

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

**на тему:**

**«Інформаційне та програмне забезпечення системи балістичного  
передбачення»**

**Завідувач**

**випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Сімоновський Ю. В.**

**Студента групи ІНз-71**

**Кашенка Я. А.**

**СУМИ 2021**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую \_\_\_\_\_

Зав. кафедри Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

## **ЗАВДАННЯ**

### **до випускної роботи**

Студента четвертого курсу, групи ІНз-71 спеціальності “Комп’ютерні науки” заочної форми навчання Кашенка Ярослава Аленовича.

**Тема:** “ Інформаційне та програмне забезпечення системи балістичного передбачення”

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 2021 р.

**Зміст пояснювальної записки:** 1) Формування математичного апарату визначення параметрів зовнішньої балістики артилерійського снаряда на траєкторії; 2) Вибір точки прицілювання з урахуванням просторових точок фіксації рухомої цілі; 3) Вибір програмного забезпечення для реалізації окремих компонентів сформованої моделі руху артилерійського снаряда та цілі; 4) Розробка графічного модуля.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

Керівник випускної роботи \_\_\_\_\_ Сімоновський Ю.В.

Завдання прийняв до виконання \_\_\_\_\_ Кашенко Я. А.

## РЕФЕРАТ

**Записка:** 62 стор., 10 рис., 1 табл., 2 додаток, 10 джерел.

**Об'єкт дослідження** — процес прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі.

**Мета роботи** — розробка додатку прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі.

**Методи дослідження** — методи проектування інформаційних систем, методи створення і оптимізації структури баз даних.

**Результати** — створено додаток, який дозволяє аналізувати вхідні дані пункту артилерійської розвідки та прогнозувати точку прицілювання артилерійської системи, яка дозволяє знищити рухому ціль.

ПРОГНАЗУВАННЯ ТОЧКИ ПРИЦІЛЮВАННЯ АРТИЛЕРІЙСЬКОЇ  
СИСТЕМИ, ДОДАТОК, DELPHI

## ЗМІСТ

<b>ВСТУП.....</b>	<b>5</b>
<b>1 АНАЛІТИЧНИЙ ОГЛЯД.....</b>	<b>6</b>
<b>1.1 Аналіз руху артилерійського снаряда під дією сил інерції, гравітації та сили аеродинамічного опору.....</b>	<b>6</b>
<b>1.2. Диференційні рівняння руху центру мас снаряда.....</b>	<b>10</b>
<b>1.3 Прогнозування точки прицілювання .....</b>	<b>29</b>
<b>1.4 Постановка задачі .....</b>	<b>29</b>
<b>2 РЕАЛІЗАЦІЯ ПОСТАВЛЕНОЇ ЗАДАЧІ.....</b>	<b>30</b>
<b>2.1 Додаток прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі .....</b>	<b>30</b>
<b>ВИСНОВОК .....</b>	<b>33</b>
<b>СПИСОК ЛІТЕРАТУРИ .....</b>	<b>34</b>
<b>ДОДАТОК А.....</b>	<b>35</b>
<b>ДОДАТОК Б .....</b>	<b>37</b>

## ВСТУП

Delphi – імперативна, структурована об'єктно орієнтована, багаторівнева мова програмування з структурованою статичною типізацією пермінних. Основною областю використання є створення прикладних додатків [1, 2].

Delphi є мовою програмування яка є похідною Object Pascal. При створенні мови програмування не ставилася мета забезпечити максимальну продуктивність виконувального кода або його лаконічність з метою економії оперативної пам'яті. Спочатку мова програмування мала на меті забезпечення структурованість кода та легкість його аналізу, бо був створений для навчання програмуванню [2, 3, 4].

Спираючись на вищезазначене було обрано саме Delphi для створення додатку прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі.

# 1 АНАЛІТИЧНИЙ ОГЛЯД

## 1.1 Аналіз руху артилерійського снаряда під дією сил інерції, гравітації та сили аеродинамічного опору

Рух снаряда в повітрі як твердого тіла в загальному випадку описується шістьма диференційними рівняннями другого порядку: трьома рівняннями руху центру мас і трьома рівняннями руху снаряда відносно центру мас. При використанні диференціальних рівнянь руху першого порядку їх кількість зростає вдвічі, тобто шість рівнянь описують рух центру мас і шість рівнянь – рух снаряда відносно центру мас [5].

У загальному вигляді розв'язання такої задачі досить складне, тому що необхідно знати значну кількість аеродинамічних коефіцієнтів снаряда, визначення яких є трудомістким і потребує суттєвих матеріальних витрат. Разом з тим, при малих кутах нутації, що має місце в більшості випадків руху артилерійських снарядів, рух центру мас снаряда не суттєво залежить від його руху відносно центру мас. У цьому випадку снаряд можна розглядати як матеріальну точку, на яку діють сила тяжіння і аеродинамічні сили. Рух снаряда описується шістьма рівняннями першого порядку.

Нижче наводиться розв'язання задачі про рух снаряда, який стабілізується під час польоту за допомогою обертання:

снаряд розглядається як матеріальна точка, що рухається під дією сили тяжіння і сил аеродинамічного опору повітря;

поверхня Землі приймається круглою;

приймаються нормальні метеорологічні умови (нормальні значення на земній поверхні та нормальні залежності розподілу метеорологічних елементів за висотою);

враховуються поздовжня  $W_x$  і бокова  $W_z$  складові швидкості вітру.

сили опору повітря при наявності кута нутації  $\delta$  включають наступні складові: силу лобового опору  $R_T$ , що направлена по дотичній до траєкторії руху центру мас снаряда; нормальну силу  $R_N$ , що прикладена в центрі мас і розташована в площині опору; перекидаючий (стабілізуєчий) момент  $M$ , що діє в площині опору (рис. 1).

Іншими складовими аеродинамічних сил будемо нехтувати [6, 7].

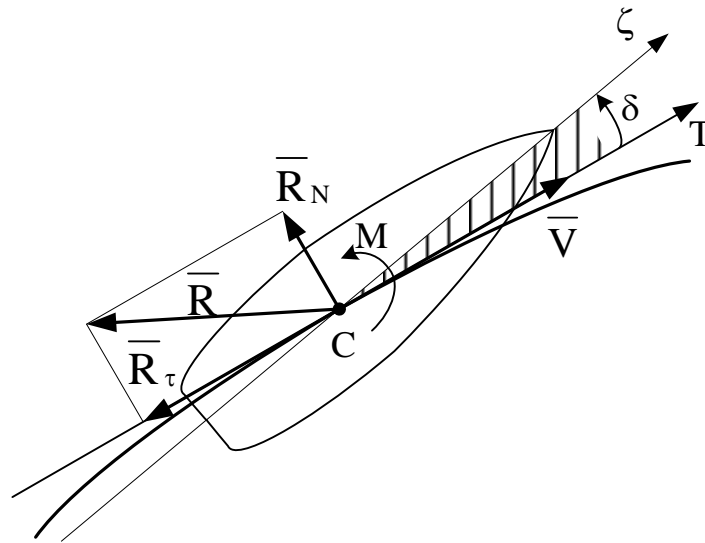


Рисунок 1 - Складові аеродинамічних сил при наявності кута нутації

Сила лобового опору, нормальна сила та перекидаючий (стабілізуєчий) момент вираховуються за наступними залежностями [1]:

$$\begin{cases} R_T = \frac{C_x \cdot \rho \cdot S \cdot V^2}{2}; \\ R_N = \frac{d \cdot l}{g} \cdot 10^3 \cdot H(Y) \cdot V^2 \cdot k_N \left( \frac{V}{a} \right) \cdot \delta; \\ M = \frac{d^2 \cdot h}{g} \cdot 10^3 \cdot H(Y) \cdot V^2 \cdot k_M \left( \frac{V}{a} \right) \cdot \delta = \beta \cdot \delta \end{cases}, \quad (1)$$

де:  $g$  – прискорення вільного падіння;  $d$  – калібр снаряда;  $l$  – довжина снаряда;  $\rho$  – густина повітря;  $S$  – площа міделевого перетину;  $h$  – плече перекидаючого моменту (відстань від центру опору до центру мас снаряда);  $V$  – швидкість;  $C_x$  – коефіцієнт лобового опору;  $H(Y)$  – функція щільності повітря, що виражає нормальну залежність розподілу щільності повітря з

висотою  $Y$ ;  $\pi(Y)$  – функція тиску, аналогічна з функцією щільності;  $F(V_\tau)$ ,  $k_N$   
 $\left(\frac{V}{a}\right)$ ,  $k_M$  $\left(\frac{V}{a}\right)$  – табличні функції опору повітря.

При цьому:

$$\pi(Y) = H(Y) \frac{\tau}{\tau_{ON}}; \quad V_\tau = V \sqrt{\frac{\tau_{ON}}{\tau}}; \quad \tau_{ON} = 288,9 \text{ } ^\circ\text{K};$$

$$F(V_\tau) = F(V) \frac{\tau_{ON}}{\tau}; \quad F(V) = 4,74 \cdot 10^{-4} \cdot V^2 \cdot C_x \left(\frac{V}{a}\right),$$

де  $C_x \left(\frac{V}{a}\right)$  - табличний коефіцієнт лобового опору снаряда.

Коефіцієнт аеродинамічного опору визначався шляхом розрахункового моделювання процесу обтікання артилерійського снаряда в програмному комплексі Ansys CFX. Розрахункове моделювання проводилося для діапазону швидкостей артилерійського снаряда 0 ... 935 м/с, що відповідає значенням чисел Маха – 0...2,9.

Під час розрахункового моделювання враховувалися наступні фактори:

моделювання проводилося для трьох режимів обтікання (дозвукового, трансзвукового та надзвукового);

для дозвукового та трансзвукового режимів обтікання створювалася розрахункова область, яка виключає вплив акустичного ефекту на результати розрахунків;

використовувалась SST модель турбулентності з додатковими налаштуваннями, які дозволяли відтворити еволюцію примежового шару (ламінарно-турбулентний перехід).

У результаті розрахункового моделювання було отримано залежності коефіцієнту аеродинамічного опору від числа Маха, див. рис 2 та табл.



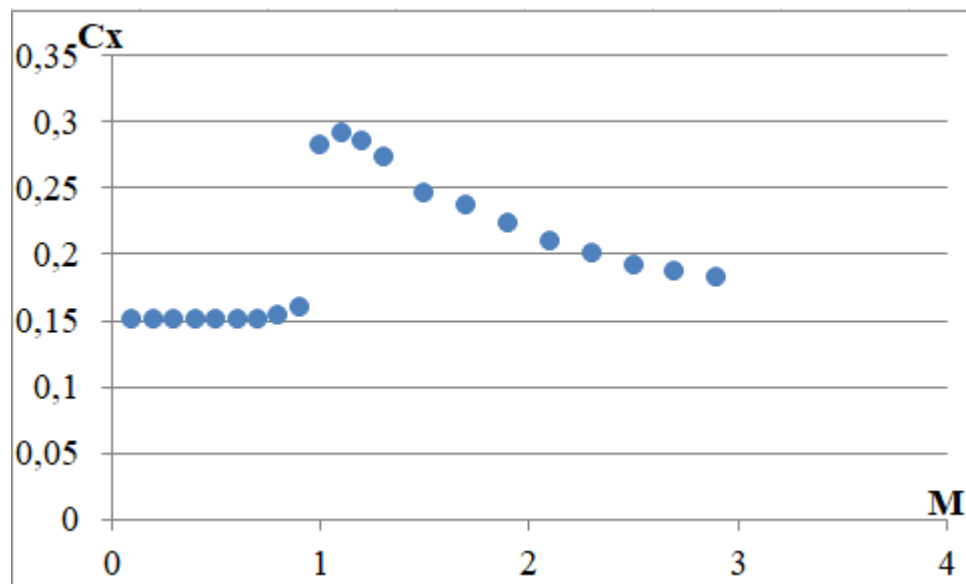


Рисунок 2 – Залежність коефіцієнта аеродинамічного опору артилерійського снаряда відносно числа Маха

Таблиця – Залежність коефіцієнта аеродинамічного опору артилерійського снаряда відносно числа Маха

М	Сх	М	Сх	М	Сх
0,1	0,150022	0,8	0,154532	1,7	0,236398
0,2	0,150031	0,9	0,159114	1,9	0,22276
0,3	0,150028	1,0	0,281687	2,1	0,209122
0,4	0,150024	1,1	0,290952	2,3	0,200029
0,5	0,150127	1,2	0,285164	2,5	0,190937
0,6	0,150118	1,3	0,272767	2,7	0,186391
0,7	0,150134	1,5	0,245491	2,9	0,181845

За нормальний розподіл температури  $\tau$  по висоті прийняті залежності:

– для висот від 0 до 9300 м:

$$\tau = \tau_{\text{ON}} - 0,006328 \cdot Y + \Delta\tau;$$

- для висот від 9300 м до 12000 м:

$$\tau = 230,0 + 0,006328 (Y - 9300) + 1,172 \cdot 10^{-6} (Y - 9300)^2 + \Delta\tau;$$

- для висот від 12000 м і вище:

$$\tau(Y) = 221,5 \text{ }^{\circ}\text{K} + \Delta\tau,$$

де  $\Delta\tau$  - відхилення температури від  $t=15^{\circ}\text{C}$  на рівні моря.

Сила опору, поділена на масу снаряда, дає прискорення опору повітря.

Прискорення лобового опору:

$$E = \frac{R_T}{m} = \frac{d^2}{m} \cdot 10^3 \cdot \pi(Y) \cdot F(V_{r\tau}), \quad (2)$$

де  $m$  – маса снаряда в кг,  $F(V_{r\tau}) = 4,74 \cdot 10^{-4} \cdot V_{r\tau}^2 \cdot C_X \left(\frac{V_{r\tau}}{a}\right)$ .

## 1.2. Диференційні рівняння руху центру мас снаряда

Виходячи із зроблених припущень, складемо диференційні рівняння руху центру мас неопереного снаряда. Для цього введемо наступні системи координатних осей (рис. 3) [5, 7, 8]:

- стартову систему координат  $OXYZ$ , що суміщена з точкою вильоту;
- паралельну до неї систему координат  $MX'Y'Z'$ , що рухається поступово разом з центром мас;
- траєкторну систему координат  $MTNB$  з початком у центрі мас снаряда.

Вісь  $T$  траєкторної системи спрямована уздовж дотичної до траєкторії руху центру мас. Вісь  $N$  перпендикулярна до дотичної та лежить у вертикальній площині, яка містить дотичну (нормаль у вертикальній площині). Третя вісь  $B$  перпендикулярна вертикальній площині, в якій розташовані осі  $T$  та  $N$  і таким чином є горизонтальною оссю. Площина  $OXY$  – площина кидання.

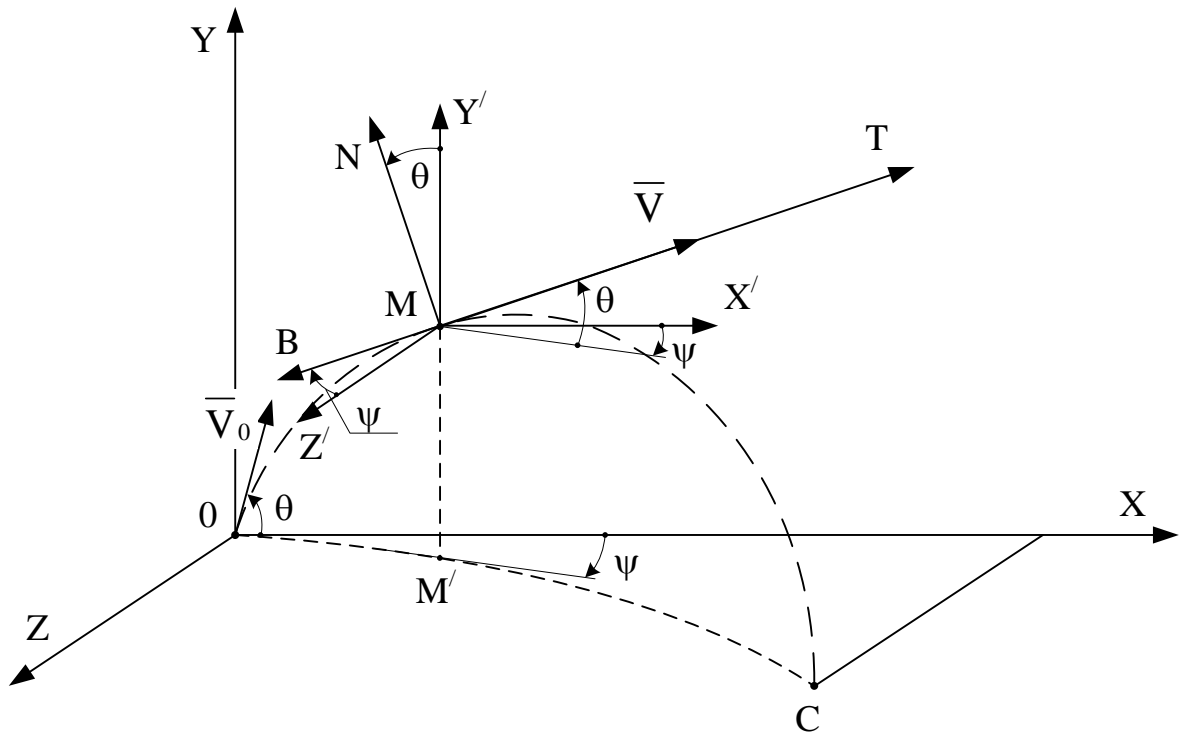


Рисунок 3 – Стартова (OXYZ), поступально рухома (MX'Y'Z') і траєкторна (MTNB) системи координат

Положення центру мас снаряда будемо визначати координатами  $X, Y, Z$  стартової системи координат. Орієнтацію вектора швидкості центру мас по відношенню до стартової системи координат визначимо двома кутами:  $\theta$  – кут між вектором швидкості центру мас  $\bar{V}$  та проекцією  $\bar{V}$  на горизонтальну площину  $MX'Z'$  (напрямок позитивного відліку кута, вказаний на рис. 3);  $\psi$  – кут між оссю  $MX'$  та проекцією  $\bar{V}$  на горизонтальну площину  $MX'Z'$  (напрямок позитивного відліку кута, вказаний на рис. 3).

Починаючи запис рівнянь руху, звернемося до рис. 3, з якого видно, що [5, 9, 10]:

$$\begin{cases} \dot{Y} = V \cdot \sin\theta; \\ \dot{X} = V \cdot \cos\theta \cdot \cos\psi; \\ \dot{Z} = V \cdot \cos\theta \cdot \sin\psi. \end{cases} \quad (3)$$

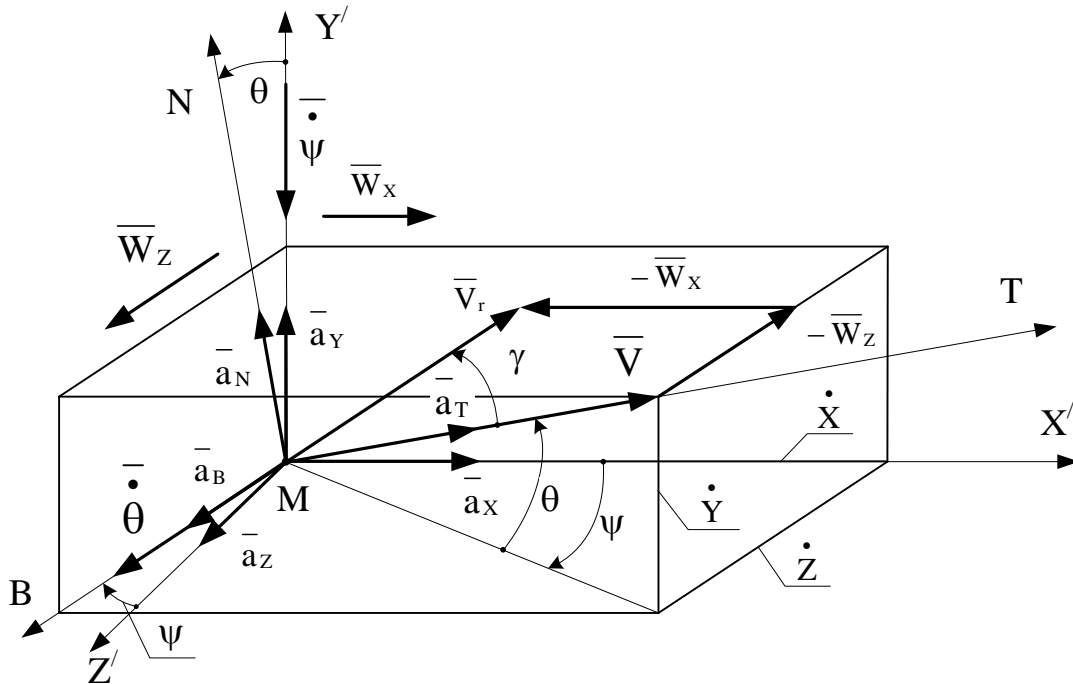


Рисунок 4 – Складові швидкості та прискорення центру мас снаряда

Грунтуючись на теоремі про похідну вектора, отримаємо:

$$\frac{d\bar{V}}{dt} = \frac{d\bar{V}'}{dt} + \bar{\Omega} \times \bar{V},$$

де  $\frac{d\bar{V}}{dt}$  – похідна від вектора швидкості в стартовій системі координат;  $\frac{d\bar{V}'}{dt}$  – похідна від вектора швидкості в траєкторній системі координат;  $\bar{\Omega}$  – вектор кутової швидкості обертання траєкторної системи координат відносно стартової.

З рис. 4 видно, що:

$$\bar{\Omega} = \dot{\theta} + \dot{\Psi}.$$

З точки зору спостерігача, який зв'язаний із траєкторною системою координат,  $\frac{d\bar{V}'}{dt} = \bar{a}_T$ . У свою чергу  $\frac{d\bar{V}}{dt} = \bar{a}_M$  – повне абсолютне прискорення

центру мас, розраховане по відношенню до стартової системи координат.

Отже, можна записати:

$$\bar{a}_M = \bar{a}_T + \bar{\Omega} \times \bar{V}, \quad (4)$$

або ж на підставі (4):

$$\bar{a}_M = \bar{a}_T + \bar{\dot{\theta}} \times \bar{V} + \bar{\dot{\psi}} \times \bar{V}.$$

Зауваживши, що два останні доданки є складовими абсолютного прискорення центру мас в напрямку нормалі  $\bar{\dot{\theta}} \times \bar{V} = \bar{a}_N$  та бінормалі до траєкторії  $\bar{\dot{\psi}} \times \bar{V} = \bar{a}_B$ , отримаємо:

$$\bar{a}_M = \bar{a}_T + \bar{a}_N + \bar{a}_B,$$

де

$$\begin{cases} a_T = \dot{V}; \\ a_N = \dot{\theta} \cdot V \cdot \sin\left(\dot{\theta}; \bar{V}\right) = \dot{\theta} \cdot V; \\ a_B = V \cdot \dot{\psi} \cdot \sin\left(\dot{\psi}; \bar{V}\right) = V \cdot \dot{\psi} \cdot \cos\theta. \end{cases}$$

Інакше:

$$\begin{cases} \dot{V} = a_T; \\ \dot{\theta} = \frac{a_N}{V}; \\ \dot{\psi} = \frac{a_B}{V \cdot \cos\theta} \end{cases}, \quad (5)$$

де  $a_T, a_N, a_B$  – проекції прискорення центру мас відповідно на дотичну T, нормаль N та бокову нормаль B траєкторної системи осей координат.

З рис. 4 видно, що:

$$\begin{cases} a_T = a_x \cdot \cos\psi \cdot \cos\theta + a_y \cdot \sin\theta + a_z \cdot \sin\psi \cdot \cos\theta; \\ a_N = -a_x \cdot \cos\psi \cdot \sin\theta + a_y \cdot \cos\theta - a_z \cdot \sin\psi \cdot \sin\theta; \\ a_B = -a_x \cdot \sin\psi + a_z \cdot \cos\psi. \end{cases} \quad (6)$$

де  $a_x, a_y, a_z$  – проекції прискорення центру мас на осі  $X, Y, Z$ .

Прискорення сили лобового опору при наявності вітру буде спрямовано у бік, зворотній вектору відносної швидкості  $\bar{V}_r$  і може бути розраховане за формулою (2) з причини малості кута  $\gamma$ . Прискорення сили опору повітря  $\vec{E}$  складає з осями координат  $OXYZ$  кути, косинуси яких дорівнюють [8]::

$$\begin{cases} \cos(\bar{E}; X) = -\frac{\dot{X} - W_x}{V_r}; \\ \cos(\bar{E}; Y) = -\frac{\dot{Y}}{V_r}; \\ \cos(\bar{E}; Z) = -\frac{\dot{Z} - W_z}{V_r}. \end{cases} \quad (7)$$

тому що  $\bar{V}_r = \bar{V} - \bar{W}_x - \bar{W}_z$ .

Проекції прискорення сили лобового опору на осі координат  $XYZ$ :

$$E_x = -E \frac{\dot{X} - W_x}{V_r}; \quad E_y = -E \frac{\dot{Y}}{V_r}; \quad E_z = -E \frac{\dot{Z} - W_z}{V_r} \quad (8)$$

Знайдемо проекції прискорення сили лобового опору на осі швидкісної системи координат. Проекція прискорення на дотичну  $T$ :

$$E_T = -E \cos\gamma = -E', \quad (9)$$

де  $\gamma$  – кут між вектором  $\vec{V}$  та  $\vec{V}_r$ . Базуючись на виразі для скалярного добутку векторів, отримаємо:

$$\cos\gamma = \frac{\vec{V} \cdot \vec{V}_r}{V \cdot V_r}, \quad \text{де} \quad V_r = \sqrt{\left(\dot{X} - W_x\right)^2 + \dot{Y}^2 + \left(\dot{Z} - W_z\right)^2} \quad (10)$$

Використовуючи вирази для проекцій  $\vec{V}$  та  $\vec{V}_r$ , запишемо:

$$\cos\gamma = \frac{\dot{X}(\dot{X} - W_x) + \dot{Y} \cdot \dot{Y} + \dot{Z}(\dot{Z} - W_z)}{V \cdot V_r} = \frac{\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2 - \dot{X} \cdot W_x - \dot{Z} \cdot W_z}{V \cdot V_r}.$$

Нехтуючи доданком  $\dot{Z}W_z$ , у зв'язку з його порівняно незначною величиною, отримуємо:

$$\cos\gamma = \frac{V^2 - \dot{X}W_x}{V \cdot V_r} = \frac{V - \cos\theta \cdot \cos\psi \cdot W_x}{V_r},$$

де  $V^2 = \dot{X}^2 + \dot{Y}^2 + \dot{Z}^2$ . У отриманому виразі  $\cos\psi \approx 1$ , тоді:

$$\cos\gamma = \frac{V - \cos\theta \cdot W_x}{V_r} \quad (11)$$

Проекція прискорення сили лобового опору на нормаль N системи осей TNB на підставі (6) і (7) може бути представлена спрощеним виразом:

$$\begin{aligned} E_N &= -E_x \cdot \sin\theta + E_y \cdot \cos\theta = \frac{E(\dot{X} - W_x)}{V_r} \cdot \sin\theta - E \frac{\dot{Y}}{V_r} \cdot \cos\theta = \\ &= \frac{EV \cdot \cos\theta \cdot \cos\psi}{V_r} \cdot \sin\theta - \frac{EW_x}{V_r} \sin\theta - \frac{EV \cdot \sin\theta \cdot \cos\theta}{V_r}, \end{aligned}$$

отже ( $\cos\psi \approx 1$ ), приймаємо:

$$E_N = -\frac{E \cdot W_x}{V_r} \cdot \sin\theta. \quad (12)$$

Проекція прискорення сили лобового опору на нормаль B системи осей TNB на підставі (6) і (7) також може бути представлена спрощеним виразом:

$$\begin{aligned} E_B &= -E_x \cdot \sin\psi + E_z \cos\psi = E \frac{\dot{X} - W_x}{V_r} \sin\psi - E \frac{\dot{Z} - W_z}{V_r} \cos\psi = \\ &= \frac{E\dot{X}\sin\psi - E\dot{Z}\cos\psi}{V_r} - \frac{EW_x}{V_r} \sin\psi + \frac{EW_z}{V_r} \cos\psi. \end{aligned}$$

Перший доданок в отриманому виразі дорівнює 0, так як:

$$E \left( \dot{X} \sin \psi - \dot{Z} \cos \psi \right) = E \cdot V (\cos \theta \cdot \cos \psi \cdot \sin \psi - \cos \theta \cdot \sin \psi \cdot \cos \psi) = 0$$

Нехтуючи другим доданком з причини малості кута  $\psi$  ( $\sin \psi \approx 0$ ), отримуємо ( $\cos \psi \approx 1$ ):

$$E_B = \frac{E \cdot W_Z}{V_r} \quad .(13)$$

Визначимо складові прискорення сили тяжіння, припускаючи, що вектор  $\vec{g}$  лежить у площині кидання, тобто нехтуючи малістю відхилення снаряда в боковому напрямку. З рис.5 видно, що:

$$g_x = -g \cdot \sin \varphi; \quad g_y = -g \cdot \cos \varphi; \quad g_z = 0.$$

Грунтуючись на виразі (7), отримуємо проекції вектора  $\vec{g}$  на вісь T траекторної системи осей:

$$g_T = g_x \cos \theta + g_y \sin \theta = -g \sin \varphi \cos \theta - g \cos \varphi \sin \theta = -g \sin \theta,$$

оскільки кут  $\varphi$  малий і  $\cos \varphi \approx 1$ ,  $\sin \varphi \approx 0$ . Аналогічно, проекції  $\vec{g}$  на осі N і B траекторної системи осей:

$$g_N = -g_x \cdot \sin \theta + g_y \cdot \cos \theta = g \cdot \sin \varphi \cdot \sin \theta - g \cdot \cos \varphi \cdot \cos \theta = -g \cdot \cos \theta,$$

$$g_B = g \cdot \sin \varphi \cdot \sin \psi.$$

Враховуючи, що [1]:  $g = g_0 \left( 1 - 2 \frac{Y}{R_3} \right)$  і  $\sin \varphi = \operatorname{tg} \varphi = \frac{X}{R + Y}$ , маємо:

$$\begin{cases} g_T = -g_0 \left( 1 - 2 \frac{Y}{R_3} \right) \cdot \sin \theta; \\ g_N = -g_0 \left( 1 - 2 \frac{Y}{R_3} \right) \cdot \cos \theta; \\ g_B = g_0 \left( 1 - 2 \frac{Y}{R_3} \right) \cdot \frac{X}{R + Y} \cdot \sin \psi \end{cases} \quad (14)$$



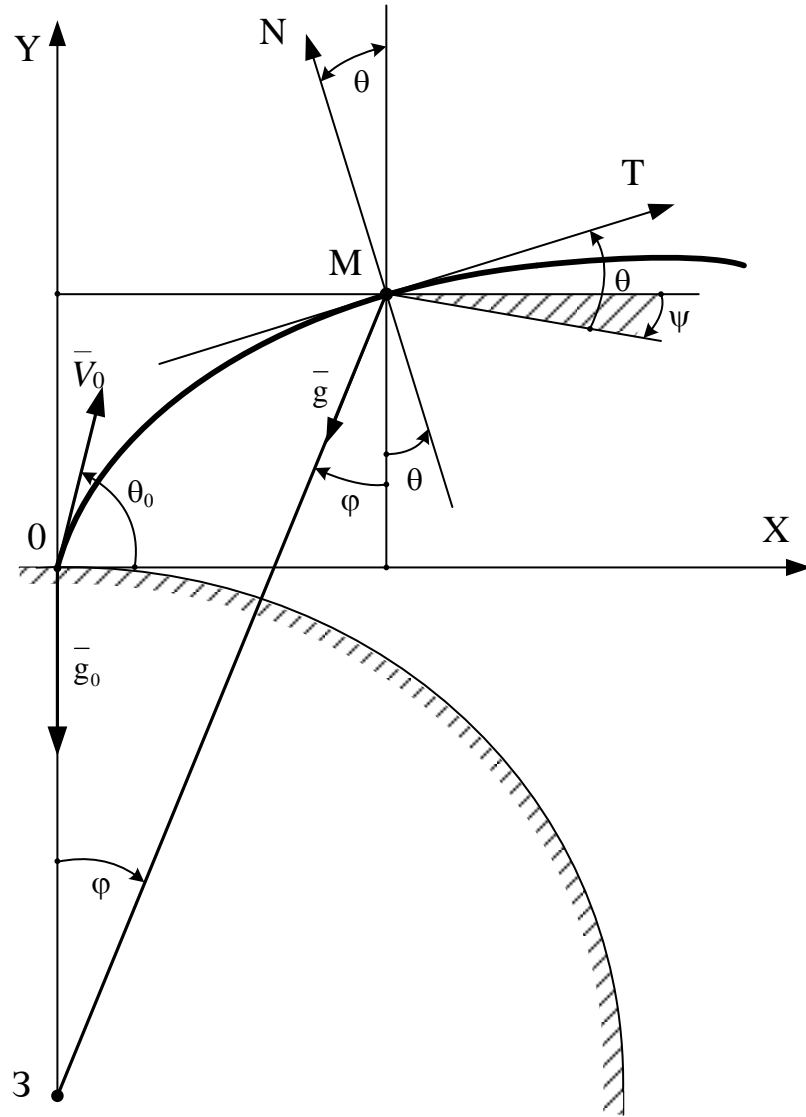


Рисунок 5 – До визначення складових прискорення сили тяжіння

Для того, щоб врахувати в рівняннях руху обертання Землі, визначимо проєкції прискорення Коріоліса на обрані осі координат (рис.5). Користуючись правилом векторної алгебри, запишемо:

$$\vec{a}_{\text{КОР}} = 2(\vec{\Omega} \times \vec{V}) = 2 \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \Omega_x & \Omega_y & \Omega_z \\ \dot{X} & \dot{Y} & \dot{Z} \end{vmatrix} \quad (15)$$

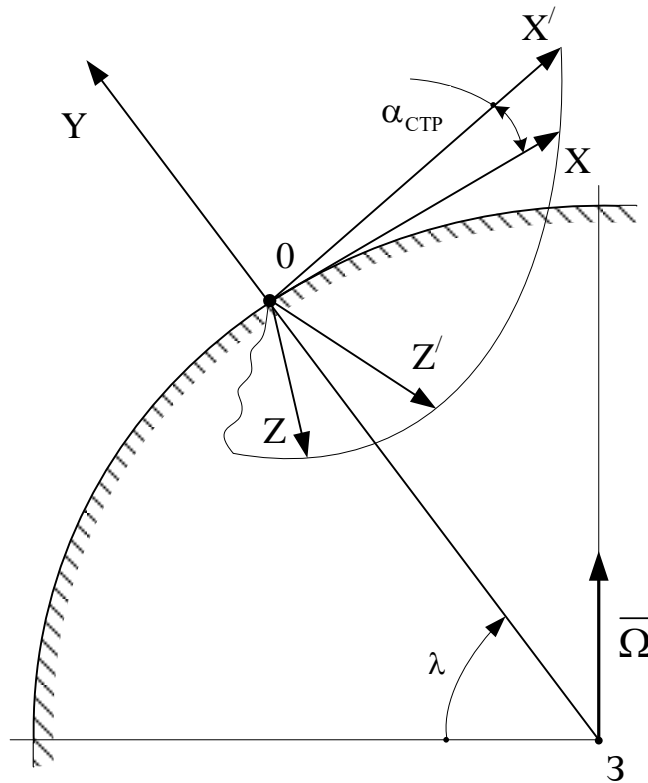


Рисунок 6 – До визначення проекції прискорення сили Коріоліса

$\bar{\Omega}$  – кутова швидкість обертання Землі;  $\lambda$  – географічна широта;  $\alpha_{\text{СТР}}$  – кут стрільби, що відраховується від меридіана у напрямку на північ (вісь  $X$  спрямована на північ)

З рис. 6 видно, що проекції кутової швидкості обертання Землі:

$$\Omega_x = \Omega \cdot \cos \lambda \cdot \cos \alpha_{\text{СТР}};$$

$$\Omega_y = \Omega \cdot \sin \lambda;$$

$$\Omega_z = -\Omega \cdot \cos \lambda \cdot \sin \alpha_{\text{СТР}}.$$

Проекції прискорення Коріоліса знайдемо як відповідні мінори визначника (15):

$$a_{\text{КХ}} = 2 \left( \Omega_y \dot{Z} - \Omega_z \dot{Y} \right), \quad a_{\text{КY}} = 2 \left( \Omega_z \dot{X} - \Omega_x \dot{Z} \right), \quad a_{\text{КZ}} = 2 \left( \Omega_x \dot{Y} - \Omega_y \dot{X} \right).$$

Нехтуючи в отриманих виразах доданками, що містять  $\dot{Z}$ , так як  $\dot{Z}$  значно менше  $\dot{Y}$ , отримаємо:

$$\begin{aligned}
a_{kx} &= -2\Omega_z \cdot \dot{Y}; \\
a_{ky} &= 2\Omega_z \cdot \dot{X}; \\
a_{kz} &= 2\left(\Omega_x \cdot \dot{Y} - \Omega_y \cdot \dot{X}\right).
\end{aligned} \tag{16}$$

Визначимо проекції прискорення Кориоліса на осі траекторної системи координат, користуючись виразами (7) і (16):

$$\begin{aligned}
a_{kt} &= -2\Omega_z \dot{Y} \cdot \cos\theta + 2\Omega_z \dot{X} \cdot \sin\theta = -2\Omega_z \left( \dot{Y} \cdot \cos\theta - \dot{X} \cdot \sin\theta \right) = \\
&= -2\Omega_z (V \cdot \sin\theta \cdot \cos\theta - V \cdot \sin\theta \cdot \cos\theta \cdot \cos\psi) = 0.
\end{aligned}$$

Результат для  $a_{kt}$  є закономірним, тому що вектор прискорення Кориоліса перпендикулярний вектору швидкості:

$$\begin{aligned}
a_{kn} &= 2\Omega_z \dot{Y} \cdot \sin\theta + 2\Omega_z \dot{X} \cdot \cos\theta = 2\Omega_z \left( \dot{Y} \cdot \sin\theta + \dot{X} \cdot \cos\theta \right) = \\
&= 2\Omega_z (V \cdot \sin^2\theta + V \cdot \cos^2\theta) = 2\Omega_z V.
\end{aligned}$$

Враховуючи вираз для  $\Omega_z$ , отримаємо:

$$a_{kn} = -2\Omega V \cdot \cos\lambda \cdot \sin\alpha_{\text{СТР}}. \tag{1.17}$$

Проекція кориолісова прискорення на бокову нормаль:

$$a_{kb} = 2\Omega_z \cdot \dot{Y} \cdot \sin\psi + 2\left(\Omega_x \cdot \dot{Y} - \Omega_y \cdot \dot{X}\right) \cos\psi = 2\left(\Omega_x \cdot \dot{Y} - \Omega_y \cdot \dot{X}\right),$$

де  $\sin\psi \approx 0$ ,  $\cos\psi \approx 1$ . Використовуючи вираз для  $\Omega_x$  і  $\Omega_y$ , маємо:

$$a_{kb} = -2\Omega \cdot \left( \dot{X} \cdot \sin\lambda - \dot{Y} \cdot \cos\lambda \cdot \cos\alpha_{\text{СТР}} \right) \tag{1.18}$$

Під час постановки задачі про рух снаряда було зроблено припущення, що його вісь не співпадає з напрямком руху. Між віссю снаряда і дотичною до траєкторії завжди існує деякий кут  $\delta$ , змінний за часом. Періодична складова кута  $\delta$  призводить до появи нормальної сили, що обертається навколо осі динамічної рівноваги і надає траєкторії снаряда спіралеподібну форму. Сама ж вісь динамічної рівноваги для снаряда, що обертається, на криволінійній ділянці траєкторії не співпадає з дотичною, а утворює з нею

деяку неперіодичну складову кута  $\delta$  – кут динамічної рівноваги  $\delta_p$ . Детально про механізм утворення кута  $\delta_p$  можна дізнатися з [1].

Згідно з [1] кут  $\delta_p$  можна представити двома складовими  $\delta_{1p}$  та  $\delta_{2p}$  (рис. 7). Кут  $\delta_{1p}$ , що значно більший за кут  $\delta_{2p}$ , лежить у площині, яка проходить через дотичну перпендикулярно вертикальній площині, в якій лежить дотична. Кут  $\delta_{2p}$  лежить у вертикальній площині, що проходить через дотичну. Наявність кутів  $\delta_{1p}$  і  $\delta_{2p}$  відхилення осі снаряда від напрямку дотичної до траєкторії призводить до появи нормальних неперіодичних складових сили опору  $\bar{R}_{\text{ДЕР}}$  і  $\bar{R}_{\text{NN}}$  (рис. 7). При цьому бокова нормальна сила  $\bar{R}_{\text{ДЕР}}$  викликає бокове дериваційне відхилення снаряда, а сила  $\bar{R}_{\text{NN}}$  дещо змінює дальність польоту. Необхідно також прийняти до уваги те, що сила опору повітря зростає зі збільшенням кута  $\delta_p$ , тому що погіршуються умови обтікання. Отже, збільшується і сила лобового опору. Зростання лобового опору можна врахувати збільшенням балістичного коефіцієнта.

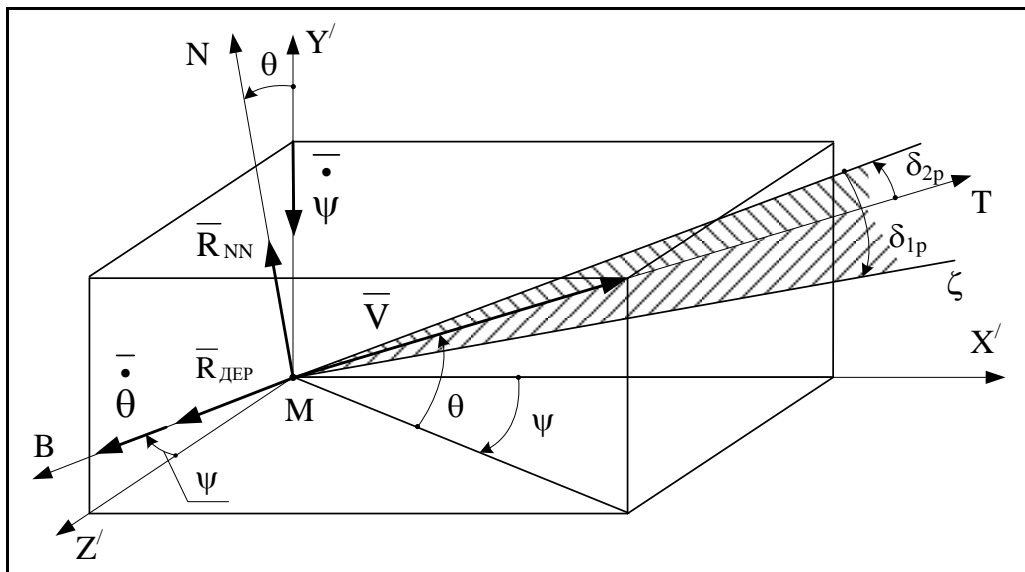


Рисунок 7 – Складові динамічного кута рівноваги  $\delta_p$

Нормальна бокова складова  $\bar{R}_{\text{ДЕР}}$  сили опору повітря, що викликана наявністю кута  $\delta_{1p}$ , лежить у площині кута  $\delta_{1p}$  та спрямована вздовж бокової нормалі В. При цьому, ґрунтуючись на (1), можна записати:

$$R_{\text{ДЕР}} = \frac{d \cdot l}{g} \cdot 10^3 \cdot H(Y) \cdot V^2 \cdot k_N \left( \frac{V}{a} \right) \cdot \delta_{1p}. \quad (19)$$

Враховуючи [1], вираз для  $\delta_{1p}$  має вигляд:

$$\delta_{1p} = \frac{\alpha}{\beta} \cdot \left| \dot{\theta} \right| \cdot \varphi(t), \quad (20)$$

де

$$\alpha = B \cdot r_0; \quad \beta = \frac{d^2 \cdot h}{g} \cdot 10^3 \cdot H(Y) \cdot V^2 \cdot k_M \left( \frac{V}{a} \right); \quad B = 0,55 \frac{Q \cdot d^2}{4g};$$

$$r_0 = \frac{2\pi V_0}{\eta d}; \quad h = h_1 + 0,57h_2 - 0,16d; \quad \varphi(t) = \frac{r}{r_0} = e^\sigma;$$

$$\sigma = -0,0598 \frac{\frac{d}{g} \cdot 10^3}{Q} \cdot \int_0^t V^{\frac{4}{5}} dt,$$

де  $\varphi(t)$  – враховує згасання кутової швидкості обертання снаряда;  $B$  – осьовий момент інерції снаряда;  $r$  – кутова швидкість обертання снаряда;  $r_0$  – початкова швидкість обертання снаряда;  $Q$  – вага снаряда (у кілограмах сили);  $l$  – довжина снаряда;  $d$  – калібр снаряда;  $\eta$  – відносна довжина ходу нарізів ствола;  $h$  – плече перекидаючого моменту, тобто відстань від центру опору до центру мас снаряда;  $h_1$  – відстань від центру мас снаряда до головної частини;  $h_2$  – довжина головної частини.

У наведених залежностях  $h$ ,  $h_1$  і  $h_2$  пов'язані формулою Гобара [1], а величина  $\sigma$  визначається теоретичною формулою Сльозкіна [1] та встановлює закон змінення кутової швидкості снаряда в результаті тертя повітря.

Коефіцієнти опору  $k_N \left( \frac{V}{a} \right)$  та  $k_M \left( \frac{V}{a} \right)$  є функціями числа Маєвського

(Маха)  $M = V/a$  і, отже, однозначними функціями приведеної швидкості, оскільки за визначенням

$$M(V, a) = \frac{V}{a} = \frac{V_\tau}{a_{ON}},$$

де  $a_{ON} = \text{const}$  – швидкість звуку при нормальних атмосферних умовах. У зв'язку з цим можна позначити:

$$k_N\left(\frac{V}{a}\right) = k_N(V_\tau); \quad k_M\left(\frac{V}{a}\right) = k_M(V_\tau).$$

Підставляючи ці вирази у (1.19) з урахуванням (1.20), отримаємо:

$$R_{\text{ДЕР}} = \frac{1}{d \cdot h} \cdot \frac{k_N(V_\tau)}{k_M(V_\tau)} \cdot \alpha \cdot |\dot{\theta}| \cdot \varphi(t) \quad (21)$$

Прискорення центру мас снаряда у напрямку нормалі  $V$ , що викликане дією сили  $R_{\text{ДЕР}}$  має вигляд:

$$E_{\text{ДЕР}} = C_N \cdot |\dot{\theta}| \cdot \varphi(t) \cdot \frac{k_N(V_\tau)}{k_M(V_\tau)}, \quad C_N = \frac{1 \cdot \alpha \cdot g}{d \cdot h \cdot Q} \quad (22)$$

Залежність відношення  $\left(\frac{k_N}{k_M}\right)$  від безмірної довжини снаряда  $\frac{1}{d}$  за

узагальненими експериментальними даними має вигляд:

$$\frac{k_N}{k_M} = \left(\frac{k_N}{k_M}\right)_{4,5} \cdot \frac{4,5}{1/d}, \quad (23)$$

де  $\left(\frac{k_N}{k_M}\right)_{4,5}$  – співвідношення аеродинамічних коефіцієнтів.

Зупинимося тепер на визначенні сили  $R_{NN}$ . Грунтуючись на [1], складову кута динамічної рівноваги знаходимо так:

$$\delta_{2p} = \frac{\alpha}{\beta} \cdot \dot{\delta}_{1p}. \quad (24)$$

Використовуючи (20), маємо:

$$\dot{\delta}_{1p} = \frac{d}{dt} \left( \frac{\alpha}{\beta} \cdot |\dot{\theta}| \cdot \varphi \right)$$

Замінімо  $\varphi$  його середнім значенням:

$$\dot{\delta}_{1p} = \varphi \cdot \alpha \cdot \frac{d}{dt} \left( \frac{|\dot{\theta}|}{\beta} \right)$$

або

$$\dot{\delta}_{1p} = \varphi \cdot \alpha \cdot \frac{g}{d^2 \cdot h \cdot 10^3} \cdot \frac{d}{dt} \left( \frac{|\dot{\theta}|}{H(Y) \cdot k_M \cdot V^2} \right).$$

Замінімо  $H(Y)$  та  $k_M$  їх середніми значеннями:

$$\dot{\delta}_{1p} = \frac{\varphi \cdot \alpha}{[H(Y)] \cdot k_M \cdot d^2 \cdot h \cdot 10^3} \cdot \frac{d}{dt} \left( \frac{|\dot{\theta}|}{V^2} \right) \quad (25)$$

Приймаючи в першому наближенні

$$|\dot{\theta}| = \frac{g \cdot \cos \theta}{V}, \text{ маємо } \frac{|\dot{\theta}|}{V^2} = \frac{g \cdot \cos \theta}{V^3}.$$

У цьому випадку похідна

$$\frac{d}{dt} \left( \frac{|\dot{\theta}|}{V^2} \right) = g \frac{d}{dt} \left( \frac{\cos \theta}{V^3} \right).$$

Подальші розрахунки дають:

$$\frac{d}{dt} \left( \frac{|\dot{\theta}|}{V^2} \right) = -g \frac{V \cdot \sin \theta \cdot \dot{\theta} + 3 \dot{V} \cdot \cos \theta}{V^4} \quad (26)$$

Підставимо вираз (26) в (25):

$$\dot{\delta}_{1p} = - \frac{\varphi \cdot \alpha \cdot g^2}{\left[ \pi(Y) \cdot \frac{\tau_{ON}}{\tau(Y)} \right] \cdot k_M} \cdot \frac{V \cdot \sin \theta \cdot \dot{\theta} + 3 \dot{V} \cdot \cos \theta}{d^2 \cdot h \cdot 10^3 \cdot V^4} \quad (27)$$

Приймаючи, що кут  $\delta_{2p}$  визначений, скористуємося виразом (1) для нормальної сили і запишемо:

$$R_{NN} = \frac{d \cdot l}{g} \cdot 10^3 \cdot H(Y) \cdot V^2 \cdot k_N \cdot \delta_{2p}.$$

Підставляючи сюди вираз (24), отримаємо після деяких перетворень:

$$R_{NN} = \frac{1}{hd} \cdot \alpha \cdot \frac{k_N(V_{rt})}{k_M(V_{rt})} \cdot \dot{\delta}_{1p}$$

Або ж, ґрунтуючись на виразі (23),

$$R_{NN} = \frac{\alpha}{h} \cdot 4,5 \cdot \left[ \frac{k_N(V_{rt})}{k_M(V_{rt})} \right]_{4,5} \cdot \dot{\delta}_{1p}, \quad (28)$$

де  $\dot{\delta}_{1p}$  визначається формулою (27).

Прискорення центру мас снаряда у напрямку нормалі N, викликане дією сили  $R_{NN}$  (Q – вага снаряда):

$$E_{NN} = \frac{\alpha \cdot g}{Q \cdot h} \cdot 4,5 \cdot \left[ \frac{k_N(V_{rt})}{k_M(V_{rt})} \right]_{4,5} \cdot \dot{\delta}_{1p} \quad (29)$$

Для того, щоб при розрахунках функції тиску повітря  $\pi(Y)$  не займатися інтерполяцією її табличних значень, знайдемо аналітичний вираз для похідної  $\dot{\pi}(Y)$ . Це дозволить, включивши  $\dot{\pi}(Y)$  до числа рівнянь руху, отримати значення  $\pi(Y)$  на кожному кроці числового інтегрування.

Продиференціюємо вираз для  $\pi(Y)$  за часом:

$$\pi(Y) = \frac{h}{h_{ON}}; \quad \dot{\pi}(Y) = \frac{\dot{h}}{h_{ON}}, \quad (30)$$

де  $h$  – тиск повітря у точці траєкторії;  $h_{ON}$  – тиск на земній поверхні в умовах нормальної атмосфери ( $h_{ON} = 750$  мм рт. ст.).

Для того, щоб визначити  $\dot{h}$ , скористуємося рівнянням вертикальної рівноваги атмосфери, котре має вигляд (R – питома газова постійна повітря):



$$\frac{dh}{h} = -\frac{1}{R} \cdot \frac{dY}{\tau}.$$

Помножимо та розділимо його ліву частину на  $dt$ . У результаті отримаємо:

$$\frac{dt}{dt} \cdot \frac{dh}{h} = -\frac{1}{R} \cdot \frac{dY}{\tau}; \quad \frac{dh}{dt} \cdot \frac{1}{h} = -\frac{1}{R \cdot \tau} \cdot \frac{dY}{dt}, \quad \text{або} \quad \frac{\dot{h}}{h} = -\frac{1}{R \cdot \tau} \cdot \dot{Y}, \quad \text{звідки}$$

$$\dot{h} = -\frac{h}{R \cdot \tau} \cdot V \cdot \sin \theta. \quad (31)$$

Підставляючи вираз (1.31) в (1.30), отримуємо:

$$\dot{\pi}(Y) = -\frac{\pi(Y)}{R \cdot \tau} \cdot V \cdot \sin \theta.$$

Якщо враховувати дійсний розподіл температури за висотою, тоді

$$\dot{\pi}(Y) = -\frac{\pi(Y)}{R \cdot [\tau(Y) + \Delta\tau]} \cdot V \cdot \sin \theta, \quad (32)$$

де  $\tau(Y)$  – нормальна залежність змінення віртуальної температури з висотою;  
 $\Delta\tau$  – відхилення віртуальної температури від нормальної залежності:

$$\Delta\tau = \frac{\Delta T}{1 - \frac{3}{8} \cdot \frac{e}{h}};$$

$e$  – тиск водяних парів;  $\Delta T$  – відхилення абсолютної температури вологого повітря від нормального закону розподілу;  $R$  – газова постійна.

Додаючи знайдені складові прискорень центру мас снаряда у напрямку осей  $T$ ,  $N$  та  $B$ , отримаємо:

$$\begin{cases} a_T = E_T + g_T; \\ a_N = E_N + g_N - a_{KN} + E_{NN}; \\ a_B = E_B + g_B - a_K + E_{ДЕР}, \end{cases} \quad (33)$$

де  $E_T$ ,  $E_N$ ,  $E_B$  – визначаються формулами (9), (12), (13);  $g_T$ ,  $g_N$ ,  $g_B$  – визначаються формулами (14);  $E_{NN}$  та  $E_{ДЕР}$  – визначаються формулами (29) та (22) відповідно.

Підставляємо (33) в (5), виконуємо необхідні перетворення та додаємо до отриманих рівнянь рівняння (3) і (32). Отримуємо наступну систему рівнянь руху центру мас снаряда (розрахуно в системі СИ):

$$\dot{Y} = V \cdot \sin \theta; \quad (34)$$

$$\dot{X} = V \cdot \cos \theta \cdot \cos \psi; \quad (35)$$

$$\dot{Z} = V \cdot \cos \theta \cdot \sin \psi; \quad (36)$$

$$\dot{V} = -E' - g_0 \left( 1 - 2 \frac{Y}{R_3} \right) \cdot \sin \theta; \quad (37)$$

$$\dot{\theta} = -g_0 \left( \frac{1 - 2 \frac{Y}{R_3}}{V} \right) \cdot \cos \theta - \frac{E \cdot W_x}{V_r \cdot V} \cdot \sin \theta + \quad (38)$$

$$+ 2\Omega \cdot \cos \lambda \cdot \sin \alpha_{\text{СТР}} + \frac{E_{\text{NN}}}{V} + \frac{V \cdot \cos \theta}{R_3 + Y};$$

$$\dot{\psi} = g_0 \left( \frac{1 - 2 \frac{Y}{R_3}}{V \cdot \cos \theta} \right) \cdot \frac{X}{R_3 + Y} \cdot \sin \psi + \frac{E \cdot W_z}{V_r \cdot V \cdot \cos \theta} + \quad (39)$$

$$+ 2\Omega \cdot (\sin \lambda - \text{tg} \theta \cdot \cos \lambda \cdot \cos \alpha_{\text{СТР}}) + \frac{E_{\text{ДЕР}}}{V \cdot \cos \theta};$$

$$\dot{\pi}(Y) = -\frac{\pi(Y)}{R \cdot \tau(Y)} \cdot V \cdot \sin \theta; \quad (40)$$

$$\dot{S}(t) = V^{\frac{4}{5}}; \quad (41)$$

$$\dot{\delta}_{\text{IP}} = -1_7 \cdot 1_8; \quad (42)$$

Тут:

$$W^2 = W_x^2 + W_z^2;$$

$$V_r = V \cdot \sqrt{1 - 2 \frac{(W_x \cdot \cos \theta \cdot \cos \psi + W_z \cdot \sin \psi)}{V} + \frac{W^2}{V^2}};$$

$$V_{\tau} = V_r \cdot \sqrt{\frac{\tau_{ON}}{\tau(Y)}}; \quad \cos \gamma = \frac{V - W_x \cdot \cos \theta}{V_r};$$

$$\tau(Y) = \begin{cases} \tau(Y) = 288,9 - 0,006328 \cdot Y + \Delta\tau, & \text{при } Y < 9300, \\ \tau(Y) = 230 - 0,006328 \cdot (Y - 9300) + 1,172 \cdot 10^{-6} \cdot (Y - 9300)^2 + \Delta\tau, & \text{при } 9300 \leq Y \leq 1200, \\ \tau(Y) = 221,5^\circ \text{K} + \Delta\tau, & \text{при } Y \geq 1200; \end{cases}$$

$$E' = E \cdot \cos \gamma; \quad C_N = \frac{l \cdot \alpha \cdot g}{d \cdot h \cdot Q}; \quad E_{\text{ДЕР}} = C_N \cdot \varphi \cdot |\dot{\theta}| \cdot \frac{k_N(V_{\tau})}{k_M(V_{\tau})},$$

$$\text{де} \quad \alpha = B \cdot r_0; \quad r_0 = \frac{2\pi V_0}{\eta d}; \quad B = 0,55 \frac{Q \cdot d^2}{4g};$$

$$h = h_1 + 0,57h_2 - 0,16d; \quad \varphi = e^{\sigma}; \quad \sigma = -0,0598 \frac{d \cdot l^{\frac{4}{5}}}{Q} S(t); \quad (Q - \text{вага в кг.})$$

$$\frac{k_N(V_{\tau})}{k_M(V_{\tau})} = \left[ \frac{k_N(V_{\tau})}{k_M(V_{\tau})} \right]_{4,5} \cdot \frac{4,5d}{1}; \quad \left( \frac{k_N}{k_M} \right)_{4,5} - \text{див. табл. 1.1};$$

$$E_{\text{NN}} = \frac{\alpha \cdot g}{Q \cdot h} \cdot 4,5 \cdot \left[ \frac{k_N(V_{\tau})}{k_M(V_{\tau})} \right]_{4,5} \cdot \dot{\delta}_{1p};$$

$$l_7 = \frac{\alpha \cdot g^2 \cdot \varphi_{\text{CP}}}{\left[ \pi(Y) \cdot \frac{\tau_{ON}}{\tau(Y)} \right]_{\text{CP}} \cdot (k_M)_{\text{CP}}}; \quad l_8 = \frac{V \cdot \sin \theta \cdot \dot{\theta} + 3 \dot{V} \cdot \cos \theta}{d^2 \cdot h \cdot 10^3 \cdot V^4};$$

$$\varphi_{\text{CP}} = \frac{1 + \varphi}{2}; \quad (k_M)_{\text{CP}} = \frac{k_M(V_{\tau})_0 + k_M(V_{\tau})}{2}.$$

Останній доданок у рівнянні (38) враховує вплив на дальність польоту

снаряда кривизни земної поверхні й являє собою швидкість пониження лінії горизонту по відношенню до стартової. Для її визначення розкладемо вектор швидкості на складові (рис.8).

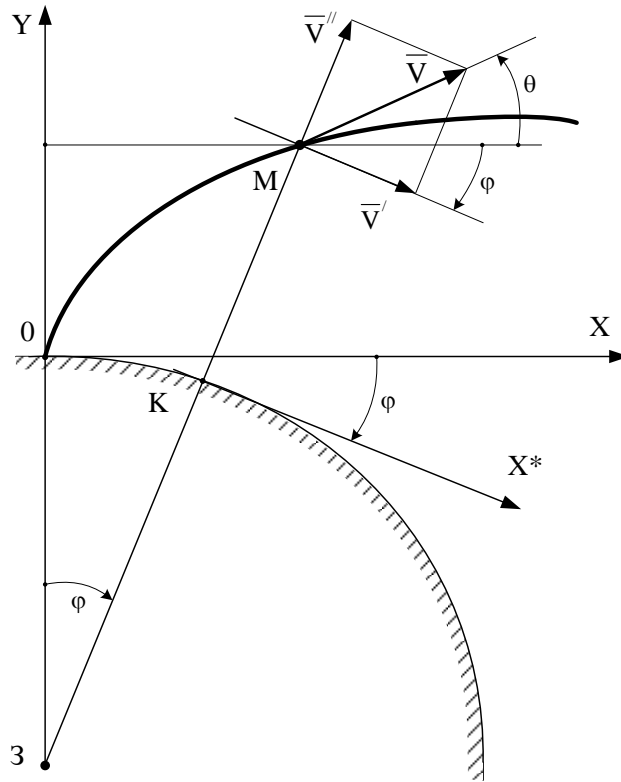


Рисунок 8 – Розкладання вектора швидкості на складові

З рис. 8 видно, що  $\dot{\varphi} \cdot (R_3 + KM) = V \cdot \cos(\theta + \varphi)$ ,

звідси 
$$\dot{\varphi} = \frac{V}{R_3 + KM} \cdot \cos(\theta + \varphi).$$

Оскільки кут  $\varphi$  малий, то можна прийняти:

$$\cos(\theta + \varphi) \approx \cos\theta; \quad R_3 + KM = R_3 + Y.$$

У такому випадку: 
$$\dot{\varphi} = \frac{V \cdot \cos\theta}{R_3 + Y}.$$

### 1.3 Прогнозування точки прицілювання

Точка прицілювання артилерійської системи залежить від наступних факторів:

- кількість точок фіксації цілі пунктом артилерійської розвідки з урахуванням часових інтервалів;
- суміщення точок фіксації з наявними топографічними даними місцевості;
- час трансферу даних з пункту артилерійської розвідки до артилерійського підрозділу;
- час підготовки артилерійської установки до пострілу;
- час польоту артилерійського снаряда до точки прицілювання.

З урахуванням вищезазначеного визначення точки прицілювання є складним ітераційним процесом.

### 1.4 Постановка задачі

В результаті проведеного аналітичного огляду фізичного процесу польоту артилерійського снаряда з урахуванням логістичних інтервалів транзиту даних необхідно створити додаток який дозволяє:

- враховувати дію сили інерції, гравітації та сили аеродинамічного опору;
- метеорологічні дані;
- кількість точок фіксації цілі пунктом артилерійської розвідки з урахуванням часових інтервалів;
- час трансферу даних з пункту артилерійської розвідки до артилерійського підрозділу;
- час підготовки артилерійської установки до пострілу;
- час польоту артилерійського снаряда до точки прицілювання.

## 2 РЕАЛІЗАЦІЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

### 2.1 Додаток прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі

Графічний інтерфейс додатку прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі представлено на рис. 9.

Вогнева позиція:

X

Y

Кіль-сть часу на наведення, сек

Пересування цілі:

Кіль-сть точок спостереження:

№	X	Y	Час
1	0	0	0
2	0	0	0
3	0	0	0

Тип визначення ТП:

Середній кут спостер., середня швидкість

Останній кут спостер., остання швидкість

Балістичні дані снаряду:

V0, м/с

P0N, кг/м3

MS, кг

d, м

LC, м

Ds, м

INA, кг·м2

INC, кг·м2

RED, м

Рисунок 9 – Область введення вхідних даних

На рис. 9 присутні наступні вхідні параметри:

- $X$  та  $Y$  – координати розташування вогневої позиції;
- точки спостереження – координати та час перебування цілі на місцевості;
- $V_0$  – початкова швидкість польоту снаряда;
- $\rho_{00}$  – густина повітря;
- $M_S$  – маса снаряда;
- $d$  – діаметр міделевого перетину снаряда;
- $LC$  – довжина снаряда;
- $D_s$  – відстань від підричника до центра маси снаряда;
- $I_{NA}$  – повздовжній момент інерції;
- $I_{NC}$  – поперечний момент інерції;
- $RED$  – максимальна дальність пострілу;
- $C_x$  – коефіцієнт аеродинамічного опору для 3-х значень кутів атаки повітряного потоку.

Для розрахунку запропоновано використовувати параметри артилерійського снаряда ОФ-29 та гармату 2А36.

В результаті розрахунку отримуємо текстову частину та графічну з позначеннями точок прицілювання в декартових координатах та візуальним відображенням, рис. 10.

Час льоту до останньої точки спостереження 27,51 сек  
Тип визначення ТП по середній значенням  
Кут руху = 4-18 (25,08°);  $V = 7,95$  м/с  
Час визначення ТП = "Час льоту" + "Час підготовки" + "20 (резерв)" = 93 сек  
Дальність від останньої точки спостереження = "Час ТП" \* "Швидкість руху" = 735 м.  
Координати ТП:  $X=5223420$ ;  $Y=7414714$   
Дальність від ВП до ТП = 13427;  
Час льоту сигналу на відстань до ТП = 29,41 сек  
Час зашта: 15:22:53;

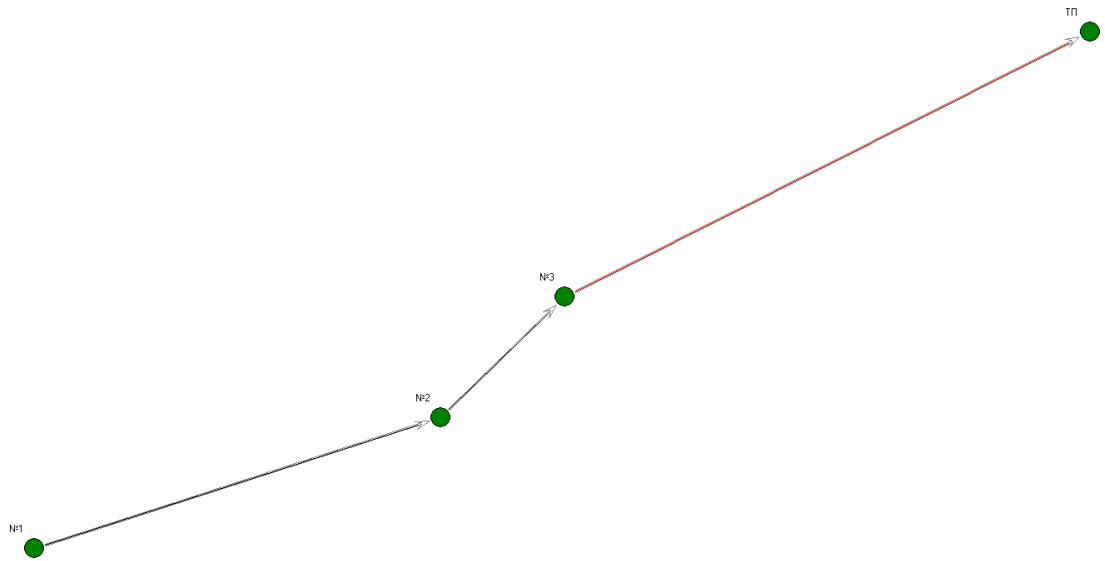


Рисунок 10 – Результати розрахунку

Алгоритм дії додатку представлено в Додатку А.

Вихідний код додатку представлено в Додатку Б.



## ВИСНОВОК

В результаті виконання поставленої задачі розроблено додаток прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі, який враховує:

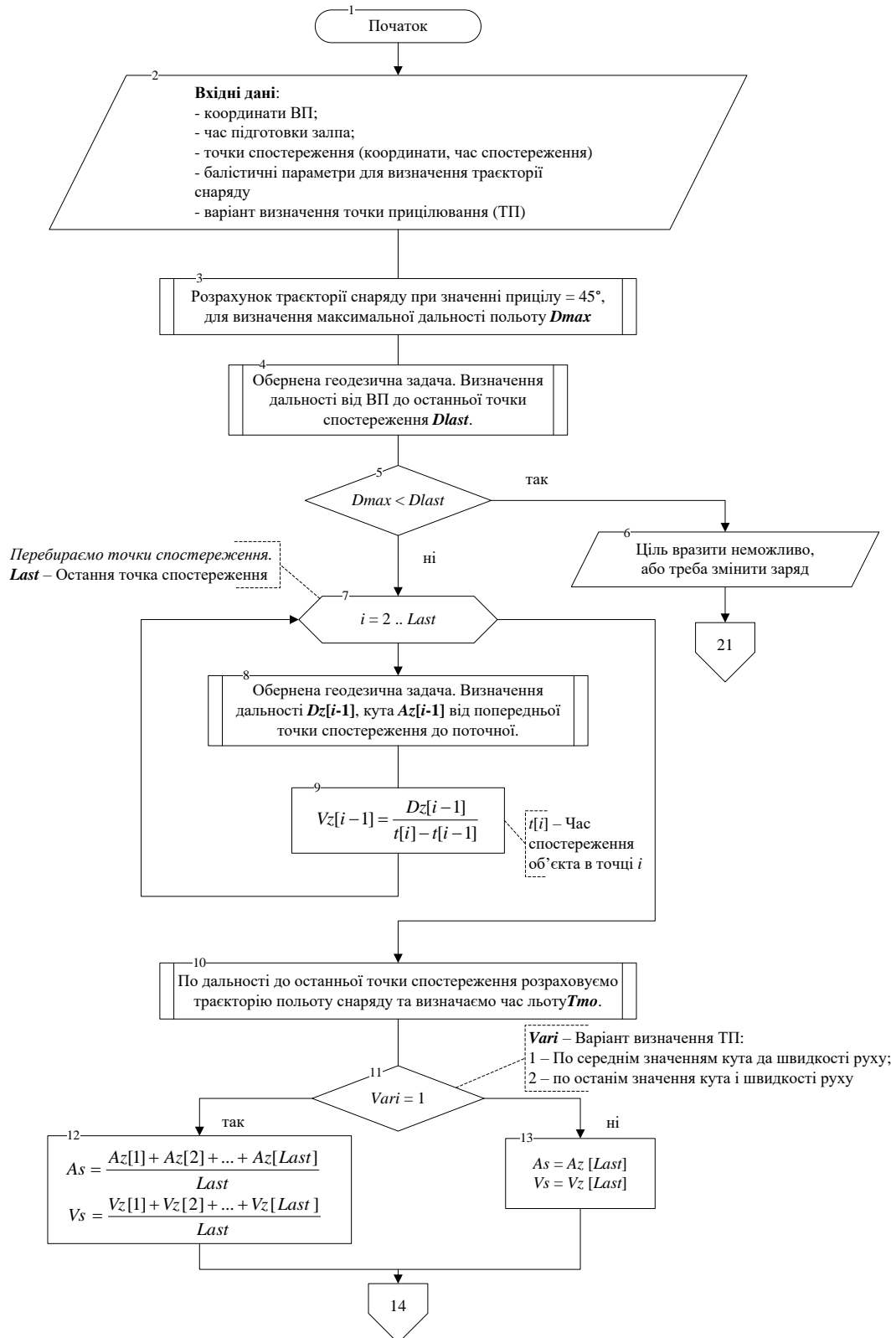
- дію сили інерції, гравітації та сили аеродинамічного опору;
- метеорологічні дані;
- кількість точок фіксації цілі пунктом артилерійської розвідки з урахуванням часових інтервалів;
- час трансферу даних з пункту артилерійської розвідки до артилерійського підрозділу;
- час підготовки артилерійської установки до пострілу.

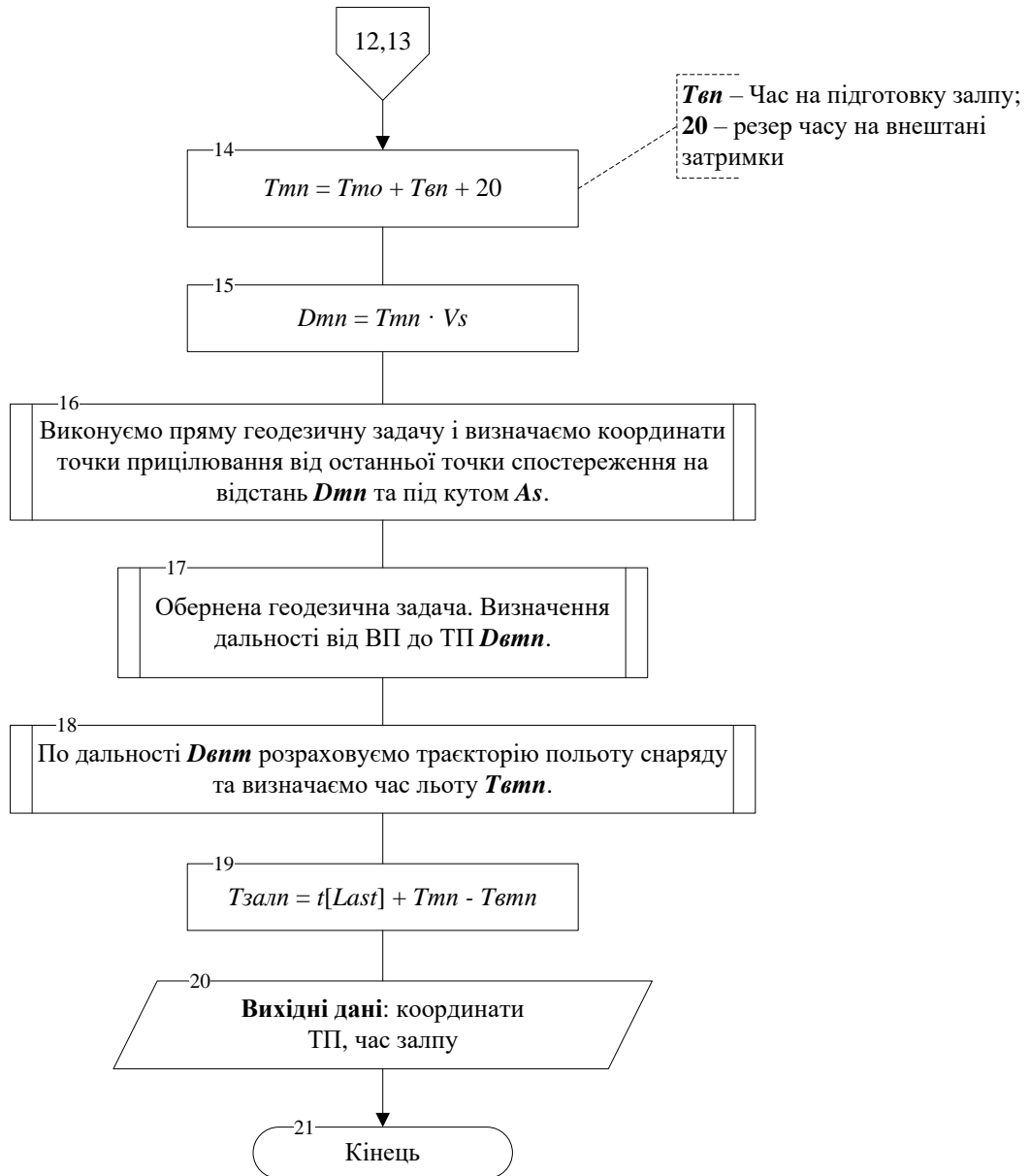
## СПИСОК ЛІТЕРАТУРИ

1. Окулов С.Н. Практикум по объектно-ориентированному программированию. – ЛитРес. – 2020 г. – 371 с.
2. Григорьев А.Б. О чем не пишут в книгах по Delphi. – 2012 г. – 576 с.
3. Пестриков В. М. Delphi на примерах.– БХВ-Петербург. – 2005 г. – 496с.
4. Культин Н. Б. Программирование в Turbo Pascal 7.0 и Delphi, 3 изд. – БХВ-Петербург. – 2012 – 400 с.
5. Пугачев В. С. Общая задача о движении вращающегося артиллерийского снаряда в воздухе // Тр. ВВИА им. Н.Е. Жуковского. – 1940. – Вып. 70. –С. 1-89.
6. Дмитриевский, А. А. Внешняя баллистика / А. А. Дмитриевский. – М.: Машиностроение. – 2005. – 582с.
7. Donald E. Carlucci and Sidney S. Jacobson. (2008). Ballistics, Theory and Design of Guns and Ammunitions. – CRC Press. – 585 p.
8. Bryan Litz. Applied Ballistics for Long-Range Shooting. Applied Ballistics,– LLC 2011. – 578 p.
9. George Klimi. (2009). Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles. – Schiffer – 328 p
10. Крылов, А.Н. Собрание трудов. Т. IV, Баллистика / А. Н. Крылов. – М: Из-во Академии Наук СССР. – 1937. – 444 с.

## ДОДАТОК А

### Блок-схема додатку прогнозування точки прицілювання артилерійської системи з урахуванням руху цілі





## Вихідний код додатку

```
nit FmMain;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, ExtCtrls, Spin, Grids,  
xmldom, XMLDoc, msxmldom, XMLIntf, Math, DateUtils, UnCalk;
```

```
type
```

```
TFormMain = class(TForm)
```

```
  PLeft: TPanel;
```

```
  GrBVP: TGroupBox;
```

```
  Label1: TLabel;
```

```
  Label2: TLabel;
```

```
  Ed_VPX: TEdit;
```

```
  Ed_VPY: TEdit;
```

```
  GrBTarg: TGroupBox;
```

```
  SGrKoor: TStringGrid;
```

```
  PTargTop: TPanel;
```

```
  Label3: TLabel;
```

```
  SE_PoinTarg: TSpinEdit;
```

```
  PAll: TPanel;
```

```
  PTop: TPanel;
```

```
  PClose: TPanel;
```

BBClose: TBitBtn;  
GrBBalistic: TGroupBox;  
Label9: TLabel;  
Ed\_V0: TEdit;  
Label7: TLabel;  
Ed\_P0N: TEdit;  
Label8: TLabel;  
Ed\_MS: TEdit;  
Label10: TLabel;  
Ed\_d: TEdit;  
Label12: TLabel;  
Ed\_LC: TEdit;  
Label13: TLabel;  
Ed\_Ds: TEdit;  
Label19: TLabel;  
Ed\_INA: TEdit;  
Label20: TLabel;  
Ed\_INC: TEdit;  
Label21: TLabel;  
Ed\_RED: TEdit;  
BBCx: TBitBtn;  
SBOpen: TSpeedButton;  
SBSave: TSpeedButton;  
OD1: TOpenDialog;  
SD1: TSaveDialog;  
SBRun: TSpeedButton;  
MRes: TMemo;  
Label14: TLabel;  
Ed\_VPPodG: TEdit;

```

    Img1: TImage;
    GrBTypeRun: TRadioGroup;
    procedure BBCloseClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SE_PoinTargChange(Sender: TObject);
    procedure EdKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);
    procedure BBCxClick(Sender: TObject);
    procedure EtalonToCx;
    procedure SBSaveClick(Sender: TObject);
    procedure DanToFile(Fl:String);
    procedure SBOpenClick(Sender: TObject);
    procedure DanFromFile(Fl:String);
    procedure SBRunClick(Sender: TObject);
    procedure ClearGraph;
    procedure ToGraph(Mass: FMPoint; TP: FPoint);
    procedure LineGraph(X1,Y1,X2,Y2:integer; const TPLine:Boolean);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    FormMain: TFormMain;

    function UglToEdit(ugl:real):string;
    function VhodEnter(Ed,Par:string;var
    TextErr:string;VarMin,VarMax:Double):double;

```

implementation

```
uses FmDictCx;
```

```
{ $R *.dfm }
```

```
// Функція відображення значення real у вигляді кутових координат
```

```
function UglToEdit(ugl:real):string;
```

```
var ugl_c,ugl_d:real;
```

```
    s:string;
```

```
begin
```

```
    result:='';
```

```
    ugl:=round(ugl*100)/100;
```

```
    ugl_c:=abs(Trunc(ugl));
```

```
    ugl_d:=abs(round(Frac(ugl)*100));
```

```
    s:=FloatToStr(ugl_d);
```

```
    if (ugl_d<10) and (ugl_d<>0) then s:='0'+FloatToStr(ugl_d);
```

```
    if ugl_d=0 then s:='00';
```

```
    s:=FloatToStr(ugl_c)+'-'+s;
```

```
    if ugl<0 then s:='-'+s;
```

```
    result:=s;
```

```
end;
```

```
// Функція для зчитування значень з Edit, та перевірка на відповідність
```

```
function VhodEnter(Ed,Par:string;var
```

```
TextErr:string;VarMin,VarMax:Double):double;
```

```
var Ed_s,PWin:string;
```

```
begin
```

```
    PWin:=DECIMALSEPARATOR;
```



```

try
  Ed_s:=StringReplace(trim(Ed),',',PWin,[rfReplaceAll, rfIgnoreCase]);
  Ed_s:=StringReplace(trim(Ed_s),',',PWin,[rfReplaceAll, rfIgnoreCase]);
  Result:=0;
  Result:=StrToFloat(Ed_s);
  if (Result<VarMin) or (Result>VarMax) then TextErr := TextErr + Par + ' - не
відповідає діапазону значень:
'+FloatToStr(VarMin)+'...' +FloatToStr(VarMax)+'"#13#10;
except
  TextErr := TextErr + Par+' - не є числом"#13#10;
end;
end; // function VhodEnter

procedure TFormMain.BBCloseClick(Sender: TObject);
begin
  Close;
end; // procedure BBCloseClick

procedure TFormMain.FormCreate(Sender: TObject);
var i: integer;
begin
  SGrKoor.Cells[0,0]:='№';
  SGrKoor.Cells[1,0]:=' X';
  SGrKoor.Cells[2,0]:=' Y';
  SGrKoor.Cells[3,0]:=' Час';
  for i:=1 to 25 do begin
    SGrKoor.Cells[0,i]:=' '+IntToStr(i);
    SGrKoor.Cells[1,i]:='0';

```

```

    SGrKoor.Cells[2,i]:='0';
    SGrKoor.Cells[3,i]:='0';
end;
EtalonToCx;
end; // procedure FormCreate

procedure TFormMain.SE_PoinTargChange(Sender: TObject);
begin
    if (SE_PoinTarg.Value>=3) and (SE_PoinTarg.Value<=25) then
        SGrKoor.RowCount:=SE_PoinTarg.Value+1;
end; // procedure SE_PoinTargChange

procedure TFormMain.EdKeyUp(Sender: TObject; var Key: Word; Shift:
TShiftState);
var AC: TWinControl;
begin
    if key=VK_RETURN then
        begin
            AC := ActiveControl;
            AC := FindNextControl(AC, True, True, False);
            AC.SetFocus;
        end;
end; // EdKeyUp

procedure TFormMain.BBCxClick(Sender: TObject);
begin
    FmCx.ShowModal;
end; // procedure BBCxClick

```

```
procedure TFormMain.EtalonToCx;
var i,i1:Integer;
begin
  for i:=0 to 3 do
    for i1:=0 to 14 do
      Cx[i,i1]:=Cz_rask[i,i1];
end; // procedure EtalonToCx

procedure TFormMain.SBSaveClick(Sender: TObject);
begin
  if SD1.Execute then DanToFile(SD1.FileName);
end; // procedure SBSaveClick

procedure TFormMain.DanToFile(Fl:String);
var
  XMLDoc: IXMLDocument;
  Node, ChNode : IXMLNode;
  i: Integer;
begin
  XMLDoc:=TXMLDocument.Create(nil);
  XMLDoc.IsEmptyDoc;
  XMLDoc.XML.Text:="";
  XMLDoc.Active := True;
  XMLDoc.Version := '1.0';
  XMLDoc.Encoding := 'utf-8';
  Node := XMLDoc.AddChild('meta_data');
```

```

ChNode := Node.AddChild('VPX');
ChNode.NodeValue:=StringReplace(Ed_VPX.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('VPY');
ChNode.NodeValue:=StringReplace(Ed_VPY.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('VPPodG');
ChNode.NodeValue:=StringReplace(Ed_VPPodG.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('TypeRun');
ChNode.NodeValue:=GrBTypeRun.ItemIndex;

```

```

ChNode := Node.AddChild('V0');
ChNode.NodeValue:=StringReplace(Ed_V0.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('P0N');
ChNode.NodeValue:=StringReplace(Ed_P0N.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('MS');
ChNode.NodeValue:=StringReplace(Ed_MS.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('d');
ChNode.NodeValue:=StringReplace(Ed_d.Text,',',',',[rfReplaceAll, rfIgnoreCase]);

```

```

ChNode := Node.AddChild('LC');
ChNode.NodeValue:=StringReplace(Ed_LC.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('Ds');
ChNode.NodeValue:=StringReplace(Ed_Ds.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('INA');
ChNode.NodeValue:=StringReplace(Ed_INA.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('INC');
ChNode.NodeValue:=StringReplace(Ed_INC.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('RED');
ChNode.NodeValue:=StringReplace(Ed_RED.Text,',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

ChNode := Node.AddChild('PTarg'); ChNode.NodeValue:=SE_PoinTarg.Value;
for i:=1 to SE_PoinTarg.Value do begin

```

```

    ChNode := Node.AddChild('PTargX_'+IntToStr(i));
ChNode.NodeValue:=StringReplace(SGrKoor.Cells[1,i],',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

    ChNode := Node.AddChild('PTargY_'+IntToStr(i));
ChNode.NodeValue:=StringReplace(SGrKoor.Cells[2,i],',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

    ChNode := Node.AddChild('PTargD_'+IntToStr(i));
ChNode.NodeValue:=StringReplace(SGrKoor.Cells[3,i],',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

end;

```

```

for i:=0 to 14 do begin

```

```

    ChNode := Node.AddChild('Cx0_'+IntToStr(i));
ChNode.NodeValue:=StringReplace(FloatToStr(Cx[0,i]),',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

    ChNode := Node.AddChild('Cx1_'+IntToStr(i));
ChNode.NodeValue:=StringReplace(FloatToStr(Cx[1,i]),',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

    ChNode := Node.AddChild('Cx2_'+IntToStr(i));
ChNode.NodeValue:=StringReplace(FloatToStr(Cx[2,i]),',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

    ChNode := Node.AddChild('Cx3_'+IntToStr(i));
ChNode.NodeValue:=StringReplace(FloatToStr(Cx[3,i]),',',',',[rfReplaceAll,
rfIgnoreCase]);

```

```

end;

```

```

XMLDoc.SaveToFile(Fl);

```

```

XMLDoc.Active:=False;

```

```

XMLDoc:=nil;

```

```
ShowMessage('Дані збережено в файл:'#13#10+F1);
end; // procedure DanToFile

procedure TFormMain.SBOpenClick(Sender: TObject);
begin
    if OD1.Execute then DanFromFile(OD1.FileName);
end; // procedure SBOpenClick

procedure TFormMain.DanFromFile(Fl:String);
var
    XMLDoc: IXMLDocument;
    Node, ChNode : IXMLNode;
    i, i1, iz, inod: integer;
    S,SErr: String;
begin
    try
        XMLDoc:=LoadXMLDocument(Fl);
        XMLDoc.Active:=True;
    except
        XMLDoc:=nil;
    // EtalonToCx;
        ShowMessage('Помилка завантаження Даних з файлу');
        exit;
    end;

    Node:=XMLDoc.DocumentElement;
    inod:=Node.ChildNodes.Count;
    for i:=0 to inod-1 do begin
        ChNode:=Node.ChildNodes.Nodes[i];
```

```

if ChNode.NodeName='VPX' then Ed_VPX.Text:=ChNode.NodeValue;
if ChNode.NodeName='VPY' then Ed_VPY.Text:=ChNode.NodeValue;
if ChNode.NodeName='VPPodG' then Ed_VPPodG.Text:=ChNode.NodeValue;
if ChNode.NodeName='TypeRun' then
GrBTypeRun.ItemIndex:=ChNode.NodeValue;
if ChNode.NodeName='V0' then Ed_V0.Text:=ChNode.NodeValue;
if ChNode.NodeName='PON' then Ed_PON.Text:=ChNode.NodeValue;
if ChNode.NodeName='MS' then Ed_MS.Text:=ChNode.NodeValue;
if ChNode.NodeName='d' then Ed_d.Text:=ChNode.NodeValue;
if ChNode.NodeName='LC' then Ed_LC.Text:=ChNode.NodeValue;
if ChNode.NodeName='Ds' then Ed_Ds.Text:=ChNode.NodeValue;
if ChNode.NodeName='INA' then Ed_INA.Text:=ChNode.NodeValue;
if ChNode.NodeName='INC' then Ed_INC.Text:=ChNode.NodeValue;
if ChNode.NodeName='RED' then Ed_RED.Text:=ChNode.NodeValue;
if ChNode.NodeName='PTarg' then SE_PoinTarg.Value:=ChNode.NodeValue;

for i1:=1 to SE_PoinTarg.Value do begin
  if ChNode.NodeName='PTargX_'+IntToStr(i1) then
SGrKoor.Cells[1,i1]:=ChNode.NodeValue;
  if ChNode.NodeName='PTargY_'+IntToStr(i1) then
SGrKoor.Cells[2,i1]:=ChNode.NodeValue;
  if ChNode.NodeName='PTargD_'+IntToStr(i1) then
SGrKoor.Cells[3,i1]:=ChNode.NodeValue;
end;

if (Pos(AnsiLowerCase('Cx0_'), AnsiLowerCase(ChNode.NodeName)) > 0)
then begin
  S:=ChNode.NodeName;
  S:=StringReplace(S,'Cx0_',",[rfReplaceAll, rfIgnoreCase]);
  try

```

```

    iz:=StrToInt(S);
    Cx[0,iz]:=VhodEnter(VarToStr(ChNode.NodeValue),"SErr,-1e20,1e20);
except
    ShowMessage('Помилка завантаження '"+ChNode.NodeName+"'");
end;
end;
if (Pos(AnsiLowerCase('Cx1_'), AnsiLowerCase(ChNode.NodeName)) > 0)
then begin
    S:=ChNode.NodeName;
    S:=StringReplace(S,'Cx1_',",[rfReplaceAll, rfIgnoreCase]);
    try
        iz:=StrToInt(S);
        Cx[1,iz]:=VhodEnter(VarToStr(ChNode.NodeValue),"SErr,-1e20,1e20);
    except
        ShowMessage('Помилка завантаження '"+ChNode.NodeName+"'");
    end;
end;
if (Pos(AnsiLowerCase('Cx2_'), AnsiLowerCase(ChNode.NodeName)) > 0)
then begin
    S:=ChNode.NodeName;
    S:=StringReplace(S,'Cx2_',",[rfReplaceAll, rfIgnoreCase]);
    try
        iz:=StrToInt(S);
        Cx[2,iz]:=VhodEnter(VarToStr(ChNode.NodeValue),"SErr,-1e20,1e20);
    except
        ShowMessage('Помилка завантаження '"+ChNode.NodeName+"'");
    end;
end;
if (Pos(AnsiLowerCase('Cx3_'), AnsiLowerCase(ChNode.NodeName)) > 0)
then begin

```



```

S:=ChNode.NodeName;
S:=StringReplace(S,'Cx3_',",[rfReplaceAll, rfIgnoreCase]);
try
  iz:=StrToInt(S);
  Cx[3,iz]:=VhodEnter(VarToStr(ChNode.NodeValue),"SErr,-1e20,1e20);
except
  ShowMessage('Помилка завантаження "'+ChNode.NodeName+'");
end;
end;

end;
XMLDoc:=nil;

ShowMessage('Дані відновлено з файлу:'#13#10+F1);
end; // procedure DanFromFile

procedure TFormMain.SBRunClick(Sender: TObject);
var SErr: String;

MdT,Mdh,Mdht,MP0N,MMS,Md,MLC,MDs,MV0,MDEL0,MRO0,MINA,MINC,
MTAUG,MTAUM,MRED,MWX,MWZ:Double;

Alf,DMax,VPPodG,oTm,oTmA,sA,sV,sD: Double;

i, TypeRun: Integer;

Res: TMass;

VP,TP: FPoint;

Koor: FMPoint;

zD, zA, oD, oA, oV, KoorT: array of double;

Ts: TTime;

h,m,sec,ms: Word;

```

```

function TimeSnfly(D1: Double):Double;
var Kut1, dKut, dKut2, dD: Double;
    Prov: Boolean;
    i: integer;
begin
    kut1:=22.5;
    dKut:=3;
    dD:=D1;
    Prov:=False;
    i:=0;
    while (dD>25) and (i<25) do begin
        i:=i+1;

        Daln(MdT,Mdh,Mdht,MP0N,MMS,Md,MLC,MDs,MV0,MDEL0,MRO0,MINA,
        MINC,kut1,MTAUM,MRED,MWX,MWZ,Res);
        dD:=abs(D1-Res[0]);
        Result:=Res[9];
        if i=1 then
            if Res[0]>D1 then dKut:=-abs(dKut) else dKut:=abs(dKut);
        if (Res[0]>D1) and (dKut>0) then dKut2:=-abs(dKut/2);
        if (Res[0]<D1) and (dKut>0) then dKut2:=abs(dKut);
        if (Res[0]>D1) and (dKut<0) then dKut2:=-abs(dKut);
        if (Res[0]<D1) and (dKut<0) then dKut2:=abs(dKut/2);
        dKut:=dKut2;
        Kut1:=Kut1+dKut;
        if Kut1>45 then Kut1:=45;
        if Kut1<2 then Kut1:=2;
    end;
end;

```

```
end;
```

```
procedure final;
```

```
begin
```

```
  Koor:=nil;
```

```
  KoorT:=nil;
```

```
  Res:=nil;
```

```
  zD:=nil;
```

```
  zA:=nil;
```

```
  oD:=nil;
```

```
  oA:=nil;
```

```
  oV:=nil;
```

```
end;
```

```
begin
```

```
// Ввод даних
```

```
  SErr:="";
```

```
  MRes.Lines.Clear;
```

```
  ClearGraph;
```

```
  VP.X:=Round(VhodEnter(Ed_VPX.Text,'ВП X',SErr,0,1e7));
```

```
  VP.Y:=Round(VhodEnter(Ed_VPY.Text,'ВП Y',SErr,0,1e7));
```

```
  if (VP.X=0) or (VP.Y=0) then SErr:=SErr+'Координати ВП не повинні бути  
  рівними 0'#13#10;
```

```
  VPPodG:=Round(VhodEnter(Ed_VPPodG.Text,'Час на підготовку  
  наведення',SErr,0,1e7));
```

```
  SetLength(Koor,SE_PoinTarg.Value);
```

```
  SetLength(KoorT,SE_PoinTarg.Value);
```

```
for i:=0 to SE_PoinTarg.Value-1 do begin
```

```
  Koor[i].X:=Round(VhodEnter(SGrKoor.Cells[1,i+1],'Точка №'+IntToStr(i+1)+'
X',SErr,0,1e7));
```

```
  Koor[i].Y:=Round(VhodEnter(SGrKoor.Cells[2,i+1],'Точка №'+IntToStr(i+1)+'
Y',SErr,0,1e7));
```

```
  try
```

```
    Ts:=StrToTime(SGrKoor.Cells[3,i+1]);
```

```
    if Ts=0 then SErr:=SErr+'Час спостереження '+IntToStr(i+1)+' точки не
повинен бути равним 0'#13#10;
```

```
    DecodeTime(Ts,h,m,sec,ms);
```

```
    KoorT[i]:=h*60*60+m*60+sec;
```

```
  except
```

```
    SErr:=SErr+'Точка '+IntToStr(i)+' час спостереження - помилка'#13#10;
```

```
  end;
```

```
end;
```

```
TypeRun:=GrBTypeRun.ItemIndex;
```

```
MV0:=VhodEnter(Ed_V0.Text,'V0',SErr,-1e20,1e20);
```

```
MRO0:=1858;
```

```
MTAUM:=0;
```

```
MdT:=0;
```

```
Mdh:=0;
```

```
MWX:=0;
```

```
MWZ:=0;
```

```
Mdht:=0;
```

```
MP0N:=VhodEnter(Ed_P0N.Text,'P0N',SErr,-1e20,1e20);
```

```
MMS:=VhodEnter(Ed_MS.Text,'MS',SErr,-1e20,1e20);
```

```
Md:=VhodEnter(Ed_d.Text,'d',SErr,-1e20,1e20);
```

```
MLC:=VhodEnter(Ed_LC.Text,'LC',SErr,-1e20,1e20);
```

```
MDs:=VhodEnter(Ed_Ds.Text,'Ds',SErr,-1e20,1e20);
```

```

MDEL0:=0.001;
MINA:=VhodEnter(Ed_INA.Text,'INA',SErr,-1e20,1e20);
MINC:=VhodEnter(Ed_INC.Text,'INC',SErr,-1e20,1e20);
MRED:=VhodEnter(Ed_RED.Text,'RED',SErr,-1e20,1e20);

if SErr<>" then begin
    ShowMessage(SErr);
    final;
    exit;
end;

// Визначення максимальної дальності стрільби
MTAUG:=45;

Daln(MdT,Mdh,Mdht,MP0N,MMS,Md,MLC,MDs,MV0,MDEL0,MRO0,MINA,
MINC,MTAUG,MTAUM,MRED,MWX,MWZ,Res);

DMax:=Res[0];

MRes.Lines.Add('Максимальна дальність стрільби для даного снаряду:
'+FloatToStr(RoundTo(DMax,-2)));

// Визначення дальності до точок маршрута

MRes.Lines.Add('Дальність та кут з ВП на точки спостереження руху:');
for i:=1 to SE_PoinTarg.Value do begin
    SetLength(zD,Length(zD)+1);
    SetLength(zA,Length(zA)+1);
    SErr:=ObrGeo(VP.X,VP.Y,Koor[i-1].X,Koor[i-1].Y,zD[Length(zD)-
1],zA[Length(zA)-1]);

    if SErr<>" then begin

```

```

ShowMessage(SErr);

final;

exit;

end;

MRes.Lines.Add(IntToStr(i)+' Д='+FloatToStr(RoundTo(zD[Length(zD)-1],-2))+'); alf='+UglToEdit(zA[Length(zA)-1]));

if i<>1 then begin

  SetLength(oD,Length(oD)+1);

  SetLength(oA,Length(oA)+1);

  SetLength(oV,Length(oV)+1);

  SErr:=ObrGeo(Koor[i-2].X,Koor[i-2].Y,Koor[i-1].X,Koor[i-1].Y,oD[Length(oD)-1],oA[Length(oA)-1]);

  if SErr<>" then begin

    ShowMessage(SErr);

    final;

    exit;

  end;

  oV[Length(oV)-1]:=oD[Length(oD)-1]/(KoorT[i-1]-KoorT[i-2]);

end;

end;

MRes.Lines.Add('Дані по точкам спостереження:');

for i:=0 to Length(oD)-1 do

  MRes.Lines.Add(IntToStr(i+1)+' діляниця Д='+FloatToStr(RoundTo(oD[i],-2))+'); alf='+UglToEdit(oA[i])+'; V='+FloatToStr(RoundTo(oV[i],-2))+ ' м/с ('+FloatToStr(RoundTo(oV[i]*3.6,-2))+ ' км/ч)');

if zD[Length(zD)-1]>DMax then begin

```

```

    ShowMessage('Дальність до останньої точки спостереження більша ніж
    МАХ стрільби');

    final;

    exit;

end;

// Розраховуємо час льоту до останньої точки спостереження
oTm:=TimeSnfly(zD[Length(zD)-1]);

MRes.Lines.Add('Час льоту до останньої точки спостереження
'+FloatToStr(RoundTo(oTm,-2))+' сек');

case TypeRun of
0: begin
    sA:=0;
    sV:=0;
    for i:=0 to Length(oA)-1 do begin
        sA:=sA+oA[i];
        sV:=sV+oV[i];
    end;
    sA:=sA/Length(oA);
    sV:=sV/Length(oV);
    MRes.Lines.Add('Тип визначення ТП: по середній значенням');
end;
1: begin
    sA:=oA[Length(oA)-1];
    sV:=oV[Length(oV)-1];
    MRes.Lines.Add('Тип визначення ТП: по останнім значенням');
end;

end;

MRes.Lines.Add('Кут руху = '+UglToEdit(sA)+' ('+FloatToStr(RoundTo(sA*6,-
2))+'°); V = '+FloatToStr(RoundTo(sV,-2))+' м/с');

```

```

oTmA:=oTm+VPPodG+20;

MRes.Lines.Add('Час визначення ТП = "Час льоту" + "Час підготовки" + "20
(резерв)" = '+FloatToStr(RoundTo(oTmA,0))+' сек');

sD:=oTmA*sV;

MRes.Lines.Add('Дальність від останньої точки спостереження = "Час ТП" *
"Швидкість руху" = '+FloatToStr(RoundTo(sD,0))+' м. ');

PrymaGeo(Koor[Length(Koor)-1].X,Koor[Length(Koor)-
1].Y,sD,sA,TP.X,TP.Y);

MRes.Lines.Add('Координати ТП: X='+FloatToStr(RoundTo(TP.X,0))+';
Y='+FloatToStr(RoundTo(TP.Y,0)));

SErr:=ObrGeo(VP.X,VP.Y,TP.X,TP.Y,sD,sA);

if SErr<>" then begin
  ShowMessage(SErr);
  final;
  exit;
end;

MRes.Lines.Add('Дальність від ВП до ТП ='+FloatToStr(RoundTo(sD,0))+';');

oTm:=TimeSnfly(sD);

MRes.Lines.Add('Час льоту снаряду на відстань до ТП =
'+FloatToStr(RoundTo(oTm,-2))+' сек');

oTm:=oTmA-oTm;

Ts:=IncSecond(Ts,Round(oTm));

MRes.Lines.Add('Час залпа: '+TimeToStr(Ts)+');

ToGraph(Koor,TP);

//

final;

end;

```



```

procedure TFormMain.ClearGraph;
var Dest: TRect;
begin
  Dest:= Rect(0, 0, Img1.Width, Img1.Height);
  Img1.Canvas.Pen.Color:=ClBlack;
  Img1.canvas.Pen.Style:=psSolid;
  Img1.Canvas.Brush.Color := clBtnFace;
  Img1.Canvas.FillRect(Dest);
end;

procedure TFormMain.ToGraph(Mass: FMPoint; TP: FPoint);
var g_minX, g_minY, g_maxX, g_maxY, x_shag,X,Y: double;
  Dest: TRect;
  MinMax:boolean;
  i: Integer;
  GrKoor: FmPoint;
  kf, kfX, kfY: Double;
  MinB, MinXB, MinYB: Double;
  S: String;
begin
  Dest:= Rect(0, 0, Img1.Width, Img1.Height);
  Img1.Canvas.Pen.Color:=ClBlack;
  Img1.canvas.Pen.Style:=psSolid;
  Img1.Canvas.Brush.Color := clBtnFace;
  Img1.Canvas.FillRect(Dest);
  GrKoor:=nil;
  for i:=0 to Length(Mass)-1 do begin
    SetLength(GrKoor,Length(GrKoor)+1);
    GrKoor[i].Y:=Mass[i].X;
    GrKoor[i].X:=Mass[i].Y;
  end;
end;

```

```

if (not MinMax) then begin
  g_minX:=GrKoor[i].X;
  g_maxX:=GrKoor[i].X;
  g_minY:=GrKoor[i].Y;
  g_maxY:=GrKoor[i].Y;
  MinMax:=True;
end;
end;
SetLength(GrKoor,Length(GrKoor)+1);
GrKoor[Length(GrKoor)-1].Y:=TP.X;
GrKoor[Length(GrKoor)-1].X:=TP.Y;
for i:=0 to Length(GrKoor)-1 do begin
  if (g_minX>GrKoor[i].X) then g_minX:=GrKoor[i].X;
  if (g_maxX<GrKoor[i].X) then g_maxX:=GrKoor[i].X;
  if (g_minY>GrKoor[i].Y) then g_minY:=GrKoor[i].Y;
  if (g_maxY<GrKoor[i].Y) then g_maxY:=GrKoor[i].Y;
end;
if (g_minY=g_maxY) then begin
  g_minY:=g_minY-1;
  g_maxY:=g_maxY+1;
end;
MinXB:=Img1.Width/10;
MinYB:=Img1.Height/10;
kfX:=(Img1.Width-2*MinXB)/(g_maxX-g_minX);
kfY:=(Img1.Height-2*MinYB)/(g_maxY-g_minY);
if kfX<kfY then begin
  kf:=kfX;
  MinB:=(Img1.Height-Img1.Width)/2;
  MinXB:=MinXB;

```

```

    MinYB:=(Img1.Height-Img1.Width)/2+MinYB;
{!}  Dest:= Rect(0, Round(MinB), Img1.Width, Round(Img1.Height-MinB));
end else begin
    kf:=kfY;
    MinB:=(Img1.Width-Img1.Height)/2;
    MinYB:=MinYB;
    MinXB:=(Img1.Width-Img1.Height)/2+MinXB;
{!}  Dest:= Rect(Round(MinB), 0, Round(Img1.Width-MinB), Img1.Height);
end;
{!}  Img1.Canvas.Pen.Color:=ClBlack;
{!}  Img1.Canvas.Pen.Style:=psSolid;
{!}  Img1.Canvas.Brush.Color := ClWhite;
{!}  Img1.Canvas.FillRect(Dest);

for i:=0 to Length(GrKoor)-1 do begin
    Img1.Canvas.Brush.Color:=clGreen;
    Img1.Canvas.Pen.Style:=psSolid;
{  Img1.Canvas.Ellipse(Round(MinXB+(GrKoor[i].X-g_minX)*kfX-12),
Round(Img1.Height-MinYB-(GrKoor[i].Y-g_minY)*kfY-12),
Round(MinXB+(GrKoor[i].X-g_minX)*kfX+12), Round(Img1.Height-MinYB-
(GrKoor[i].Y-g_minY)*kfY+12));}

    Img1.Canvas.Ellipse(Round(MinXB+(GrKoor[i].X-g_minX)*kf-12),
Round(Img1.Height-MinYB-(GrKoor[i].Y-g_minY)*kf-12),
Round(MinXB+(GrKoor[i].X-g_minX)*kf+12), Round(Img1.Height-MinYB-
(GrKoor[i].Y-g_minY)*kf+12));

    if i>0 then
{
    if i<>Length(GrKoor)-1
        then LineGraph(Round(MinXB+(GrKoor[i-1].X-g_minX)*kfX),
Round(Img1.Height-MinYB-(GrKoor[i-1].Y-g_minY)*kfY),
Round(MinXB+(GrKoor[i].X-g_minX)*kfX), Round(Img1.Height-MinYB-
(GrKoor[i].Y-g_minY)*kfY), False)

```

```

    else LineGraph(Round(MinXB+(GrKoor[i-1].X-g_minX)*kfX),
Round(Img1.Height-MinYB-(GrKoor[i-1].Y-g_minY)*kfY),
Round(MinXB+(GrKoor[i].X-g_minX)*kfX), Round(Img1.Height-MinYB-
(GrKoor[i].Y-g_minY)*kfY), True);
}
    if i<>Length(GrKoor)-1
        then LineGraph(Round(MinXB+(GrKoor[i-1].X-g_minX)*kf),
Round(Img1.Height-MinYB-(GrKoor[i-1].Y-g_minY)*kf),
Round(MinXB+(GrKoor[i].X-g_minX)*kf), Round(Img1.Height-MinYB-
(GrKoor[i].Y-g_minY)*kf), False)
        else LineGraph(Round(MinXB+(GrKoor[i-1].X-g_minX)*kf),
Round(Img1.Height-MinYB-(GrKoor[i-1].Y-g_minY)*kf),
Round(MinXB+(GrKoor[i].X-g_minX)*kf), Round(Img1.Height-MinYB-
(GrKoor[i].Y-g_minY)*kf), True);
    if i<>Length(GrKoor)-1
        then S:='№'+IntToStr(i+1)
        else S:='TII';
    Img1.Canvas.Brush.Color:=clWhite;
    Img1.Canvas.TextOut(Round(MinXB+(GrKoor[i].X-g_minX)*kf-
30),Round(Img1.Height-MinYB-(GrKoor[i].Y-g_minY)*kf-30),S);
end;
//
GrKoor:=nil;
end;

procedure TFormMain.LineGraph(X1,Y1,X2,Y2:integer; const TPLine:Boolean);
var alfa, d: Double;
    P1,P2: FPoint;
    R1,R2,R3,R4,R5: FPoint;
begin
    P1.X:=X1;
    P1.Y:=Y1;

```

P2.X:=X2;

P2.Y:=Y2;

d:=sqrt(sqr(P2.X-P1.X)+sqr(P2.Y-P1.Y));

if d<40 then exit;

if (P2.X-P1.X)<>0 then alfa:=arctan((P2.Y-P1.Y)/(P2.X-P1.X))

    else alfa:=pi/2;

Img1.canvas.Pen.Color:= clGray;

R1.X:=P1.X+14\*cos(alfa); R1.Y:=P1.Y+14\*Sin(alfa);

    Img1.canvas.MoveTo(Round(R1.X),Round(R1.Y));

R2.X:=R1.X+1\*cos(alfa+pi/2); R2.Y:=R1.Y+1\*Sin(alfa+pi/2);

    Img1.canvas.LineTo(Round(R2.X),Round(R2.Y));

R3.X:=R2.X+(d-38)\*cos(alfa); R3.Y:=R2.Y+(d-38)\*Sin(alfa);

    Img1.canvas.LineTo(Round(R3.X),Round(R3.Y));

R4.X:=R3.X+8\*cos(alfa+5/6\*pi); R4.Y:=R3.Y+8\*Sin(alfa+5/6\*pi);

    Img1.canvas.LineTo(Round(R4.X),Round(R4.Y));

R5.X:=P1.X+(d-14)\*cos(alfa); R5.Y:=P1.Y+(d-14)\*Sin(alfa);

    Img1.canvas.LineTo(Round(R5.X),Round(R5.Y));

    Img1.canvas.MoveTo(Round(R1.X),Round(R1.Y));

R2.X:=R1.X+1\*cos(alfa-pi/2); R2.Y:=R1.Y+1\*Sin(alfa-pi/2);

    Img1.canvas.LineTo(Round(R2.X),Round(R2.Y));

R3.X:=R2.X+(d-38)\*cos(alfa); R3.Y:=R2.Y+(d-38)\*Sin(alfa);

    Img1.canvas.LineTo(Round(R3.X),Round(R3.Y));

R4.X:=R3.X+8\*cos(alfa-5/6\*pi); R4.Y:=R3.Y+8\*Sin(alfa-5/6\*pi);

    Img1.canvas.LineTo(Round(R4.X),Round(R4.Y));

R5.X:=P1.X+(d-14)\*cos(alfa); R5.Y:=P1.Y+(d-14)\*Sin(alfa);

```
Img1.canvas.LineTo(Round(R5.X),Round(R5.Y));
```

```
    if TPLine then Img1.canvas.Pen.Color:=clRed else  
Img1.canvas.Pen.Color:=clBlack;
```

```
    Img1.canvas.MoveTo(Round(R1.X),Round(R1.Y));
```

```
    R5.X:=P1.X+(d-28)*cos(alfa); R5.Y:=P1.Y+(d-28)*Sin(alfa);
```

```
    Img1.canvas.LineTo(Round(R5.X),Round(R5.Y));
```

```
end;
```

```
end.
```