимизированной. Также, при расчете значений в Excel пользователь видит, как в таблице, так и на графике, в какой точке расположен снаряд в тот или иной момент времени, что в программе скрыто от пользователя.

Таким образом, каждый из способов имеет свои преимущества и недостатки, поэтому при решении задач на баллистику следует использовать оба способа: решение при помощи языка программирования для того, чтобы понять можно ли вообще попасть в цель и для выявления в таком случае значения угла, и электронные таблицы в качестве визуализации полученных данных и построения траектории, если это требуется.

# 

# ЛИТЕРАТУРА

1. Леденёв А.Н. Физика. Механика. - ISBN: 5-9221-0461-6 изд. - М: "Физматлит", 2005. - 240 с.

1. Горбунов А.К., Панаиотти Э.Д., Силаева Н.А. Курс физики для довузовской подготовки: учебное пособие. - ISBN: 978-5-7038-3156-4 изд. - М: Московский государственный технический университет имени Н.Э. Баумана, 2008. - 340 с.

1. Бобров П.П Основные понятия механики: учебное пособие. - Омск: Омский государственный педагогический университет, 2013. - 180 с.

1. Аксенова Е.Н Общая физика. Механика (главы курса). - 2 изд. - М, СПб, Краснодар: Издательство Лань, 2018. - 128 с

1. Дмитриевский А.А, Лысенко Л.Н Внешняя баллистика. - 4 изд. - М.: Издательство Машиностроение, 2005. - 608 с.
2. Миркин В.В. Подготовка документов с использованием табличного процессора ms excel 2013 (пособие к практическим занятиям). - Томск: Национальный исследовательский Томский государственный университет, 2015. - 36 с.

1. Ерёмен О.Ф Методическое пособие по программированию на языке Pascal ABC. - Моздок: 2009. - 49 с.

1. Долинер Л.И. Основы программирования в среде PascalABC.NET. - ISBN 978-5-7996-1260-3 изд. - Екатеринбург: Издательство Уральского университета , 2014. - 130 с.

1. Патрон 7.62х53R // Оружие и боеприпасы URL: http://weaponland.ru/board/patron\_762kh53r/43-1-0-229 (дата обращения: 10.12.2018).
2. Движение тела, брошенного под углом к горизонту // Научная библиотека избранных естественно-научных изданий URL: http://sernam.ru/book\_phis\_t1.php?id=116 (дата обращения: 10.12.2018).

# ПРИЛОЖЕНИЕ А

const  
 g = 9.80665;// ускорение свободного падения   
  
var  
 S, x2, y2, a, diam, t, d, v, x1, y1, vist, e, h: real;  
 p, k1, k2, k3: Boolean;  
  
begin  
 write('| Введите скорость полета пули в м/с '); readln(v);// скорость полета пули  
 write('| Введите диаметр пули '); readln(diam); //диаметр пули   
 write('| Введите расстояние до мишени в метрах '); readln(x1); // расстояние до мишени  
 write('| Введите высоту до нижней границы мишени в метрах '); readln(y1); // нижняя граница мишени  
 write('| Введите высоту мишени в метрах '); readln(d); // высота мишени, расстояние от нижней границы до верхней  
 write('| Введите длину мишени в метрах '); readln(vist); // длина мишени   
 write('| Введите погрешность для расчета угла в градусах '); read(e);//шаг изменения угла;  
 writeln;  
 k1 := false; //начальное значение 1 переменной проверки y2  
 k2 := false; //начальное значение 2 переменной проверки y2  
 k3 := false; //начальное значение 3 переменной проверки y2  
 p := false; //начальное значение основной переменной проверки y2  
 h := 0; //начальное значение угла в градусах  
 repeat//начало внешнего цикла  
 //обнуление переменных  
 x2 := 0;   
 y2 := 0;  
 t := 0;  
 a := h \* pi / 180; // угол в радианах  
 S := (v \* v \* sin(2 \* a)) /g; //дальность полета снаряд   
 if (x1 + vist) <= S then  
 repeat//начало внутреннего цикла  
 x2 := v \* cos(a) \* t; // координаты пули по x  
 y2 := v \* sin(a) \* t - (g \* t \* t) / 2; // координаты пули по y  
 if (x2 >= x1) and (x2 <= (x1 + vist)) then begin  
 k1 := true;  
 if (y2 + diam / 2 > (y1 + d)) and ((y2 + diam / 2 - y1 - d) < 1) then k2 := true else   
 if (y2 + diam / 2 < y1) and ((y1 - y2 + diam / 2) < 1) then k3 := true else   
 if (y2 + diam / 2 >= y1) and (y2 + diam / 2 <= (y1 + d)) then  
 begin  
 p := true;  
 break  
 end;  
 end;  
 t := t + 0.0001; //изменение времени   
 until (y2 < 0); //конец внутреннего цикла  
 if p = true then break else h := h + e;  
 until h >= 90; //конец внешнего цикла  
 writeln;  
 if (p = false) then   
 if k1 = false then writeln('Попасть в цель невозможно, так как не достигается необходимая дистанция') else   
 begin  
 if (k2 = true) and (k3 = true) then write('Нет попадания. Нужная дистанция достигается, но траектория полета снаряда не достигает мишени менее, чем на метр') else   
 if (k2 = true) then write('Нет попадания. Нужная дистанция достигается, но траектория полета снаряда превышает мишень более, чем на метр') else   
 write('Нужная дистанция достигается, но траектория полета снаряда не достигает мишень на значение меньше метра')  
 end  
 else begin  
 writeln('полное преодоление дистанция в пределах ', x1, '-', x1 + vist, ' м. с учетом длины мишени ', vist, ' м. достигается на дистанции ', x2:2:3, ' м. с расстоянием от начала мишени ', abs(x1 - x2):2:3, ' м.');  
 writeln('при значении угла ', h, ' в градусах и высоте мишени ', y1, '-', y1 + d, 'м. при высоте попадания ', y2:2:3, ' м. с учетом расстояния от центра мишени ', abs((y1 + d / 2) - y2):2:3, ' м.');  
 end;  
end.

# ПРИЛОЖЕНИЕ Б

