## ЛАБОРАТОРНАЯ РАБОТА № 3. Расчет внутрибаллистического процесса выстрела с использованием модели Лагранжа. Расчет внешебаллистической траектории полета снаряда

### 1. Цель работы

Научиться рассчитывать параметры выстрела из пушки в приближении Лагранжа и рассчитывать оптимальную траекторию полета снаряда.

### 2. Постановка задачи

#### 2.1. Основная задача внутренней баллистики

Основная задача внутренней баллистики состоит в нахождении закона изменения давления пороховых газов и скорости движения снаряда в зависимости от пути или времени при заданных условиях заряжания (прямая задача), и в нахождении рациональных конструктивных данных камеры, канала ствола и условий заряжания, при которых снаряд заданного калибра и массы получит заданную скорость (задача баллистического проектирования орудия).

Если решение прямой задачи однозначно определяет значения баллистических элементов выстрела (давление *,* путь  и скорость ) в точке максимума давления, а также в точке конца горения пороха () и в момент прохождения снарядом дульного среза (), то задача баллистического проектирования допускает множество вариантов решений, окончательный выбор которого зависит от предъявляемых к проектируемой системе тактико-технических требований. Составной и обязательной частью задачи баллистического проектирования является обратная задача внутренней баллистики. Она заключается в определении параметров заряжания орудия, обеспечивающих получение заданного наибольшего давления или заданной дульной скорости снаряда для пороха данной природы.

Приведем систему уравнений внутренней баллистики в осредненных параметрах, когда диаметр камеры заряжания совпадает с диаметром ствола орудия:

, ,



, , (1)

,

,

, , ,

, ,

, ,

где время; относительная доля сгоревшего порохового элемента; относительная толщина сгоревшего свода порохового элемента; толщина свода порохового элемента, сгоревшего к моменту времени ; максимальная толщина горящего свода порохового элемента; зависимость линейной скорости горения пороха от давления; скорость снаряда; масса снаряда в стволе; давление на дно снаряда; площадь поперечного сечения снаряда, на которую воздействуют пороховые газы ( полагается равной площади сечения цилиндрической части канала ствола ); путь, пройденный снарядом к моменту ; среднее давление в заснарядном объеме при выстреле; средняя температура пороховых газов в заснарядном объеме при выстреле; объем камеры; коволюм пороховых газов; масса заряда; плотность материала пороха; сила пороха; удельная газовая постоянная продуктов горения пороха; температура пороховых газов при сжигании пороха в постоянном объеме; коэффициент фиктивности; ; теплоемкость пороховых газов при постоянном давлении; теплоемкость пороховых газов при постоянном объеме; коэффициенты формы порохового элемента; относительная доля сгоревшего порохового элемента к моменту его распада.

Начальные условия:

при , 

, , , , ,

где плотность заряжания; давление форсирования.

Система обыкновенных дифференциальных уравнений может быть решена одним из численных методов. В частности, может быть применен Метод Рунге-Кутта 4-го порядка.

Рассмотрим систему уравнений, записанную в векторной форме:

.  (2)

Пусть отрезок интегрирования  разбит на частичные отрезки ,  с постоянным шагом . Для сеточной функции , аппроксимирующей искомое решение , имеем:

, ,

,



.

Шаг интегрирования  выбирается автоматически, исходя из заданной точности при помощи принципа Рунге, суть которого состоит в следующем. Численное решение задачи осуществляется последовательно с шагом  и с шагом :

Если для заданного  будут выполняться неравенства



где полное число частичных отрезков при соответствующем шаге интегрирования, время движения снаряда по каналу ствола (время выстрела), то получим

.

Для метода Рунге-Кутта 4-го порядка .

Начальное значение шага удобно выбирать по формуле:

, (4)

где ; длина дульной части канала ствола (полный путь снаряда в стволе при выстреле).

Для векторной функции  погрешность вычисляется в заданной норме. Например, в эвклидовой относительной среднеквадратичной норме имеем:

 (5)

и

.

При определенных допущениях решение системы (1) может быть получено аналитически. Так если принять, что порох сгорает мгновенно в начальный момент времени, т. е.

при , 

, , , , ,

то приходим к задаче **Лагранжа** в термодинамическом приближении о разгоне снаряда в цилиндрической трубе предварительно сжатым газом:

,

, (6)

,

,

, .

Приняв в качестве независимой переменной длину пути, пройденного снарядом , и разделяя переменные, из системы уравнений (6) получим:

,

откуда можно получить решение последнего уравнения в квадратурах:

. (7)

Если в качестве начальных значений принять значения параметров газа и снаряда, соответствующих моменту конца горения заряда:

при , 

, , , , ,

то решение ОЗВБ после конца горения заряда  примет вид:

. (8)

#### 2.2. Основная задача внешней баллистики

Основная задача внешней баллистики состоит в нахождении закона изменения траектории полета снаряда от времени (прямая задача) в зависимости от его начальных значений скорости  и угла стрельбы . Обратная задача внешней баллистики состоит в определении указанных выше параметров, если заданы координаты цели.

Приведем систему уравнений внешней баллистики, описывающее движение снаряда в неподвижной атмосфере.

Траектория движения снаряда строится в плоской стартовой системе координат , связанной с точкой расположения орудия и ориентированной по направлению стрельбы (рис. 1). Координаты центра масс снаряда определяются уравнениями:

, , (9)

где – продольная координата; – высота полета; – угол наклона траектории, отсчитываемый от положительного направления оси  против часовой стрелки до направления вектора скорости снаряда; – скорость центра масс снаряда.

Параметры движения снаряда определяются в траекторной системе координат , связанной с центром масс снаряда и ориентированной по вектору скорости (см. рис. 1):

, (10)

, (11)

, .



Рис. 1. Ориентация стартовой  и траекторной  систем координат

Здесь – ускорение силы тяжести; – коэффициент лобового сопротивления снаряда; – скоростной напор воздуха; число Маха; – скорость звука в воздухе; – площадь миделева сечения снаряда; – калибр снаряда; – полетная масса снаряда. Для стандартных условий стрельбы (, мм рт.ст.) скорость звука , плотность воздуха .

Коэффициент лобового сопротивления снаряда определяется по формуле:

,

где коэффициент аэродинамической формы снаряда; эталонный коэффициент сопротивления, выбирается в соответствии с аппроксимацией закона 1943 г. (для осколочно-фугасных снарядов) или закона 1958 г. (для бронебойно-подкалиберных снарядов) (см. табл. 1).

Табл. 1. Зависимости коэффициента лобового сопротивления от числа Маха [[1]](#footnote-1)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *M* | 0,5 | 0,8 | 0,9 | 1,0 | 1,1 | 1,2 | 1,5 | 1,75 | 2,0 | 2,5 | 3,0 |
| Закон 1943 г. | 0,157 | 0,159 | 0,183 | 0,326 | 0,378 | 0,384 | 0,361 | 0,337 | 0,317 | 0,288 | 0,270 |
| Закон 1958 г. | 0,306 | 0,334 | 0,372 | 0,354 | 0,616 | 0,620 | 0,558 | 0,514 | 0,478 | 0,416 | 0,369 |

Численное решение задачи получить методом Рунге-Кутта с автоматическим выбором шага при помощи принципа Рунге, изложенным в п. 2.1.

При отсутствии сопротивления воздуха  система (9)-(11) допускает аналитическое решение:

, , , .

Для дальности полета  и времени полета  имеем

, .

В качестве начального значения шага интегрирования численного решения задачи внешней баллистики принять

.

### 3. Задание на лабораторную работу

1. Ознакомиться с постановкой задачи.
2. По согласованию с преподавателем выбрать конфигурацию артиллерийской системы и параметры заряжания.
3. Определить начальные условия для задачи Лагранжа.
4. Решить задачу Лагранжа численно, приняв , и сравнить с аналитическим решением.

import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import solve\_ivp

# Параметры

m = 0.85 # масса снаряда, кг

Cd = 0.58 # коэффициент сопротивления

A = 0.01 # площадь поперечного сечения, м^2

rho = 1.225 # плотность воздуха, кг/м^3

g = 9.81 # ускорение свободного падения, м/с^2

# Начальные условия

v0 = 800 # начальная скорость, м/с

alpha = 45 # угол вылета, градусы

vx0 = v0 \* np.cos(np.radians(alpha))

vy0 = v0 \* np.sin(np.radians(alpha))

initial\_conditions = [0, 0, vx0, vy0]

# Уравнения движения

def equations(t, y):

x, y\_pos, vx, vy = y

v = np.sqrt(vx\*\*2 + vy\*\*2)

dvxdt = -0.5 \* Cd \* A \* rho \* v \* vx / m

dvydt = -g - 0.5 \* Cd \* A \* rho \* v \* vy / m

return [vx, vy, dvxdt, dvydt]

# Решение

t\_span = (0, 100) # время в секундах

t\_eval = np.linspace(t\_span[0], t\_span[1], 500)

sol = solve\_ivp(equations, t\_span, initial\_conditions, t\_eval=t\_eval)

# График

plt.plot(sol.y[0], sol.y[1])

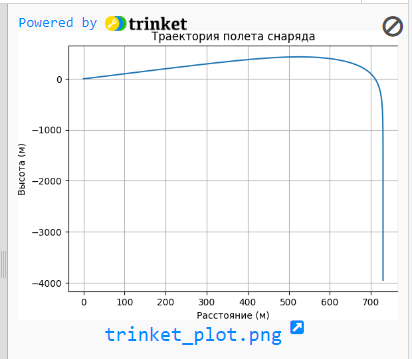
plt.title("Траектория полета снаряда")

plt.xlabel("Расстояние (м)")

plt.ylabel("Высота (м)")

plt.grid()

plt.show()



1. Построить зависимости давления на дно канала и дно снаряда, а также скорости снаряда от времени выстрела.

import numpy as np

import matplotlib.pyplot as plt

# Исходные данные

P0 = 1e5 # Начальное давление, Па

T0 = 2000 # Начальная температура, К

R = 287 # Газовая постоянная, J/(kg·K)

V = 0.0005 # Объем, м^3

m = 0.85 # Масса снаряда, кг

A = 0.005 # Площадь поперечного сечения, м^2 (примерно 5 см^2)

k = 1000 # Коэффициент сгорания

# Время

t = np.linspace(0, 0.01, 100) # Время от 0 до 0.01 с

# Давление в канале (предположим, давление уменьшается с течением времени)

P = P0 \* np.exp(-k \* t) # Давление в канале

# Масса газов (предположим, что она пропорциональна времени)

mass\_gas = (P0 \* V) / (R \* T0) # Масса газов при начальных условиях

# Скорость снаряда (изменяется по времени)

# F = P \* A - mg

# a = F/m = (P \* A - mg) / m

# v(t) = v(0) + ∫a dt

velocity = np.zeros\_like(t)

for i in range(1, len(t)):

a = (P[i] \* A - m \* 9.81) / m # Ускорение

velocity[i] = velocity[i-1] + a \* (t[i] - t[i-1]) # Интегрирование для получения скорости

# Построение графиков

plt.figure(figsize=(12, 6))

# График давления

plt.subplot(1, 2, 1)

plt.plot(t, P, color='blue')

plt.title('Давление в канале от времени')

plt.xlabel('Время (с)')

plt.ylabel('Давление (Па)')

plt.grid()

# График скорости

plt.subplot(1, 2, 2)

plt.plot(t, velocity, color='red')

plt.title('Скорость снаряда от времени')

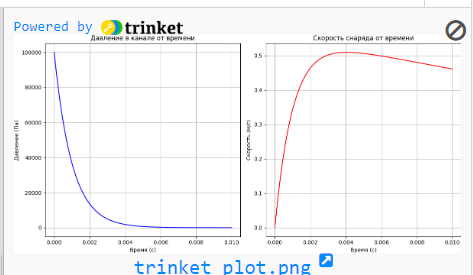
plt.xlabel('Время (с)')

plt.ylabel('Скорость (м/с)')

plt.grid()

plt.tight\_layout()

plt.show()



1. Построить зависимости плотности и температуры пороховых газов от времени выстрела.

import numpy as np

import matplotlib.pyplot as plt

# Исходные данные

T0 = 2000 # Начальная температура, К

rho0 = 1 # Начальная плотность, кг/м^3

k = 100 # Коэффициент охлаждения, s^-1

R = 287 # Газовая постоянная для воздуха, J/(kg·K)

# Время

t = np.linspace(0, 0.01, 100) # Время от 0 до 0.01 с

# Температура

T = T0 \* np.exp(-k \* t)

# Давление (предположим, что оно остается постоянным на короткий период)

P = 1e5 # Давление, Па (например, атмосферное давление)

# Плотность газа по уравнению состояния

rho = P / (R \* T)

# Построение графиков

plt.figure(figsize=(12, 6))

# График температуры

plt.subplot(1, 2, 1)

plt.plot(t, T, color='red')

plt.title('Температура пороховых газов от времени')

plt.xlabel('Время (с)')

plt.ylabel('Температура (К)')

plt.grid()

# График плотности

plt.subplot(1, 2, 2)

plt.plot(t, rho, color='blue')

plt.title('Плотность пороховых газов от времени')

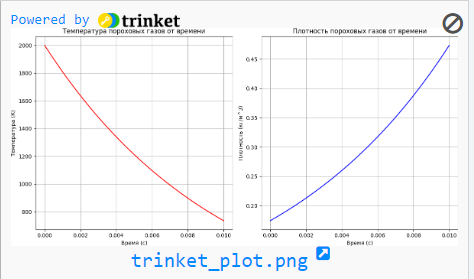
plt.xlabel('Время (с)')

plt.ylabel('Плотность (кг/м^3)')

plt.grid()

plt.tight\_layout()

plt.show()



1. Решить задачу внешне баллистики, приняв , и сравнить с аналитическим решением при  и .

import numpy as np

import pandas as pd

from scipy.integrate import solve\_ivp

# Параметры

g = 9.81 # Ускорение свободного падения, м/с^2

m = 0.85 # Масса снаряда, кг

v0 = 800 # Начальная скорость, м/с

theta\_c0 = 45 # Угол вылета, градусы

# Функция для расчета траектории

def equations(t, y):

x, y\_pos, vx, vy = y

v = np.sqrt(vx\*\*2 + vy\*\*2)

# Сопротивление воздуха учтено

dvxdt = -0.5 \* Cx \* A \* rho \* v \* vx / m

dvydt = -g - 0.5 \* Cx \* A \* rho \* v \* vy / m

return [vx, vy, dvxdt, dvydt]

# Угол в радианах

theta\_rad = np.radians(theta\_c0)

vx0 = v0 \* np.cos(theta\_rad)

vy0 = v0 \* np.sin(theta\_rad)

initial\_conditions = [0, 0, vx0, vy0]

# Сопротивление воздуха

Cx = 10\*\*-3 # Коэффициент сопротивления

A = 0.01 # Площадь поперечного сечения, м^2

rho = 1.225 # Плотность воздуха, кг/м^3

# Решение уравнений с учетом сопротивления

t\_span = (0, 100) # Время в секундах

t\_eval = np.linspace(t\_span[0], t\_span[1], 500)

sol = solve\_ivp(equations, t\_span, initial\_conditions, t\_eval=t\_eval)

# Максимальное расстояние

numerical\_distance = max(sol.y[0])

# Аналитическое решение

analytical\_distance = (v0\*\*2 \* np.sin(2 \* theta\_rad)) / g

# Сравнение результатов

results\_df = pd.DataFrame({

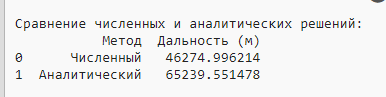
'Метод': ['Численный', 'Аналитический'],

'Дальность (м)': [numerical\_distance, analytical\_distance]

})

print("\nСравнение численных и аналитических решений:")

print(results\_df)



Сравнение численных и аналитических решений:

Метод Дальность (м)

Численный 46274.996214

Аналитический 65239.551478

1. Построить траекторию полета снаряда при различных заданных значениях . Исследовать влияние угла вылета снаряда на дальность стрельбы. Принять , с шагом . Построить таблицу и график.

Табл. 2. Параметры артиллерийской системы[[2]](#footnote-2)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Тип  орудия | ,  дм | ,  дм2 | ,  кг | ,  кг | ,  дм3 | ,  дм | ,  дм | ,  кг |  |
| 4 | 85-мм  ЗП - 1944 | 0,85 | 0,58 | 3,08 | 9,20 | 3,87 | 6,64 | 46,5 | 9,20 | 1,0 |

Теплофизические характеристики выстрела приведены в табл. 3

Табл. 3. Теплофизические характеристики выстрела

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Марка  пороха | ,  МДж/кг | ,  Дж/кг/К | ,  Дж/кг/К | ,  Дж/кг/К |  | ,  дм3/кг | ,  кг/дм3 |
| 4 | 14/7 ВА | 1,013 | 345,1 | 1885,7 | 1540,6 | 0,224 | 1,010 | 1,60 |

import numpy as np

import matplotlib.pyplot as plt

# Параметры

g = 9.81 # Ускорение свободного падения, м/с²

C\_d = 0.5 # Коэффициент сопротивления

rho = 1.225 # Плотность воздуха, кг/м³

A = 0.01 # Площадь поперечного сечения, м² (примерно)

m = 1.0 # Масса снаряда, кг

# Функция для вычисления силы сопротивления

def drag\_force(v):

return 0.5 \* C\_d \* rho \* A \* v\*\*2

# Метод Рунге-Кутты 4-го порядка

def runge\_kutta(v0, theta, dt, total\_time):

theta\_rad = np.radians(theta) # Преобразование угла в радианы

v0\_x = v0 \* np.cos(theta\_rad) # Начальная скорость по x

v0\_y = v0 \* np.sin(theta\_rad) # Начальная скорость по y

# Инициализация переменных

t = 0

x, y = 0, 0

vx, vy = v0\_x, v0\_y

trajectory = [(x, y)] # Список для хранения координат

while y >= 0: # Пока снаряд не приземлится

# Расчет величин для Рунге-Кутты

v = np.sqrt(vx\*\*2 + vy\*\*2)

F\_d = drag\_force(v)

# Определение значений k1, k2, k3, k4

k1vx = -F\_d/m \* (vx/v) \* dt

k1vy = -g - (F\_d/m \* (vy/v)) \* dt

k1x = vx \* dt

k1y = vy \* dt

k2vx = -F\_d/m \* ((vx + k1vx/2)/v) \* dt

k2vy = -g - (F\_d/m \* ((vy + k1vy/2)/v)) \* dt

k2x = (vx + k1vx/2) \* dt

k2y = (vy + k1vy/2) \* dt

k3vx = -F\_d/m \* ((vx + k2vx/2)/v) \* dt

k3vy = -g - (F\_d/m \* ((vy + k2vy/2)/v)) \* dt

k3x = (vx + k2vx/2) \* dt

k3y = (vy + k2vy/2) \* dt

k4vx = -F\_d/m \* ((vx + k3vx)/v) \* dt

k4vy = -g - (F\_d/m \* ((vy + k3vy)/v)) \* dt

k4x = (vx + k3vx) \* dt

k4y = (vy + k3vy) \* dt

# Обновление значений

vx += (k1vx + 2\*k2vx + 2\*k3vx + k4vx) / 6

vy += (k1vy + 2\*k2vy + 2\*k3vy + k4vy) / 6

x += (k1x + 2\*k2x + 2\*k3x + k4x) / 6

y += (k1y + 2\*k2y + 2\*k3y + k4y) / 6

trajectory.append((x, y))

t += dt

return trajectory

# Параметры запуска

v0 = 800 # Начальная скорость, м/с

theta\_values = np.arange(10, 71, 5) # Углы от 10 до 70 с шагом 5

results = []

# Главный цикл для разных углов

for theta in theta\_values:

trajectory = runge\_kutta(v0, theta, 0.01, 10) # dt = 0.01, total\_time = 10

max\_distance = max([point[0] for point in trajectory]) # Максимальная дальность

results.append((theta, max\_distance))

# Вывод результатов

print("Угол (градусы) | Дальность (м)")

for result in results:

print(f"{result[0]:<16} | {result[1]:.2f}")

# График

angles, distances = zip(\*results)

plt.figure(figsize=(10, 6))

plt.plot(angles, distances, marker='o')

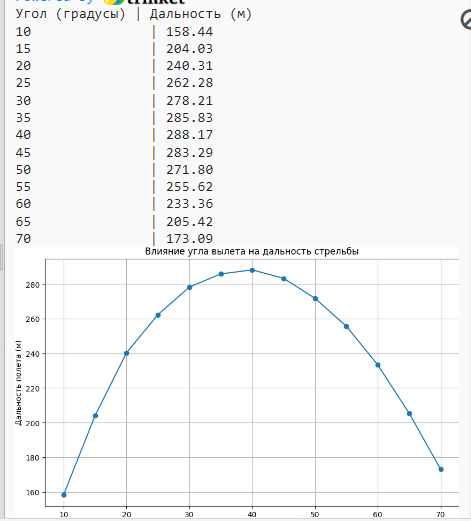
plt.title('Влияние угла вылета на дальность стрельбы')

plt.xlabel('Угол вылета (градусы)')

plt.ylabel('Дальность полета (м)')

plt.grid()

plt.show()



1. Для определения промежуточных значений использовать линейную аппроксимацию. [↑](#footnote-ref-1)
2. Для первых восьми вариантов использовать закон сопротивления 1943 года, для двух последних – 1958 года. [↑](#footnote-ref-2)