

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

ВИПУСКНА РОБОТА

на тему:

**«Моделювання руху некерованих літальних апаратів в збуреному
середовищі»**

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Проценко О.Б.

Студента групи ІН – 63

Шимко Є.А.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедри Довбиш А.С.

“ _____ ” _____ 2020 р.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи ІН-63 спеціальності “Інформатика” денної форми навчання Шимко Євгена Анатолійовича.

Тема: “Моделювання руху некерованих літальних апаратів в збуреному середовищі”

Затверджена наказом по СумДУ

№ _____ от _____ 2020 р.

Зміст пояснювальної записки: 1) інформаційний огляд моделювання руху некерованих літальних апаратів в збуреному середовищі; 2) постановка завдання та визначення цілей роботи; 3) огляд вибраних рішень для створення зручної та якісної програми на ОС WINDOWS 4) створення математичної моделі; 5) перевірка роботи програми.

Дата видачі завдання “ _____ ” _____ 2020р.

Керівник випускної роботи _____ Проценко О.Б.

Завдання прийняв до виконання _____ Шимко Є. А.

РЕФЕРАТ

Записка: 93 стор., 19 рис., 7 табл., 1 додаток, 15 джерел.

Об'єкт дослідження — моделювання руху некерованих літальних апаратів в збуреному середовищі.

Мета роботи — розробка математичної моделі та програма для визначення вирахованих установок для стрільби на ураження різноманітних цілей мінометними комплексами 120-мм мінометом або 82 мм. мінометом з метою скорочення часу розрахунків на полі бою.

Методи дослідження — математична модель для визначення вирухуваних установок.

Результати — розроблено алгоритм та програмне забезпечення моделювання руху некерованих літальних апаратів в збуреному середовищі. Програма вміє самостійно визначати топографічні дані по цілі на основі вхідних даних за допомогою математичних алгоритмів. Реалізований механізм складання бюлетеня «Метеосередній», який використовують в артилерії. За допомогою даних отриманих в результаті виконання алгоритмів для знаходження топографічних даних та бюлетеня «Метеосередній», програма вміє знаходити вираховані установки, які потрібні для точної стрільби із мінометів.

МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ ВИЗНАЧЕННЯ
ТОПОГРАФІЧНИХ ДАНИХ ПО ЦІЛІ, СКЛАДАННЯ БЮЛЕТЕНЯ
«МЕТЕОСЕРЕДНІЙ», ВИЗНАЧЕННЯ ВИРАХУВАНИХ
УСТАНОВОК ДЛЯ СТРІЛЬБИ ІЗ 120-ММ АБО 82-ММ
МІНОМЕТІВ, ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ
МОДЕЛІ НА ОС WINDOWS

ЗМІСТ

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД	6
1.1 Складова повної аеродинамічної сили	6
1.2 Аеродинамічні моменти	9
1.3 Рух центра мас некерованих реактивних ЛА на активній ділянці траєкторії.....	12
1.4 Рух центра мас некерованих літальних апаратів на пасивній ділянці траєкторії.....	15
1.5 Постановка завдання.....	18
2 ВИБІР МЕТОДУ РІШЕННЯ	19
2.1 Основні переваги та недоліки мови програмування C++.....	19
2.2 Вибір середовища розробки.....	24
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	30
3.1 Математична модель	30
3.2 Інформаційна модель	35
3.3 Перевірка роботи програми.....	36
ВИСНОВКИ.....	41
СПИСОК ЛІТЕРАТУРИ.....	42
ДОДАТОК	44

ВСТУП

На даний момент у Збройних силах України (ЗСУ) існує потреба в блискавичній артилерійській відповіді агресору, саме тому була розроблена спеціальна програма для ОС Windows (вибір впав на Windows, бо вона найбільш розповсюджена та через легкість розробки програмного забезпечення для цієї ОС).

Мета роботи – розробка математичної моделі та програми для визначення вирахованих установок для стрільби на ураження різноманітних цілей мінометними комплексами 120-мм мінометом або 82 мм. мінометом з метою скорочення часу розрахунків на полі бою.

Завданням роботи є: запропонувати математичну модель та розробити програму руху некерованих літальних апаратів в збуреному середовищі; розробити алгоритм і програму розрахунку бюлетня «метеосередній» по даним наземних вимірювань, топографічним даних по цілям (об'єктам) ураження; програма повинна дозволяти розраховувати вираховані установки по цілям (об'єктам) ураження для 120-мм мінометом або 82-мм мінометом з високою точністю.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Складова повної аеродинамічної сили

У залежності від умов руху літального апарата (ЛА) або реактивних ЛА вектор повної аеродинамічної сили або сили опору повітря орієнтована деяким чином у просторі відносно ЛА. Тому прийнято визначати не саму повну аеродинамічну силу, а її складові у „швидкісній”, і „пов’язаній” системах координат (рис. 1.1) [1].

„Швидкісна” або поточна система координат $OXYZ$ має початок у центрі мас ЛА, вісь OX співпадає з вектором швидкості центра мас ЛА (наприклад, у плоскості вертикальної пари стабілізаторів) перпендикулярно осі OY , вісь OZ спрямована праворуч перпендикулярно плоскості OXY .

„Пов’язана” система координат $OX_1Y_1Z_1$ має початок у центрі мас ЛА O , вісь OX_1 – поздовжня вісь ЛА (направлена до вершини ЛА), вісь OY_1 – розташована у головній плоскості симетрії перпендикулярно вісі OX_1 вісь OZ_1 спрямована праворуч перпендикулярно головній плоскості симетрії ЛА OX_1Y_1 .

Взаємне положення швидкісної і пов’язаної систем координат визначається кутами атаки і ковзання (рис. 1.1). Кут атаки α – це кут у головній плоскості симетрії ЛА між поздовжньою віссю ЛА OX_1 і проекцією вектора V швидкості центра мас ЛА на її головну плоскість симетрії OX_1Y_1 . Кут ковзання β – кут між вектором швидкості центра мас ЛА та його проекцією на головну плоскість симетрії ЛА.

У балістиці некерованих ЛА прийнято також розглядати кут між поздовжньою віссю ЛА і вектором швидкості центра мас, названий кутом нутації δ . Кути атаки і ковзання α і β є складові кута нутації δ .

У швидкісній системі координат повна аеродинамічна сила (повна сила опору повітря) має наступні складові (рис. 1.1): по вісі OX – силу лобового опору R_x , по вісі OY – підймальну силу R_y , по вісі OZ – бокову силу R_z .

Для вісісиметричних некерованих ЛА у якості головної площини симетрії зручно приймати площину кута нутації δ , що відповідає випадку польоту ЛА тільки з кутом атаки α , який дорівнює куту нутації δ при відсутності кута

ковзання ($\alpha=\delta; \beta=0$). Тому для вісісиметричних некерованих ЛА приймають, що вісь OY підйомальної сили і підйомальна сила R_y , розташовані у плоскості кута нутації перпендикулярно вектору швидкості V (вісі OX швидкісної системи координат), а бокова вісь OZ спрямована праворуч перпендикулярно плоскості кута нутації (рис. 1.1).

Із розгляду фізичних причин виникнення опору повітря слідує, що величини повної аеродинамічної сили R , а отже і її складових R_x, R_y, R_z залежать від ряду параметрів, які характеризують умови руху ЛА.

Виходячи з співвідношень теорії подібності, приймаємо наступні залежності для складових повної аеродинамічної сили у швидкісній системі координат:

$$\begin{cases} R_x = c_x \frac{\rho V^2}{2} S_M \\ R_y = c_y \frac{\rho V^2}{2} S_M \\ R_z = c_z \frac{\rho V^2}{2} S_M \end{cases} \quad (1)$$

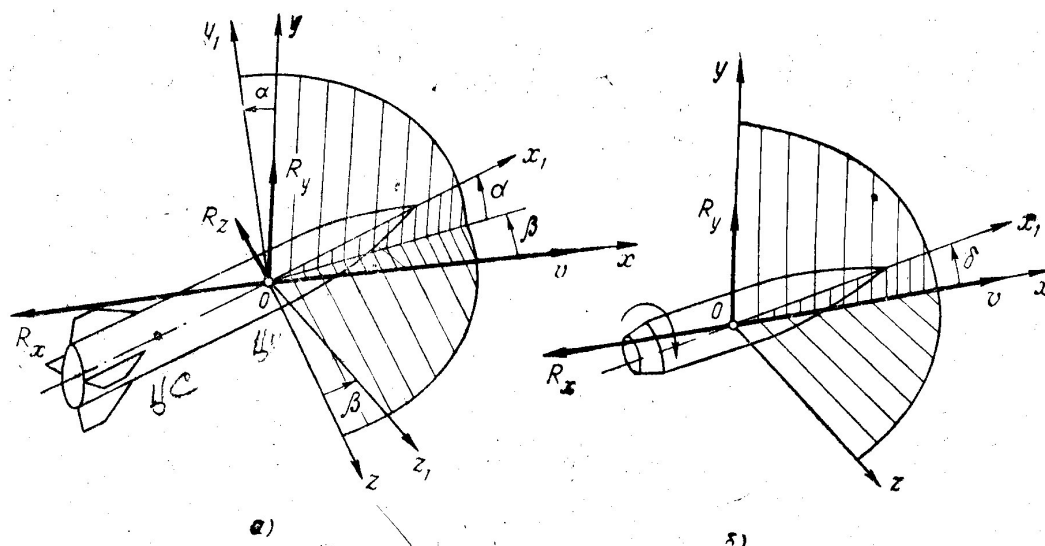


Рисунок 1.1 - Швидкісна і пов'язана системи координат

де V – швидкість ЛА, м/сек.; ρ – масова щільність повітря, кг·сек²/м⁴;
 $\frac{\rho V^2}{2}$ – так званий швидкісний напір, який має розмірність тиску кг/м², являє собою кінетичну енергію одиниці об'єму повітря, що має щільність ρ і швидкість,

рівну швидкості ЛА V ; $SM = \frac{\pi d^2}{4}$ - площа найбільшого поперечного перетину корпусу ЛА (площа міделя); c_x, c_y, c_z - безрозмірні аеродинамічні коефіцієнти, які мають найменування відповідних сил (коефіцієнти сили лобового опору c_x , підйимальної сили c_y , бокової сили c_z).

Аеродинамічні коефіцієнти c_x, c_y, c_z , які визначаються, як правило, щодо дослідних даних, опосередковано враховують вплив усіх тих факторів, які характеризують умови польоту ЛА, що безпосередньо не входять до залежності (1). Вони суттєво залежать від форми ЛА, числа Маха $M = \frac{V}{a}$ і кутів α, β або δ , які характеризують орієнтацію поздовжньої вісі ЛА відносно вектора швидкості.

При малих кутах α і β коефіцієнти c_y і c_z пропорційні відповідним кутам: $c_y = \alpha$; $c_z = \beta$, причому коефіцієнти пропорційності (градієнти), які залежать від c_y^α форми c_z^β ЛА і числа M .

Прийнято, що повна аеродинамічна сила (сила опору повітря), як рівнодіюча усіх елементарних сил тертя і тиску повітря, розподілених по поверхні ЛА, має точку прикладання на вісь літального апарата, названу центром тиску. Центр тиску (ЦТ) опереного літального апарата розташовано позаду центру мас (ЦМ), а центр тиску неопереного літального апарата - попереду центру мас, як показано на (рис. 1.2), де складові повної аеродинамічної сили R_x і R_y , відповідають польоту ЛА з кутом нутації δ .

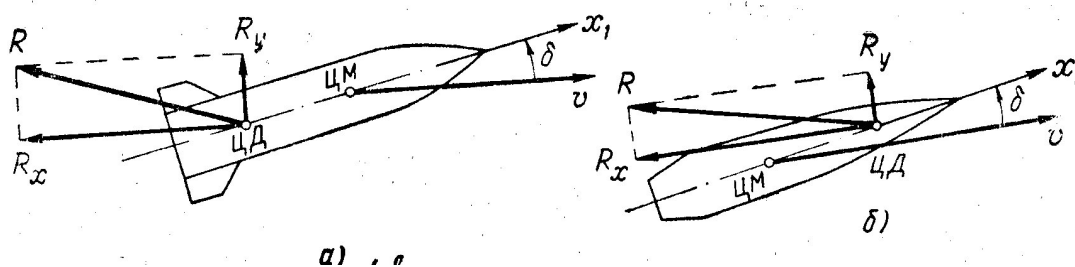


Рисунок 1.2 - Положення центру тиску

a - для опереного літального апарата; b - для літального апарата, що обертається

Згідно правилу теоретичної механіки про переніс сил аеродинамічні сили R, R_x, R_y, R_z можна вважати прикладеними у центрі мас літального апарата (рис. 1) але при цьому слід враховувати моменти цих сил відносно центра мас. [2]

1.2 Аеродинамічні моменти

Стабілізуючий аеродинамічний момент опереного літального апарата – це момент повної аеродинамічної сили відносно центра мас літального апарата, які виникають при кутовому відхиленні поздовжньої вісі ЛА від напрямку вектора швидкості. Він прагне зменшити це кутове відхилення, тим самим повернути літальний апарат у положення рівноваги, при якому поздовжня вісь ЛА співпадає з напрямком вектора швидкості і обтікання ЛА потоком повітря, відбувається симетрично.

Виникнення стабілізуючого моменту пов'язано з тим, що у оперених ЛА центр тиску розташований позаду центра мас таким же чином як і точка прикладання вектора повної аеродинамічної сили. Тому з появою кута атаки α (кута нутації δ) сила лобового опору R_x і підймальна сила R_y створюють момент відносно центра мас, який прагне повернути ЛА у напрямку, протилежному кутовому відхиленню α (δ), тобто усунути це кутове відхилення і стабілізувати поздовжню вісь ЛА у положення, яке співпадає з напрямком вектора швидкості (рис. 1.3)

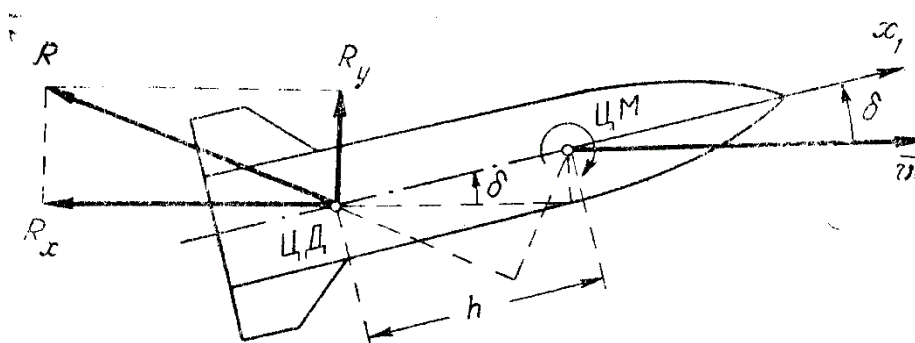


Рисунок 1.3 - Дія стабілізуючого моменту

Стабілізуючий момент відносно пов'язаної поперечної (екваторіальної) вісі z_1 , яка проходить через центр мас опереного Л.А., визначається залежністю:

$$M_{z_1}^{CT} = -m_{z_1}^{CT} \frac{\rho V^2}{2} S_M L_K \quad (2)$$

де $m_{z_1}^{CT} = m_{z_1}^{CT}(M, \alpha)$ - безмірний аеродинамічний коефіцієнт стабілізуючого моменту, який залежить від числа $M = \frac{V}{a}$ і кута атаки α ; L_K – довжина РС.

Знак „мінус” у формулі (2) показує, що стабілізуючий момент спрямовано так, щоб зменшити кут атаки α або кут нутації δ . Вектор стабілізуючого моменту перпендикулярний площині кута атаки α і при позитивному куті α спрямовано у бік негативної напіввісі z_1 , тобто повертає ЛА за ходом часової стрілки, зменшуючи кут атаки.

При малих кутах атаки коефіцієнт стабілізуючого моменту для даного числа M пропорційно куту атаки:

$$m_{z_1}^{CT} = m_{z_1}^{CT} \alpha \quad (3)$$

$$m_{z_1}^{\alpha} = \frac{\partial m_{z_1}^{CT}}{\partial \alpha} \quad \text{- градієнт (похідна) коефіцієнта стабілізуючого моменту по куту атаки, який залежить від числа } M \text{ і визначається дослідним шляхом.}$$

Коефіцієнт $m_{z_1}^{\alpha}$, а отже, і величина стабілізуючого моменту суттєво залежать від відстані h між центром тиску і центром мас опереного літальних апаратів, який називається плечем стабілізуючого моменту (рис. 1.3). На активній ділянці траєкторії плече h змінюється, так як по мірі вигорання палива зміниться положення центра мас ЛА, а по мірі збільшення швидкості польоту – положення центра тиску. Оскільки центр тиску залишається позаду центру мас, стабілізуючий момент прагне зменшити кут нутації, тобто оперений ЛА при цих умовах володіє статистичною стійкістю.

Тому при випадковому відхиленні вісі ЛА від напрямку вектора швидкості на кут нутації δ повна аеродинамічна сила створює не стабілізуючий, а перекидаючий момент відносно центру мас, який прагне збільшити кут нутації δ (рис. 1.4).

Вектор перекидаючого моменту перпендикулярний площині кута нутації δ і при позитивному куті δ спрямовано у бік позитивної вісі z_1 , тобто сам по собі є позитивним.

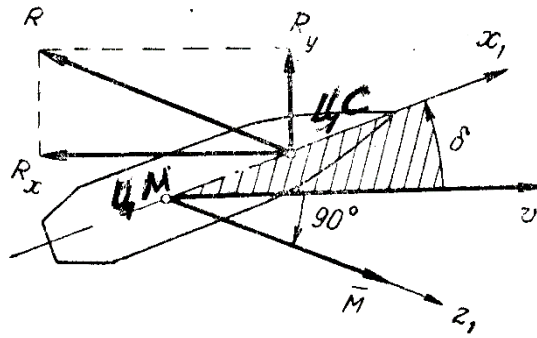


Рисунок 1.4 - Перекидаючий момент ЛА без оперення

Для протидії перекидаючого моменту і стабілізації поздовжньої вісі ЛА відносно вектора швидкості ЛА надається швидкий обертаючий рух навколо поздовжньої вісі.

Аеродинамічні погасаючі моменти – це моменти аеродинамічних сил, які виникають при наявності у літаючого апарату або ЛА кутових швидкостей обертання відносно пов'язаних висей. Аеродинамічні погасаючі моменти є наслідком опору повітря обертання літаючого апарату і спрямовані у бік, протилежний напрямку кутової швидкості обертання літаючого апарату, тобто зменшують її. Для некерованих ЛА розрізняють екваторіальний і полярний погасаючі моменти, які обумовлені відповідною кутовою швидкістю обертання відносно екваторіальних OZ_1 , OY_1 і полярною OX_1 , пов'язаних висей (рис. 1.1).

Виникнення погасаючих аеродинамічних моментів і їх величини залежать від зміни швидкостей обтікання ЛА повітрям у різних розрізах при наявності кутової швидкості обертання. При цьому екваторіальний погасаючий момент обумовлюється перерозподілом тиску повітря на поверхню ЛА, а полярний погасаючий момент – діями сил тертя і тиску або тільки силами тертя (у випадку застосування ЛА без оперення) спрямованими протилежно обертаючій швидкості точок поверхні корпусу ЛА.

1.3 Рух центра мас некерованих реактивних ЛА на активній ділянці траєкторії

Розглянемо поступовий рух або рух центру мас некерованого ЛА на активній ділянці траєкторії (АДТ) у вертикальній площині кидання при наступних припущеннях.[14]

1. ЛА являє собою матеріальну точку змінної маси, яка рухається під впливом сили тяги двигуна P , сили лобового опору R_x і сили ваги G . Це припущення означає, що не враховуються коливання ЛА (кут нутації $\delta = 0$), повздожжня вісь ЛА і вектор сили тяги співпадають з напрямком вектора швидкості ЛА.

2. Прискорення сили тяги постійні по величині та напрямку і співпадають з напрямком вертикалі в початковій точці траєкторії.

3. Поверхня Землі приймається за площину, яка співпадає з площиною горизонту у початковій точці траєкторії.

4. Обертання Землі не враховується.

5. Вітер відсутній.

При зроблених припущеннях ЛА здійснює тільки поздовжній поступальний рух, тобто його траєкторія буде розташована у вертикальній площині кидання.

Виходячи із зроблених припущень, складемо диференціальні рівняння руху центру мас ЛА (поступового руху ЛА) у проєкціях на дотичну і нормаль до траєкторії.[15]

У якості вихідної залежності візьмемо векторне рівняння поступового руху ЛА під впливом трьох сил: тяги P , лобового опору R_x і ваги G :

$$m\overline{\omega} = \overline{P} + \overline{R}_x + \overline{G}, \quad (4)$$

де $\overline{\omega}$ - вектор повного прискорення центру мас ЛА; m – маса ЛА у момент часу, що розглядається.

Векторне рівняння (4) виражає другий закон механіки: добуток маси РС на його лінійне прискорення (вектор) дорівнює сумі діючих на ЛА сил (векторів).

Спроектуємо векторне рівняння руху центру мас ЛА на дотичну τ і нормаль n до траєкторії (рис. 1.5). У рівнянні поступового руху ЛА в проекції на будь яку вісь ліворуч записується добуток маси ЛА на складову (проекцію) повного прискорення ЛА по даній вісі, а праворуч – сума проекцій на дану вісь усіх діючих на ЛА сил. [3]

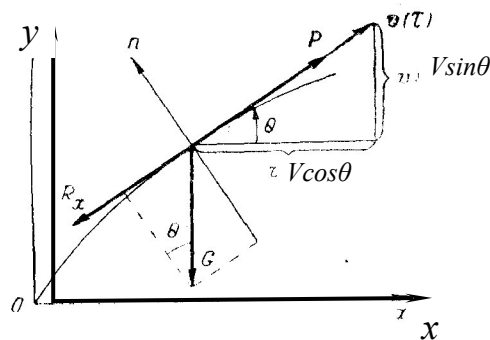


Рисунок 1.5 - Сили, які діють на літальний апарат

Складові (проекції) повного прискорення центру мас ЛА по дотичній w_τ і по нормалі w_n до траєкторії визначаються наступними вираженнями:

$$w_\tau = \partial V / \partial t; \quad w_n = \partial \theta / \partial t. \quad (5)$$

Дотичне прискорення w_τ - є похідна модуля вектора швидкості по часу, а нормальне прискорення w_n можна представити як результат повертання вектора швидкості у вертикальній площині з кутовою швидкістю $\partial \theta / \partial t$, обумовленою безперервною зміною кута θ нахилу дотичної до траєкторії τ (зниженням дотичної до траєкторії τ під впливом сили ваги).

Враховуючи проекції сил \bar{P} , \bar{R}_x , \bar{G} на вісі τ і n (рис. 1.5)

$$\begin{aligned} P_\tau &= P; & R_{x\tau} &= -R_x; & G_\tau &= -G \sin \theta; \\ P_n &= 0; & R_{xn} &= 0; & G_n &= -G \cos \theta, \end{aligned}$$

отримуємо рівняння поступового руху ЛА у проекціях на дотичну і нормаль до траєкторії

$$\begin{cases} m \frac{\partial V}{\partial t} = P - R_x - G \sin\theta \\ mV \frac{\partial \theta}{\partial t} = -G \cos\theta \end{cases} \quad (6)$$

До рівняння поступового руху ЛА необхідно додати рівняння зв'язку між координатами x, y земної стартової системи координат OXY і проекціями вектора поступової швидкості ЛА (рис. 1.5);

$$\frac{\partial y}{\partial t} = V \sin\theta; \quad \frac{\partial x}{\partial t} = V \cos\theta \quad (7)$$

Ці рівняння зв'язку показують, що похідна будь якої координати по часу, тобто швидкість переміщення ЛА по напрямку даної вісі, дорівнює проекції швидкості центру мас ЛА на дану координатну вісь.

Поділити ліву і праву частини першого рівняння (6) на масу ЛА m , а другого – на добуток маси і швидкості mV і враховуючи, що $G/m = g$, запишемо систему диференціальних рівнянь руху центру мас некерованого ЛА на активній ділянці траєкторії у наступному вигляді:

$$\begin{cases} \frac{\partial V}{\partial t} = \frac{1}{m}(P - R_x) - g \sin\theta \\ \frac{\partial \theta}{\partial t} = -\frac{g \cos\theta}{V} \\ \frac{\partial y}{\partial t} = V \sin\theta \\ \frac{\partial x}{\partial t} = V \cos\theta \end{cases} \quad (8)$$

У загальному вигляді система чотирьох диференціальних рівнянь (8), яка утримує в собі чотири шукаючі функції часу – елементи траєкторії V , θ , y , і x , вирішується методом чисельного інтегрування. Для інтегрування отриманої системи диференціальних рівнянь руху центра мас ЛА на активній ділянці траєкторії маємо наступні початкові умови в момент сходу ЛА з направляючої при $t = 0$:

$$V = V_0; \quad \theta = \theta_0; \quad y = y_0; \quad x = x_0 = 0.$$

У результаті чисельного інтегрування системи (8) диференціальних рівнянь руху центру мас ЛА для послідовних значень часу польоту t будуть визначені чисельні значення усіх елементів поточних точок активної ділянки траєкторії. [4]

1.4 Рух центра мас некерованих літальних апаратів на пасивній ділянці траєкторії

Поступовий рух, тобто рух центру мас некерованого ЛА на пасивній ділянці траєкторії, розглядається у вигляді руху матеріальної точки с постійною масою під впливом двох сил: сили лобового опору повітря і сили ваги при відсутності вітру і обертання Землі. [5] При цьому поверхня Землі приймається плоскою, а напрямок прискорення сили ваги – постійним (по вертикалі початкової точки траєкторії, рис. 1.6). При цих припущеннях траєкторія ЛА буде в вертикальній площині кидання. Рівняння руху центру мас ЛА на пасивній ділянці траєкторії отримуємо, як частковий випадок рівнянь руху ЛА на активній ділянці траєкторії (6) або (8), виключаючи із них силу тяги двигуна P , тому що на пасивній ділянці траєкторії двигун не працює (у реактивного ЛА відсутній).

Відмітимо, що у системі рівнянь (8) сила тяги двигуна входить тільки до рівняння руху в проекції на дотичну до траєкторії, а рівняння руху в проекції на нормаль до траєкторії і рівняння зв'язку між похідними координат y і x по часу і

проекціями вектора швидкості на координатні вісі y і x зберігають незмінний вигляд як для активної, так і для пасивної ділянки траєкторії (див. рис. 1.5 і 1.6).

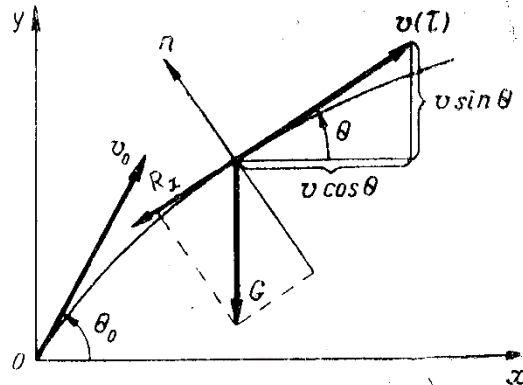


Рисунок 1.6 - Сили, які діють на ЛА на пасивній ділянці траєкторії

Вважаючи у першому рівнянні системи (8) $P = 0$, отримуємо систему диференціальних рівнянь руху центру мас ЛА на пасивній ділянці траєкторії (в проекції на дотичну τ і нормаль n до траєкторії):

$$\begin{cases} \frac{\partial V}{\partial t} = -cH(y) \cdot F(V) - g \sin \theta \\ \frac{\partial \theta}{\partial t} = -\frac{g \cos \theta}{V} \\ \frac{\partial y}{\partial t} = V \sin \theta \\ \frac{\partial x}{\partial t} = V \cos \theta \end{cases} \quad (9)$$

У правій частині першого рівняння системи (9) підставлено значення прискорення сили лобового опору повітря згідно залежності

$$J = cH(y) \cdot F(V),$$

де $c = id^2/q \cdot 10^3$ – балістичний коефіцієнт ЛА, отримуємо

$$R_x/m = J = cH(y) \cdot F(V).$$

У загальному вигляді система чотирьох рівнянь (9), яка містить чотири шукані функції часу V , θ , y , x , вирішується методом чисельного інтегрування аналогічно системі рівнянь (8). [6]

У випадку розрахунку пасивної ділянки траєкторії некерованого РС необхідно знати початкові умови, які визначаються елементами точки K – кінця активної ділянки траєкторії при $t = t_k$ (рис. 1.7): $V = V_k$; $\theta = \theta_k$; $y = y_k$; $x = x_k$.

Отже, при припущеннях, які покладені до системи рівнянь (9) поступового руху ЛА, траєкторія ЛА і всі її елементи повністю визначаються трьома балістичними параметрами: кутом кидання θ_0 , початковою швидкістю V_0 (початкові умови) і балістичним коефіцієнтом c , який входить до рівняння руху.

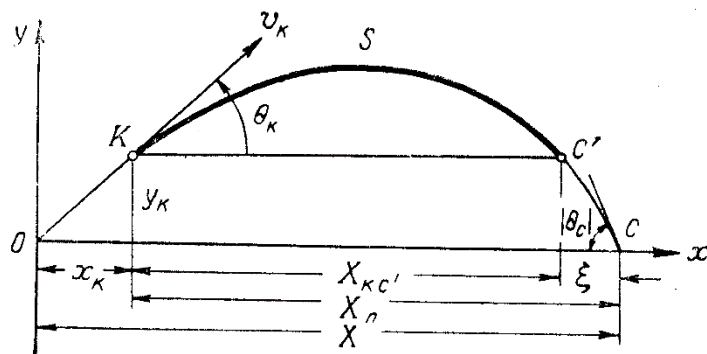


Рисунок 1.7 - До розрахунку дальності і височини пасивної ділянки траєкторії методом двох дуг

Пасивна ділянка траєкторії ЛА у цілому і всі її елементи повністю визначаються чотирма балістичними параметрами в кінці активної ділянки траєкторії: кутом θ_k , швидкістю V_k , балістичним коефіцієнтом c_k , і ординатою y_k , (від абсциси кінця активної ділянки траєкторії x_k пасивна ділянка траєкторії не залежить).

Для ЛА з дальністю польоту в декілька десятків кілометрів при розрахунку траєкторій (пасивних ділянок) необхідно враховувати кривизну поверхні Землі і зміни напрямку прискорення сили ваги вздовж траєкторії.

1.5 Постановка завдання

Завданнями роботи є:

- Запропонувати математичну модель та розробити програму руху некерованих літальних апаратів в збуреному середовищі;
- Розробити алгоритм і програму розрахунку бюлетня «Метеосередній» по даним наземних вимірювань,
- Розробити алгоритм і реалізувати програму для знаходження топографічних даних;
- Розробити алгоритм для знаходження вирахованих установок
- За допомогою знайдених раніше топографічних даних та даних бюлетня «Метеосередній» програма повинна дозволяти розраховувати вираховані установки по цілям (об'єктам) ураження для 120-мм мінометом або 82-мм мінометом з високою точністю та в найкоротші терміни.

2 ВИБІР МЕТОДУ РІШЕННЯ

2.1 Основні переваги та недоліки мови програмування C++

Мова програмування C++ є універсальною, тому що тут допускається не єдина парадигма. Наприклад, C++ підтримує процедурне та об'єктно-орієнтоване програмування, а також абстракцію даних. C++ створив американський розробник Б'ярн Страуструп ще у 70-х роках минулого століття. Спершу вона мала ім'я «С з класами», а згодом автор змінив назву на сучасну. Взагалі, C++ розроблена на основі мови програмування Сі.

Щорічно професії пов'язані з розробкою програмного забезпечення стають все більш популярними. Програмісти-початківці дедалі частіше стоять перед вибором, яку мову програмування слід опанувати першочергово. Більшість українських вишів, так само як і досвідчені програмісти, рекомендують саме C++, як першу мову програмування.

За даними сервісу [TIOBE Index](#), на травень 2020 року мова C++ посідає четверте місце у рейтингу [12], див. рис. 2.1.

May 2020	May 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	17.07%	+2.82%
2	1	▼	Java	16.28%	+0.28%
3	4	▲	Python	9.12%	+1.29%
4	3	▼	C++	6.13%	-1.97%
5	6	▲	C#	4.29%	+0.30%
6	5	▼	Visual Basic	4.18%	-1.01%
7	7		JavaScript	2.68%	-0.01%
8	9	▲	PHP	2.49%	-0.00%
9	8	▼	SQL	2.09%	-0.47%
10	21	▲	R	1.85%	+0.90%

Рисунок 2.1 - Місце мови програмування в рейтингу TIOBE Index на травень 2020 року

Згідно із цими даними, C++ на сьогоднішній день поступається лише мовам програмування Java, C та Python. Що стосується історії, то за даними

того ж сервісу, мова програмування C++ лідирувала у 1995 році, а з початку 2000-х її рейтинг коливався з 2 до 4 місця, див. рис. 2.2.

Programming Language	2020	2015	2010	2005	2000	1995	1990	1985
Java	1	2	1	2	3	-	-	-
C	2	1	2	1	1	2	1	1
Python	3	7	6	7	23	22	-	-
C++	4	4	4	3	2	1	3	9
C#	5	5	5	8	8	-	-	-
JavaScript	6	8	8	9	5	-	-	-
PHP	7	6	3	4	25	-	-	-
SQL	8	-	-	-	-	-	-	-
Swift	9	95	-	-	-	-	-	-
Ruby	10	11	9	23	31	-	-	-

Рисунок 2.2 - Історія рейтингу мов програмування за даними сервісу TIOBE Index

Динаміка зміни рейтингу мови програмування C++ у порівнянні з іншими графічно представлена на рис. 2.3.

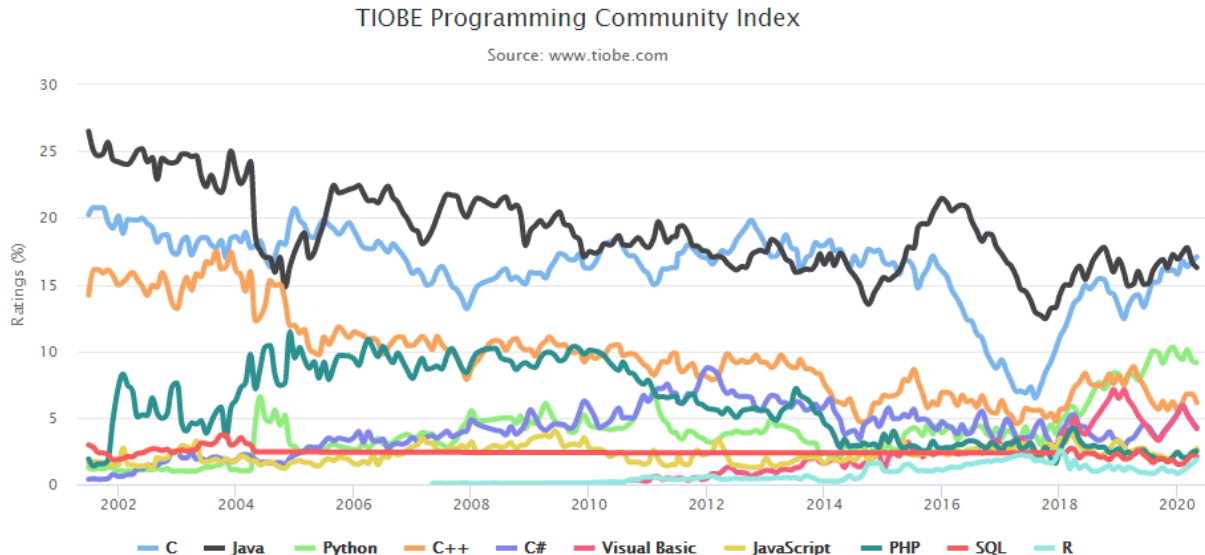


Рисунок 2.3 - Динаміка зміни рейтингу основних мов програмування за даними TIOBE Index

Перш ніж розглянути основні переваги та недоліки C++, варто ознайомитися з правилами, які використовував Б'ярн Страуструп при проектуванні даної мови програмування [13].

Отже, згідно з Б. Страуструпом, мова програмування C++:

1. Створена як універсальна мова зі статичними типами даних, ефективністю та сумісністю з мовою програмування Сі.
2. Розроблена так, аби безпосередньо та всебічно підтримувати багато стилів програмування, наприклад процедурне програмування, об'єктно-орієнтоване програмування, абстракцію даних, узагальнене програмування.
3. Побудована так, щоб надати програмісту свободу вибору, незважаючи на те, що цей вибір може бути й помилковим.
4. Створена таким чином, щоб максимально зберегти сумісність з Сі, роблячи тим самим легкий перехід з однієї мови програмування на іншу.
5. Не має особливостей, що роблять мову програмування залежною від платформи, тобто С++ є універсальною.
6. Не створює жодних надмірних навантажень на програму, якщо не використовуються всі можливості мови програмування.
7. Розроблена таким чином, щоб не вимагати надто ускладненого середовища програмування.

Тобто, перше правило вказує на одну з ключових переваг мови програмування С++, а саме те, що змінна не може міняти свій тип даних. Це означає, що при автодоповненні інтегроване середовище розробки буде пропонувати виключно ті методи, які можна застосувати до певного типу даних. Статична типізація також значно зменшує кількість помилок, тому що більшість з них виключаються вже на стадії компілювання додатку. Але в той же час застосування статичної типізації створює деякі труднощі з реалізацією алгоритмів та читанням кодів.

Ще одна суттєва перевага мови програмування С++ - це її універсальність. Розробник програмного забезпечення, який опанував цю мову, може створювати прості десктопні додатки для системи Windows , використовуючи Windows

Forms. Більш складні програми можуть виконуватися для систем Windows та Linux. Також можна робити ігрові програми для Windows, Android та ін.

Характеристики, що викладені у другому правилі, теж можна віднести до плюсів C++. Так, програміст-початківець може спершу опанувати процедурне програмування, а потім перейти до об'єктно-орієнтованого.

В третьому правилі мова йде про свободу вибору, але її важко охарактеризувати, як суто позитивну, чи негативну сторону C++. Справа в тому, що на перших етапах знайомства з цією мовою програмування такі помилки, як несумісність методів з типом змінних, спроби привести одну й ту ж змінну до різних типів даних є неможливими, бо виключаються ще на стадії компілювання. Тобто, про так звану свободу вибору говорити не доводиться, принаймні, якщо мова йде про початківців.

Четверте правило повністю вказує на переваги C++, адже сумісність с Cі, з набуттям досвіду програмування користувачами, відіграє значну роль. Слід відзначити зокрема такі функції, що підтримуються обома мовами програмування як то введення/виведення на екран інформації `scanf()` та `printf()`, бібліотека `stdio.h` тощо. Як наслідок, багато програмістів називають C++ як «Ц з класами».

У п'ятому правилі йдеться про кросплатформенність мови програмування C++. І це також її вагомий плюс. Існує багато компіляторів та IDE що мають версії для платформ Windows, Linux та MacOS. Тобто, розробник може створювати додатки, які працюватимуть на різних платформах, використовуючи «рідне» інтегроване середовище розробки та компілятор. Це завдання ще більше спростив QT-framework, що був створений ще у 1996 році.

Шосте правило говорить про ефективність програмного забезпечення, створеного за допомогою C++. При цьому програма не буде перевантаженою, якщо увесь її функціонал не задіяний. Так вона споживатиме набагато менше ресурсів системи та відповідно працюватиме швидше. Це додатково підтверджується тестуванням IBM. Були створені програми, які знаходять числа Фібоначчі. Програма створена на базі C++ склалася за 0,014 секунди,

поступившись тільки Сі на 0,01 секунди. Тобто мова програмування С++ швидко компілює програми, завдяки статичній типізації. Адже компілятор не витрачає час на визначення типу змінної, як це відбувається в JavaScript, PHP або Python. До речі, тоді алгоритм Фібоначчі виконався повністю за 0,69 секунди на С++, 1,11 секунди на JavaScript та 1,31 секунди на Python та 2 секунди на PHP.

Стосовно того, що С++ не вимагає ускладненого середовища програмування, про що йде мова у сьомому правилі Б. Страуструпа, це важко назвати зараз суттєвою перевагою. Тому що інші мови програмування також пропонують зручні інтегровані середовища розробки. Крім того, С++ саме за цим показником значно поступається, наприклад, JavaScript. Тобто для розробки програм на JavaScript потрібен мінімальний інструментарій: достатньо блокнота та веб-оглядача, які зазвичай є вбудованими в систему. А для С++ обов'язково потрібен також компілятор, який необхідно додатково завантажувати та налаштовувати.

Ще одним суттєвим недоліком С++ можна назвати високий поріг входження. В.М. Бондарев у своїй книзі «Основи програмування» образно охарактеризував С++ так: «Програми на С++, обганяють програми, створені на інших мовах, як гоночний автомобіль обганяє звичайні. Але вчитися програмувати на них – все одно, що вчитися основам водіння, сівши за кермо гоночного авто – дорого та довго».

Отже, враховуючи все вище перелічене можна зробити висновок, що С++ є низькорівневою кросплатформенною мовою програмування, яка тривалий час залишається однією з найбільш популярних. С++ буде актуальною ще довго завдяки своїй універсальності та ефективності. З іншого боку, у порівнянні, наприклад, з JavaScript, початківцям вона може здатися складною через статичну типізацію, яка обмежує роботу зі змінними. Але, тим не менш, ця мова програмування рекомендується саме молодим програмістам, тому що, С++ дає змогу опанувати С-подібний синтаксис, а потім і парадигми об'єктно-

орієнтованого програмування, яке в свою чергу дасть змогу в подальшому реалізовувати більш складні проекти.

2.2 Вибір середовища розробки

Borland C++ Builder 6 – це середовище для створення програм мовою C++. Воно було розроблене у 2002 році компанією Borland, яка з 2009 року є частиною великого ІТ-розробника Micro Focus. Загалом існує 19 версій середовища Borland C++ Builder, перша з яких вийшла ще 1997 році, див. Таблицю 2.1.

Таблиця 2.1 Версії Borland C++ Builder 6 за роками

Рік	Версія
1997	1
1998	3
1999	4 (випущений як Inprise)
2000	5
2002	6
2003	X
2005	2006
2007	2007 (v11)
Вересень 2008	2009 (v12)
25 серпня 2009	2010 (v14)
2010	XE
2011	XE2
2012	XE3
Квітень 2013	XE4
Вересень 2013	XE5
Квітень 2014	XE6
Вересень 2014	XE7
Квітень 2015	XE8
31 серпня 2015 р.	10 Seattle

У цьому середовищі програми створюються на основі сучасного методу, що називається об'єктно-орієнтованим програмуванням. Тут можна здійснювати,

так зване, візуальне програмування, тобто створювати програми, які під час виконання взаємодіють з користувачем завдяки багатовіконному графічному інтерфейсу. Такий інтерфейс є дуже зручним тому що фактично програми в момент виконання можуть на екрані мати вигляд графічних об'єктів. Серед них:

- Кнопки, на які можна натискати мишею, після чого відбуваються деякі, прив'язані до них дії;
- Поля введення-виводу даних;
- Списки, з яких можна обирати дані для подальших розрахунків у програмі;
- Різноманітні меню, що дозволяють обирати та виконувати певні дії;
- Елементи, що контролюють, стан об'єктів, наприклад увімкнений - вимкнений, обраний – не обраний тощо;
- Елементи, які дозволяють обирати дати з календаря;
- Елементи, що дозволяють слідкувати за ходом деяких процесів по мірі їх виконання;
- Елементи, які забезпечують стандартний вибір файлів, шрифтів, кольорів, налаштувань принтерів і т.д.
- Елементи, які дозволяють додавати в програму, що розроблюється, інші програми-об'єкти, наприклад Word, Excel, анімаційні файли, діаграми, кліпи мультимедіа тощо [8].

Крім того, на екрані можуть з'являтися різні зображення, ієрархічні дерева, що допомагають краще зрозуміти хід виконання програми та керувати ним. Середовище Borland C++ Builder дозволяє працювати як з простими локальними та віддаленими базами даних, так і з багатоланковими розподіленими базами даних. Крім того, воно може встановлювати взаємодію програми, що створюється, з мережею Інтернет [9].

Такі можливості надають переваги Borland C++ Builder перед іншими середовищами. Так, за потужністю та зручністю використання з цим середовищем може конкурувати наразі тільки Borland Delphi. Але останній використовує алгоритмічну мову об'єктний Паскаль, чим все-таки поступається Borland C++ Builder, який використовує C++ - більш зручну та прогресивну мову програмування [10].

Всі програми, які створюються в Borland C++ Builder називають додатками, тому що вони додаються до самого середовища та оформлюються у вигляді спеціальних структур, які називаються проектами. Жодна програма не може існувати поза структурою – проектом. Дії по управлінню проектами здійснює спеціальний програмний комплекс, який називають менеджером проектів.

Загалом Borland C++ Builder являє собою додаток з однодокументним інтерфейсом (SDI). У головному вікні розташовується палітра компонентів та панель інструментів. Крім того, за замовчуванням, після завантаження середовища відразу з'являється вікно інспектора об'єктів та форма додатку, що буде створюватися.

Основою для розробки додатку в Borland C++ Builder є форма. Вона є головним контейнером, в якому будуть розміщуватися компоненти самого середовища. А вже потім за допомогою цих компонентів буде реалізуватися якийсь конкретний алгоритм певної задачі. Все побудовано так, що треба спершу відкрити пусту форму при створенні нового проекту з нуля, або коли треба додати нову пусту форму до тих, що вже створені, якщо цього вимагає алгоритм [11].

Вже потім, у відповідності до алгоритму, сюди будь поміщені необхідні компоненти з палітри середовища. В свою чергу серед компонентів в Borland C++ Builder можна умовно виокремити видимі та невидимі. По іншому їх називають візуальними та невізуальними відповідно. Видимі – це ті, які можна побачити на самому початку користування, наприклад кнопки, поля вводу,

списки і т.д. Але деякі є невидимими під час виконання, хоча вони відіграють суттєву роль, наприклад, надають доступ до даних.

Для включення до форми варто обрати певний компонент, натиснути ліву клавішу миші і тоді, коли він з'явиться, його можна буде переміщувати, зменшувати, збільшувати і т.д.

Всі компоненти цього середовища мають три головних показники – властивості, методи, події [7]. Якщо додати до форми певний компонент, відразу на екрані відобразиться вікно інспектора об'єктів, який покаже події та властивості, які є допустимими до цього компонента, див. рис. 2.4.

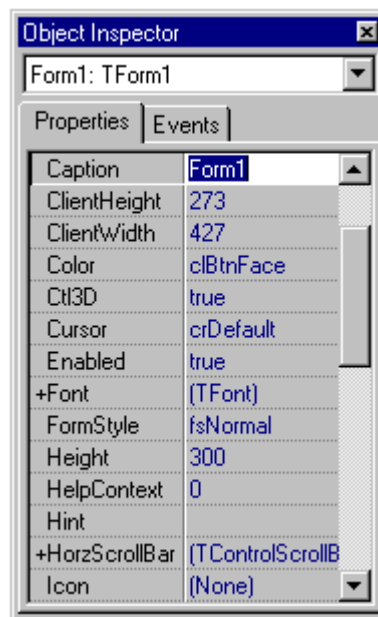


Рисунок 2.4 - Вигляд інспектора об'єктів.

Інспектор об'єктів, разом з іншими засобами, є доступним на всіх етапах роботи над додатком. Це робить можливим внесення змін у будь-який час.

Щоб розробити додаток в середовищі Borland C++ Builder треба спершу створити проект, файли якого вже будуть мати автоматично згенерований текст, який буде частиною програми після компіляції та підготовки до виконання. Зазвичай його назва - Project1 .cpp, див. рис. 2.5.

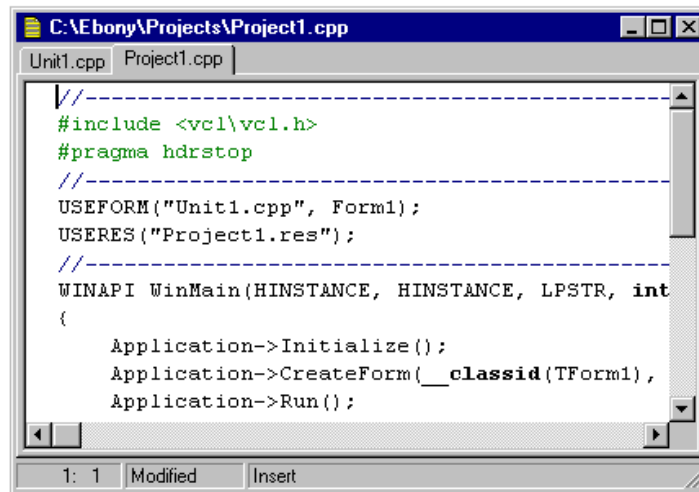


Рисунок 2.5 - Вигляд файлу нового проекту

При додаванні нової форми, відбувається автоматичне оновлення файлу проекту. Для прикріплення 1-ї, або декількох форм до проекту, треба в меню Файл обрати нову форму File/New, див. рис. 2.6.

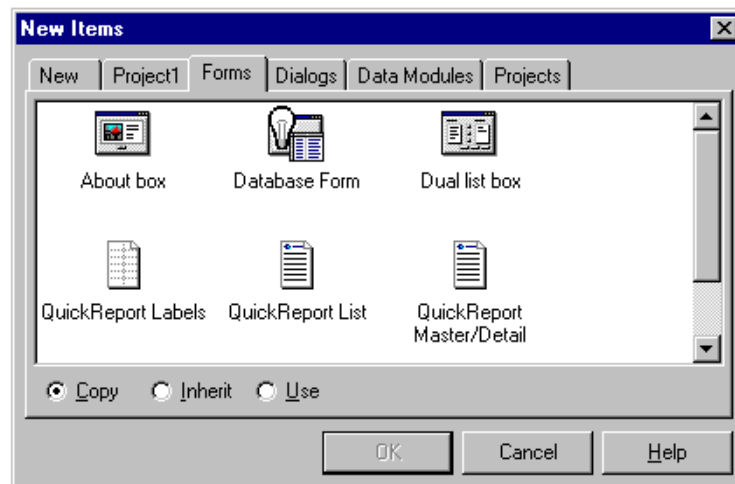


Рисунок 2.6 - Нові форми (шаблони)

Для простого компілювання поточного проекту досить буде обрати розділ Compile, а щоб допрацювати файл для виконання слід натиснути Run. Якщо виникає помилка, то середовище Borland C++ Builder зупиняє виконання програми, натомість з'являється редактор коду, де курсор вже знаходиться у місці, де є проблема. Це дуже корисна функція, особливо для програмістів-початківців. Щоб відредагувати код з помилкою, треба спершу перезапустити додаток, так буде витратитися менше ресурсів системи.

Отже, Borland C++ Builder – це зручне, зрозуміле, надійне та продуктивне середовище для програмування. Крім того, йому властива висока швидкодія при компіляції додатків для сучасних операційних систем. Програми, розроблені у цьому середовищі, є добре оптимізованими за швидкістю виконання та витратами пам'яті. Саме тому він був обраний для реалізації практичної частини даної роботи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Математична модель

За допомогою даних математичних алгоритмів програма, може точно зробити підготовку для негайної артилерійської відповіді. Всі зазначені формули в даній секції, були програмно реалізовані та повністю інтегровані в робочу програму для визначення поправочних графів, таким чином ці алгоритми розширили функціонал існуючого рішення. Дані в програмі зчитуються та відображають за допомогою таблиць.

3.1.1 Визначення топоданих по цілі аналітично

Дано: $X_{Ц}, Z_{Ц}, h_{Ц}, X_{б}, Z_{б}, h_{б}, \alpha_{осі}$

Визначити: $D_T^Ц, \delta_T^Ц, \alpha^Ц, \Delta h_{Ц}, \epsilon_{Ц}$

$$а) \Delta X = X_{Ц} - X_{б}$$

$$б) \Delta Z = Z_{Ц} - Z_{б}$$

$$в) \Delta h = h_{Ц} - h_{б}$$

$$г) H = \arctg \left| \frac{\Delta Z}{\Delta X} \right|$$

г) Знаходимо $\alpha^Ц$ в полярних координатах:

$$I \text{ чверть: якщо } \Delta Z \geq 0, \Delta X \geq 0, \text{ то } \alpha^Ц = \frac{H * 30}{\pi}.$$

Якщо $(\Delta Z > 0, \Delta X = 0)$, то $\alpha^Ц = 15.0$.

$$II \text{ чверть: якщо } \Delta Z \geq 0, \Delta X < 0, \text{ то } \alpha^Ц = \frac{(\pi - H) * 30}{\pi}.$$

$$III \text{ чверть: якщо } \Delta Z \leq 0, \Delta X < 0, \text{ то } \alpha^Ц = \frac{(\pi + H) * 30}{\pi}.$$

Якщо $(\Delta Z < 0, \Delta X = 0)$, то $\alpha^Ц = 45.0$.

$$IV \text{ чверть: якщо } \Delta Z \leq 0, \Delta X > 0, \text{ то } \alpha^Ц = \frac{(2\pi - H) * 30}{\pi}.$$

$$д) D_T^Ц = \sqrt{\Delta Z^2 + \Delta X^2}$$

$$е) \delta_T^Ц = \alpha^Ц - \alpha_{осі}$$

Якщо $|\delta_T^{\Pi'}| \leq 30.0$, то $\delta_T^{\Pi} = \delta_T^{\Pi'}$

Якщо $|\delta_T^{\Pi'}| > 30.0$ і $\delta_T^{\Pi'} < 0$ то, $\delta_T^{\Pi} = \delta_T^{\Pi'} + 60.0$

Якщо $|\delta_T^{\Pi'}| > 30.0$ і $\delta_T^{\Pi'} > 0$ то, $\delta_T^{\Pi} = \delta_T^{\Pi'} - 60.0$.

$$e) \varepsilon_{\Pi} = \frac{\Delta h_{\Pi} * 0.95}{0.1 * D_T^{\Pi}}$$

3.1.2. Складання наближеного бюлетеня “Метеонаближений” за даними наземних вимірів

1. Проводять наземні виміри та записують в таблицю М, див. табл. 3.1.

$H_0 \rightarrow$ п. 1 $t_0 \rightarrow$ п. 2 $\alpha_{w0} \rightarrow$ п. 3 $W_0 \rightarrow$ п. 4 $h_{mc} \rightarrow$ п. 5

№ з/п	Параметри	Значення
1	H_0 , мм. рт. ст.	736
2	t_0 , °C	3
3	α_{w0} , под. кут.	17,00
4	W_0 , м/с	9
5	h_{mc}	210

Таблиця 3.1. Таблиця М

2. Визначають відхилення:

- $\Delta H_0 = H_0 - 750$ та записують в таблицю МД п.3, див. табл. 3.2.
- $\Delta H_B = \Delta H_0 + (h_{mc} - h_{\sigma})$, де h_{σ} – висота вогневої позиції міномета, що поступає в комп’ютер з навігаційної апаратури

3. За величиною t_0 з табл. 3.2 визначають ΔT_V і записують в табл. 2 п. 5.

t_0 , °C	нижче 0	0	5	10	15	20	25	30	35	40	45	50
ΔT_V , °C	0	+0, 3	+0, 5	+0,6	+0,9	+1,3	+1,8	+2,4	+3,3	+4,4	+5,8	+7,4

Таблиця 3.2. Таблиця для визначення віртуальних поправок

1	H_0	736	4	t_0	+3	7	τ_0	+3,4	10	α_{W_0}	17,00
2	$-H_{N_0}$	750	5	ΔT_V	+0,4	8	$-\tau_{N_0}$	15,9			
3	ΔH_0	-14	6	τ_0	+3,4	9	$\Delta\tau_{0МП}$	-12,5	11	W_0	9

Таблиця 3.3. Таблиця МД

4. Визначають $\tau_0 = t_0 + \Delta T_V$ і записують в табл. 3.3 п. 6, 7.
5. Визначають $\Delta\tau_{0МП} = \tau_0 - 15,9$ і записують в табл. 3.3 п. 9.
6. За стандартними висотами $Y_{\text{бюл}}$ та $\Delta\tau_{0МП}$ з табл. 3.4 методом інтерполяції визначаємо $\Delta\tau$ і записуємо в табл. 3.5 для кожної стандартної висоти Y_M , стовпчик 2.

Y_M	$\Delta\tau_{0МП}, ^\circ\text{C}$													
	1	2	3	4	5	6	7	8	9	10	20	30	40	50
200	-1	-2	-3	-4	-5	-6	-7	-8	-8	-9	-20	-29	-39	-49
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
400	-1	-2	-3	-4	-5	-6	-6	-7	-8	-9	-19	-29	-38	-48
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
800	-1	-2	-3	-4	-5	-6	-6	-7	-7	-8	-18	-28	-37	-46
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
1200	-1	-2	-3	-4	-4	-5	-5	-6	-7	-8	-17	-26	-35	-44
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
1600	-1	-2	-3	-3	-4	-4	-5	-6	-7	-8	-17	-25	-34	-42
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
2000	-1	-2	-3	-3	-4	-4	-5	-6	-6	-7	-16	-24	-32	-40
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
2400	-1	-2	-2	-3	-4	-4	-5	-5	-6	-7	-15	-23	-31	-38
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
3000	-1	-2	-2	-3	-4	-4	-4	-5	-5	-6	-15	-22	-30	-37
	1	2	3	4	5	6	7	8	9	10	20	30	-	-
4000	-1	-2	-2	-3	-4	-4	-4	-4	-5	-6	-14	-20	-27	-34
	1	2	3	4	5	6	7	8	9	10	20	30	-	-

Таблиця 3.4. Визначення відхилення температури Δt_Y для стандартних висот в залежності від $\Delta t_{0мп}$

№ з/П	1	2	3	4	5	6
	$Y_{M(бюл)}$, м	$\Delta t_Y, ^\circ C$ 3 табл.4	$\Delta \alpha_{wY}$, п.к	α_{wY} , п.к. $\alpha_{w0} + \Delta \alpha_{wY}$	W_Y , м/с 3 табл. 3	Наближений бюлетень 22 “Метео 11 наближений – 11120-0210-51463 -02-621814 -04-621917 -08-612018 -12-612019 -16-612120 -20-602120 -24-592121 -30-582221 -40-582222
12	0			17,00		
13	200	-12	1-00	18,00	14	
14	400	-12	2-00	19,00	17	
15	800	-11	3-00	20,00	18	
16	1200	-11	3-00	20,00	19	
17	1600	-10	4-00	21,00	20	
18	2000	-10	4-00	21,00	20	
19	2400	-9	4-00	21,00	21	
20	3000	-8	5-00	22,00	21	
21	4000	-8	5-00	22,00	22	

Таблиця 3.5. Наближений бюлетень

7. Визначаємо дирекційний кут вітру α_{wY} для кожної стандартної висоти $Y_{бюл}$ та записують у табл. 3.5 стовпчик 4.

$$\alpha_{wY} = \alpha_{w0} + \Delta \alpha_{wY}$$

де $\Delta \alpha_{wY}$ – беруть з табл. 3.5 стовпчик 3

α_{w0} – табл. 3.3 п. 10.

8. Визначають W_Y . для чого, з табл. 3.6 по стандартним висотам $Y_{бюл}$ та W_0 (W_0 - табл. 3.1 п.4) визначають швидкість вітру на стандартних висотах W_Y та записують в табл. 3.5 стовпчик 5.

Y, м	W ₀ , м/с													
	2	3	4	5	6	7	8	9	10	11	12	13	14	15
200	3	4	6	8	9	10	12	14	15	16	18	20	21	22
400	4	5	7	10	11	12	14	17	18	20	22	23	25	27
800	4	5	8	10	11	13	15	18	19	21	23	25	27	28
1200	4	5	8	11	12	13	16	19	20	22	24	26	28	30
1600	4	6	8	11	13	14	17	20	21	23	25	27	29	32
2000	4	6	9	11	13	14	17	20	21	24	26	28	30	32
2400	4	6	9	12	14	15	18	21	22	25	27	29	32	34
3000	5	6	9	12	14	15	18	21	23	25	28	30	32	36
4000	5	6	10	12	14	16	19	22	24	26	29	32	34	36

Таблиця 3.6. Визначення швидкості середнього вітру W_y в залежності від швидкості наземного вітру W_0

9. Визначення складових вітру: W_x , W_z

Визначаємо кут вітру A_w по залежності

$A_w = \alpha_{\Pi} - \alpha_{wy}$, якщо $\alpha_{\Pi} > \alpha_{wy}$. (полярні координати)

якщо $\alpha_{\Pi} < \alpha_{wy}$ необхідно до $\alpha_{\Pi} + 60.0$

$$A_w(\text{град}) = \frac{A_{w(\text{п.к.})} * 3.6}{60}$$

11. Обчислюють W_x , W_z :

$$W_x = -W_y * \text{Cos}(A_w)$$

$$W_z = W_y * \text{Sin}(A_w)$$

3.2 Інформаційна модель

Для представлення програмної реалізації роботи була створена інформаційна модель системи, яка зображена на рисунку 3.1.

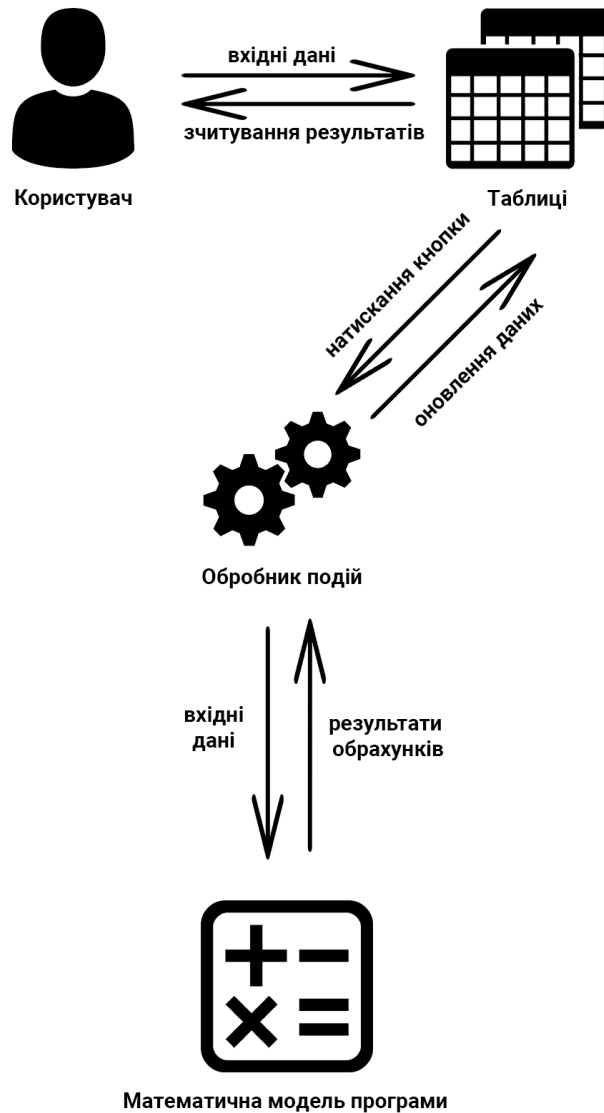


Рисунок 3.1 - Інформаційна модель системи

Данну модель можна описати наступними кроками:

1. Користувач вводить дані в спеціально заготовлені поля таблиць програми.
2. Користувач натискає на кнопку «Розрахувати» для запуску математичних алгоритмів.

3. Відбувається визначення топографічних даних по цілі, складання наближеного бюлетеня «Метеосередній» та обрахування інших даних
4. Результати обрахунків виводяться у спеціальні таблиці.

Розробку додатку можна умовно поділити на два етапи: створення інтерфейсу (завдяки Borland C++ Builder 6 це зробити дуже просто, досить лише вибрати потрібний елемент та перетягнути його до програми, таким чином можна додавати поля, таблиці, підписи, кнопки та все що необхідно), розробка математичної моделі та її подальша реалізація. Розроблений інтерфейс програми можна переглянути на рис. 3.2.

WarZone Shooting Tables - Mass Center

Подготовка | Исходные данные | Результаты расчётов | Помощь

Таблиця БП

Гармат	6	5	4	3 осн.	2	1
Xб,м				37235		
Zб,м				94830		
hб,м				180		
Alpha(оси)				52,00		
Зарад				Дальноб.		

Таблиця М

№ з/п	Параметри	Значення
1	Но, мм. рт. ст.	783
2	to, С	-24
3	Alpha wo, под. кгт.	1,00
4	Wo, м/с	5
5	hмс, м	200

Таблиця МН

delta Hа	delta tau_y	Wx	Wz
35	-32	-6	-9

Таблиця В

Пер	Дце	Xр	Zр

Таблиця Б

Параметр	DeltaVo	Ta	Delta q
Значення	-0,7	-16	3

Таблиця ДЦ

№ з/п	Параметри	№	№	№	№	№	№	№	№
1	Xц, м	40150							
2	Yц=Zц, м	93200							
3	hц, м	190							
4	Дтц, м	3339							
5	DeltaTц, м	3,13							
6	Alpha ц, м	55,13							
7	delta hц, м	10							
8	Yбюл	1800							

Таблиця МД

1	Но	783	4	to	-24	7	tauo	-24	10	Alpha wo	1,00
2	Hно	750	5	+deltaTv	0	8	-tauNo	15,9	11	Wo	5
3	deltaHo	33	6	tauo	-24	9	delta tauomт	-40			

Таблиця 5

№	1	2	3	4	5	6
	Yм(бюл, м	delta_tau_y	delta_alpha	alpha_vuy, п.	Wу, м/с	Наближення
12	0	-40		100	5	
13	200	-39	100	200	8	
14	400	-38	200	300	10	
15	800	-37	300	400	10	
16	1200	-35	300	400	11	
17	1600	-34	400	500	11	
18	2000	-32	400	500	11	
19	2400	-31	400	500	12	
20	3000	-30	500	600	12	
21	4000	-27	500	600	12	

Таблиця ММ

Мете ос.	Дата, час	hмс	dHm	dTau	Ym (Yбюл)	dTauy	AlphaWY	Wу
1	2	3	4	5	6	7	8	9
					02			
					04			
					08			
					12			
					16			
					20			
					24			
					30			
					40			

Таблиця ММН

dH6	dTauy	Wx	Wz

Расчёт Метеосреднее

Расчёт TC по метеосреднему

Рисунок 3.2 - Головне вікно програми із реальними бойовими даними та результатами обрахунків

3.3 Перевірка роботи програми

Для того щоб визначити вирахувані установки за допомогою програми необхідно виконати наступні дії, перше необхідно запусити програму, потім перейти на вкладку «Початкові дані» та натиснути на кнопку «Завантажити початкові дані» як показано на рис. 3.3.

Подготовка **Исходные данные** Результаты расчётов Помощь

WarZone ST - Mass Center
Таблицы стрельбы
СДУ Центра масс

Система:

Снаряд:

Заряд:

Загрузить исходные данные

Сохранить исходные данные

Дальностей: >>

Подбор коэффициента формы

Расчёт

Рисунок 3.3 - Вкладка «Початкові дані»

Після того, як відбулася дія натиску на кнопку «Завантажити початкові дані», з'явиться вікно провідника Windows див. рис. 3.4 за допомогою якого необхідно вибрати задалегіть підготовлений для цієї програми файл, для того типу заряду який потрібен.

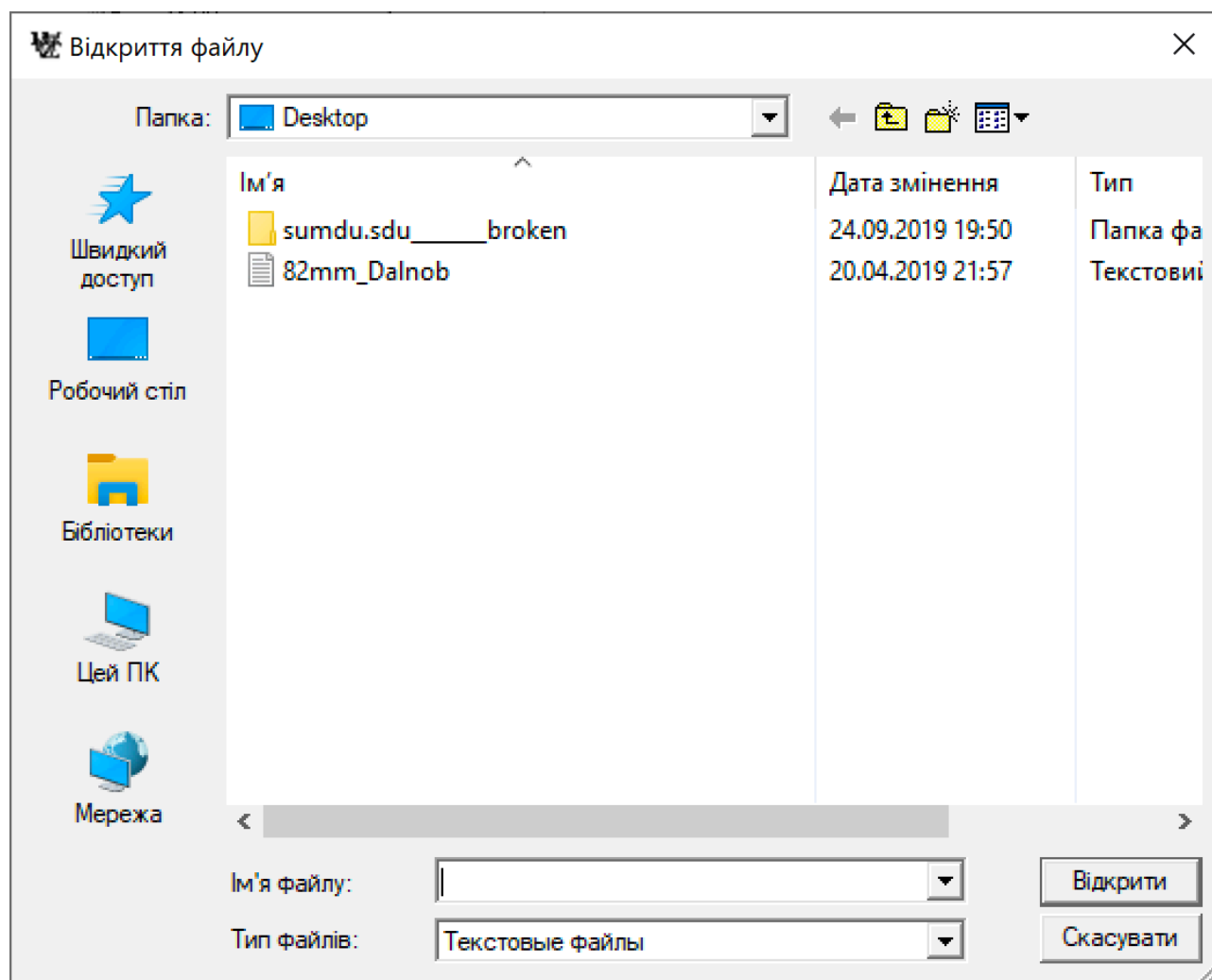


Рисунок 3.4 - Вікно провідника Windows для вибору файла із початковими даними

Після вибору файлу, програма автоматично завантажить дані, та відобразить їх на вкладці «Початкові дані», в таблиці 1 та таблиці 2 як це показано на рис. 3.5, також на цьому скріншоті, окрім двох таблиць заповнилися такі поля як: система, снаряд та заряд, всі ці дані знаходяться у файлі який був завантажений до програми.

Далі необхідно натиснути на кнопку «Розрахувати» та очікувати результату, зазвичай це триває від 10 до 20 секунд, в залежності від потужності вашого комп'ютера.

WarZone Shooting Tables - Mass Center

Подготовка | Исходные данные | Результаты расчётов | Помощь

WarZone ST - Mass Center
Таблицы стрельбы
СДУ Центра масс

Система:

Снаряд:

Заряд:

Дальностей:

Подбор коэффициента формы

Таблица 1			Таблица 2		
Д, м	ТЕТА ₀ , град.мкс	i	1	X ₀ , м	0
640.0	85.00	0.943275	2	Y ₀ , м	0
700.0	84.32	0.943275	3	Z ₀ , м	0
800.0	83.44	0.949525	4	X _t , м	0
900.0	82.57	0.9479625	5	teta ₀ , град	0
1000.0	82.10	0.9448375	6	K _{вр}	0
1100.0	81.22	0.9479625	7	V ₀ , м/с	261
1200.0	80.33	0.95265	8	dP, мм.рт.ст.	10
1300.0	79.45	0.9510875	9	dtau, град.	10
1400.0	78.56	0.9510875	10	W _x , м/с	10
1500.0	78.06	0.9542125	11	W _z , м/с	10
1600.0	77.16	0.9542125	12	T _z , град.	25
1700.0	76.26	0.95265	13	dq, 1в/эн	1
1800.0	75.35	0.95265	14	dV ₀ , %V ₀	1
1900.0	74.43	0.95265	15	d, м	0,082
2000.0	73.50	0.95343125	16	q ₀ , кг	3,1
2100.0	72.58	0.949525	17	t ₀ , с	0
2200.0	72.03	0.9510875	18	tau ₀ N, с	0
2300.0	71.08	0.949525	19	IIN	0
2400.0	70.12	0.9479625	20	w ₀	0
2500.0	69.13	0.949525	21	K1	0
2600.0	68.13	0.949525	22	K2	0
2700.0	67.11	0.95030625	23	tau_ON, К	288.9
2800.0	66.08	0.949134375	24	q ₀ , м/с ²	9,81
2900.0	65.02	0.949525	25	R, м/град	29,27
3000.0	63.52	0.9510875	26	trs	0,0007
3100.0	62.39	0.952259375	27	lq	0,33
3200.0	61.22	0.9542125	28	P ₀ n, мм.рт.ст	750
3300.0	60	0.955775	29	deta, рад	0,0010424
3400.0	58.32	0.957896625	30	eta	1
3500.0	56.56	0.96038225	31	L	0,32893
3600.0	55.10	0.96292125	32	A	0,417
3700.0	53.06	0.9674314375	33	riteta ₀ , тыс	0,5
3800.0	50.37	0.97201525	34	rv ₀ , %	0,5
3900.0	47.05	0.9802074375	35	rc, %	1,6
3922.0	45.00	0.994484375	36	rw, тыс	3,3
			37	rz, рад	0,004
			38	Вш, рад.	0.872664626

Рисунок 3.5 - Вкладка «Початкові дані» із завантаженими даними

На рис. 3.6 зображені успішно обчислені дані, за допомогою продуманої та описаної вище математичної моделі та вдалої програмної реалізації із використанням мови програмування C++.

№ з/п	Параметри	№	№	№	№	№	№	№	№
1	Xц, м	40150							
2	Yц=Zц, м	93200							
3	hц, м	190							
4	Дтц, м	3339							
5	Deltaтц, м	3,13							
6	Alpha ц, м	55,13							
7	delta hц, м	10							
8	Yбюл	1800							

№	Но	783	4	to	-24	7	tauo	-24	10	Alpha wo	1,00
2	-Hno	750	5	+deltaTv	0	8	-tauNo	15,9	11	Wo	5
3	deltaHo	33	6	tauo	-24	9	delta tauomn	-40			

№	1	2	3	4	5	6
	Yм(бюл), м	delta_tau_y,	delta_alpha_	alpha_wy, п.	Wy, м/с	Наближений
12	0	-40		100	5	
13	200	-39	100	200	8	
14	400	-38	200	300	10	
15	800	-37	300	400	10	
16	1200	-35	300	400	11	
17	1600	-34	400	500	11	
18	2000	-32	400	500	11	
19	2400	-31	400	500	12	
20	3000	-30	500	600	12	
21	4000	-27	500	600	12	

Рисунок 3.6 - Топографічні дані по цілі та бюлетень «метеосередній»

За допомогою цих даних артилерист може обстріляти цілі (об'єкти) із високою точністю до 5-7 метрів, тому ця програма принесе користь нашим збройним силам.

ВИСНОВКИ

Виконана робота присвячена розробці математичної моделі та програми для визначення вирахуваних установок для стрільби на ураження різноманітних цілей мінометними комплексами 120 мм. ММК (мобільний мінометний комплекс) і 82 мм. мінемет з метою скорочення часу розрахунків на полі бою. Результатом роботи є програма, що максимально пришвидшує цю дію.

В даній роботі запропонована математична модель та програма руху некерованих літальних апаратів в збуреному середовищі.

Розроблений алгоритм і програма розрахунку бюлетня «метеосередній» по даним наземних вимірювань, топографічним даних по цілям (об'єктам) ураження.

Також дана програма дозволяє розраховувати вираховані установки по цілям (об'єктам) ураження для 120-мм мінометом або 82-мм мінометом з високою точністю (5-7 метрів), час на розрахунок установок складає 3-5 секунд.

СПИСОК ЛІТЕРАТУРИ

1. Дмитриевский А. А., Лысенко Л.Н. Внешняя баллистика. - М.: Издательство «Машиностроение», 2005. - 607 с.
2. Гонтмахер Ф. Р., Левин Л. М. Теория полета неуправляемых ракет. - М.: Издательство физико-математической литературы, 1959. - 355с.
3. Макеев В. И. и др. математична модель просторового руху літального апарату на твердому паливі в атмосфері. Вісник СумДУ №2, 2008.
4. Костенко Н. Ф. Ракетные двигатели твердого топлива (основы теории и проектирования). - ПВАИУ. - Пенза - 1966. - 264 с.
5. Бурлов В. В. Лысенко Л.Н. Баллистика ствольных систем. М. Машиностроения, 2006. – 459 с.
6. Дмитриевский А. А, Лысенко Л. Н. Прикладные задачи теории оптимального управления движением беспилотных летательных аппаратов. М. Машиностроения, 1978 – 403с.
7. Архангельский А. Я. Решение типовых задач в C++Builder 6. — М: ЗАО«Издательство БИНОМ», 2003
8. Джаррод Холингворт, Боб Сворт, Марк Кэшмэн, Поль Густавсон Borland C++ Builder 6. Руководство разработчика = Borland C++ Builder 6 Developer's Guide. — М.: «Вильямс», 2004. — С. 976.
9. Теллес М. Borland C++ Builder: библиотека программиста Пер. с англ. - СПб.: Питер Ком, 1998. - 512 с.; ил.; компакт-диск.
10. Пахомов Б. И. C/C++ и Borland C++ Builder для студента. — СПб.: БХВ-Петербург, 2006. — 448 с.: ил.
11. Шамис В.А. Borland C++ Builder 6. Для профессионалов/В.А. Шамис. -- СПб.:Питер, 2003.-798 с.:ил.
12. TIOBE Index for May 2020 [Электронный ресурс] – Режим доступа: <https://www.tiobe.com/tiobe-index/> Назва з екрану.
13. Страуструп Б. Дизайн и эволюция C++/ Страуструп Б. - М.: ДМК Пресс; Спб.: Питер, 2006. - 448 с.

14. Макеєв В.І. Дослідження впливу вітру на політ некерованих оперених реактивних снарядів і реактивних снарядів, що обертаються // Військово – технічний збірник. Вип. 1. – Львів: Львівський ін-т сухопутних війск. —2009.— С. 12—21.
15. Макеєв В.І. та ін. Математична модель просторового руху літального апарату на твердому паливі в атмосфері // Вісник Сум ДУ. Вип. 2. — Суми: Сумський державний університет, 2008. — С. 5 — 12.

ДОДАТОК

```
//-----
//#include <iostream>
//#include <string>

#include <vcl.h>
#include <math.h>
#include <fstream.h>
#include <string.h>
#include <iostream> //только длѐ даты и времени
#include <ctime> //только длѐ даты и времени
#pragma hdrstop

#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "SDU.cpp"
#include "mymath.cpp"
#include <sstream>
#include <cstdlib>
#include <stdio.h>

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
using namespace std; //только длѐ даты и времени

TForm1 *Form1;
enum indata1 {_D,_tetao,_i,_ybul,INTABL1COLNUM};
enum
indata2 {_Xn,_Yn,_Zn,_Xtn,_teta0n,_Kvr,_vo,_dP,_dtau,_Wx,_Wz,_Tz,_dq,_dvo,_d,
_qo,_tn,_tauaN,_IIN,_w0,_K1,_K2,_tauoN,_go,_R,_ltz,_lq,_PoN,_dteta,_eta,_L,_A,
_rteta0,_rv0,_rc,_rw,_rz,_Bsh,_ag,_Omegaz,_Rz,_dt1,_dt2,_dovorot,_yrr,
INTABL2ROWNUM};

enum outdata1 {_Xt,_Yt,_Zt,_N,_tr,_vr,_tetar,OUTTABL1COLNUM};
// [old order parameters] enum
outdata2 {_teta0,_Xt4,_Pdeleniy,_dXtys,_dPdel,_Zd,_dXw,_dXc,_dXh,_dXt,_dXtz,_
dXvo,_dXq,_dZw,_dXvr,_dZvr,_Pri,_Xr,_Yrm,_Zr,_Vdteta0,_Vdv0,_Vdc,_Vd,_Vv,
_Vb,OUTTABL2COLNUM};
enum outdata2 {_Xt4,_Pdeleniy,_dPdel,_Vd,_Vb,_dZw,_dXw,/*_Zd,*/
_dXh,_dXt,_dXtz,_dXvo,_dXq,/*_dXvr,_dZvr,*/
_teta0,_Pri,_tetac,_vc,_tc,_Xr,_Yrm,_Zr,_Ybul,_dXtys,_dXc,/*
*_Vdteta0,_Vdv0,_Vdc,_Vv,*/OUTTABL2COLNUM};
```

```

AnsiString invarname1[INTABL1COLNUM],invarname2[INTABL2ROWNUM],
    outvarname1[OUTTABL1COLNUM],outvarname2[OUTTABL2COLNUM]/
*,outvarname3[OUTTABL3COLNUM]*/;

```

```

double vd_opitnie[2]

```

```

[28]={1000,2000,3000,4000,5000,6000,7000,8000,9000,10000,11000,12000,13000,
14000,15000,16000,17000,18000,19000,20000,21000,22000,23000,24000,25000,26
000,27000,28000,
43,40,37,35,33,32,31,31,31,32,33,35,37,40,43,46,49,53,56,60,63,67,71,75,80,86,92,1
00};

```

```

double vv_opitnie[2]

```

```

[28]={1000,2000,3000,4000,5000,6000,7000,8000,9000,10000,11000,12000,13000,
14000,15000,16000,17000,18000,19000,20000,21000,22000,23000,24000,25000,26
000,27000,28000,
0.3,0.5,0.8,1.1,1.4,1.7,2.2,2.7,3.4,4.3,5.5,7,9.1,12,15,19,24,30,37,45,54,64,77,91,109,
132,161,207};

```

```

double vb_opitnie[2]

```

```

[28]={1000,2000,3000,4000,5000,6000,7000,8000,9000,10000,11000,12000,13000,
14000,15000,16000,17000,18000,19000,20000,21000,22000,23000,24000,25000,26
000,27000,28000,
0.3,0.5,0.8,1,1.3,1.6,1.9,2.1,2.4,2.7,3,3.3,3.6,4,4.3,4.7,5.2,5.7,6.3,6.9,7.6,8.3,9.2,10,1
1,12,14,15};

```

```

//double indata_[INDATA1NUM];

```

```

//double outdata_[OUTDATANUM];

```

```

/*double invartabl1[INTABL1ROWNUM]={0};

```

```

double invartabl2[INTABL2ROWNUM]={0};

```

```

double outvartabl1[OUTTABL1COLNUM]={0};

```

```

double outvartabl2[OUTTABL2COLNUM]={0};*/

```

```

AnsiString temps;

```

```

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----

```

```

void readindata(const char* fname)

```

```

{

```

```

    ifstream file(fname);

```

```

    char str[100];

```

```

file>>str;
if(strcmp(str,"none")!=0)
Form1->Edit2->Text=str;

```

```

file>>str;
if(strcmp(str,"none")!=0)
Form1->Edit3->Text=str;

```

```

file>>str;
if(strcmp(str,"none")!=0)
Form1->Edit4->Text=str;

```

```

int RC;
file>>RC;
//ShowMessage(RC);

```

```

Form1->StringGrid1->RowCount=RC+1;
for(int i=0;i<Form1->StringGrid1->RowCount-1;++i)
for(int j=0;j<INTABL1COLNUM;++j)
{
    file>>str;
    if(strcmp(str,"none")!=0)
    Form1->StringGrid1->Cells[j][i+1]=str;
    else
    Form1->StringGrid1->Cells[j][i+1]="";
}
for(int i=0;i<INTABL2ROWNUM;++i)
{
    file>>str;
    if(strcmp(str,"none")!=0)
    Form1->StringGrid2->Cells[2][i]=str;
    else
    Form1->StringGrid2->Cells[2][i]="";
}
file.close();
}

```

```

void saveindata(const char* fname)
{
    ofstream file(fname);

```

```

file<<Form1->Edit2->Text.c_str()<<endl;
file<<Form1->Edit3->Text.c_str()<<endl;
file<<Form1->Edit4->Text.c_str()<<endl;
file<<Form1->StringGrid1->RowCount-1<<"\t";
for(int i=0;i<Form1->StringGrid1->RowCount-1;++i)
for(int j=0;j<INTABL1COLNUM;++j)
{
    if(Form1->StringGrid1->Cells[j][i+1]== "")
        file<<"none"<<endl;
    else
        file<<Form1->StringGrid1->Cells[j][i+1].c_str()<<endl;
}
for(int i=0;i<INTABL2ROWNUM;++i)
{
    if(Form1->StringGrid2->Cells[2][i]== "")
        file<<"none"<<endl;
    else
        file<<Form1->StringGrid2->Cells[2][i].c_str()<<endl;
}
file.close();
}

```

```

void __fastcall TForm1::FormCreate(TObject *Sender)
{

```

```

    invarname1[_D]="f, м";
    invarname1[_tetao]="ТЕТАо, град,ммсс";
    invarname1[_i]="i";
    invarname1[_ybul]=""" бюл";

```

```

    invarname2[_Xn]="Xо, м";
    invarname2[_Yn]="Yo, м";
    invarname2[_Zn]="Zo, м";
    invarname2[_Xtn]="Xt, м";
    invarname2[_teta0n]="teta0, град";
    invarname2[_Kvr]="вр";
    invarname2[_vo]="Vo, м/с";
    invarname2[_dP]="dP, мм.рт.ст.";
    invarname2[_dtau]="dtau, град.";
    invarname2[_Wx]="Wx, м/с";
    invarname2[_Wz]="Wz, м/с";
    invarname2[_Tz]="Tз, град.";
    invarname2[_dq]="dq, 1в/3н";
    invarname2[_dvo]="dVo, %Vo";
    invarname2[_d]="d, м";

```

```

invarname2[_qo]="qo, кг";
invarname2[_tn]="t0, с";
invarname2[_tauaN]="tauaN, с";
invarname2[_I1N]="I1N";
invarname2[_w0]="w0";
invarname2[_K1]="K1";
invarname2[_K2]="K2";
invarname2[_tauoN]="tau_ON, ";
invarname2[_go]="go, м/с2";
invarname2[_R]="R, м/град";
invarname2[_ltz]="ltz";
invarname2[_lq]="lq";
invarname2[_PoN]="Pon, мм.рт.ст";
invarname2[_dteta]="dteta, рад";
invarname2[_eta]="eta";
invarname2[_L]="L";
invarname2[_A]="A";
invarname2[_rteta0]="rteta0, тыс";
invarname2[_rv0]="rv0, %";
invarname2[_rc]="rc, %";
invarname2[_rw]="rw, тыс";
invarname2[_rz]="rz, рад";
invarname2[_Bsh]="ш, рад. ";
invarname2[_ag]="ag, рад. ";
invarname2[_Omegaz]="Omega_з, рад/с";
invarname2[_Rz]="Rз, м";
invarname2[_dt1]="deltat1";
invarname2[_dt2]="deltat2";
invarname2[_dovorot]="dovorot";
invarname2[_yrr]="ысота разрыва";

```

```

outvarname1[_Xt]="Т, м";
outvarname1[_Yt]="Yт, м";
outvarname1[_Zt]="Zт, м";
outvarname1[_N]="N";
outvarname1[_tetar]="TETAг";
outvarname1[_vr]="Vr";
outvarname1[_tr]="tr";

```

//в четвёртой таблице дублируется значение Xт чтоб можно было сравнивать результаты с третьей таблицей

```

outvarname2[_teta0]="Alpha, град,мин";
outvarname2[_Xt4]="Т, м";
    outvarname2[_Pdeleniy]="s, дел";
outvarname2[_dPdel]="ds, дел";

```



```

outvarname2[_dXtys]="dX, тыс";
// [hide unused] outvarname2[_Zd]="Z, тыс";
outvarname2[_dXw]="dXw, м";
outvarname2[_dXh]="dXh, м";
outvarname2[_dXt]="dXt, м";
outvarname2[_dXtz]="dXtz, м";
outvarname2[_dXvo]="dXvo, м";
outvarname2[_dXq]="dXq, м";
outvarname2[_dZw]="dZw, тыс";
outvarname2[_tetac]="ТЕТАс, град";
outvarname2[_vc]="Vс, м/с";
outvarname2[_tc]="Tс, с";
// [hide unused] outvarname2[_dXvr]="dXvr, м";
// [hide unused] outvarname2[_dZvr]="dZvr, тыс";
outvarname2[_Xr]="р, м";
outvarname2[_Yrm]="Ys, м";
outvarname2[_Zr]="Zp, м";
outvarname2[_dXc]="dXc, м";
outvarname2[_Pri]="s, тыс";
outvarname2[_Ybul]="Ybul, м";
outvarname2[_Vd]="¬д, м";
// [hide unused] outvarname2[_Vv]="¬в, м";
outvarname2[_Vb]="¬б, м";
// [hide unused] outvarname2[_Vdteta0]="¬д_тета0, м";
// [hide unused] outvarname2[_Vdv0]="¬д_V0, м";
// [hide unused] outvarname2[_Vdc]="¬д_с, м";

/* outvarname3[_x]="", м"; //динамика полёта
   [_y]="Y, м";
   outvarname3[_zoutvarname3]="Z, м";
   outvarname3[_v]="v, м/с";
   outvarname3[_psi1]="psi1";
   outvarname3[_psi2]="psi2";
   outvarname3[_x1]="x1";
   outvarname3[_x2]="x2";
   outvarname3[_alpha1]="alpha1";
   outvarname3[_alpha2]="alpha2";
   outvarname3[_r]="r";
   outvarname3[_pi]="pi";
   outvarname3[_deltaz]="deltaz"; */

StringGrid1->ColCount=INTABL1COLNUM;
for(int i=0;i<StringGrid1->ColCount;++i)
{
    StringGrid1->Cells[i][0]=invarname1[i];
}

```

```

}

StringGrid2->RowCount=INTABL2ROWNUM;
for(int i=0;i<StringGrid2->RowCount;++i)
{
    StringGrid2->Cells[0][i]=IntToStr(i+1);
    StringGrid2->Cells[1][i]=invarname2[i];
}

StringGrid3->ColCount=OUTTABL1COLNUM;
for(int i=0;i<StringGrid3->ColCount;++i)
StringGrid3->Cells[i][0]=outvarname1[i];

StringGrid4->ColCount=OUTTABL2COLNUM;
for(int i=0;i<StringGrid4->ColCount;++i)
StringGrid4->Cells[i][0]=outvarname2[i];

/* Form2->StringGrid1->ColCount=OUTTABL3COLNUM;           //динамика
полста:таблица
for(int i=0;i<Form2->StringGrid1->ColCount;++i)
Form2->StringGrid1->Cells[i][0]=outvarname3[i];           */

//readindata("resources\indata__.txt");

StringGrid5->Cells[0][0]="√арматы";
StringGrid5->Cells[1][0]=6;
StringGrid5->Cells[2][0]=5;
StringGrid5->Cells[3][0]=4;
StringGrid5->Cells[4][0]="3 осн.";
StringGrid5->Cells[5][0]=2;
StringGrid5->Cells[6][0]=1;
StringGrid5->Cells[0][1]="'б,м";
StringGrid5->Cells[0][2]="Zб,м";
StringGrid5->Cells[0][3]="hб,м";
StringGrid5->Cells[0][4]="Alpha(оси)";
StringGrid5->Cells[0][5]="«ар€д";

StringGrid6->Cells[0][0]="€ з/п";
StringGrid6->Cells[1][0]="сараметри";
StringGrid6->Cells[2][0]="«наченн€";
StringGrid6->Cells[0][1]=1;
StringGrid6->Cells[0][2]=2;
StringGrid6->Cells[0][3]=3;

```

```
StringGrid6->Cells[0][4]=4;
StringGrid6->Cells[0][5]=5;
StringGrid6->Cells[1][1]="Но, мм. рт. ст.";
StringGrid6->Cells[1][2]="to, C";
StringGrid6->Cells[1][3]="Alpha wo, под. кут.";
StringGrid6->Cells[1][4]="Wo, м/с";
StringGrid6->Cells[1][5]="hmc, м";
```

```
StringGrid7->Cells[0][0]="ε з/π";
StringGrid7->Cells[1][0]="сараметри";
StringGrid7->Cells[2][0]="ε";
StringGrid7->Cells[3][0]="ε";
StringGrid7->Cells[4][0]="ε";
StringGrid7->Cells[5][0]="ε";
StringGrid7->Cells[6][0]="ε";
StringGrid7->Cells[7][0]="ε";
StringGrid7->Cells[8][0]="ε";
StringGrid7->Cells[9][0]="ε";
StringGrid7->Cells[0][1]=1;
StringGrid7->Cells[0][2]=2;
StringGrid7->Cells[0][3]=3;
StringGrid7->Cells[0][4]=4;
StringGrid7->Cells[0][5]=5;
StringGrid7->Cells[0][6]=6;
StringGrid7->Cells[0][7]=7;
StringGrid7->Cells[0][8]=8;
StringGrid7->Cells[1][1]="Xц, м";
StringGrid7->Cells[1][2]="Yц=Zц, м";
StringGrid7->Cells[1][3]="hц, м";
StringGrid7->Cells[1][4]="fтц, м";
StringGrid7->Cells[1][5]="Deltaтц, м";
StringGrid7->Cells[1][6]="Alpha ц, м";
StringGrid7->Cells[1][7]="delta hц, м";
StringGrid7->Cells[1][8]="Yбюл";
```

```
StringGrid8->Cells[0][0]=1;
StringGrid8->Cells[0][1]=2;
StringGrid8->Cells[0][2]=3;
StringGrid8->Cells[1][0]="Ho";
StringGrid8->Cells[1][1]="-Hno";
StringGrid8->Cells[1][2]="deltaHo";
StringGrid8->Cells[3][0]=4;
StringGrid8->Cells[3][1]=5;
StringGrid8->Cells[3][2]=6;
StringGrid8->Cells[4][0]="to";
```

```

StringGrid8->Cells[4][1]="+deltaTv";
StringGrid8->Cells[4][2]="tauo";
StringGrid8->Cells[6][0]=7;
StringGrid8->Cells[6][1]=8;
StringGrid8->Cells[6][2]=9;
StringGrid8->Cells[7][0]="tauo";
StringGrid8->Cells[7][1]="-tauNo";
StringGrid8->Cells[7][2]="delta tauoMn";
StringGrid8->Cells[9][0]=10;
StringGrid8->Cells[9][1]=11;
StringGrid8->Cells[10][0]="Alpha wo";
StringGrid8->Cells[10][1]="Wo";

```

```

StringGrid9->Cells[0][0]="e";
StringGrid9->Cells[1][0]=1;
StringGrid9->Cells[2][0]=2;
StringGrid9->Cells[3][0]=3;
StringGrid9->Cells[4][0]=4;
StringGrid9->Cells[5][0]=5;
StringGrid9->Cells[6][0]=6;
StringGrid9->Cells[7][0]=7;
StringGrid9->Cells[1][1]="Ym(бюл), м";
StringGrid9->Cells[2][1]="delta_tau_y, C";
StringGrid9->Cells[3][1]="delta_alpha_wy, п.к.";
StringGrid9->Cells[4][1]="alpha_wy, п.к.";
StringGrid9->Cells[5][1]="Wy, м/с";
StringGrid9->Cells[6][1]="Каблiжений бюлетень";
StringGrid9->Cells[0][2]=12;
StringGrid9->Cells[0][3]=13;
StringGrid9->Cells[0][4]=14;
StringGrid9->Cells[0][5]=15;
StringGrid9->Cells[0][6]=16;
StringGrid9->Cells[0][7]=17;
StringGrid9->Cells[0][8]=18;
StringGrid9->Cells[0][9]=19;
StringGrid9->Cells[0][10]=20;
StringGrid9->Cells[0][11]=21;
StringGrid9->Cells[1][2]=0;
StringGrid9->Cells[1][3]=200;
StringGrid9->Cells[1][4]=400;
StringGrid9->Cells[1][5]=800;
StringGrid9->Cells[1][6]=1200;
StringGrid9->Cells[1][7]=1600;
StringGrid9->Cells[1][8]=2000;

```

```
StringGrid9->Cells[1][9]=2400;
StringGrid9->Cells[1][10]=3000;
StringGrid9->Cells[1][11]=4000;
StringGrid9->Cells[3][3]="100";
StringGrid9->Cells[3][4]="200";
StringGrid9->Cells[3][5]="300";
StringGrid9->Cells[3][6]="300";
StringGrid9->Cells[3][7]="400";
StringGrid9->Cells[3][8]="400";
StringGrid9->Cells[3][9]="400";
StringGrid9->Cells[3][10]="500";
StringGrid9->Cells[3][11]="500";
```

```
StringGrid10->Cells[0][0]="delta Ha";
StringGrid10->Cells[1][0]="delta tau_y";
StringGrid10->Cells[2][0]="Wx";
StringGrid10->Cells[3][0]="Wz";
```

```
StringGrid16->Cells[0][0]="sbp";
StringGrid16->Cells[1][0]="ДЦВ";
StringGrid16->Cells[2][0]="Xp";
StringGrid16->Cells[3][0]="Zp";
```

```
StringGrid8->Cells[2][1]=750;
StringGrid8->Cells[8][1]=15.9;
```

```
StringGrid11->Cells[0][0]="параметр";
StringGrid11->Cells[0][1]="«наченн€";
StringGrid11->Cells[1][0]="DeltaVo";
StringGrid11->Cells[2][0]="“з";
StringGrid11->Cells[3][0]="Delta q";
```

```
StringGrid12->Cells[0][0]="sbp";
StringGrid12->Cells[1][0]="ДЦВ";
```

```
StringGrid11->Cells[1][1]=-0.7;
StringGrid11->Cells[2][1]=-16;
StringGrid11->Cells[3][1]=3;
```

```
StringGrid5->Cells[4][1]=37235;
StringGrid5->Cells[4][2]=94830;
StringGrid5->Cells[4][3]=180;
StringGrid5->Cells[4][4]="52,00";
StringGrid5->Cells[4][5]="фальноб.";
```

```
StringGrid6->Cells[2][1]=783;
StringGrid6->Cells[2][2]=-24;
StringGrid6->Cells[2][3]="1,00";
StringGrid6->Cells[2][4]=5;
StringGrid6->Cells[2][5]=200;
```

```
StringGrid7->Cells[2][1]=40150;
StringGrid7->Cells[2][2]=93200;
StringGrid7->Cells[2][3]=190;
```

```
//StringGrid8->Cells[2][0]=736;
//StringGrid8->Cells[11][0]=17;
//StringGrid8->Cells[11][1]=9;
//StringGrid8->Cells[5][0]=3;
```

```
//-----
```

```
// гетеосреднее
```

```
StringGrid13->Cells[0][0]="гете ос.";
StringGrid13->Cells[1][0] = "фага, час";
StringGrid13->Cells[2][0] = "hmc";
StringGrid13->Cells[3][0] = "dHm";
StringGrid13->Cells[4][0] = "dTau";
```

```
StringGrid13->Cells[0][1]="1";
StringGrid13->Cells[1][1] = "2";
StringGrid13->Cells[2][1] = "3";
StringGrid13->Cells[3][1] = "4";
StringGrid13->Cells[4][1] = "5";
```

```
StringGrid14->Cells[0][0]="Ym (Yбюл)";
StringGrid14->Cells[1][0] = "dTauy";
StringGrid14->Cells[2][0] = "AlphaWY";
StringGrid14->Cells[3][0] = "Wy";
```

```
StringGrid14->Cells[0][1]="6";
StringGrid14->Cells[1][1] = "7";
StringGrid14->Cells[2][1] = "8";
StringGrid14->Cells[3][1] = "9";
```

```
StringGrid14->Cells[0][2]="02";
StringGrid14->Cells[0][3]="04";
StringGrid14->Cells[0][4]="08";
StringGrid14->Cells[0][5]="12";
StringGrid14->Cells[0][6]="16";
StringGrid14->Cells[0][7]="20";
StringGrid14->Cells[0][8]="24";
StringGrid14->Cells[0][9]="30";
StringGrid14->Cells[0][10]="40";
```

```
StringGrid15->Cells[0][0]="dH6";
StringGrid15->Cells[1][0] = "dTauy";
StringGrid15->Cells[2][0] = "Wx";
StringGrid15->Cells[3][0] = "Wz";
```

```
//-----
```

```
OpenDialog1->InitialDir=ExtractFileDir(Application->ExeName);
SaveDialog1->InitialDir=ExtractFileDir(Application->ExeName);
```

```
Edit1->Text=StringGrid1->RowCount-1;
```

```
}
```

```
//-----
```

```
void __fastcall TForm1::Button1Click(TObject *Sender)
```

```
{ //Вот тут должен быть код показа динамики полста
```

```
/* char str[50];
```

```
ifstream file("data.txt");
```

```
for(int i=0;i<INDATA1NUM;++i)
```

```
{
```

```
file>>str;
```

```
point2comma(str);
```

```
Form1->StringGrid1->Cells[2][i]=str;
```

```
}
```

```
//StringGrid1->Cells[2][_tetao]=delugl2degstr(StringGrid1->Cells[2][_tetao]);
```

```
file.close(); */
```

```
//Form2->Show();
```

```
}
```

```
//-----
```

```
void __fastcall TForm1::Button2Click(TObject *Sender)
```

```
{
```

```
if(!Form1->OpenDialog1->Execute())return;
```

```
readindata(Form1->OpenDialog1->FileName.c_str());
```

```
}
```

```
//-----
```

```

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    if(!Form1->SaveDialog1->Execute())return;
    saveindata(Form1->SaveDialog1->FileName.c_str());
}
//-----
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    saveindata("resources\\indata__.txt");
}
//-----
void __fastcall TForm1::Button4Click(TObject *Sender)
{
    float Xc,Zc,hc,Xb,Zb,hb,alpha_osi;
    float
delta_X,delta_Z,delta_h_c,H,alpha_c,D_c_t,delta_c_t,delta_c_t_shryh,sigma_c,Ho,to
,alpha_wo,Wo,hmc,delta_Ho,delta_Hb,Y_byl,delta_Tv,tau0,delta_tau_0mp,delta_tau
_y,Wy,Aw,Aw_grad,Wx,Wz;
    float table_1[2][11] =
    {
        { 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 },
        { 0.3, 0.5, 0.6, 0.9, 1.3, 1.8, 2.4, 3.3, 4.4, 5.8, 7.4 }
    };
    int table_4_col_3[9] = {100, 200, 300, 300, 400, 400, 400, 500, 500};
    int table_4_col_1[10] = {0, 200, 400, 800, 1200, 1600, 2000, 2400, 3000, 4000};
    float table_4[19][15] =
    {
        {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50},
        {200, -1, -2, -3, -4, -5, -6, -7, -8, -8, -9, -20, -29, -39, -49},
        {200, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {400, -1, -2, -3, -4, -5, -6, -6, -7, -8, -9, -19, -29, -38, -48},
        {400, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {800, -1, -2, -3, -4, -5, -6, -6, -7, -7, -8, -18, -28, -37, -46},
        {800, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {1200, -1, -2, -3, -4, -4, -5, -5, -6, -7, -8, -17, -26, -35, -44},
        {1200, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {1600, -1, -2, -3, -3, -4, -4, -5, -6, -7, -8, -17, -25, -34, -42},
        {1600, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {2000, -1, -2, -3, -3, -4, -4, -5, -6, -6, -7, -16, -24, -32, -40},
        {2000, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {2400, -1, -2, -2, -3, -4, -4, -5, -5, -6, -7, -15, -23, -31, -38},
        {2400, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {3000, -1, -2, -2, -3, -4, -4, -4, -5, -5, -6, -15, -22, -30, -37},
        {3000, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
        {4000, -1, -2, -2, -3, -4, -4, -4, -4, -5, -6, -14, -20, -27, -34},
    }
}

```



```
{4000, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0}
};
```

```
float table_3[10][15] = {
    {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15},
    {200, 3, 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22},
    {400, 4, 5, 7, 10, 11, 12, 14, 17, 18, 20, 22, 23, 25, 27},
    {800, 4, 5, 8, 10, 11, 13, 15, 18, 19, 21, 23, 25, 27, 28},
    {1200, 4, 5, 8, 11, 12, 13, 16, 19, 20, 22, 24, 26, 28, 30},
    {1600, 4, 6, 8, 11, 13, 14, 17, 20, 21, 23, 25, 27, 29, 32},
    {2000, 4, 6, 9, 11, 13, 14, 17, 20, 21, 24, 26, 28, 30, 32},
    {2400, 4, 6, 9, 12, 14, 15, 18, 21, 22, 25, 27, 29, 32, 34},
    {3000, 5, 6, 9, 12, 14, 15, 18, 21, 23, 25, 28, 30, 32, 36},
    {4000, 5, 6, 10, 12, 14, 16, 19, 22, 24, 26, 29, 32, 34, 36}
};
```

```
int table_4_col_2[9] = {200, 400, 800, 1200, 1600, 2000, 2400, 3000, 4000};
int table_4_row_1[14] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50};
```

```
float table_4_minus[9][14] =
{
    {-1, -2, -3, -4, -5, -6, -7, -8, -8, -9, -20, -29, -39, -49},
    {-1, -2, -3, -4, -5, -6, -6, -7, -8, -9, -19, -29, -38, -48},
    {-1, -2, -3, -4, -5, -6, -6, -7, -7, -8, -18, -28, -37, -46},
    {-1, -2, -3, -4, -4, -5, -5, -6, -7, -8, -17, -26, -35, -44},
    {-1, -2, -3, -3, -4, -4, -5, -6, -7, -8, -17, -25, -34, -42},
    {-1, -2, -3, -3, -4, -4, -5, -6, -6, -7, -16, -24, -32, -40},
    {-1, -2, -2, -3, -4, -4, -5, -5, -6, -7, -15, -23, -31, -38},
    {-1, -2, -2, -3, -4, -4, -4, -5, -5, -6, -15, -22, -30, -37},
    {-1, -2, -2, -3, -4, -4, -4, -4, -5, -6, -14, -20, -27, -34}
};
```

```
float table_4_plus[9][14] =
{
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 0, 0}
};
```

```
StringGrid8->Cells[2][0]=Form1->StringGrid6->Cells[2][1];
StringGrid8->Cells[11][0]=Form1->StringGrid6->Cells[2][3];
StringGrid8->Cells[11][1]=Form1->StringGrid6->Cells[2][4];
```

```
StringGrid8->Cells[5][0]=Form1->StringGrid6->Cells[2][2];
```

```
Xc=StrToFloat(Form1->StringGrid7->Cells[2][1]);
Zc=StrToFloat(Form1->StringGrid7->Cells[2][2]);
hc=StrToFloat(Form1->StringGrid7->Cells[2][3]);
Xb=StrToFloat(Form1->StringGrid5->Cells[4][1]);
Zb=StrToFloat(Form1->StringGrid5->Cells[4][2]);
hb=StrToFloat(Form1->StringGrid5->Cells[4][3]);
alpha_osi=StrToFloat(Form1->StringGrid5->Cells[4][4]);
```

```
delta_X = Xc - Xb;
delta_Z = Zc - Zb;
delta_h_c = hc - hb;
StringGrid7->Cells[2][7]=delta_h_c;
H=atan(-1*(delta_Z/delta_X));
if (delta_Z>=0 && delta_X>0){
    alpha_c = (H*30)/M_PI;
} else if (delta_Z>0 && delta_X==0) {
    alpha_c = 15.0;
} else if (delta_Z>=0 && delta_X<0) {
    alpha_c = ((M_PI-H)*30)/M_PI;
} else if (delta_Z<=0 && delta_X<0) {
    alpha_c = ((M_PI+H)*30)/M_PI;
} else if (delta_Z<0 && delta_X==0) {
    alpha_c = 45.00;
} else if (delta_Z<=0 && delta_X>0) {
    alpha_c = ((2*M_PI-H)*30)/M_PI;
}
D_c_t = sqrt(pow(delta_Z,2)+pow(delta_X,2));
D_c_t = floor(((float)(int)(D_c_t * 100 + 0.5)) / 100);
StringGrid7->Cells[2][4]=D_c_t;
```

```
delta_c_t_shryh = alpha_c - alpha_osi;
StringGrid7->Cells[2][5]=(((float)(int)(delta_c_t_shryh * 100 + 0.5)) / 100);
```

```
StringGrid7->Cells[2][6]=(((float)(int)(alpha_c * 100 + 0.5)) / 100);
```

```
if (abs(delta_c_t_shryh) <= 30){
    delta_c_t = delta_c_t_shryh;
} else if ((abs(delta_c_t_shryh) > 30) && (delta_c_t_shryh < 0)){
    delta_c_t = alpha_c - alpha_osi + 60;
} else if ((abs(delta_c_t_shryh) > 30) && (delta_c_t_shryh > 0)){
    delta_c_t = alpha_c - alpha_osi - 60;
}
}
```

```

sigma_c = (delta_h_c * 0,95)/(0,001*D_c_t*100);

Ho = StrToFloat(Form1->StringGrid6->Cells[2][1]);
to = StrToFloat(Form1->StringGrid6->Cells[2][2]);
alpha_wo = StrToFloat(Form1->StringGrid6->Cells[2][3])*100;
Wo = StrToFloat(Form1->StringGrid6->Cells[2][4]);
hmc = StrToFloat(Form1->StringGrid6->Cells[2][5]);

delta_Ho = Ho - 750;
StringGrid8->Cells[2][2]=delta_Ho;
delta_Hb = delta_Ho + (hmc - hb)/10;
//StringGrid2->Cells[2][7]=delta_Hb;
for(int i=1;i<(StringGrid1->RowCount)-1;++i){
    if (StrToFloat(StringGrid1->Cells[0][i]) <= D_c_t && StrToFloat(StringGrid1-
>Cells[0][i+1]) >= D_c_t){
        Y_byl = (((D_c_t - StrToFloat(StringGrid1->Cells[0]
[i]))*(StrToFloat(StringGrid1->Cells[3][i+1])-StrToFloat(StringGrid1->Cells[3][i])))/
(StrToFloat(StringGrid1->Cells[0][i+1]) - StrToFloat(StringGrid1->Cells[0][i])))
+StrToFloat(StringGrid1->Cells[3][i]);
    }
}

// for(temp1 = 1; D_c_t > StrToFloat(StringGrid1->Cells[temp1][0]); temp1++){
//     if((StrToFloat(StringGrid1->Cells[temp1][0])+StrToFloat(StringGrid1-
>Cells[temp1-1][0]))/2 < D_c_t){
//         continue;
//     }
//     else{
//         temp1=temp1-1;
//     }
// }
// Y_byl = StrToFloat(StringGrid1->Cells[2][temp1]);
StringGrid7->Cells[2][8]=Y_byl;

for(int i=0;i<11-1;++i){
    if (table_1[0][i] <= to && table_1[0][i+1] >= to){
        delta_Tv = ( ( to - table_1[0][i] )*( table_1[1][i+1] - table_1[1][i] ) /
( table_1[0][i+1] - table_1[0][i] ) + table_1[1][i];
    }
}
delta_Tv;
StringGrid8->Cells[5][1]=((float)(int)(delta_Tv * 100 + 0.5)) / 100;
tau0 = to + delta_Tv;
StringGrid8->Cells[5][2]=floor((((float)(int)(tau0 * 100 + 0.5)) / 100);

```

```
StringGrid8->Cells[8][0]=floor((((float)(int)(tau0 * 100 + 0.5)) / 100);
delta_tau_0mp=tau0-15.9;
StringGrid8->Cells[8][2]=(((float)(int)(delta_tau_0mp * 100 + 0.5)) / 100);
// Y = ( ( X - X0 )*( Y1 - Y0 ) / ( X1 - X0 ) ) + Y0
```

```
StringGrid8->Cells[8][2] = floor(StrToFloat(StringGrid8->Cells[8][2]));
StringGrid9->Cells[2][2] = StringGrid8->Cells[8][2];
```

```
int count=0;
for(int i=0;i<19-1;++i){
float local_delta_tau_0mp;
if (delta_tau_0mp < 0){
//local_delta_tau_0mp = (delta_tau_0mp-StrToFloat(StringGrid9->Cells[3]
[3+count])) * -1;
local_delta_tau_0mp = delta_tau_0mp * -1;
}
for(int j=0;j<15-1;++j){
if (table_4[i][j] <= local_delta_tau_0mp && table_4[i][j+1] >=
local_delta_tau_0mp){
if (delta_tau_0mp < 0){
delta_tau_y = ( ( local_delta_tau_0mp - table_4[i][j] )*( table_4[i+1]
[j+1] - table_4[i+1][j]) / ( table_4[i][j+1] - table_4[i][j] ) ) + table_4[i+1][j];
} else {
delta_tau_y = ( ( local_delta_tau_0mp - table_4[i][j] )*( table_4[i+2]
[j+1] - table_4[i+2][j]) / ( table_4[i][j+1] - table_4[i][j] ) ) + table_4[i+2][j];
}
i++;
}
}
count++;
StringGrid9->Cells[2][2+count]=floor(delta_tau_y+0.5);
}
delta_tau_0mp = floor(delta_tau_0mp);
int trp=0;
count=0;
int tmp;
for(int i=0;i<9;++i){
//StringGrid7->Cells[3][1+i]=i;
//StringGrid7->Cells[6][1+i]=count;
if (delta_tau_0mp > 0){
trp=0;
for(int j=0;j<14;j++){
if(delta_tau_0mp == table_4_row_1[j]){
StringGrid9->Cells[2][3+count]=table_4_plus[count][j];
trp=1;
}
```

```

        break;
    }
}
if(trp==0){
    for(int j=9;j<13;j++){
        if(delta_tau_0mp > table_4_row_1[j] && delta_tau_0mp <
table_4_row_1[j+1]){
            tmp = delta_tau_0mp - table_4_row_1[j];
            StringGrid9->Cells[2][3+count]=table_4_plus[count][j]
+table_4_plus[count][tmp];
            break;
        }
    }
}
} else{
    trp=0;
    for(int j=0;j<14;j++){
        if((delta_tau_0mp*-1) == table_4_row_1[j]){
            StringGrid9->Cells[2][3+count]=table_4_minus[count][j];
            trp=1;
            break;
        }
    }
    if(trp==0){
        for(int j=11;j<14;j++){
            if(delta_tau_0mp > table_4_row_1[j] && delta_tau_0mp <
table_4_row_1[j+1]){
                tmp = delta_tau_0mp - table_4_row_1[j];
                StringGrid9->Cells[2][3+count]=table_4_minus[count][j]
+table_4_minus[count][tmp];
                break;
            }
        }
    }
}
}

count++;
}
StringGrid9->Cells[4][2]=alpha_wo;
StringGrid9->Cells[5][2]=Wo;
for(int i=0;i<9;++i){
    StringGrid9->Cells[4][3+i]=alpha_wo+table_4_col_3[i];
}
for(int j=0;j<9;++j){
    for(int i=1;i<16-1;++i){

```

```

    if (table_3[0][i] <= Wo && table_3[0][i+1] >= Wo){
        Wy = ( ( Wo - table_3[0][i] )*( table_3[j+1][i] - table_3[j+1][i] ) / ( table_3[0]
[i+1] - table_3[0][i] ) ) + table_3[j+1][i];
    }
}
StringGrid9->Cells[5][3+j]=Wy;
}

```

```

int temp;
for(temp = 0; Y_byl > table_4_col_1[temp]; temp++){
    if((table_4_col_1[temp]+table_4_col_1[temp-1])/2 < Y_byl){
        continue;
    }
    else{
        temp=temp-1;
    }
}

```

```

if(alpha_c*100>=StrToFloat(StringGrid9->Cells[4][2+temp])){
    Aw = alpha_c * 100 - StrToFloat(StringGrid9->Cells[4][2+temp]);
} else if(alpha_c*100<StrToFloat(StringGrid9->Cells[4][2+temp])){
    Aw = alpha_c * 100 + 60 - StrToFloat(StringGrid9->Cells[4][2+temp]);
}
/*ofstream foutf;
foutf.open("log.txt");
foutf << StrToFloat(StringGrid9->Cells[5][2+temp]);
foutf.close();*/

```

```

Aw_grad = (Aw*3.6)/60;
Aw_grad = floor(Aw_grad+0.5);

```

```

Wx = -1 * StrToFloat(StringGrid9->Cells[5][2+temp]) * cos(Aw_grad*3.14/180);
Wz = StrToFloat(StringGrid9->Cells[5][2+temp]) * sin(Aw_grad*3.14/180);
float tds = StrToFloat(StringGrid9->Cells[5][2+temp]);

```

```

StringGrid10->Cells[0][1]=delta_Hb;
StringGrid10->Cells[1][1]=StringGrid9->Cells[2][2+temp];
StringGrid10->Cells[2][1]=floor(Wx+0.5);
StringGrid10->Cells[3][1]=floor(Wz+0.5);

```

```

//StringGrid2->Cells[2][8]=StringGrid9->Cells[2][2+temp];
//StringGrid2->Cells[2][9]=floor(Wx+0.5);
//StringGrid2->Cells[2][10]=floor(Wz+0.5);

```

```
//StringGrid2->Cells[2][11]=floor(StrToFloat(StringGrid11->Cells[2]
[1].c_str()-15);
//StringGrid2->Cells[2][12]=StringGrid11->Cells[3][1].c_str();
//StringGrid2->Cells[2][13]=StringGrid11->Cells[1][1].c_str();
```

```
for(int i=1;i<StringGrid1->RowCount;++i)
point2comma(StringGrid1->Cells[2][i].c_str());
```

```
for(int i=0;i<StringGrid2->RowCount;++i)
point2comma(StringGrid2->Cells[2][i].c_str());
```

```
TSDUSolver S;
TSDU Sm;
```

```
StringGrid3->RowCount=StringGrid1->RowCount;
StringGrid4->RowCount=StringGrid1->RowCount;
Form4->StringGrid1->ColCount=StringGrid1->RowCount;
```

```
//чтение исходных данных, общих для всех дальностей
S.v0=StrToFloat(StringGrid2->Cells[2][_vo]);
S.dP=StrToFloat(StringGrid2->Cells[2][_dP]);
S.dtau=StrToFloat(StringGrid2->Cells[2][_dtau]);
S.Wx=StrToFloat(StringGrid2->Cells[2][_Wx]);
S.Wz=StrToFloat(StringGrid2->Cells[2][_Wz]);
S.Tz=StrToFloat(StringGrid2->Cells[2][_Tz]);
S.dq=StrToFloat(StringGrid2->Cells[2][_dq]);
S.dvo=StrToFloat(StringGrid2->Cells[2][_dvo]);
S.d=StrToFloat(StringGrid2->Cells[2][_d]);
S.qo=StrToFloat(StringGrid2->Cells[2][_qo]);
S.tauoN=StrToFloat(StringGrid2->Cells[2][_tauoN]);
S.go=StrToFloat(StringGrid2->Cells[2][_go]);
S.R=StrToFloat(StringGrid2->Cells[2][_R]);
S.ltz=StrToFloat(StringGrid2->Cells[2][_ltz]);
S.lq=StrToFloat(StringGrid2->Cells[2][_lq]);
S.PoN=StrToFloat(StringGrid2->Cells[2][_PoN]);
S.dteta=StrToFloat(StringGrid2->Cells[2][_dteta]);
S.eta=StrToFloat(StringGrid2->Cells[2][_eta]);
S.L=StrToFloat(StringGrid2->Cells[2][_L]);
S.A=StrToFloat(StringGrid2->Cells[2][_A]);
S.rteta0=StrToFloat(Form1->StringGrid2->Cells[2][_rteta0]);
S.rv0=StrToFloat(Form1->StringGrid2->Cells[2][_rv0]);
S.rc=StrToFloat(Form1->StringGrid2->Cells[2][_rc]);
S.rw=StrToFloat(Form1->StringGrid2->Cells[2][_rw]);
```

```

S.rz=StrToFloat(Form1->StringGrid2->Cells[2][_rz]);
S.Bsh=StrToFloat(Form1->StringGrid2->Cells[2][_Bsh]);
S.ag=StrToFloat(Form1->StringGrid2->Cells[2][_ag]);
S.Omegaz=StrToFloat(Form1->StringGrid2->Cells[2][_Omegaz]);
S.Rz=StrToFloat(Form1->StringGrid2->Cells[2][_Rz]);
S.x0=StrToFloat(Form1->StringGrid2->Cells[2][_Xn]);
S.y0=StrToFloat(Form1->StringGrid2->Cells[2][_Yn]);
S.z0=StrToFloat(Form1->StringGrid2->Cells[2][_Zn]);
S.Xtn=StrToFloat(Form1->StringGrid2->Cells[2][_Xtn]);
S.teta0n=StrToFloat(Form1->StringGrid2->Cells[2][_teta0n]);
S.Kvrs=StrToFloat(Form1->StringGrid2->Cells[2][_Kvr]);
S.tn=StrToFloat(Form1->StringGrid2->Cells[2][_tn]);
S.tauaN=StrToFloat(Form1->StringGrid2->Cells[2][_tauaN]);
S.I1N=StrToFloat(Form1->StringGrid2->Cells[2][_I1N]);
S.wo=StrToFloat(Form1->StringGrid2->Cells[2][_w0]);
S.K1=StrToFloat(Form1->StringGrid2->Cells[2][_K1]);
S.K2=StrToFloat(Form1->StringGrid2->Cells[2][_K2]);
S.deltat1=StrToFloat(Form1->StringGrid2->Cells[2][_dt1]); //boop
S.deltat2=StrToFloat(Form1->StringGrid2->Cells[2][_dt2]); //boop
S.dovorot=StrToFloat(Form1->StringGrid2->Cells[2][_dovorot]);
//S.yrr=StrToFloat(Form1->StringGrid2->Cells[2][_yrr]);
yrr=StrToFloat(Form1->StringGrid2->Cells[2][_yrr]);

```

```

/*Form2->Series1->Clear();
Form2->Series2->Clear();
Form2->Series3->Clear();
Form2->Series4->Clear(); */

```

```

Form3->Series1->Clear();
Form3->Series2->Clear();
Form3->Series3->Clear();

```

```

StringGrid7->Cells[5][1]=S.dP;
StringGrid7->Cells[5][2]=S.dtau;
StringGrid7->Cells[5][3]=S.Wx;
StringGrid7->Cells[5][4]=S.Wz;
StringGrid7->Cells[5][5]=S.Tz;
StringGrid7->Cells[5][6]=S.dq;
StringGrid7->Cells[5][7]=S.dvo;

```

```

ofstream file1 ("parameters\\Vd.txt");
ofstream file2 ("parameters\\Vdteta0.txt");
ofstream file3 ("parameters\\Vdv0.txt");

```



```

ofstream file4 ("parameters\\Vdc.txt");
ofstream file5 ("parameters\\Vv.txt");
ofstream file6 ("parameters\\Vb.txt");
ofstream file7 ("parameters\\Xt.txt");
ofstream file8 ("parameters\\Yt.txt");
ofstream file9 ("parameters\\Zt.txt");
ofstream file10 ("parameters\\tetac.txt");
ofstream file11 ("parameters\\vc.txt");
ofstream file12 ("parameters\\tc.txt");
ofstream file13 ("parameters\\dXtys.txt");
ofstream file14 ("parameters\\Zd.txt");
ofstream file15 ("parameters\\dXw.txt");
ofstream file16 ("parameters\\dXh.txt");
ofstream file17 ("parameters\\dXt.txt");
ofstream file18 ("parameters\\dXtz.txt");
ofstream file19 ("parameters\\dXvo.txt");
ofstream file20 ("parameters\\dXvo1proc.txt");
ofstream file21 ("parameters\\dXq.txt");
ofstream file22 ("parameters\\dZw.txt");
ofstream file23 ("parameters\\dXvr.txt");
ofstream file24 ("parameters\\dzvr.txt");
ofstream file25 ("parameters\\dXtzt.txt");
ofstream file26 ("parameters\\Xr.txt");
ofstream file27 ("parameters\\Zr.txt");
ofstream file28 ("parameters\\dXc.txt");
ofstream file29 ("parameters\\Ptys.txt");
ofstream file30 ("parameters\\Yrm.txt");
    ofstream file31 ("parameters\\dt.txt");

```

```

ofstream file89 ("parameters\\Cx(vrtau).html");
file89<<"<center><h2>вод данных</h2></center>\t" <<endl;
file89<<"<center><h3>”глы бросани€</h3></center><table align='center'
border=1>\t" <<endl;
file89<<"<tr>\t<td>vrtau</td>\t<td>Cx</td>\t</tr>" <<endl;

```

```

Form4->StringGrid1->Cells[0][1]="д, м расч.";
Form4->StringGrid1->Cells[0][2]="в, м расч.";
Form4->StringGrid1->Cells[0][3]="б, м расч.";
Form4->StringGrid1->Cells[0][4]="д, м ОПЫТ.";
Form4->StringGrid1->Cells[0][5]="в, м ОПЫТ.";
Form4->StringGrid1->Cells[0][6]="б, м ОПЫТ.";

```

```

/* Form2->StringGrid1->Cells[0][1]="х,м"; //таблица динамики полста

```

```

Form2->StringGrid1->Cells[0][2]="x,MM";
Form2->StringGrid1->Cells[0][3]="x,MMM";
Form2->StringGrid1->Cells[0][4]="x,MMMM";    */
int p;
if (S.teta0n==0){
p=StringGrid1->RowCount-1;
}
else{
/// my choose
p=1;
}

/// reduce to one flow
for(int k=0;k<p;++k)
{
    S.D = StrToFloat(StringGrid1->Cells[_D][k+1]);

    double teta0;
    int p;
    if (S.teta0n==0){
        teta0=StrToFloat(StringGrid1->Cells[_tetao][k+1]);
        S.teta0=(int(teta0)+(teta0-int(teta0))*100/60)*M_PI/180;
    }
    else{
        /// my choose
        teta0=S.teta0n;
        S.teta0=(int(teta0)+(teta0-int(teta0))*100/60)*M_PI/180;
    }

//    double teta0=StrToFloat(StringGrid1->Cells[_tetao][k+1]);
//    S.teta0=(int(teta0)+(teta0-int(teta0))*100/60)*M_PI/180;
if ((StringGrid1->Cells[_i][k+1])!=""){
    S.i=StrToFloat(StringGrid1->Cells[_i][k+1]);
}
else
{

    /*if (g1>0 && g2<p)*/

    /*double proc=(StrToFloat(StringGrid1->Cells[_D][k+1])-
StrToFloat(StringGrid1->Cells[_D][k])/
(StrToFloat(StringGrid1->Cells[_D][k+2])-StrToFloat(StringGrid1-
>Cells[_D][k]));

```

```

        S.i=StrToFloat(StringGrid1->Cells[_i][k])+(StrToFloat(StringGrid1-
>Cells[_i][k+2])-
        StrToFloat(StringGrid1->Cells[_i][k]))*proc;*/
//////////
        double temptet=StrToFloat(StringGrid1->Cells[_tetao][k+1]);;
double pred;

        //Sm.yrr=StrToFloat(Form1->StringGrid2-
>Cells[2][_yrr]);

        //Sm.yrr=S.yrr;

ifstream filem("resources\\152-мм_—√_2—3-к‘540-солный-коэф.dat");
filem>>pred;
        for(int i=0;i<pred;++i){
                filem>>Sm.mtetk[i];
                filem>>Sm.mkform[i];

        }
        filem.close();

//////////f-30

/* if (Form1->ComboBox1->ItemIndex==1)
if (Form1->ComboBox2->ItemIndex==0)
if (Form1->ComboBox3->ItemIndex==0)
if (Form1->ComboBox4->ItemIndex==0){
ifstream filem("resources\\122-мм_√_f30-солный-коэф.dat");
filem>>pred;
        for(int i=0;i<pred;++i){
                filem>>Sm.mtetk[i];
                filem>>Sm.mkform[i];

        }
        filem.close(); }*/

```

```

        for(int i=0;i<(pred);++i){
            if(temptet>=Sm.mtetk[i] &&
temptet<=Sm.mtetk[i+1])
                {
                    double proc=(temptet-Sm.mtetk[i])/
(Sm.mtetk[i+1]-Sm.mtetk[i]);
                    S.i=(Sm.mkform[i]+(Sm.mkform[i+1]-
Sm.mkform[i])*proc);
                }
            /*else{
                if (temptet<Sm.mtetk[0]) S.i=Sm.mkform[0];
            }*/
            }
            if(temptet<Sm.mtetk[0]){
                S.i=Sm.mkform[0];
            }
            if(temptet>Sm.mtetk[3]){
                S.i=Sm.mkform[3];
            }
        }

//////////
        StringGrid1->Cells[_i][k+1]=(FloatToStr(S.i));
    }

    S.calculate(Form1->autoFindCoefFormy->Checked);

    /// not necessary
    if (Form1->autoFindCoefFormy->Checked) {
        StringGrid1->Cells[_i][k+1] = FloatToStr(S.i);
    }

    StringGrid3->Cells[_Xt][k+1]=FloatToStr(okr(S.Xt));
    StringGrid3->Cells[_Yt][k+1]=FloatToStr(okr(S.Yt));
    StringGrid3->Cells[_Zt][k+1]=FloatToStr(okr(S.Zt));
    StringGrid3->Cells[_N][k+1]=FloatToStr(okr(S.N));
    StringGrid3->Cells[_tetar][k+1]=FloatToStr(okr(S.tetar*180/M_PI));
    StringGrid3->Cells[_vr][k+1]=FloatToStr(okr(S.vr));
    S.tr = 0;StringGrid3->Cells[_tr][k+1]=FloatToStr(okr(S.tr));

```

```

StringGrid4->Cells[_Xt4][k+1]=FloatToStr(okr(S.Xt));
StringGrid4->Cells[_Pdelenyi][k+1]=FloatToStr((1500-okr(S.Ptys))+250);
    StringGrid4->Cells[_dPdel][k+1]=FloatToStr(okr(S.dPdel,0));
    StringGrid4->Cells[_dXtys][k+1]=FloatToStr(okr(S.dXtys,0));
// [hide unused] StringGrid4->Cells[_Zd][k+1]=FloatToStr(okr(S.Zd));
StringGrid4->Cells[_dXw][k+1]=FloatToStr(okr(S.dXw));
StringGrid4->Cells[_dXh][k+1]=FloatToStr(okr(S.dXh,0));
StringGrid4->Cells[_dXt][k+1]=FloatToStr(okr(S.dXt,0));
StringGrid4->Cells[_dXtz][k+1]=FloatToStr(okr(S.dXtz,0));
StringGrid4->Cells[_dXvo][k+1]=FloatToStr(okr(S.dXvo,0));
StringGrid4->Cells[_dXq][k+1]=FloatToStr(okr(S.dXq,0));
StringGrid4->Cells[_dZw][k+1]=FloatToStr(okr(S.dZw));
// [hide unused] StringGrid4->Cells[_dXvr][k+1]=FloatToStr(okr(S.dXvr));
// [hide unused] StringGrid4->Cells[_dZvr][k+1]=FloatToStr(okr(S.dZvr));
StringGrid4->Cells[_Xr][k+1]=FloatToStr(okr(S.Xr));
StringGrid4->Cells[_Yrm][k+1]=FloatToStr(okr(S.Yrm));
StringGrid4->Cells[_Zr][k+1]=FloatToStr(okr(S.Zr));
StringGrid4->Cells[_dXc][k+1]=FloatToStr(okr(S.dXc));

StringGrid4->Cells[_teta0][k+1]=StringGrid1->Cells[_teta0][k+1];

StringGrid4->Cells[_Pri][k+1]=FloatToStr(okr(S.Ptys));
StringGrid4->Cells[_tetac][k+1]=FloatToStr(okr(S.tetac*180/M_PI)*-1);
StringGrid4->Cells[_vc][k+1]=FloatToStr(okr(S.vc));
StringGrid4->Cells[_tc][k+1]=FloatToStr(okr(S.tc));
StringGrid4->Cells[_Vd][1+k]=FloatToStr(okr(S.Vd));

StringGrid4->Cells[_Ybul][1+k]=StringGrid1->Cells[_ybul][k+1];
// [hide unused] StringGrid4->Cells[_Vv][1+k]=FloatToStr(okr(S.Vv));
StringGrid4->Cells[_Vb][1+k]=FloatToStr(okr(S.Vb));
// [hide unused] StringGrid4->Cells[_Vdteta0]
[1+k]=FloatToStr(okr(S.Vdteta0));
// [hide unused] StringGrid4->Cells[_Vdv0][1+k]=FloatToStr(okr(S.Vdv0));
// [hide unused] StringGrid4->Cells[_Vdc][1+k]=FloatToStr(okr(S.Vdc));
// [hide unused] StringGrid4->Cells[_teta0][k+1]=FloatToStr(teta0);

//Form2->Series1->AddXY(S.Xt,S.Vd);
//Form2->Series2->AddXY(S.Xt,S.Vdteta0);
//Form2->Series3->AddXY(S.Xt,S.Vdv0);
//Form2->Series4->AddXY(S.Xt,S.Vdc);

Form3->Series1->AddXY(S.Xt,S.Vd);
Form3->Series2->AddXY(S.Xt,S.Vv);
Form3->Series3->AddXY(S.Xt,S.Vb);

```

```

Form4->StringGrid1->Cells[k+1][0]=FloatToStr(okr(S.Xt));
Form4->StringGrid1->Cells[k+1][1]=FloatToStr(okr(S.Vd));
Form4->StringGrid1->Cells[k+1][2]=FloatToStr(okr(S.Vv));
Form4->StringGrid1->Cells[k+1][3]=FloatToStr(okr(S.Vb));

```

```

for (int vi=0;vi<28;vi++)
if (okr(S.Xt)==vd_opitnie[0][vi])
{
Form4->StringGrid1->Cells[k+1][4]=vd_opitnie[1][vi];
Form4->StringGrid1->Cells[k+1][5]=vv_opitnie[1][vi];
Form4->StringGrid1->Cells[k+1][6]=vb_opitnie[1][vi];

}
//динамика полЄта

```

```
//конец динамики полЄта
```

```

file1<<okr(S.Xt)<<"\t"<<okr(S.Vd)<<endl;
file2<<okr(S.Xt)<<"\t"<<S.Vdteta0<<endl;
file3<<okr(S.Xt)<<"\t"<<S.Vdv0<<endl;
file4<<okr(S.Xt)<<"\t"<<S.Vdc<<endl;
file5<<okr(S.Xt)<<"\t"<<okr(S.Vv)<<endl;
file6<<okr(S.Xt)<<"\t"<<okr(S.Vb)<<endl;
file7<<okr(S.Xt)<<endl;
file8<<okr(S.Xt)<<"\t"<<okr(S.Yt)<<endl; ("parameters\Yt.txt");
file9<<okr(S.Xt)<<"\t"<<okr(S.Zt)<<endl; ("parameters\Zt.txt");
file10<<okr(S.Xt)<<"\t"<<okr(S.tetac*180/M_PI)<<endl; ("parameters\
\tetac.txt");
file11<<okr(S.Xt)<<"\t"<<okr(S.vc)<<endl; ("parameters\vc.txt");
file12<<okr(S.Xt)<<"\t"<<okr(S.tc)<<endl; ("parameters\tc.txt");
file13<<okr(S.Xt)<<"\t"<<okr(S.dXtys)<<endl; ("parameters\dXtys.txt");
file14<<okr(S.Xt)<<"\t"<<okr(S.Zd)<<endl; ("parameters\Zd.txt");
file15<<okr(S.Xt)<<"\t"<<okr(S.dXw)<<endl; ("parameters\dXw.txt");
file16<<okr(S.Xt)<<"\t"<<okr(S.dXh)<<endl; ("parameters\dXh.txt");
file17<<okr(S.Xt)<<"\t"<<okr(S.dXt)<<endl; ("parameters\dXt.txt");
file18<<okr(S.Xt)<<"\t"<<okr(S.dXtz)<<endl; ("parameters\dXtz.txt");
file19<<okr(S.Xt)<<"\t"<<okr(S.dXvo)<<endl; ("parameters\dXvo.txt");
file20<<okr(S.Xt)<<"\t"<<okr(S.dXvo1proc)<<endl; ("parameters\
\dXvo1proc.txt");
file21<<okr(S.Xt)<<"\t"<<okr(S.dXq)<<endl; ("parameters\dXq.txt");
file22<<okr(S.Xt)<<"\t"<<okr(S.dZw)<<endl; ("parameters\dZw.txt");
file23<<okr(S.Xt)<<"\t"<<okr(S.dXvr)<<endl; ("parameters\dXvr.txt");
file24<<okr(S.Xt)<<"\t"<<okr(S.dZvr)<<endl; ("parameters\dZvr.txt");
file25<<okr(S.Xt)<<"\t"<<okr(S.dXtzt)<<endl; ("parameters\dXtzt.txt");

```

```

file26<<okr(S.Xt)<<"\t"<<okr(S.Xr)<<endl; ("parameters\\Xr.txt");
file27<<okr(S.Xt)<<"\t"<<okr(S.Zr)<<endl; ("parameters\\Zr.txt");
file28<<okr(S.Xt)<<"\t"<<okr(S.dXc)<<endl; ("parameters\\dXc.txt");
file29<<okr(S.Xt)<<"\t"<<okr(S.Ptys)<<endl; ("parameters\\Ptys.txt");
file30<<okr(S.Xt)<<"\t"<<okr(S.Yrm)<<endl; ("parameters\\Yrm.txt");
file31<<okr(S.deltat1)<<"<<okr(S.deltat2)<<endl; ("parameters\\dt.txt");
}
    file89<<S.Cx.c_str()<<endl; ("parameters\\current_f(Mtau).txt");
    temps=/*temps+*/"<td>"+S.Cx+"</td>\t</tr>";
S.vrtau="";
S.Cx="";
file1.close();
file2.close();
file3.close();
file4.close();
file5.close();
file6.close();
file7.close();
file8.close();
file9.close();
file10.close();
file11.close();
file12.close();
file13.close();
file14.close();
file15.close();
file16.close();
file17.close();
file18.close();
file19.close();
file20.close();
file21.close();
file22.close();
file23.close();
file24.close();
file25.close();
file26.close();
file29.close();
file30.close();
    file31.close();
file89<<"</table>\t"<<endl;
file89.close();

```

```
/* outdata_[Xt_]=S.Xt;
```

```

outdata_[Zt_]=S.Zt;
outdata_[dXtys_]=S.dXtys;
outdata_[Zd_]=S.Zd/100;
outdata_[dXw_]=S.dXw;
outdata_[dXh_]=S.dXh;
outdata_[dXt_]=S.dXt;
outdata_[dXtz_]=S.dXtz;
outdata_[dXvo_]=S.dXvo;
outdata_[dXq_]=S.dXq;
outdata_[dZw_]=S.dZw/100;
outdata_[dXtzt_]=0;
outdata_[Xr_]=S.Xr;
outdata_[Zr_]=S.Zr;

ofstream file("data2.txt");
for(int i=0;i<OUTDATANUM;++i)
file<<outdata_[i]<<endl;
file<<S.dXvo1proc<<endl;
file.close(); */
PageControl1->ActivePageIndex=1;
}
//-----
void __fastcall TForm1::Button5Click(TObject *Sender)
{
    int RC=StrToInt(Edit1->Text);
    if(RC>0)
        StringGrid1->RowCount=RC+1;
}
//-----

void __fastcall TForm1::Button6Click(TObject *Sender)
{
    Form4->Show();
}
//-----

void __fastcall TForm1::Button7Click(TObject *Sender)
{
    Form3->Show();
}
//-----

void __fastcall TForm1::Save1Click(TObject *Sender)

```



```

{
    ofstream file99 ("Rezult.html");

    char dateandtime[50];           //датавремѐ
    time_t seconds = time(NULL);
    tm* timeinfo = localtime(&seconds);
    char* format = "%B, %d, %Y года. %I:%M:%S ";
    setlocale(LC_ALL, "");
    strftime(dateandtime, 50, format, timeinfo);
    file99<<"<p><i><font size=\\"3\">"<<dateandtime<<"</font></i></p>"<<endl; //
дата и времѐ

    file99<<"<p align=\\"center\"><b><font size=\\"4\">—f” ÷ентра масс</font></
b></p>"<<endl;
    file99<<"<p align=\\"center\"><b><font size=\\"4\">“аблицы стрельбы</font></
b></p>"<<endl;
    file99<<"<p align=\\"center\"><b><font size=\\"4\">"<<Form1->Edit2-
>Text.c_str()<<" , запѐд "<<Form1->Edit4->Text.c_str()<<" , снарѐд
"<<Form1->Edit3->Text.c_str()<<"</font></b></p>"<<endl;

    //if (Form1->StringGrid2->Cells[0][4]==0){
    file99<<"<center><h2>”вод данных</h2></center><table align='center'
border=1>\t"<<endl;
    file99<<"<center><h3>”глы бросаниѐ</h3></center>"<<endl;
    file99<<"<table align='center' border=1>"<<endl;
    for(int i=0;i<=Form1->StringGrid1->RowCount-1;i++){
        file99<<"<tr>"<<endl;
        file99<<"<td>"<<(Form1->StringGrid1->Cells[0][i].c_str())<<"</td><td>"
<<(Form1->StringGrid1->Cells[1][i].c_str())<<"</td><td>"
<<(Form1->StringGrid1->Cells[2][i].c_str())<<"</td>"<<endl;
        file99<<"</tr>"<<endl;
    }
    file99<<"</table>"<<endl;
    file99<<"<br>"<<endl;
    file99<<"<br>"<<endl;
    //}

    file99<<"<center><h3>«начениѐ констант длѐ расчѐта табличной
траектории</h3></center>"<<endl;
    file99<<"<table align='center' border=1>"<<endl;
    for(int i=0;i<=Form1->StringGrid2->RowCount-1;i++){
        file99<<"<tr>"<<endl;
        file99<<"<td>"<<(Form1->StringGrid2->Cells[0][i].c_str())<<"</td><td>"
<<(Form1->StringGrid2->Cells[1][i].c_str())<<"</td><td>"
<<(Form1->StringGrid2->Cells[2][i].c_str())<<"</td>"<<endl;

```

```

    file99<<"</tr>"<<endl;
}
file99<<"</table>"<<endl;
file99<<"<br>"<<endl;
file99<<"<br>"<<endl;

file99<<"<center><h3>“абличнаѹ траекториѹ</h3></center>"<<endl;
file99<<"<table align='center' border=1>"<<endl;
for(int i=0;i<=Form1->StringGrid3->RowCount-1;i++){
    file99<<"<tr>"<<endl;
    file99<<"<td>"
    <<(Form1->StringGrid3->Cells[0][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid3->Cells[1][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid3->Cells[2][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid3->Cells[3][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid3->Cells[4][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid3->Cells[5][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid3->Cells[6][i].c_str())<<"</td>"<<endl;
    file99<<"</tr>"<<endl;
}
file99<<"</table>"<<endl;
file99<<"<br>"<<endl;
file99<<"<br>"<<endl;

file99<<"<center><h3>сопpавочные графы</h3></center>"<<endl;
/* file99<<"“ѹта0, град,ммсс"<<"\t"
<<"’т, м"<<"\t"
<<"s, тыс"<<"\t"
<<"dXтыс, м"<<"\t"
<<"Z, тыс"<<"\t"
<<"dXw, м"<<"\t"
<<"dXс, м"<<"\t"
<<"dXн, м"<<"\t"
<<"dXт, м"<<"\t"
<<"dXтз, м"<<"\t"
<<"dXvo, м"<<"\t"
<<"dXq, м"<<"\t"
<<"dZw, тыс"<<"\t"
<<"dXvr, м"<<"\t"
<<"dZvr, тыс"<<"\t"
<<"’р, м"<<"\t"
<<"Ys, м"<<"\t"
<<"Zp, м"<<"\t"
<<"¬д_ѹта0, м"<<"\t"
<<"¬д_V0, м"<<"\t"

```

```

<<"\u0414_c, M"<<"\t"
<<"\u0414, M"<<"\t"
<<"\u0411, M"<<"\t"
<<"\u0416, M"<<endl;*/
file99<<"<table align='center' border=1>"<<endl;

```

```

for(int i=0;i<=0;i++){
    file99<<"<tr>"<<endl;
    file99<<"<td>"
    <<(Form1->StringGrid4->Cells[0][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[1][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[2][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[3][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[4][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[5][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[6][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[7][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[8][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[9][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[10][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[11][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[12][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[13][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[14][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[15][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[16][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[17][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[18][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[19][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[20][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[21][i].c_str())<<"</td><td>"
    <<(Form1->StringGrid4->Cells[22][i].c_str())<<"</td>"<<endl;
    file99<<"</tr>"<<endl;
}
file99<<"<tr>"<<endl;
for(int i=1;i<=23;i++){
    file99<<"<td>"<<(i)<<"</td>"<<endl;
}
file99<<"</tr>"<<endl;

for(int i=1;i<=Form1->StringGrid1->RowCount-1;i++){
    file99<<"<tr>"<<endl;
    file99<<"<td>"
    <<(Form1->StringGrid4->Cells[0][i].c_str())<<"</td><td>"

```

```

<<(Form1->StringGrid4->Cells[1][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[2][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[3][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[4][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[5][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[6][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[7][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[8][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[9][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[10][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[11][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[12][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[13][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[14][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[15][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[16][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[17][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[18][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[19][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[20][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[21][i].c_str())<<"</td><td>"
<<(Form1->StringGrid4->Cells[22][i].c_str())<<"</td>"<<endl;
file99<<"</tr>"<<endl;
}
file99<<"</table>"<<endl;
//file99<<(Form1->StringGrid4->Cells[0][1])<<"</td><td>"<<(Form1-
>StringGrid4->Cells[1][1])<<endl;
//file99<<okr(S.Xt)<<"</td><td>"<<okr(S.dXc)<<endl;

file99<<"<center><h3>яэродинамика</h3></center><table align='center'
border=1>\t"<<endl;
file99<<"<tr>\t<td>vrtau</td>\t<td>Cx</td>\t</tr>"<<endl;
file99<<temps.c_str()<<endl; ("parameters\\current_f(Mtau).txt");

file99<<"</table>"<<endl;

file99.close();
}

void __fastcall TForm1::Button8Click(TObject *Sender)
{
float table_coef_row_one[6] = {0, 1000, 2000, 0, -1000, -2000};

```

```
float table_coef_col_one[33] = {350, 370, 390, 410, 430, 450, 470, 490, 510, 530,
550, 570, 590, 610, 630, 650, 670, 690, 710, 730, 750, 770, 790, 810, 830, 850, 870,
890, 910, 930, 950, 970, 990};
```

```
float table_coef[33][6] = {
    {0.15, 0.14, 0.14, 0, 0, 0},
    {0.17, 0.16, 0.16, 0, 0, 0},
    {0.19, 0.18, 0.18, 0, 0, 0},
    {0.21, 0.2, 0.2, 0, 0, 0.1},
    {0.23, 0.22, 0.22, 0, 0, 0.12},
    {0.25, 0.24, 0.24, 0, 0, 0.14},
    {0.27, 0.27, 0.26, 0, 0.16, 0.16},
    {0.3, 0.29, 0.29, 0, 0.18, 0.18},
    {0.32, 0.32, 0.31, 0, 0.21, 0.21},
    {0.35, 0.35, 0.34, 0, 0.23, 0.23},
    {0.38, 0.38, 0.37, 0, 0.26, 0.26},
    {0.42, 0.41, 0.41, 0, 0.29, 0.29},
    {0.45, 0.44, 0.44, 0, 0.32, 0.32},
    {0.5, 0.5, 0.49, 0.36, 0.35, 0.35},
    {0.55, 0.54, 0.53, 0.41, 0.4, 0.4},
    {0.6, 0.59, 0.59, 0.45, 0.45, 0.44},
    {0.6, 0.6, 0.59, 0.5, 0.5, 0.49},
    {0.74, 0.73, 0.72, 0.5, 0.5, 0.5},
    {0.81, 0.8, 0.79, 0.63, 0.63, 0.62},
    {0.9, 0.88, 0.88, 0.7, 0.7, 0.69},
    {0.99, 0.98, 0.98, 0.78, 0.78, 0.77},
    {1.11, 1.1, 1.09, 0.88, 0.87, 0.87},
    {1.24, 1.23, 1.23, 0.98, 0.98, 0.98},
    {1.4, 1.39, 1.39, 1.11, 1.11, 1.1},
    {1.59, 1.58, 1.58, 1.26, 1.26, 1.26},
    {1.82, 1.82, 1.82, 1.44, 1.44, 1.44},
    {2.11, 2.11, 2.12, 1.66, 1.67, 1.67},
    {2.48, 2.5, 2.51, 1.94, 1.95, 1.96},
    {2.97, 3, 3.03, 2.29, 2.31, 2.33},
    {3.73, 3.81, 3.9, 2.76, 2.79, 2.83},
    {5.16, 5.33, 5.54, 3.4, 3.44, 3.49},
    {0, 0, 0, 4.08, 4.16, 4.24},
    {0, 0, 0, 5.09, 5.22, 5.38}
};
```

```
float
```

```
Alpha_o,Xp,dv=0,Zp,Tetav_1,Tetav_2,Pr_tys,Pr_tys_1,Pr_tys_2,Pr_del,delta_dv,delt
a_h_c,delta_Ph,K_Ph,P_vr,d_cv;
```

```
int temp2;
```

```
for(temp2 = 2; StrToFloat(StringGrid7->Cells[2][4]) > StrToFloat(StringGrid1-
>Cells[0][temp2]); temp2++){
```

```

    if((StrToFloat(StringGrid1->Cells[0][temp2])+StrToFloat(StringGrid1->Cells[0][temp2-1]))/2 < StrToFloat(StringGrid7->Cells[2][4])){
        continue;
    }
    else{
        temp2=temp2-1;
    }
}

```

```

std::string s = (StringGrid1->Cells[1][temp2-1]).c_str();
std::replace(s.begin(), s.end(), ',', '.');

```

```

Alpha_o=atof(s.data());

```

```

Xp=StrToFloat(StringGrid4->Cells[17][temp2-1]);
Zp=StrToFloat(StringGrid4->Cells[19][temp2-1]);
Zp=okr(Zp);
StringGrid7->Cells[6][2]=Xp;
StringGrid7->Cells[6][3]=Zp;
int i;
float tmp;
for(i=1;i<(StringGrid1->RowCount)-1;++i){
    if (StrToFloat(StringGrid1->Cells[0][i]) <= Xp && StrToFloat(StringGrid1->Cells[0][i+1]) >= Xp){
        Tetav_1 = StrToFloat(StringGrid1->Cells[1][i]);
        Tetav_2 = StrToFloat(StringGrid1->Cells[1][i+1]);
        break;
        // Tetav_1 = (((Xp - StrToFloat(StringGrid1->Cells[0][i]))*(StrToFloat(StringGrid1->Cells[1][i+1])-StrToFloat(StringGrid1->Cells[1][i])))/
        (StrToFloat(StringGrid1->Cells[0][i+1]) - StrToFloat(StringGrid1->Cells[0][i])))
        +StrToFloat(StringGrid1->Cells[1][i]);
    }
}
// StringGrid7->Cells[6][1]=((float)(int)(Tetav_1 * 100)) / 100;
StringGrid7->Cells[6][1]=Tetav_1;
// StringGrid7->Cells[7][1]=((float)(int)(Tetav_2 * 100)) / 100;
tmp=Xp-StrToFloat(StringGrid1->Cells[0][i]);
StringGrid7->Cells[7][1]=tmp;
StringGrid7->Cells[6][2]=((float)(int)(Tetav_1 * 1)) / 1;
StringGrid7->Cells[6][3]=((((float)(int)(Tetav_1 * 100)) / 100)-((float)(int)(Tetav_1 * 1)) / 1)*100;

```

```

Pr_tys_1=(((float)(int)(Tetav_1 * 1)) / 1)*60+((((float)(int)(Tetav_1 * 100)) /
100)-(((float)(int)(Tetav_1 * 1)) / 1))*100;
Pr_tys_2=(((float)(int)(Tetav_2 * 1)) / 1)*60+((((float)(int)(Tetav_1 * 100)) /
100)-(((float)(int)(Tetav_1 * 1)) / 1))*100;
Pr_tys=(((float)(int)(Pr_tys_1 * 1)) / 1)-(((float)(int)(Pr_tys_2 * 1)) / 1);
StringGrid7->Cells[6][2]=Tetav_2;
// StringGrid7->Cells[7][2]=Pr_tys;
StringGrid7->Cells[7][3]=Pr_tys_1;
StringGrid7->Cells[7][4]=Pr_tys_2;
StringGrid7->Cells[7][5]=(((float)(int)(Tetav_1 * 1)) / 1);
StringGrid7->Cells[7][6]=(((float)(int)(Tetav_1 * 1)) / 1)*60;
StringGrid7->Cells[7][7]=((((float)(int)(Tetav_1 * 100)) / 100)-(((float)(int)
(Tetav_1 * 1)) / 1))*100;

```

```

Pr_tys=(((float)(int)(Pr_tys * 1)) / 1);
StringGrid7->Cells[6][4]=Pr_tys;
Pr_del=(1500-Pr_tys)+250;
StringGrid7->Cells[6][5]=Pr_del;
delta_dv=(Zp*955)/StrToFloat(StringGrid7->Cells[2][4]);
StringGrid7->Cells[6][6]=delta_dv;
int temp,temp3;
for(temp = 0; Pr_del > table_coef_col_one[temp]; temp++){
    if(table_coef_col_one[temp]+table_coef_col_one[temp-1]/2 < Pr_del){
        temp++;
        continue;
    }
    else{
        temp++;
        temp=temp-1;
    }
}
delta_h_c=StrToFloat(StringGrid7->Cells[2][7]);
for(temp3 = 0; delta_h_c > table_coef_row_one[temp3]; temp3++){
    if((table_coef_row_one[temp3]+table_coef_row_one[temp3-1])/2 < delta_h_c){
        continue;
    }
    else{
        temp3=temp3-1;
    }
}
K_Ph=table_coef[temp][temp3];
delta_Ph=0.1*K_Ph*delta_h_c;

```

```

ofstream foutf;
foutf.open("log.txt");
foutf << Alpha_o;
foutf.close();

// StringGrid7->Cells[7][1]=delta_Ph;
P_vr=Pr_del+delta_Ph;
StringGrid12->Cells[0][1]=((float)(int)(P_vr * 1)) / 1;
d_cv=okr(StrToFloat(StringGrid7->Cells[2][5])*100+delta_dv);
StringGrid12->Cells[1][1]=((float)(int)(d_cv * 100)) / 100;
}
//-----

```

```

double round (double x) {
return ((x - floor(x)) >= 0.5) ? ceil(x) : floor(x);
}

```

```

void __fastcall TForm1::Button9Click(TObject *Sender)
{

```

```

//ShowMessage("Opa");
//float x = StrToFloat(StringGrid14->Cells[3][2]);
// StringGrid15->Cells[3][1] = x;

```

```

// rabotaet -----

```

```

float ybul = StrToFloat(StringGrid7->Cells[2][8])*0.01;
float alphaTsili = StrToFloat(StringGrid7->Cells[2][6]);
float hmc = StrToFloat(StringGrid13->Cells[2][2]);
float dHm = StrToFloat(StringGrid13->Cells[3][2]);
float dTauOsv = StrToFloat(StringGrid13->Cells[4][2]);
float dhm = 0;
if (dHm*0.01 >= 5)
{
string out_string;
stringstream sss;
sss << dHm;
out_string = sss.str();
dhm = -atof(out_string.substr(1).c_str());
}
else
{
string out_string;
stringstream sss;
sss << dHm;

```



```

    out_string = sss.str();
    dhm = atof(out_string.substr(1).c_str());
}

    int Ym [9] = { 2, 4, 8, 12, 16, 20, 24, 30, 40 };
    float dTauY[9] = { StrToFloat(StringGrid14->Cells[1]
[2]),StrToFloat(StringGrid14->Cells[1][3]),StrToFloat(StringGrid14->Cells[1][4]),
    StrToFloat(StringGrid14->Cells[1][5]), StrToFloat(StringGrid14-
>Cells[1][6]),StrToFloat(StringGrid14->Cells[1][7]),
    StrToFloat(StringGrid14->Cells[1][8]),StrToFloat(StringGrid14-
>Cells[1][9]), StrToFloat(StringGrid14->Cells[1][10])};
    float AlphaWY[9] = {StrToFloat(StringGrid14->Cells[2]
[2]),StrToFloat(StringGrid14->Cells[2][3]),StrToFloat(StringGrid14->Cells[2][4]),
    StrToFloat(StringGrid14->Cells[2][5]), StrToFloat(StringGrid14-
>Cells[2][6]),StrToFloat(StringGrid14->Cells[2][7]),
    StrToFloat(StringGrid14->Cells[2][8]),StrToFloat(StringGrid14-
>Cells[2][9]), StrToFloat(StringGrid14->Cells[2][10]) };
    float Wy[9] = { StrToFloat(StringGrid14->Cells[3]
[2]),StrToFloat(StringGrid14->Cells[3][3]),StrToFloat(StringGrid14->Cells[3][4]),
    StrToFloat(StringGrid14->Cells[3][5]),StrToFloat(StringGrid14-
>Cells[3][6]),StrToFloat(StringGrid14->Cells[3][7]),
    StrToFloat(StringGrid14->Cells[3][8]),StrToFloat(StringGrid14-
>Cells[3][9]),StrToFloat(StringGrid14->Cells[3][10]) };
    int YmResult, YmResultIndex;
    int temp;
    for (temp = 0; ybul > Ym[temp]; temp++)
    {
    }
    if ((Ym[temp] + Ym[temp - 1]) / 2 < ybul)
    {

        YmResult = Ym[temp];
        YmResultIndex = temp;

    }
    else
    {

        YmResult = Ym[temp - 1];
        YmResultIndex = temp;
    }
    int dTauY_ = dTauY[YmResultIndex];
    double dTauY_output;
    if (dTauY_*0.1 >= 5)

```

```

{
    string out_string;
    stringstream sss;
    sss << dTauY_;
    out_string = sss.str();
    dTauY_output = -atof(out_string.substr(1).c_str());
}
else
{
    string out_string;
    stringstream sss;
    sss << dTauY_;
    out_string = sss.str();
    dTauY_output = atof(out_string.substr(1).c_str());
}
int AlphaWy = AlphaWY[YmResultIndex];
int WY = Wy[YmResultIndex];
float hb = StrToFloat(StringGrid14->Cells[3][8]);
int dHb = dhm + (hmc - hb) / 10;

double AwPK = (alphaTsili - AlphaWy)*100;
double Aw = (AwPK*3.6) / 60;
double Wx = round(-WY * cos( M_PI *Aw/180));
double Wz = round(WY * sin(M_PI *Aw / 180));

StringGrid15->Cells[0][1] = dHb;
StringGrid15->Cells[1][1] = dTauY_output;
StringGrid15->Cells[2][1] = Wx;
StringGrid15->Cells[3][1] = Wz;

//double Wz = 50.5703;

// StringGrid15->Cells[3][1] = round(Wz);
}
//-----

void __fastcall TForm1::Button10Click(TObject *Sender)
{
    // передача метеосреднего
    StringGrid2->Cells[2][_dP]=StringGrid15->Cells[0][1];
    StringGrid2->Cells[2][_dtau]=StringGrid15->Cells[1][1];
    StringGrid2->Cells[2][_Wx]=StringGrid15->Cells[2][1];
    StringGrid2->Cells[2][_Wz]=StringGrid15->Cells[3][1];
}

```

```
for(int i=1;i<StringGrid1->RowCount;++i)
point2comma(StringGrid1->Cells[2][i].c_str());
```

```
for(int i=0;i<StringGrid2->RowCount;++i)
point2comma(StringGrid2->Cells[2][i].c_str());
```

```
TSDUSolver S;
TSDU Sm;
```

```
StringGrid3->RowCount=StringGrid1->RowCount;
StringGrid4->RowCount=StringGrid1->RowCount;
Form4->StringGrid1->ColCount=StringGrid1->RowCount;
```

```
//чтение исходных данных, общих для всех дальностей
S.v0=StrToFloat(StringGrid2->Cells[2][_vo]);
S.dP=StrToFloat(StringGrid2->Cells[2][_dP]);
S.dtau=StrToFloat(StringGrid2->Cells[2][_dtau]);
S.Wx=StrToFloat(StringGrid2->Cells[2][_Wx]);
S.Wz=StrToFloat(StringGrid2->Cells[2][_Wz]);
S.Tz=StrToFloat(StringGrid2->Cells[2][_Tz]);
S.dq=StrToFloat(StringGrid2->Cells[2][_dq]);
S.dvo=StrToFloat(StringGrid2->Cells[2][_dvo]);
S.d=StrToFloat(StringGrid2->Cells[2][_d]);
S.qo=StrToFloat(StringGrid2->Cells[2][_qo]);
S.tauoN=StrToFloat(StringGrid2->Cells[2][_tauoN]);
S.go=StrToFloat(StringGrid2->Cells[2][_go]);
S.R=StrToFloat(StringGrid2->Cells[2][_R]);
S.ltz=StrToFloat(StringGrid2->Cells[2][_ltz]);
S.lq=StrToFloat(StringGrid2->Cells[2][_lq]);
S.PoN=StrToFloat(StringGrid2->Cells[2][_PoN]);
S.dteta=StrToFloat(StringGrid2->Cells[2][_dteta]);
S.eta=StrToFloat(StringGrid2->Cells[2][_eta]);
S.L=StrToFloat(StringGrid2->Cells[2][_L]);
S.A=StrToFloat(StringGrid2->Cells[2][_A]);
S.rteta0=StrToFloat(Form1->StringGrid2->Cells[2][_rteta0]);
S.rv0=StrToFloat(Form1->StringGrid2->Cells[2][_rv0]);
S.rc=StrToFloat(Form1->StringGrid2->Cells[2][_rc]);
S.rw=StrToFloat(Form1->StringGrid2->Cells[2][_rw]);
S.rz=StrToFloat(Form1->StringGrid2->Cells[2][_rz]);
S.Bsh=StrToFloat(Form1->StringGrid2->Cells[2][_Bsh]);
S.ag=StrToFloat(Form1->StringGrid2->Cells[2][_ag]);
S.Omegaz=StrToFloat(Form1->StringGrid2->Cells[2][_Omegaz]);
S.Rz=StrToFloat(Form1->StringGrid2->Cells[2][_Rz]);
S.x0=StrToFloat(Form1->StringGrid2->Cells[2][_Xn]);
```

```

S.y0=StrToFloat(Form1->StringGrid2->Cells[2][_Yn]);
S.z0=StrToFloat(Form1->StringGrid2->Cells[2][_Zn]);
S.Xtn=StrToFloat(Form1->StringGrid2->Cells[2][_Xtn]);
S.teta0n=StrToFloat(Form1->StringGrid2->Cells[2][_teta0n]);
S.Kvrs=StrToFloat(Form1->StringGrid2->Cells[2][_Kvr]);
S.tn=StrToFloat(Form1->StringGrid2->Cells[2][_tn]);
S.tauaN=StrToFloat(Form1->StringGrid2->Cells[2][_tauaN]);
S.I1N=StrToFloat(Form1->StringGrid2->Cells[2][_I1N]);
S.wo=StrToFloat(Form1->StringGrid2->Cells[2][_w0]);
S.K1=StrToFloat(Form1->StringGrid2->Cells[2][_K1]);
S.K2=StrToFloat(Form1->StringGrid2->Cells[2][_K2]);
S.deltat1=StrToFloat(Form1->StringGrid2->Cells[2][_dt1]); //boop
S.deltat2=StrToFloat(Form1->StringGrid2->Cells[2][_dt2]); //boop
S.dovorot=StrToFloat(Form1->StringGrid2->Cells[2][_dovorot]);
yrr=StrToFloat(Form1->StringGrid2->Cells[2][_yrr]);

```

```

Form3->Series1->Clear();
Form3->Series2->Clear();
Form3->Series3->Clear();

```

```
//-----
```

```

ofstream file1 ("parameters\Vd.txt");
ofstream file2 ("parameters\Vdteta0.txt");
ofstream file3 ("parameters\Vdv0.txt");
ofstream file4 ("parameters\Vdc.txt");
ofstream file5 ("parameters\Vv.txt");
ofstream file6 ("parameters\Vb.txt");
ofstream file7 ("parameters\Xt.txt");
ofstream file8 ("parameters\Yt.txt");
ofstream file9 ("parameters\Zt.txt");
ofstream file10 ("parameters\tetac.txt");
ofstream file11 ("parameters\vc.txt");
ofstream file12 ("parameters\tc.txt");
ofstream file13 ("parameters\dXtys.txt");
ofstream file14 ("parameters\Zd.txt");
ofstream file15 ("parameters\dXw.txt");
ofstream file16 ("parameters\dXh.txt");
ofstream file17 ("parameters\dXt.txt");
ofstream file18 ("parameters\dXtz.txt");
ofstream file19 ("parameters\dXvo.txt");
ofstream file20 ("parameters\dXvo1proc.txt");
ofstream file21 ("parameters\dXq.txt");

```

```

ofstream file22 ("parameters\\dZw.txt");
ofstream file23 ("parameters\\dXvr.txt");
ofstream file24 ("parameters\\dzvr.txt");
ofstream file25 ("parameters\\dXtzt.txt");
ofstream file26 ("parameters\\Xr.txt");
ofstream file27 ("parameters\\Zr.txt");
ofstream file28 ("parameters\\dXc.txt");
ofstream file29 ("parameters\\Ptys.txt");
ofstream file30 ("parameters\\Yrm.txt");
    ofstream file31 ("parameters\\dt.txt");

```

```

ofstream file89 ("parameters\\Cx(vrtau).html");
file89<<"<center><h2>вод данных</h2></center>\t"<<endl;
file89<<"<center><h3>”глы бросани€</h3></center><table align='center'
border=1>\t"<<endl;
file89<<"<tr>\t<td>vrtau</td>\t<td>Cx</td>\t</tr>"<<endl;

```

```

Form4->StringGrid1->Cells[0][1]="¬д, м расч.";
Form4->StringGrid1->Cells[0][2]="¬в, м расч.";
Form4->StringGrid1->Cells[0][3]="¬б, м расч.";
Form4->StringGrid1->Cells[0][4]="¬д, м ОПЫТ.";
Form4->StringGrid1->Cells[0][5]="¬в, м ОПЫТ.";
Form4->StringGrid1->Cells[0][6]="¬б, м ОПЫТ.";

```

```

int p;
if (S.teta0n==0){
p=StringGrid1->RowCount-1;
}
else{
/// my choose
p=1;
}

/// reduce to one flow
for(int k=0;k<p;++k)
{
    S.D = StrToFloat(StringGrid1->Cells[_D][k+1]);

    double teta0;
    int p;
    if (S.teta0n==0){
        teta0=StrToFloat(StringGrid1->Cells[_tetao][k+1]);
        S.teta0=(int(teta0)+(teta0-int(teta0))*100/60)*M_PI/180;
    }
}

```

```

else{
    // my choose
    teta0=S.teta0n;
    S.teta0=(int(teta0)+(teta0-int(teta0))*100/60)*M_PI/180;
}

if ((StringGrid1->Cells[_i][k+1])!=""){
    S.i=StrToFloat(StringGrid1->Cells[_i][k+1]);
}
else
{

//////////

    double temptet=StrToFloat(StringGrid1->Cells[_tetao][k+1]);;
    double pred;

    //Sm.yrr=StrToFloat(Form1->StringGrid2-
>Cells[2][_yrr]);
    //Sm.yrr=S.yrr;

    ifstream filem("resources\\152-мм_—√_2—3-к'540-солный-коэф.dat");
    filem>>pred;
    for(int i=0;i<pred;++i){
        filem>>Sm.mtetk[i];
        filem>>Sm.mkform[i];
    }
    filem.close();

    for(int i=0;i<(pred);++i){
        if(temptet>=Sm.mtetk[i] &&
temptet<=Sm.mtetk[i+1])
        {

```

```

                                double proc=(temptet-Sm.mtetk[i]/
(Sm.mtetk[i+1]-Sm.mtetk[i]);
                                S.i=(Sm.mkform[i]+(Sm.mkform[i+1]-
Sm.mkform[i])*proc);
                                }
                                }
                                if(temptet<Sm.mtetk[0]){
                                    S.i=Sm.mkform[0];
                                }
                                if(temptet>Sm.mtetk[3]){
                                    S.i=Sm.mkform[3];
                                }
                                }

```

```
//////////
```

```

    StringGrid1->Cells[_i][k+1]=(FloatToStr(S.i));
}

S.calculate(Form1->autoFindCoefFormy->Checked);

```

```
/// not necessary
```

```

if (Form1->autoFindCoefFormy->Checked) {
    StringGrid1->Cells[_i][k+1] = FloatToStr(S.i);
}

```

```

StringGrid3->Cells[_Xt][k+1]=FloatToStr(okr(S.Xt));
StringGrid3->Cells[_Yt][k+1]=FloatToStr(okr(S.Yt));
StringGrid3->Cells[_Zt][k+1]=FloatToStr(okr(S.Zt));
StringGrid3->Cells[_N][k+1]=FloatToStr(okr(S.N));
StringGrid3->Cells[_tetar][k+1]=FloatToStr(okr(S.tetar*180/M_PI));
StringGrid3->Cells[_vr][k+1]=FloatToStr(okr(S.vr));
S.tr = 0;StringGrid3->Cells[_tr][k+1]=FloatToStr(okr(S.tr));

```

```

StringGrid4->Cells[_Xt4][k+1]=FloatToStr(okr(S.Xt));
StringGrid4->Cells[_Pdelenyi][k+1]=FloatToStr((1500-okr(S.Ptys))+250);
    StringGrid4->Cells[_dPdel][k+1]=FloatToStr(okr(S.dPdel,0));
    StringGrid4->Cells[_dXtys][k+1]=FloatToStr(okr(S.dXtys,0));
StringGrid4->Cells[_dXw][k+1]=FloatToStr(okr(S.dXw));
StringGrid4->Cells[_dXh][k+1]=FloatToStr(okr(S.dXh,0));
StringGrid4->Cells[_dXt][k+1]=FloatToStr(okr(S.dXt,0));
StringGrid4->Cells[_dXtz][k+1]=FloatToStr(okr(S.dXtz,0));

```

```

StringGrid4->Cells[_dXvo][k+1]=FloatToStr(okr(S.dXvo,0));
StringGrid4->Cells[_dXq][k+1]=FloatToStr(okr(S.dXq,0));
StringGrid4->Cells[_dZw][k+1]=FloatToStr(okr(S.dZw));
StringGrid4->Cells[_Xr][k+1]=FloatToStr(okr(S.Xr));
StringGrid4->Cells[_Yrm][k+1]=FloatToStr(okr(S.Yrm));
StringGrid4->Cells[_Zr][k+1]=FloatToStr(okr(S.Zr));
StringGrid4->Cells[_dXc][k+1]=FloatToStr(okr(S.dXc));

StringGrid4->Cells[_teta0][k+1]=StringGrid1->Cells[_tetao][k+1];

StringGrid4->Cells[_Pri][k+1]=FloatToStr(okr(S.Ptys));
StringGrid4->Cells[_tetac][k+1]=FloatToStr(okr(S.tetac*180/M_PI)*-1);
StringGrid4->Cells[_vc][k+1]=FloatToStr(okr(S.vc));
StringGrid4->Cells[_tc][k+1]=FloatToStr(okr(S.tc));
StringGrid4->Cells[_Vd][1+k]=FloatToStr(okr(S.Vd));

StringGrid4->Cells[_Ybul][1+k]=StringGrid1->Cells[_ybul][k+1];
StringGrid4->Cells[_Vb][1+k]=FloatToStr(okr(S.Vb));

Form3->Series1->AddXY(S.Xt,S.Vd);
Form3->Series2->AddXY(S.Xt,S.Vv);
Form3->Series3->AddXY(S.Xt,S.Vb);

Form4->StringGrid1->Cells[k+1][0]=FloatToStr(okr(S.Xt));
Form4->StringGrid1->Cells[k+1][1]=FloatToStr(okr(S.Vd));
Form4->StringGrid1->Cells[k+1][2]=FloatToStr(okr(S.Vv));
Form4->StringGrid1->Cells[k+1][3]=FloatToStr(okr(S.Vb));

for (int vi=0;vi<28;vi++)
if (okr(S.Xt)==vd_opitnie[0][vi])
{
Form4->StringGrid1->Cells[k+1][4]=vd_opitnie[1][vi];
Form4->StringGrid1->Cells[k+1][5]=vv_opitnie[1][vi];
Form4->StringGrid1->Cells[k+1][6]=vb_opitnie[1][vi];

}
//динамика полЄта

//конец динамики полЄта

file1<<okr(S.Xt)<<"\t"<<okr(S.Vd)<<endl;
file2<<okr(S.Xt)<<"\t"<<S.Vdteta0<<endl;
file3<<okr(S.Xt)<<"\t"<<S.Vdv0<<endl;
file4<<okr(S.Xt)<<"\t"<<S.Vdc<<endl;
file5<<okr(S.Xt)<<"\t"<<okr(S.Vv)<<endl;

```



```

file6<<okr(S.Xt)<<"\t"<<okr(S.Vb)<<endl;
file7<<okr(S.Xt)<<endl;
file8<<okr(S.Xt)<<"\t"<<okr(S.Yt)<<endl; ("parameters\\Yt.txt");
file9<<okr(S.Xt)<<"\t"<<okr(S.Zt)<<endl; ("parameters\\Zt.txt");
file10<<okr(S.Xt)<<"\t"<<okr(S.tetac*180/M_PI)<<endl; ("parameters\
\tetac.txt");
file11<<okr(S.Xt)<<"\t"<<okr(S.vc)<<endl; ("parameters\\vc.txt");
file12<<okr(S.Xt)<<"\t"<<okr(S.tc)<<endl; ("parameters\\tc.txt");
file13<<okr(S.Xt)<<"\t"<<okr(S.dXtys)<<endl; ("parameters\\dXtys.txt");
file14<<okr(S.Xt)<<"\t"<<okr(S.Zd)<<endl; ("parameters\\Zd.txt");
file15<<okr(S.Xt)<<"\t"<<okr(S.dXw)<<endl; ("parameters\\dXw.txt");
file16<<okr(S.Xt)<<"\t"<<okr(S.dXh)<<endl; ("parameters\\dXh.txt");
file17<<okr(S.Xt)<<"\t"<<okr(S.dXt)<<endl; ("parameters\\dXt.txt");
file18<<okr(S.Xt)<<"\t"<<okr(S.dXtz)<<endl; ("parameters\\dXtz.txt");
file19<<okr(S.Xt)<<"\t"<<okr(S.dXvo)<<endl; ("parameters\\dXvo.txt");
file20<<okr(S.Xt)<<"\t"<<okr(S.dXvo1proc)<<endl; ("parameters\
\dXvo1proc.txt");
file21<<okr(S.Xt)<<"\t"<<okr(S.dXq)<<endl; ("parameters\\dXq.txt");
file22<<okr(S.Xt)<<"\t"<<okr(S.dZw)<<endl; ("parameters\\dZw.txt");
file23<<okr(S.Xt)<<"\t"<<okr(S.dXvr)<<endl; ("parameters\\dXvr.txt");
file24<<okr(S.Xt)<<"\t"<<okr(S.dZvr)<<endl; ("parameters\\dZvr.txt");
file25<<okr(S.Xt)<<"\t"<<okr(S.dXtzt)<<endl; ("parameters\\dXtzt.txt");
file26<<okr(S.Xt)<<"\t"<<okr(S.Xr)<<endl; ("parameters\\Xr.txt");
file27<<okr(S.Xt)<<"\t"<<okr(S.Zr)<<endl; ("parameters\\Zr.txt");
file28<<okr(S.Xt)<<"\t"<<okr(S.dXc)<<endl; ("parameters\\dXc.txt");
file29<<okr(S.Xt)<<"\t"<<okr(S.Ptys)<<endl; ("parameters\\Ptys.txt");
file30<<okr(S.Xt)<<"\t"<<okr(S.Yrm)<<endl; ("parameters\\Yrm.txt");
file31<<okr(S.deltat1)<<"<<okr(S.deltat2)<<endl; ("parameters\\dt.txt");
}
file89<<S.Cx.c_str)<<endl; ("parameters\\current_f(Mtau).txt");
temps=/*temps+*/"<td>"+S.Cx+"</td>\t</tr>";
S.vrtau="";
S.Cx="";
file1.close();
file2.close();
file3.close();
file4.close();
file5.close();
file6.close();
file7.close();
file8.close();
file9.close();
file10.close();
file11.close();
file12.close();

```

```

file13.close();
file14.close();
file15.close();
file16.close();
file17.close();
file18.close();
file19.close();
file20.close();
file21.close();
file22.close();
file23.close();
file24.close();
file25.close();
file26.close();
file29.close();
file30.close();
    file31.close();
file89<<"</table>\t"<<endl;
file89.close();

```

```

PageControl1->ActivePageIndex=1;

```

```

}

```

```

//-----

```

```

void __fastcall TForm1::Button11Click(TObject *Sender)

```

```

{

```

```

    float table_coef_row_one[6] = {0, 1000, 2000, 0, -1000, -2000};

```

```

    float table_coef_col_one[33] = {350, 370, 390, 410, 430, 450, 470, 490, 510, 530,
550, 570, 590, 610, 630, 650, 670, 690, 710, 730, 750, 770, 790, 810, 830, 850, 870,
890, 910, 930, 950, 970, 990};

```

```

    float table_coef[33][6] = {
        {0.15, 0.14, 0.14, 0, 0, 0},
        {0.17, 0.16, 0.16, 0, 0, 0},
        {0.19, 0.18, 0.18, 0, 0, 0},
        {0.21, 0.2, 0.2, 0, 0, 0.1},
        {0.23, 0.22, 0.22, 0, 0, 0.12},
        {0.25, 0.24, 0.24, 0, 0, 0.14},
        {0.27, 0.27, 0.26, 0, 0.16, 0.16},
        {0.3, 0.29, 0.29, 0, 0.18, 0.18},
        {0.32, 0.32, 0.31, 0, 0.21, 0.21},
        {0.35, 0.35, 0.34, 0, 0.23, 0.23},
        {0.38, 0.38, 0.37, 0, 0.26, 0.26},
        {0.42, 0.41, 0.41, 0, 0.29, 0.29},

```

```

    {0.45, 0.44, 0.44, 0, 0.32, 0.32},
    {0.5, 0.5, 0.49, 0.36, 0.35, 0.35},
    {0.55, 0.54, 0.53, 0.41, 0.4, 0.4},
    {0.6, 0.59, 0.59, 0.45, 0.45, 0.44},
    {0.6, 0.6, 0.59, 0.5, 0.5, 0.49},
    {0.74, 0.73, 0.72, 0.5, 0.5, 0.5},
    {0.81, 0.8, 0.79, 0.63, 0.63, 0.62},
    {0.9, 0.88, 0.88, 0.7, 0.7, 0.69},
    {0.99, 0.98, 0.98, 0.78, 0.78, 0.77},
    {1.11, 1.1, 1.09, 0.88, 0.87, 0.87},
    {1.24, 1.23, 1.23, 0.98, 0.98, 0.98},
    {1.4, 1.39, 1.39, 1.11, 1.11, 1.1},
    {1.59, 1.58, 1.58, 1.26, 1.26, 1.26},
    {1.82, 1.82, 1.82, 1.44, 1.44, 1.44},
    {2.11, 2.11, 2.12, 1.66, 1.67, 1.67},
    {2.48, 2.5, 2.51, 1.94, 1.95, 1.96},
    {2.97, 3, 3.03, 2.29, 2.31, 2.33},
    {3.73, 3.81, 3.9, 2.76, 2.79, 2.83},
    {5.16, 5.33, 5.54, 3.4, 3.44, 3.49},
    {0, 0, 0, 4.08, 4.16, 4.24},
    {0, 0, 0, 5.09, 5.22, 5.38}
};
int temp2;
float
Xp,Zp,Dsum,deltaXw,Wx,Wz,deltaXh,delta_Hb,deltaXt,delta_tauY,deltaXtz,deltaTz,
deltaXvo,deltaVo,deltaXq,deltaQ,deltaDsum,deltaZw,D_c_t,delta_c_t,D_c_v,Pr_tys,
Pr_dol;
for(temp2 = 2; StrToFloat(StringGrid7->Cells[2][4]) > StrToFloat(StringGrid1-
>Cells[0][temp2]); temp2++){
    if((StrToFloat(StringGrid1->Cells[0][temp2])+StrToFloat(StringGrid1->Cells[0]
[temp2-1]))/2 < StrToFloat(StringGrid7->Cells[2][4])){
        continue;
    }
    else{
        temp2=temp2-1;
    }
}
Wx = StrToFloat(StringGrid10->Cells[2][1]);
Wz = StrToFloat(StringGrid10->Cells[3][1]);
delta_Hb = StrToFloat(StringGrid10->Cells[0][1]);
delta_tauY = StrToFloat(StringGrid10->Cells[1][1]);
deltaXw = StrToFloat(StringGrid4->Cells[6][temp2-1]);
deltaXh = StrToFloat(StringGrid4->Cells[7][temp2-1]);
deltaXt = StrToFloat(StringGrid4->Cells[8][temp2-1]);
deltaXtz = StrToFloat(StringGrid4->Cells[9][temp2-1]);

```

```

deltaTz = StrToFloat(StringGrid11->Cells[2][1])-15;
deltaXvo = StrToFloat(StringGrid4->Cells[10][temp2-1]);
deltaVo = StrToFloat(StringGrid11->Cells[1][1]);
deltaXq = StrToFloat(StringGrid4->Cells[11][temp2-1]);
deltaQ = StrToFloat(StringGrid11->Cells[3][1]);
Dsum = (0.1 * deltaXw * Wx) + (0.1 * deltaXh * delta_Hb) + (0.1 * deltaXt *
delta_tauY) + (0.1 * deltaXtz * deltaTz) + (deltaXvo * deltaVo) + (deltaXq * deltaQ);
Dsum = floor(Dsum+0.5);
deltaZw = StrToFloat(StringGrid4->Cells[5][temp2-1]);
deltaDsum = 0.1 * deltaZw * Wz;
deltaDsum = floor(deltaDsum+0.5);
D_c_t = StrToFloat(StringGrid7->Cells[2][4]);
Zp=(deltaDsum*D_c_t)/955;
Zp = floor(Zp+0.5);
Xp=D_c_t+Dsum;
delta_c_t = StrToFloat(StringGrid7->Cells[2][5]);
D_c_v = delta_c_t + deltaDsum/100;
D_c_v = ((float)(int)(D_c_v * 100)) / 100;
int i;
float tmp,Tetav_1,Tetav_2;
for(i=1;i<(StringGrid1->RowCount)-1;++i){
    if (StrToFloat(StringGrid1->Cells[0][i]) <= Xp && StrToFloat(StringGrid1-
>Cells[0][i+1]) >= Xp){
        Tetav_1 = StrToFloat(StringGrid1->Cells[1][i]);
        Tetav_2 = StrToFloat(StringGrid1->Cells[1][i+1]);
        break;
    }
}
float tdmp,tdmp2;
tdmp = Xp - StrToFloat(StringGrid1->Cells[0][i]);
float Tetav_1_first, Tetav_1_second, Tetav_2_first, Tetav_2_second,
Tatav_rad,Tatav_1_min,Tatav_2_min,Tetav_min,Tetav_grad;
Tetav_1_first = floor(Tetav_1);
Tetav_1_second = round((Tetav_1-(floor(Tetav_1)))*100);
Tetav_2_first = floor(Tetav_2);
Tetav_2_second = round((Tetav_2-(floor(Tetav_2)))*100);
Tatav_1_min = (Tetav_1_first*60)+ Tetav_1_second;
Tatav_2_min = (Tetav_2_first*60)+ Tetav_2_second;

Tatav_rad = Tatav_1_min - Tatav_2_min;
tdmp2 = floor((Tatav_rad/100)*tdmp+0.5);
Tetav_grad = floor((Tatav_1_min-tdmp2)/60);
Tetav_min = (Tatav_1_min-(Tetav_grad*60))-tdmp2;

Pr_tys = (Tetav_grad * 60 + Tetav_min)/3.6;

```

```

Pr_tys = floor(Pr_tys+0.5);
Pr_dol = (1500 - Pr_tys) + 250;
StringGrid16->Cells[2][1] = Pr_tys;
StringGrid16->Cells[1][1] = Pr_dol;
StringGrid16->Cells[0][1] = round(Tetav_1*10)/10;
int j=0;
float delta_h_c,Kph,delta_P_h,P_v_r,delta_d_sum;
delta_d_sum = (Zp*955)/D_c_t;
if (j > delta_h_c){
    j = 3;
}
j = 5;

for(i=0;i<33;++i){
    if (table_coef_col_one[i] <= Pr_dol && table_coef_col_one[i+1] >= Pr_dol){
        Kph = table_coef[i+1][j];
        break;
    }
}
delta_h_c = StrToFloat(StringGrid7->Cells[2][7]);
delta_P_h = 0.1 * Kph * delta_h_c;
delta_P_h = floor(delta_P_h+0.5);
//delta_P_h = floor(delta_P_h+0.5);
P_v_r = Pr_dol + delta_P_h;
P_v_r = floor(P_v_r+0.5);
    delta_h_c = StrToFloat(StringGrid7->Cells[2][5])+delta_d_sum;
//Tetav_1
//1 ' = 0.00029088820866572 rad
//grad* (pi/180)
//180/pi*rad
StringGrid12->Cells[0][1] = delta_h_c;
StringGrid12->Cells[1][1] = delta_d_sum;
StringGrid16->Cells[0][1] = P_v_r;
StringGrid16->Cells[1][1] = D_c_v;
StringGrid16->Cells[2][1] = Xp;
StringGrid16->Cells[3][1] = Zp;
}
//-----

```