



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования

«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**Кафедра информатики, математического
и компьютерного моделирования**


Оценка координат и времени атмосферного электрического разряда в
программном комплексе региональной системе геолокации молний.

КУРСОВАЯ РАБОТА

образовательной программе подготовки бакалавров
по направлению 01.03.02 «Прикладная математика и информатика»

Работа защищена с оценкой

Выполнил студент группы Б8117-01.03.02

 _____ Горячкин Д. М.
(подпись)

« ____ » _____ 2020г.

Регистрационный номер _____

« ____ » _____ 2020г

Руководитель _____
(должность, ученое звание)

_____ Пермяков М.С.
(подпись)

« ____ » _____ 2020г.

г. Владивосток
2020

Оглавление

Введение.....	2
Постановка задачи.....	2
Формализация задачи.	2
Методы решения.	3
Прямой метод.	3
Векторный метод.....	10
Метод четырёхугольника.	13
Реализация.....	14
Тестирование.	16
Прямой метод.	17
Векторный метод.....	22
Метод четырёхугольника.	27
Моделирование.....	33
Разрешение неопределённостей.	48
Вывод.....	48
Источники.	49

Введение.

Молнией называют метеорологическое явление происходящее во время грозы и представляющее собой мощный электрический разряд среди облаков или между облаками и землёй. Фиксация возникновения молнии в пространстве и времени является важной задачей, используемой в прогнозировании погоды.

С момента субъективных наблюдений за молниями прошло достаточно много времени, так что в текущий момент во многих странах используют автоматические системы обнаружения и определения координат молнии. В основном они представляют из себя системы пассивной локации, состоящие из n детекторов ($n \geq 3$) и центрального компьютера, обрабатывающего данные с них.

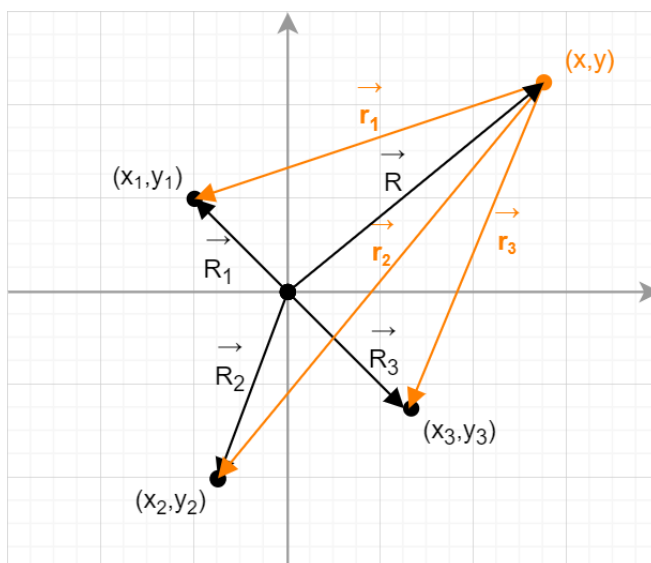
Примером такой системы является WWLLN (World Wide Lightning Location Network), которая позволяет анализировать поля распределений молний с высоким пространственным разрешением на обширных территориях материков и в Мировом океане. WWLLN используются для проведения различных исследований в некоторых районах Российской Федерации, в том числе в её азиатской части. Однако аналогичные системы, но учитывающие особенности своего местоположения, отсутствуют в Приморском крае.

Постановка задачи.

Разработать программу способную за минимальное время с достаточной точностью определить время и местоположение на плоскости возникшей молнии, использующую данные с различного числа детекторов.

Формализация задачи.

Происходящий процесс можно представить в следующем виде:



Где:

- x, y – координаты молнии на плоскости,
- x_i, y_i – координаты i детектора, зафиксировавшего молнию,
- \vec{R} – радиус-вектор местоположения молнии,
- \vec{R}_i – радиус-вектор местоположения i детектора,
- \vec{r}_i – вектор из точки возникновения молнии в местоположение i детектора.

Помимо этого, введём следующие обозначения:

- t – время возникновения молнии,
- t_i – время фиксации молнии i детектором.

Также будем считать, что $|\vec{r}_i| = c(t_i - t)$, где c – скорость распространения сигнала, порождаемого молнией. Тогда, отметим очевидный факт, что $t \leq t_i \forall i$. Данное утверждение всегда выполняется благодаря тому, что зафиксировать сигнал, порождаемый молнией, невозможно до её появления.

Следовательно, нашу задачу можно представить в следующем виде: зная x_i, y_i, t_i необходимо определить x, y, t .

Методы решения.

Из геометрии следует, что однозначно определить точку на плоскости можно путём пересечения трёх окружностей, при условии, что их радиусы и центры известны, при этом последние различны. Поэтому изначально рассмотрим два метода детекции для 3х детекторов ($n = 3$). Они будут иметь общий недостаток – наличие зон неопределённости (пространственные области, в которых невозможно однозначно определить время и место появления молнии), возникающие из-за незнания величины удалённости молнии от детектора. А также рассмотрим метод, работающий при $n = 4$, который лишён данного изъяна.

Стоит отметить, что нижеуказанные методы можно масштабировать и на большее число детекторов, перебирая все возможные сочетания без повторений из 3х / 4х детекторов (в зависимости от метода) и правильно объединяя полученные результаты.

Прямой метод.

Воспользовавшись физическим смыслом скорости, а также формулой нахождения расстояния на плоскости, можем прийти к наиболее очевидному решению поставленной задачи:

$$\begin{cases} \sqrt{(x_1 - x)^2 + (y_1 - y)^2} = c(t_1 - t) \\ \sqrt{(x_2 - x)^2 + (y_2 - y)^2} = c(t_2 - t) \\ \sqrt{(x_3 - x)^2 + (y_3 - y)^2} = c(t_3 - t) \end{cases}$$

Попробуем решить полученную систему. Для этого, введём новую систему координат таким образом, чтобы x_1, y_1 – являлось её началом, а x_2 стало бы равно 0. Тогда, в новой СК получим следующую систему:

$$\begin{cases} \sqrt{x^{*2} + y^{*2}} = c(t_1 - t) \\ \sqrt{x^{*2} + (y_2^* - y^*)^2} = c(t_2 - t) \\ \sqrt{(x_3^* - x^*)^2 + (y_3^* - y^*)^2} = c(t_3 - t) \end{cases},$$

где $(x^*, y^*), (x_i^*, y_i^*)$ – соответственно координаты молнии и i детектора в новой СК. Таким образом нам удалось уменьшить число квадратов разностей с неизвестными. Преобразуем систему, возведя левую и правую часть каждого уравнения в квадрат:

$$\begin{cases} x^{*2} + y^{*2} = c^2(t_1 - t)^2 \\ x^{*2} + (y_2^* - y^*)^2 = c^2(t_2 - t)^2 \\ (x_3^* - x^*)^2 + (y_3^* - y^*)^2 = c^2(t_3 - t)^2 \end{cases}.$$

Обозначим её как систему (*). Выразив из первого уравнения x^{*2} , получим:

$$x^{*2} = c^2(t_1 - t)^2 - y^{*2}.$$

Подставив его во второе уравнение системы (*), имеем:

$$c^2 * (t_1 - t)^2 - y^{*2} + (y_2^* - y^*)^2 = c^2(t_2 - t)^2.$$

Приведя подобные, получим:

$$y_2^{*2} - 2y_2^*y^* = c^2((t_2 - t)^2 - (t_1 - t)^2).$$

Из данного уравнения легко можно выразить y^* :

$$y^* = \frac{y_2^{*2} - c^2((t_2 - t)^2 - (t_1 - t)^2)}{2y_2^*}.$$

Преобразуем 3 уравнение системы (*), раскрыв квадраты разности пространственных координат:

$$x_3^{*2} - 2x_3^*x^* + x^{*2} + y_3^{*2} - 2y_3^*y^* + y^{*2} = c^2(t_3 - t)^2.$$

Вычтем из него первое уравнение системы (*), тем самым сократив квадраты неизвестных пространственных координат:

$$x_3^{*2} - 2x_3^*x^* + y_3^{*2} - 2y_3^*y^* = c^2((t_3 - t)^2 - (t_1 - t)^2).$$

Выразим x^* из данного уравнения:

$$x^* = \frac{x_3^{*2} + y_3^{*2} - 2y_3^*y^* - c^2((t_3 - t)^2 - (t_1 - t)^2)}{2x_3^*} = \frac{x_3^{*2}}{2x_3^*} + \frac{y_3^{*2}}{2x_3^*} - \frac{2y_3^*y^*}{2x_3^*} - \frac{c^2((t_3 - t)^2 - (t_1 - t)^2)}{2x_3^*}.$$

Тогда, возведя полученное выражение в квадрат и упростив, имеем:

$$\begin{aligned} x^{*2} &= \left(\frac{x_3^{*2}}{2x_3^*} + \frac{y_3^{*2}}{2x_3^*} - \frac{2y_3^*y^*}{2x_3^*} - \frac{c^2((t_3 - t)^2 - (t_1 - t)^2)}{2x_3^*} \right)^2 = \frac{c^4((t_3 - t)^2 - (t_1 - t)^2)^2}{4x_3^{*2}} - \\ &\frac{2x_3^{*2}c^2((t_3 - t)^2 - (t_1 - t)^2)}{4x_3^{*2}} - \frac{2y_3^{*2}c^2((t_3 - t)^2 - (t_1 - t)^2)}{4x_3^{*2}} + \frac{4y_3^*y^*c^2((t_3 - t)^2 - (t_1 - t)^2)}{4x_3^{*2}} - \\ &\frac{4x_3^{*2}y_3^*y^*}{4x_3^{*2}} - \frac{4y_3^{*3}y^*}{4x_3^{*2}} + \frac{4y_3^{*2}y^{*2}}{4x_3^{*2}} + \frac{y_3^{*4}}{4x_3^{*2}} + \frac{x_3^{*4}}{4x_3^{*2}} + \frac{2x_3^{*2}y_3^{*2}}{4x_3^{*2}} = \frac{c^4((t_3 - t)^2 - (t_1 - t)^2)^2}{4x_3^{*2}} + \\ &\frac{-2(x_3^{*2} + y_3^{*2})c^2((t_3 - t)^2 - (t_1 - t)^2)}{4x_3^{*2}} + \frac{4y_3^*y^*c^2((t_3 - t)^2 - (t_1 - t)^2)}{4x_3^{*2}} + \frac{-4y_3^*(x_3^{*2} + y_3^{*2})y^*}{4x_3^{*2}} + \\ &\frac{4y_3^{*2}y^{*2}}{4x_3^{*2}} + \frac{x_3^{*4} + 2x_3^{*2}y_3^{*2} + y_3^{*4}}{4x_3^{*2}}. \end{aligned}$$

Подставив x^{*2} в первое уравнение системы (*) получим:

$$\begin{aligned} &c^4((t_3 - t)^2 - (t_1 - t)^2)^2 - 2(x_3^{*2} + y_3^{*2})c^2((t_3 - t)^2 - (t_1 - t)^2) + \\ &4y_3^*y^*c^2((t_3 - t)^2 - (t_1 - t)^2) - 4y_3^*(x_3^{*2} + y_3^{*2})y^* + 4(x_3^{*2} + y_3^{*2})y^{*2} + \\ &(x_3^{*2} + y_3^{*2})^2 = 4x_3^{*2}c^2(t_1 - t)^2. \end{aligned}$$

Расписав второй множитель первого слагаемого, имеем:

$$(t_3 - t)^2 - (t_1 - t)^2 = t_3^2 - 2t_3t + t^2 - t_1^2 + 2t_1t - t^2 = t_3^2 - t_1^2 - 2(t_3 - t_1)t.$$

Тогда, всё выражение примет вид:

$$\begin{aligned} &c^4(t_3^2 - t_1^2 - 2(t_3 - t_1)t)^2 - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) + \\ &4y_3^*y^*c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) - 4y_3^*(x_3^{*2} + y_3^{*2})y^* + 4(x_3^{*2} + y_3^{*2})y^{*2} + \\ &(x_3^{*2} + y_3^{*2})^2 = 4x_3^{*2}c^2(t_1 - t)^2. \end{aligned}$$

Раскроем первое слагаемое левой части:

$$\begin{aligned} &c^4(t_3^2 - t_1^2 - 2(t_3 - t_1)t)^2 = c^4(t_3^4 - 2t_1^2t_3^2 - 4(t_3 - t_1)t_3^2t + t_1^4 + \\ &4(t_3 - t_1)t_1^2t + 4(t_3 - t_1)^2t^2) = c^4(4(t_3 - t_1)^2t^2 - 4(t_3 - t_1)(t_3^2 - t_1^2)t + \\ &(t_3^2 - t_1^2)^2). \end{aligned}$$

Следовательно:

$$c^4(4(t_3 - t_1)^2 t^2 - 4(t_3 - t_1)(t_3^2 - t_1^2)t + (t_3^2 - t_1^2)^2) - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) + 4y_3^* y^* c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) - 4y_3^*(x_3^{*2} + y_3^{*2})y^* + 4(x_3^{*2} + y_3^{*2})y^{*2} + (x_3^{*2} + y_3^{*2})^2 = 4x_3^{*2} c^2(t_1 - t)^2.$$

Расписав правую часть уравнения и сделав её перенос в левую часть, получим:

$$c^4(4(t_3 - t_1)^2 t^2 - 4(t_3 - t_1)(t_3^2 - t_1^2)t + (t_3^2 - t_1^2)^2) - 4x_3^{*2} c^2(t_1^2 - 2t_1 t + t^2) - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) + 4y_3^* y^* c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) - 4y_3^*(x_3^{*2} + y_3^{*2})y^* + 4(x_3^{*2} + y_3^{*2})y^{*2} + (x_3^{*2} + y_3^{*2})^2 = 0.$$

Подставим y^* в данное уравнение:

$$c^4(4(t_3 - t_1)^2 t^2 - 4(t_3 - t_1)(t_3^2 - t_1^2)t + (t_3^2 - t_1^2)^2) - 4x_3^{*2} c^2(t_1^2 - 2t_1 t + t^2) - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) + 4y_3^* c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) * \frac{y_2^{*2} - c^2((t_2 - t)^2 - (t_1 - t)^2)}{2y_2^*} - 4y_3^*(x_3^{*2} + y_3^{*2}) * \frac{y_2^{*2} - c^2((t_2 - t)^2 - (t_1 - t)^2)}{2y_2^*} + 4(x_3^{*2} + y_3^{*2}) * (\frac{y_2^{*2} - c^2((t_2 - t)^2 - (t_1 - t)^2)}{2y_2^*})^2 + (x_3^{*2} + y_3^{*2})^2 = 0.$$

Распишем $(t_2 - t)^2 - (t_1 - t)^2$:

$$(t_2 - t)^2 - (t_1 - t)^2 = t_2^2 - 2t_2 t + t^2 - t_1^2 + 2t_1 t - t^2 = t_2^2 - t_1^2 - 2(t_2 - t_1)t.$$

Тогда:

$$c^4(4(t_3 - t_1)^2 t^2 - 4(t_3 - t_1)(t_3^2 - t_1^2)t + (t_3^2 - t_1^2)^2) - 4x_3^{*2} c^2(t_1^2 - 2t_1 t + t^2) - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) + 4y_3^* c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) * \frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*} - 4y_3^*(x_3^{*2} + y_3^{*2}) * \frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*} + 4(x_3^{*2} + y_3^{*2}) * (\frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*})^2 + (x_3^{*2} + y_3^{*2})^2 = 0.$$

Отметим, что:

$$\begin{aligned} \left(\frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*}\right)^2 &= \frac{y_2^{*4}}{4y_2^{*2}} - 2 * \frac{y_2^{*2}}{2y_2^*} * \frac{c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*} + \\ \frac{c^4(t_2^2 - t_1^2 - 2(t_2 - t_1)t)^2}{4y_2^{*2}} &= \frac{y_2^{*2}}{4} - \frac{c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2} + \\ \frac{c^4(t_2^4 - 2t_1^2 t_2^2 - 4t_2^2(t_2 - t_1)t + t_1^4 + 4t_1^2(t_2 - t_1)t + 4(t_2 - t_1)^2 t^2)}{4y_2^{*2}} &= \frac{y_2^{*2}}{4} - \frac{c^2(t_2^2 - t_1^2)}{2} + \end{aligned}$$

$$c^2(t_2 - t_1)t + \frac{c^4(4(t_2 - t_1)^2 t^2 - 4(t_2 - t_1)(t_2^2 - t_1^2)t + (t_2^2 - t_1^2)^2)}{4y_2^{*2}} = \frac{y_2^{*2}}{4} - \frac{c^2(t_2^2 - t_1^2)}{2} +$$

$$c^2(t_2 - t_1)t + \frac{c^4(t_2 - t_1)^2 t^2}{y_2^{*2}} - \frac{c^4(t_2 - t_1)(t_2^2 - t_1^2)t}{y_2^{*2}} + \frac{c^4(t_2^2 - t_1^2)^2}{4y_2^{*2}}.$$

Следовательно, подставив полученные результат в вышеупомянутое уравнение получим:

$$c^4(4(t_3 - t_1)^2 t^2 - 4(t_3 - t_1)(t_3^2 - t_1^2)t + (t_3^2 - t_1^2)^2) - 4x_3^{*2} c^2(t_1^2 - 2t_1 t + t^2) - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) + 4y_3^* c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) * \frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*} - 4y_3^*(x_3^{*2} + y_3^{*2}) * \frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*} + 4(x_3^{*2} + y_3^{*2}) \left(\frac{c^4(t_2 - t_1)^2 t^2}{y_2^{*2}} + c^2(t_2 - t_1)t - \frac{c^4(t_2 - t_1)(t_2^2 - t_1^2)t}{y_2^{*2}} + \frac{y_2^{*2}}{4} - \frac{c^2(t_2^2 - t_1^2)}{2} + \frac{c^4(t_2^2 - t_1^2)^2}{4y_2^{*2}} \right) + (x_3^{*2} + y_3^{*2})^2 = 0.$$

Представим данное уравнение в виде:

$$At^2 + Bt + C = 0.$$

Для этого распишем каждое слагаемое, содержащие t :

- $c^4(4(t_3 - t_1)^2 t^2 - 4(t_3 - t_1)(t_3^2 - t_1^2)t + (t_3^2 - t_1^2)^2) = 4c^4(t_3 - t_1)^2 t^2 - 4c^4(t_3 - t_1)(t_3^2 - t_1^2)t + c^4(t_3^2 - t_1^2)^2,$
- $-4x_3^{*2} c^2(t_1^2 - 2t_1 t + t^2) = -4x_3^{*2} c^2 t^2 + 8x_3^{*2} c^2 t_1 t - 4x_3^{*2} c^2 t_1^2,$
- $-2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) = 4(x_3^{*2} + y_3^{*2})c^2(t_3 - t_1)t - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2),$
- $4y_3^* c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) * \frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*} = 2y_3^* y_2^* c^2(t_3^2 - t_1^2 - 2(t_3 - t_1)t) - 2\frac{y_3^*}{y_2^*} c^4(t_3^2 - t_1^2 - 2(t_3 - t_1)t)(t_2^2 - t_1^2 - 2(t_2 - t_1)t) = 2y_3^* y_2^* c^2(t_3^2 - t_1^2) - 4y_3^* y_2^* c^2(t_3 - t_1)t - 2\frac{y_3^*}{y_2^*} c^4(t_3^2 - t_1^2 - 2(t_3 - t_1)t)(t_2^2 - t_1^2) + 4\frac{y_3^*}{y_2^*} c^4(t_3^2 - t_1^2 - 2(t_3 - t_1)t)(t_2 - t_1)t = 2y_3^* y_2^* c^2(t_3^2 - t_1^2) - 4y_3^* y_2^* c^2(t_3 - t_1)t - 2\frac{y_3^*}{y_2^*} c^4(t_3^2 - t_1^2)(t_2^2 - t_1^2) + 4\frac{y_3^*}{y_2^*} c^4(t_2^2 - t_1^2)(t_3 - t_1)t + 4\frac{y_3^*}{y_2^*} c^4(t_3^2 - t_1^2)(t_2 - t_1)t - 8\frac{y_3^*}{y_2^*} c^4(t_3 - t_1)(t_2 - t_1)t^2 = -8\frac{y_3^*}{y_2^*} c^4(t_3 - t_1)(t_2 - t_1)t^2 + \left(4\frac{y_3^*}{y_2^*} c^4(t_3^2 - t_1^2)(t_2 - t_1) + 4\frac{y_3^*}{y_2^*} c^4(t_2^2 - t_1^2)(t_3 - t_1) - 4y_3^* y_2^* c^2(t_3 - t_1) \right) t + 2y_3^* y_2^* c^2(t_3^2 - t_1^2) - 2\frac{y_3^*}{y_2^*} c^4(t_3^2 - t_1^2)(t_2^2 - t_1^2),$

- $-4y_3^*(x_3^{*2} + y_3^{*2}) * \frac{y_2^{*2} - c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t)}{2y_2^*} = -2y_3^*y_2^*(x_3^{*2} + y_3^{*2}) +$
 $2\frac{y_3^*}{y_2^*}(x_3^{*2} + y_3^{*2})c^2(t_2^2 - t_1^2 - 2(t_2 - t_1)t) = -4\frac{y_3^*}{y_2^*}(x_3^{*2} + y_3^{*2})c^2(t_2 -$
 $t_1)t - 2y_3^*y_2^*(x_3^{*2} + y_3^{*2}) + 2\frac{y_3^*}{y_2^*}(x_3^{*2} + y_3^{*2})c^2(t_2^2 - t_1^2),$
- $4(x_3^{*2} + y_3^{*2})\left(\frac{c^4(t_2 - t_1)^2 t^2}{y_2^{*2}} + c^2(t_2 - t_1)t - \frac{c^4(t_2 - t_1)(t_2^2 - t_1^2)t}{y_2^{*2}} + \frac{y_2^{*2}}{4} -$
 $\frac{c^2(t_2^2 - t_1^2)}{2} + \frac{c^4(t_2^2 - t_1^2)^2}{4y_2^{*2}}\right) = 4\frac{(x_3^{*2} + y_3^{*2})}{y_2^{*2}}c^4(t_2 - t_1)^2 t^2 + 4(x_3^{*2} + y_3^{*2})c^2(t_2 -$
 $t_1)(1 - \frac{1}{y_2^{*2}}c^2(t_2^2 - t_1^2))t + 4(x_3^{*2} + y_3^{*2})\left(\frac{y_2^{*2}}{4} - \frac{c^2(t_2^2 - t_1^2)}{2} + \frac{c^4(t_2^2 - t_1^2)^2}{4y_2^{*2}}\right).$

Теперь мы легко можем представить вышеуказанное уравнение в искомом виде, со следующими коэффициентами:

$$A = 4c^4(t_3 - t_1)^2 - 4x_3^{*2}c^2 - 8\frac{y_3^*}{y_2^*}c^4(t_3 - t_1)(t_2 - t_1) + 4\frac{(x_3^{*2} + y_3^{*2})}{y_2^{*2}}c^4(t_2 - t_1)^2,$$

$$B = -4c^4(t_3 - t_1)(t_3^2 - t_1^2) + 8x_3^{*2}c^2t_1 + 4(x_3^{*2} + y_3^{*2})c^2(t_3 - t_1) +$$

$$\left(4\frac{y_3^*}{y_2^*}c^4(t_3^2 - t_1^2)(t_2 - t_1) + 4\frac{y_3^*}{y_2^*}c^4(t_2^2 - t_1^2)(t_3 - t_1) - 4y_3^*y_2^*c^2(t_3 - t_1)\right) -$$

$$4\frac{y_3^*}{y_2^*}(x_3^{*2} + y_3^{*2})c^2(t_2 - t_1) + 4(x_3^{*2} + y_3^{*2})c^2(t_2 - t_1)(1 - \frac{1}{y_2^{*2}}c^2(t_2^2 - t_1^2)),$$

$$C = c^4(t_3^2 - t_1^2)^2 - 4x_3^{*2}c^2t_1^2 - 2(x_3^{*2} + y_3^{*2})c^2(t_3^2 - t_1^2) + 2y_3^*y_2^*c^2(t_3^2 -$$

$$t_1^2) - 2\frac{y_3^*}{y_2^*}c^4(t_3^2 - t_1^2)(t_2^2 - t_1^2) - 2y_3^*y_2^*(x_3^{*2} + y_3^{*2}) + 2\frac{y_3^*}{y_2^*}(x_3^{*2} +$$

$$y_3^{*2})c^2(t_2^2 - t_1^2) + 4(x_3^{*2} + y_3^{*2})\left(\frac{y_2^{*2}}{4} - \frac{c^2(t_2^2 - t_1^2)}{2} + \frac{c^4(t_2^2 - t_1^2)^2}{4y_2^{*2}}\right) + (x_3^{*2} + y_3^{*2})^2.$$

Введём обозначения:

- $d_2 = t_2 - t_1,$
- $d_3 = t_3 - t_1,$
- $q_2 = t_2^2 - t_1^2,$
- $q_3 = t_3^2 - t_1^2,$
- $v = \frac{y_3^*}{y_2^*},$

а также вспомним, что $|\vec{R}_3|^2 = x_3^{*2} + y_3^{*2}$. Тогда:

$$A = 4c^4 d_3^2 - 4x_3^{*2} c^2 - 8vc^4 d_3 d_2 + 4 \frac{|\vec{R}_3|^2}{y_2^{*2}} c^4 d_2^2 = 4c^2 (c^2 (d_3 (d_3 - 2vd_2) + \frac{|\vec{R}_3|^2}{y_2^{*2}} d_2^2) - x_3^{*2}),$$

$$B = -4c^4 d_3 q_3 + 8x_3^{*2} c^2 t_1 + 4|\vec{R}_3|^2 c^2 d_3 + 4vc^4 q_3 d_2 + 4vc^4 q_2 d_3 - 4y_3^* y_2^* c^2 d_3 - 4v|\vec{R}_3|^2 c^2 d_2 + 4|\vec{R}_3|^2 c^2 d_2 \left(1 - \frac{1}{y_2^{*2}} c^2 q_2\right) = 4c^2 \left(-c^2 d_3 q_3 + 2x_3^{*2} t_1 + |\vec{R}_3|^2 d_3 + vc^2 q_3 d_2 + vc^2 q_2 d_3 - y_3^* y_2^* d_3 + |\vec{R}_3|^2 d_2 \left(1 - \frac{1}{y_2^{*2}} c^2 q_2 - v\right) \right) = 4c^2 \left(d_3 \left(c^2 (vq_2 - q_3) - y_3^* y_2^* + |\vec{R}_3|^2 \right) + 2x_3^{*2} t_1 + d_2 \left(vc^2 q_3 + |\vec{R}_3|^2 \left(1 - \frac{1}{y_2^{*2}} c^2 q_2 - v\right) \right) \right),$$

$$C = c^4 q_3^2 - 4x_3^{*2} c^2 t_1^2 - 2|\vec{R}_3|^2 c^2 q_3 + 2y_3^* y_2^* c^2 q_3 - 2vc^4 q_3 q_2 - 2y_3^* y_2^* |\vec{R}_3|^2 + 2v|\vec{R}_3|^2 c^2 q_2 + 4d \left(\frac{y_2^{*2}}{4} - \frac{c^2 q_2}{2} + \frac{c^4 q_2^2}{4y_2^{*2}} \right) + |\vec{R}_3|^4 = c^2 \left(c^2 q_3^2 - 4x_3^{*2} t_1^2 - 2|\vec{R}_3|^2 q_3 + 2y_3^* y_2^* q_3 - 2vc^2 q_3 q_2 + 2v|\vec{R}_3|^2 q_2 - 2|\vec{R}_3|^2 q_2 + \frac{|\vec{R}_3|^2}{y_2^{*2}} c^2 q_2^2 \right) + |\vec{R}_3|^2 \left(y_2^* (y_2^* - 2y_3^*) + |\vec{R}_3|^2 \right) = c^2 \left(2(q_3 (y_3^* y_2^* - |\vec{R}_3|^2) - 2x_3^{*2} t_1^2 + q_2 (|\vec{R}_3|^2 (v - 1) - vc^2 q_3)) + c^2 \left(\frac{|\vec{R}_3|^2}{y_2^{*2}} q_2^2 + q_3^2 \right) \right) + |\vec{R}_3|^2 \left(y_2^* (y_2^* - 2y_3^*) + |\vec{R}_3|^2 \right).$$

Таким образом, решив получившиеся квадратное уравнение, мы найдём время возникновения молнии. При решении данного уравнения возможно получение следующих сочетаний корней:

- 1) $t_+ \geq \min(t_1, t_2, t_3)$ и $t_- \geq \min(t_1, t_2, t_3)$;
- 2) $t_+ \geq \min(t_1, t_2, t_3)$ и $t_- < \min(t_1, t_2, t_3)$ или $t_+ < \min(t_1, t_2, t_3)$ и $t_- \geq \min(t_1, t_2, t_3)$;
- 3) $t_+ < \min(t_1, t_2, t_3)$ и $t_- < \min(t_1, t_2, t_3)$.

Рассмотрим дальнейшее поведение для каждого случая. В ситуации, описанной под номер 3, нам требуются дополнительные сведения, помогающие выбрать один из корней, поэтому дальнейшие вычисления не могут быть продолжены. Случай под номером 1 с точки зрения физики невозможен, поэтому, нам необходимо вывести сообщение об ошибке входных данных. Таким образом, дальнейшее выполнение программы имеет

смысл только в ситуации под номером 2, в которой за t будем считать $\max(t_-, t_+)$.

Тогда, из выше сказанного, введя обозначение $t_1^{*2} = (t_1 - t)^2$, получаем: $x^* = \frac{|\vec{R}_3|^2 - 2y_3^*y^* - c^2((t_3 - t)^2 - t_1^{*2})}{2x_3^*}$, $y^* = \frac{y_2^{*2} - c^2((t_2 - t)^2 - t_1^{*2})}{2y_2^*}$. С

помощью данных формул мы сможем найти смещённые координаты положения молнии в пространстве. После чего необходимо сделать перевод в начальную систему координат. При этом рекомендуется не находить угол поворота СК, а высчитать \sin и \cos по определению, так как это позволит повысит точность и снизить время выполнения программы.

Таким образом нам удалось получить метод, решающий поставленную задачу, но при этом имеющий зоны неопределённости.

Векторный метод.

Обратим внимание на разность радиус-векторов:

$$\vec{r}_i = \vec{R}_i - \vec{R}.$$

Она представляет собой вектор, направленный из точки возникновения молнии в местоположение i детектора. Модулем данного вектора является расстояние от источника сигнала до i детектора. Тогда, воспользовавшись определением скорости получаем:

$$c^2(t_i - t)^2 = |\vec{r}_i|^2.$$

Из определения скалярного произведения и вектора \vec{r}_i следует:

$$|\vec{r}_i|^2 = \frac{\langle \vec{r}_i, \vec{r}_i \rangle}{\cos 0} = \langle \vec{R}_i - \vec{R}, \vec{R}_i - \vec{R} \rangle.$$

Расписав разности квадратов, получим:

$$\begin{aligned} c^2(t_i^2 - 2t_it + t^2) &= \langle \vec{R}_i, \vec{R}_i \rangle - 2\langle \vec{R}_i, \vec{R} \rangle + \langle \vec{R}, \vec{R} \rangle, \\ \frac{c^2t_i^2}{2} - c^2t_it + \frac{c^2t^2}{2} &= \frac{\langle \vec{R}_i, \vec{R}_i \rangle}{2} - \langle \vec{R}_i, \vec{R} \rangle + \frac{\langle \vec{R}, \vec{R} \rangle}{2}, \\ \frac{\langle \vec{R}_i, \vec{R}_i \rangle - c^2t_i^2}{2} &= \langle \vec{R}_i, \vec{R} \rangle - c^2t_it - \frac{\langle \vec{R}, \vec{R} \rangle - c^2t^2}{2}, \\ \frac{|\vec{R}_i|^2 - c^2t_i^2}{2} &= x_ix + y_iy - c^2t_it - \frac{|\vec{R}|^2 - c^2t^2}{2}. \end{aligned}$$

Обозначим последнее уравнение как (#).

Введём дополнительные условия: $x_1 = y_1 = t_1 = 0$. Если они не выполняются, то можно произвести смещение СК и продолжить решение в

новых переменных, после чего, полученный ответ представить в изначальной СК. Благодаря данным условиям получаем:

- $t < 0$, так как время появления молнии произошло раньше, чем её зафиксировал 1 датчик,
- $c(-t) = |\vec{R}|$.

Тогда:

$$\frac{|\vec{R}|^2 - c^2 t^2}{2} = \frac{|\vec{R}|^2 - c^2 (-t)^2}{2} = \frac{|\vec{R}|^2 - |\vec{R}|^2}{2} = 0.$$

Из чего следует, что в данных условиях уравнение (#) принимает вид:

$$\frac{|\vec{R}_i|^2 - c^2 t_i^2}{2} = x_i x + y_i y + |\vec{R}| c t_i.$$

Следовательно, мы можем составить систему (*):

$$\begin{cases} x^2 + y^2 = |\vec{R}|^2 \\ \frac{|\vec{R}_2|^2 - c^2 t_2^2}{2} = x_2 x + y_2 y + |\vec{R}| c t_2. \\ \frac{|\vec{R}_3|^2 - c^2 t_3^2}{2} = x_3 x + y_3 y + |\vec{R}| c t_3 \end{cases}$$

Для дальнейших рассуждений стоит отметить, что $t = -\frac{|\vec{R}|}{c}$, т. к. $t < 0$ из выше сказанного, а $|\vec{R}| > 0$ и $c > 0$. Также введём обозначения:

- $q'_i \equiv \frac{|\vec{R}_i|^2 - c^2 t_i^2}{2}$,
- $q_{i-1} \equiv q'_i - |\vec{R}| c t_i$,
- $\vec{q} = (q_1, q_2)$,
- $\vec{p} = (x, y)$.

Тогда $\vec{q} = Q\vec{p}$, где $Q = \begin{pmatrix} x_2 & y_2 \\ x_3 & y_3 \end{pmatrix}$. Следовательно, $Q^{-} = \frac{1}{x_2 y_3 - x_3 y_2} \begin{pmatrix} y_3 & -y_2 \\ -x_3 & x_2 \end{pmatrix}$ в $\vec{p} = Q^{-} \vec{q}$. То есть получаем систему (**):

$$\begin{cases} x = \frac{y_3 q_1 - y_2 q_2}{x_2 y_3 - x_3 y_2} \\ y = \frac{(x_2 q_2 - x_3 q_1)}{(x_2 y_3 - x_3 y_2)}. \\ t = -\frac{|\vec{R}|}{c} \end{cases}$$

Подставив в 1 уравнение системы (*) x, y из системы (**) получим:

$$\frac{(y_3 q_1 - y_2 q_2)^2}{(x_2 y_3 - x_3 y_2)^2} + \frac{(x_2 q_2 - x_3 q_1)^2}{(x_2 y_3 - x_3 y_2)^2} = |\vec{R}|^2.$$

Преобразуем получившиеся уравнение к виду $A|\vec{R}|^2 + B|\vec{R}| + C = 0$.
 Домножим на $(x_2y_3 - x_3y_2)^2$ и распишем квадраты в левой части:

$$(y_3^2q_1^2 - 2y_2y_3q_1q_2 + y_2^2q_2^2) + (x_2^2q_2^2 - 2x_2x_3q_1q_2 + x_3^2q_1^2) = |\vec{R}|^2(x_2y_3 - x_3y_2)^2.$$

Подставим q_1^2 и приведём к вышеуказанному виду:

$$\begin{aligned} & (y_3^2(q'_2 - |\vec{R}|ct_2)^2 - 2y_2y_3(q'_2 - |\vec{R}|ct_2)(q'_3 - |\vec{R}|ct_3) + y_2^2(q'_3 - |\vec{R}|ct_3)^2) + \\ & (x_2^2(q'_3 - |\vec{R}|ct_3)^2 - 2x_2x_3(q'_2 - |\vec{R}|ct_2)(q'_3 - |\vec{R}|ct_3) + x_3^2(q'_2 - |\vec{R}|ct_2)^2) = \\ & |\vec{R}|^2(x_2y_3 - x_3y_2)^2, \end{aligned}$$

$$\begin{aligned} & (x_3^2 + y_3^2)(q'_2 - |\vec{R}|ct_2)^2 - 2(x_2x_3 + y_2y_3)(q'_2 - |\vec{R}|ct_2)(q'_3 - |\vec{R}|ct_3) + \\ & (x_2^2 + y_2^2)(q'_3 - |\vec{R}|ct_3)^2 = |\vec{R}|^2(x_2y_3 - x_3y_2)^2, \end{aligned}$$

$$\begin{aligned} & (x_3^2 + y_3^2)(q_2'^2 - 2q'_2|\vec{R}|ct_2 + |\vec{R}|^2c^2t_2^2) - 2(x_2x_3 + y_2y_3)(q'_2q'_3 - q'_2|\vec{R}|ct_3 - \\ & q'_3|\vec{R}|ct_2 + |\vec{R}|^2c^2t_2^2) + (x_2^2 + y_2^2)(q_3'^2 - 2q'_3|\vec{R}|ct_3 + |\vec{R}|^2c^2t_3^2) = \\ & |\vec{R}|^2(x_2y_3 - x_3y_2)^2, \end{aligned}$$

$$\begin{aligned} & ((x_3^2 + y_3^2)t_2^2 - 2(x_2x_3 + y_2y_3)t_2t_3 + (x_2^2 + y_2^2)t_3^2)c^2 - (x_2y_3 - \\ & x_3y_2)^2|\vec{R}|^2 + 2(-(x_3^2 + y_3^2)q'_2t_2 + (x_2x_3 + y_2y_3)(q'_2t_3 + q'_3t_2) - \\ & (x_2^2 + y_2^2)q'_3t_3)|\vec{R}|c + ((x_3^2 + y_3^2)q_2'^2 - 2(x_2x_3 + y_2y_3)q'_2q'_3 + (x_2^2 + y_2^2)q_3'^2) = \\ & 0. \end{aligned}$$

Таким образом, обозначив $k = (x_2x_3 + y_2y_3)$ и заменив $|\vec{R}_i|^2 = x_i^2 + y_i^2$, получаем:

$$A = (|\vec{R}_3|^2t_2^2 - 2kt_2t_3 + |\vec{R}_2|^2t_3^2)c^2 - (x_2y_3 - x_3y_2)^2,$$

$$B = 2c(-|\vec{R}_3|^2q'_2t_2 + k(q'_2t_3 + q'_3t_2) - |\vec{R}_2|^2q'_3t_3),$$

$$C = |\vec{R}_3|^2q_2'^2 - 2kq'_2q'_3 + |\vec{R}_2|^2q_3'^2.$$

Решив данное квадратное уравнение, мы получаем следующие случаи корней:

- 1) $|\vec{R}|_- < 0$ и $|\vec{R}|_+ < 0$,
- 2) $|\vec{R}|_- < 0$ и $|\vec{R}|_+ \geq 0$ или $|\vec{R}|_- \geq 0$ и $|\vec{R}|_+ < 0$,

$$3) |\vec{R}|_- \geq 0, |\vec{R}|_+ \geq 0.$$

Первый случай, с точки зрения физики невозможен, поэтому необходимо прекратить дальнейшие вычисления и вывести сообщение о не правильных входных данных. В ситуации под номером три возникает неоднозначность, для решения которой требуются дополнительные данные. Во втором случае $|\vec{R}| = \max(|\vec{R}|_-, |\vec{R}|_+)$, при подстановке которого в систему (**) мы найдём x , y и t .

Таким образом мы видим, что данный метод, по сравнению с прямым методом, требует меньшее число арифметических операций, что позволяет с большей скоростью получать ответ на поставленный вопрос.

Метод четырёхугольника.

Пусть $x_1 = y_1 = t_1 = x_2 = 0$. Если это не так, то необходимо ввести новую систему координат, в которой данное условие будет выполняться.

Тогда, аналогично прямому методу получим систему:

$$\begin{cases} x^2 + y^2 = c^2 t^2 \\ x^2 + (y - y_2)^2 = c^2 (t_2 - t)^2 \\ (x - x_3)^2 + (y - y_3)^2 = c^2 (t_3 - t)^2 \end{cases}.$$

Вычитая первое уравнение из второго и третьего, получаем:

$$\begin{cases} x^2 + y^2 = c^2 t^2 \\ -2y_2 y + y_2^2 = c^2 (-2t_2 t + t_2^2) \\ -2x_3 x + x_3^2 - 2y_3 y + y_3^2 = c^2 (-2t_3 t + t_3^2) \end{cases}.$$

Заметим, что если мы хотим избавиться от квадратов над неизвестными, то нам необходимо ввести четвёртое уравнение. Его составим на основе данных, поступающих ещё с одного детектора:

$$(x - x_4)^2 + (y - y_4)^2 = c^2 (t_4 - t)^2.$$

Тогда, вычитая из него первое уравнение системы получаем следующую СЛАУ:

$$\begin{cases} -2y_2 y + y_2^2 = c^2 (-2t_2 t + t_2^2) \\ -2x_3 x + x_3^2 - 2y_3 y + y_3^2 = c^2 (-2t_3 t + t_3^2), \\ -2x_4 x + x_4^2 - 2y_4 y + y_4^2 = c^2 (-2t_4 t + t_4^2) \end{cases}$$

которая без труда может быть решена методом Крамера.

Таким образом мы получили линейный метод, отвечающий на поставленный вопрос и не имеющий областей не определённости, но при этом требующий большего числа детекторов.

Реализация.

С целью дальнейшего развития и увеличения скорости расчётов было принято решение реализовывать проект на ЯП C++ 20.

С проектом можно ознакомиться по ссылке:

<https://github.com/DanilaG/LDtc>.

Рассмотрим структуру проекта:

- *LDtc* – директория, содержащая основные файлы библиотеки детекции молний. Она включает в себя:
 - *Debug* – директория, содержащая файлы для отладки программы. В ней находятся:
 - *Log.h* – файл, описывающий singleton класс логирования, с помощью которого можно оставлять логи, а также перенаправлять выходной поток логирования.
 - *Log.cpp* – файл, реализующий singleton класс логирования,
 - *Detection* – директория, содержащая файлы различных методов детекции. Она включает в себя:
 - *DetectionMath.h* – содержит различные функции, являющиеся общими для многих методов. Такие как: *sqr* – возведение в квадрат и *update_mass_data* – обновляет данные с их коэффициентом уверенности.
 - *DirectMethod.h* – описывает функцию детекции прямым методом (*direct_detect*),
 - *DirectMethod.cpp* – реализует функцию детекции прямым методом,
 - *QuadMethod.h* – описывает функцию детекции методом четырёхугольника (*quad_detect*),
 - *QuadMethod.cpp* – реализует функцию детекции методом четырёхугольника,
 - *VectorMethod.h* – описывает функцию детекции векторным методом (*vector_detect*),
 - *VectorMethod.cpp* – реализует функцию детекции векторным методом,
 - *Structures* – директория, содержащая файлы описывающие основные структуры. Она состоит из:
 - *Geometric* - директория, содержащая файлы описывающие геометрические структуры. Она включает в себя:

- *Points* - директория, содержащая файлы описывающие структуры различных точек. В ней содержатся следующие файлы:
 - *Point.h* – описывает структуру Point, а также вспомогательные для неё функции,
 - *TimePoint.h* - описывает структуру TimePoint (точки со временем), а также вспомогательные для неё функции,
 - *CMakeLists.txt* – описывает параметры сборки данной директории,
 - *Modeling* - директория, содержащая вспомогательные программы, для моделирования различных процессов. Она включает в себя:
 - *DetectorsLocation* – директория, содержащая в себе программу, позволяющую моделировать процесс детекции молнии разными методами при различном числе детекторов с заданной погрешность во времени и в скорости. Она состоит из:
 - *Src* – директория с вспомогательными файлами программы. Она содержит в себе:
 - *DetectionMap.h* – описывает класс DetectionMap, который хранит в себе матрицу погрешности расчётов,
 - *DetectionMap.cpp* – реализует класс DetectionMap,
 - *MapIsolineVisualization.h* – описывает функцию визуализации DetectionMap с помощью изолиний,
 - *MapIsolineVisualization.cpp* – реализует функцию визуализации DetectionMap с помощью изолиний,
 - *MapVisualization.h* – описывает функцию визуализации DetectionMap,
 - *MapVisualization.cpp* – реализует функцию визуализации DetectionMap,
 - *Mat.h* – описывает и реализует класс Mat,
 - *CMakeLists.txt* – описывает параметры сборки данной программы,
 - *main.cpp* – основной файл данной программы,
 - *Tests* – директория, включающая в себя всё необходимое для тестирования. Она содержит:
 - *CMakeLists.txt* – описывает параметры сборки тестов,
 - *Tests.cpp* – реализует в себе все тесты,
 - *TestTools.h* – описывает все вспомогательные инструменты для тестирования,
 - *TestTools.cpp* – реализует вспомогательные инструменты для тестирования,

- *.clang-format* – описывает code style проекта,
- *CMakeLists.txt* – описывает общие параметры сборки всего проекта,
- *Doc.pdf* – файл описывающий проект (данный файл).

Так как проект получился довольно большим, то не будем описывать каждый класс / функцию, рассмотрим только основные моменты его реализации и сборки:

- 1) Все функции детекции принимают на вход ссылку на константный вектор структур TimePoint, содержащих координаты и время срабатывания детектора. Они выполняют вызов вычислительной функции данного метода, перебирая все возможные сочетания итерационным методом (с целью увеличения скорости работы программы), после чего объединяют полученные ответы методом взятия среднего.
- 2) Вычислительные функции всех методов имеют внутренние наименования согласно вышеприведённым выкладкам.
- 3) Вычислительные функции всех методов принимают наименьшее возможное число детекторов, в качестве отдельных аргументов.
- 4) Вычислительные функции всех методов стремятся минимизировать вызов иных функций и количество производимых в них операций с целью увеличения скорости работы программы.
- 5) Для исполнения программы, находящийся в директории DetectorsLocation необходимо убедиться, что на исполняемом устройстве установлена библиотека OpenCV.
- 6) При первой компиляции тестов необходим доступ в интернет, так как будет выполнена автоматическая загрузка Google C++ Testing Framework.

Таким образом, для копирования проекта и дальнейшего его улучшения требуются начальные знания ЯП C++ и умение пользоваться исполняемой ОС, а также доступ в интернет.

Тестирование.

Для проведения тестов будем использовать Google C++ Testing Framework, так как он:

- 1) поддерживается большинством сред разработки,
- 2) имеет хорошую скорость компиляции по сравнению с конкурентами на любых ОС,
- 3) легко дорабатывается.

Входными данными для всех тестов являются:

- 1) вектор структур TimePoint (x, y, time),
- 2) скорость.

Результатами всех тестов является структура TimePoint (x, y, time).

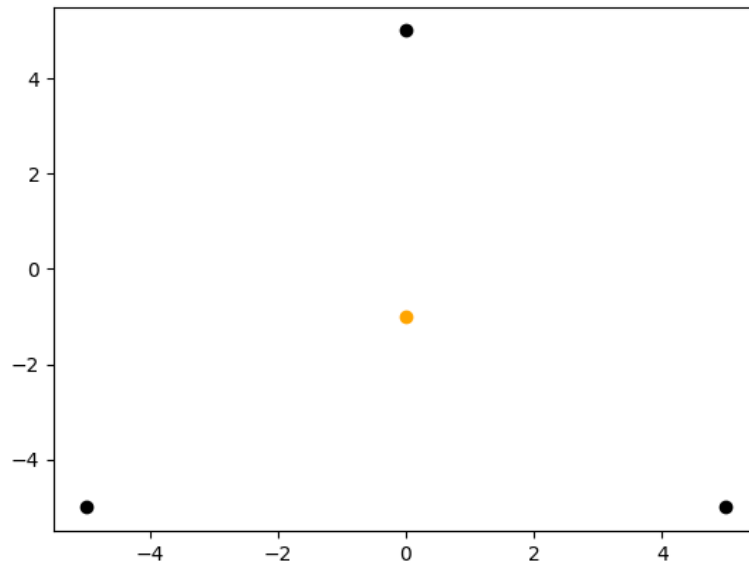
Google C++ Testing Framework и реализованные для него инструменты позволяют производить тесты и сравнивать ожидаемые и полученные результаты с заданной точностью. В случае, если тест был не пройден, фреймворк покажет его, а также выведет в консоль ожидаемое и реальное значение отличающихся переменных.

Далее будут приведены тесты для каждого метода. В описании которых содержится:

- 1) визуализация, в которой:
 - чёрные точки – место положение детекторов,
 - жёлтая точка – место положение молнии,
- 2) входные данные, в формате:
{ {x координата 1 детектора, y координата 1 детектора, время срабатывания 1 детектора}, ..., {x координата i детектора, y координата i детектора, время срабатывания i детектора} }, скорость,
- 3) реальный результат, возвращаемый программой в формате:
{ x координата молнии, y координата молнии, время возникновения молнии }
- 4) ожидаемый результат в таком же формате, как и реальные,
- 5) вывод.

Прямой метод.

- 1) Проверка детекции молнии внутри треугольника детекторов.
Визуализация:



Входные данные: $\{\{-5, -5, 15.640312423743286\},$
 $\{5, -5, 15.640312423743286\},$
 $\{0, 5, 15.6\}\}, 10.$

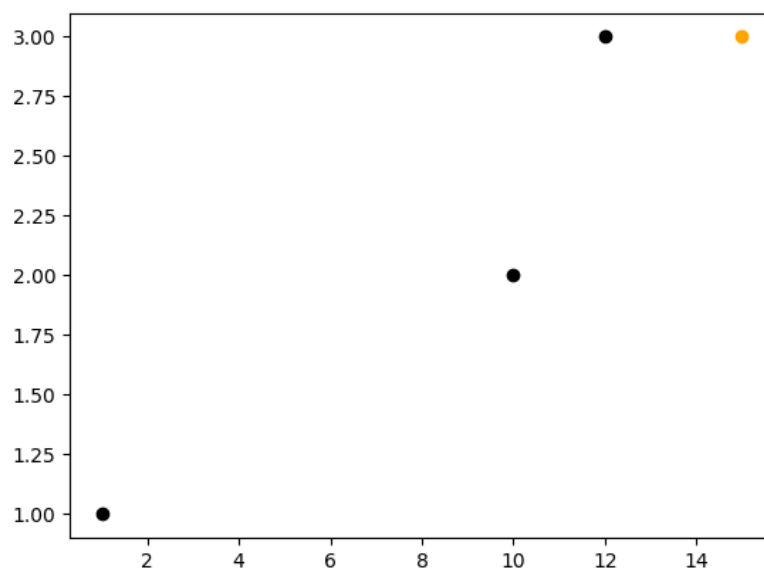
Реальный результат: $\{0, -1, 15\}.$

Ожидаемый результат: $\{0, -1, 15\},$

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 2) Проверка детекции молнии, расположенной вне треугольника детекторов со стороны наименьшего ребра.

Визуализация:



Входные данные: $\{\{1, 1, 11.17851130197758\},$

$\{10, 2, 10.424918292799399\},$
 $\{12, 3, 10.25\}\}, 12.$

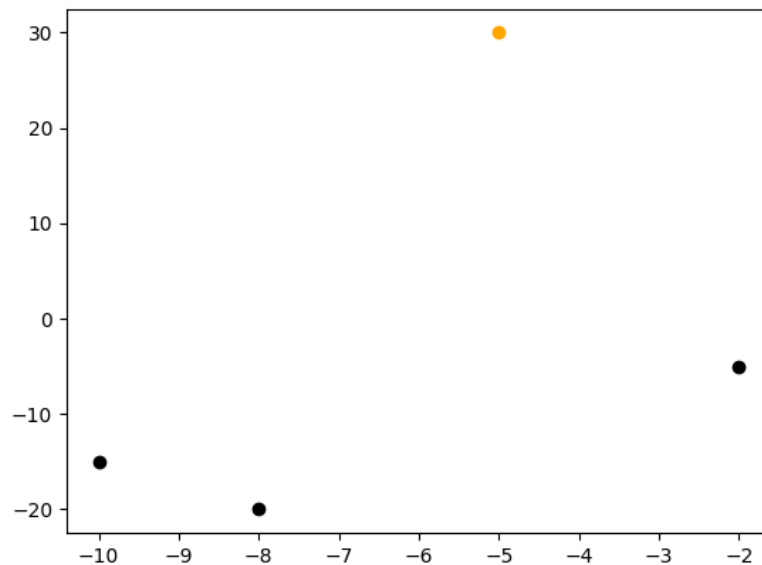
Реальный результат: $\{15, 3, 10\}.$

Ожидаемый результат: $\{15, 3, 10\}.$

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 3) Проверка детекции молнии, расположенной вне треугольника детекторов со стороны наибольшего ребра.

Визуализация:



Входные данные: $\{\{-10, -15, 13.77307714089059\},$
 $\{-2, -5, 12.927361345041716\},$
 $\{-8, -20, 14.174159928789397\}\}, 12.$

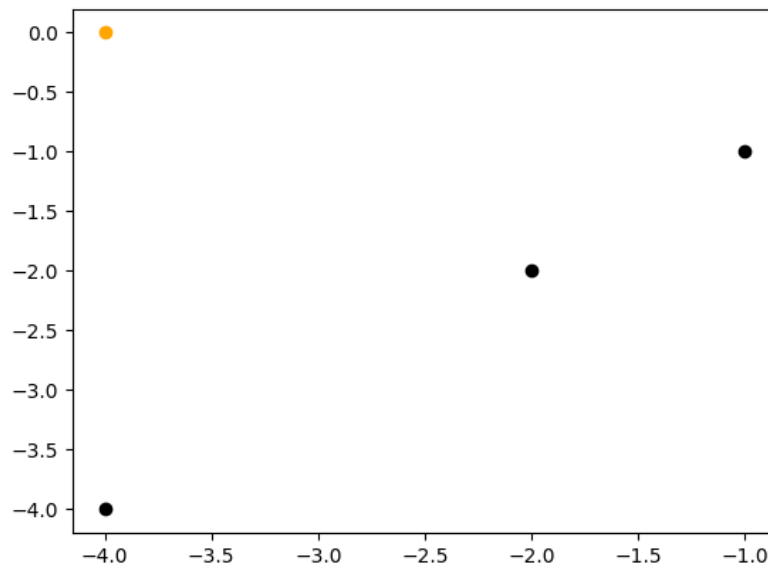
Реальный результат: $\{-5, 30, 10\}.$

Ожидаемый результат: $\{-5, 30, 10\}.$

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 4) Проверка детекции молнии, при условии, что все детекторы лежат на одной прямой.

Визуализация:



Входные данные: $\{\{-1, -1, 4.58113883008419\},$
 $\{-2, -2, 4.414213562373095\},$
 $\{-4, -4, 5\}\}, 2.$

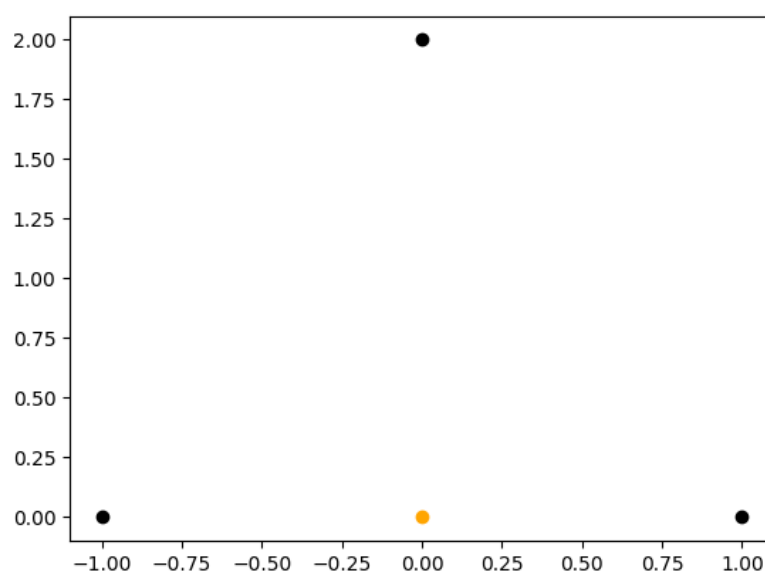
Реальный результат: $\{\text{NAN}, \text{NAN}, \text{NAN}\}.$

Ожидаемый результат: $\{\text{NAN}, \text{NAN}, \text{NAN}\}.$

Вывод: нам не удалось определить место положение молнии, что и было ожидаемым, так как получаемая система является вырожденной.

5) Проверка детекции молнии, расположенной на ребре треугольника детекторов.

Визуализация:



Входные данные: $\{\{-1, 0, 3.5\}, \{1, 0, 3.5\}, \{0, 2, 4\}\}, 2.$

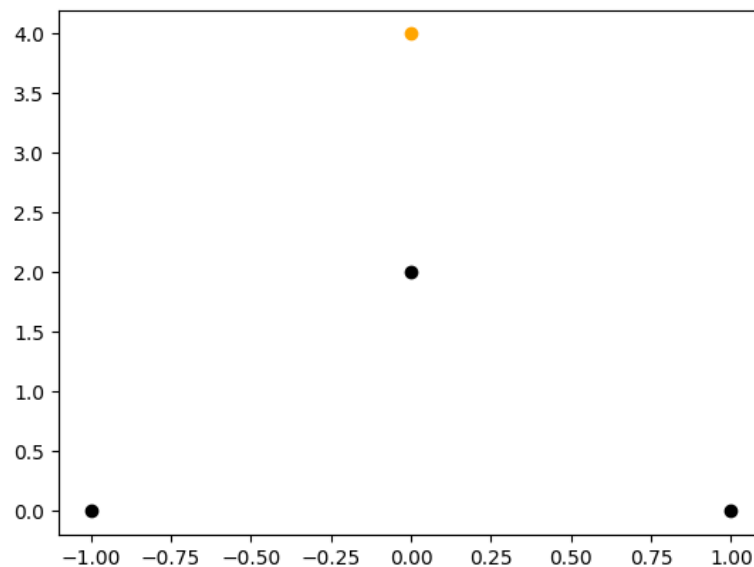
Реальный результат: $\{0, 2.22045e-16, 3\}$.

Ожидаемый результат: $\{0, 0, 3\}$.

Вывод: нам удалось определить местоположение молнии с достаточной точностью.

- 6) Проверка детекции молнии, расположенной за вершиной треугольника детекторов.

Визуализация:



Входные данные: $\{-1, 0, 5.061552812808831\}$,
 $\{1, 0, 5.061552812808831\}$,
 $\{0, 2, 4\}$, 2.

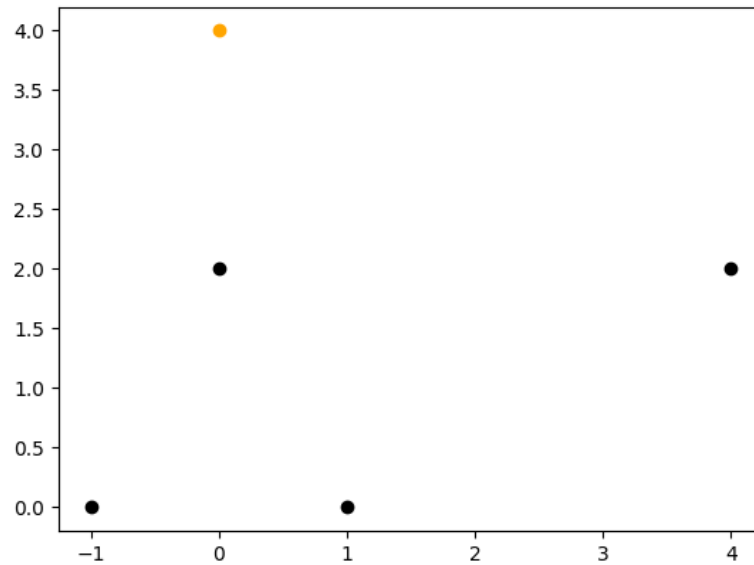
Реальный результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}$.

Ожидаемый результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}$.

Вывод: тест пройден успешно, хотя и не удалось определить место положение молнии. Это было ожидаемо, так как молния находится в зоне неопределённости.

- 7) Проверка детекции молнии при числе детекторов более трёх.

Визуализация:



Входные данные: $\{ \{-1, 0, 5.061552812808831\},$
 $\{1, 0, 5.061552812808831\},$
 $\{0, 2, 4\},$
 $\{4, 2, 5.23606797749979\} \}, 2$

Реальный результат: $\{7.77156e-16, 4, 3\}.$

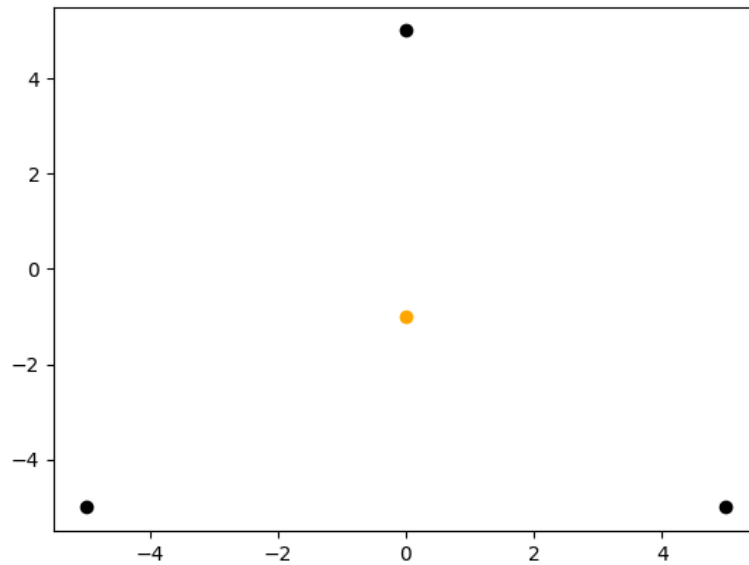
Ожидаемый результат: $\{0, 4, 3\}.$

Вывод: нам удалось определить местоположение молнии с достаточной точностью.

Таким образом видно, что прямой метод реализован корректно.

Векторный метод.

- 1) Проверка детекции молнии внутри треугольника детекторов.
 Визуализация:



Входные данные: $\{\{-5, -5, 15.640312423743286\},$
 $\{5, -5, 15.640312423743286\},$
 $\{0, 5, 15.6\}\}, 10.$

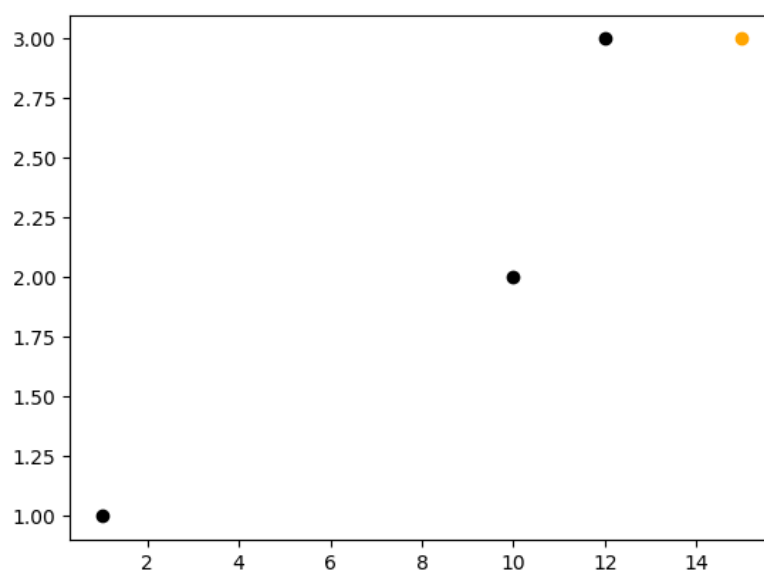
Реальный результат: $\{0, -1, 15\}.$

Ожидаемый результат: $\{0, -1, 15\}.$

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 2) Проверка детекции молнии, расположенной вне треугольника детекторов со стороны наименьшего ребра.

Визуализация:



Входные данные: $\{\{1, 1, 11.17851130197758\},$

$\{10, 2, 10.424918292799399\},$
 $\{12, 3, 10.25\}\}, 12.$

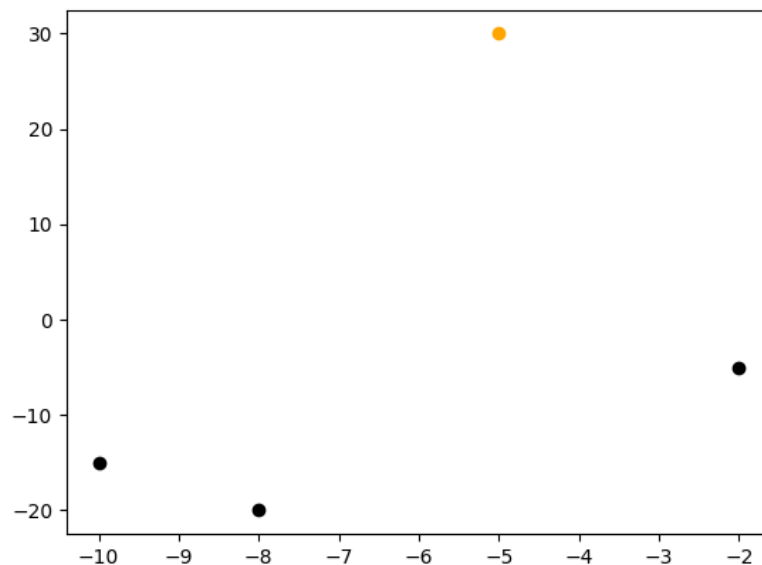
Реальный результат: $\{15, 3, 10\}.$

Ожидаемый результат: $\{15, 3, 10\}.$

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 3) Проверка детекции молнии, расположенной вне треугольника детекторов со стороны наибольшего ребра.

Визуализация:



Входные данные: $\{\{-10, -15, 13.77307714089059\},$
 $\{-2, -5, 12.927361345041716\},$
 $\{-8, -20, 14.174159928789397\}\}, 12.$

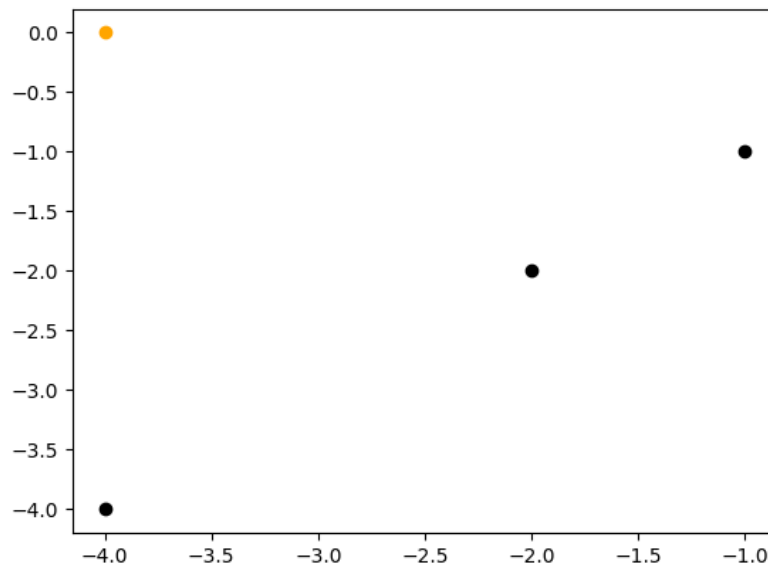
Реальный результат: $\{-5, 30, 10\}.$

Ожидаемый результат: $\{-5, 30, 10\}.$

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 4) Проверка детекции молнии, при условии, что все детекторы лежат на одной прямой.

Визуализация:



Входные данные: $\{\{-1, -1, 4.58113883008419\},$
 $\{-2, -2, 4.414213562373095\},$
 $\{-4, -4, 5\}\}, 2.$

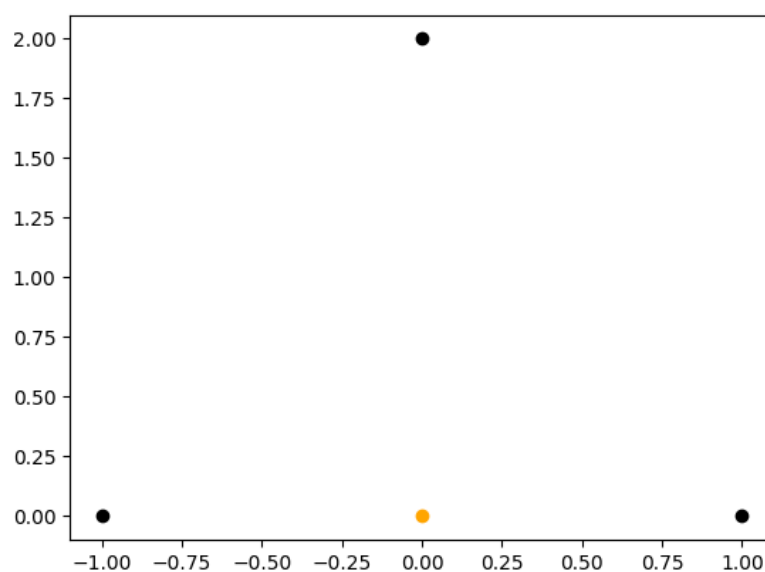
Реальный результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}.$

Ожидаемый результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}.$

Вывод: нам не удалось определить место положение молнии, что и было ожидаемым, так как получаемая система является вырожденной.

5) Проверка детекции молнии, расположенной на ребре треугольника детекторов.

Визуализация:



Входные данные: $\{\{-1, 0, 3.5\}, \{1, 0, 3.5\}, \{0, 2, 4\}\}, 2.$

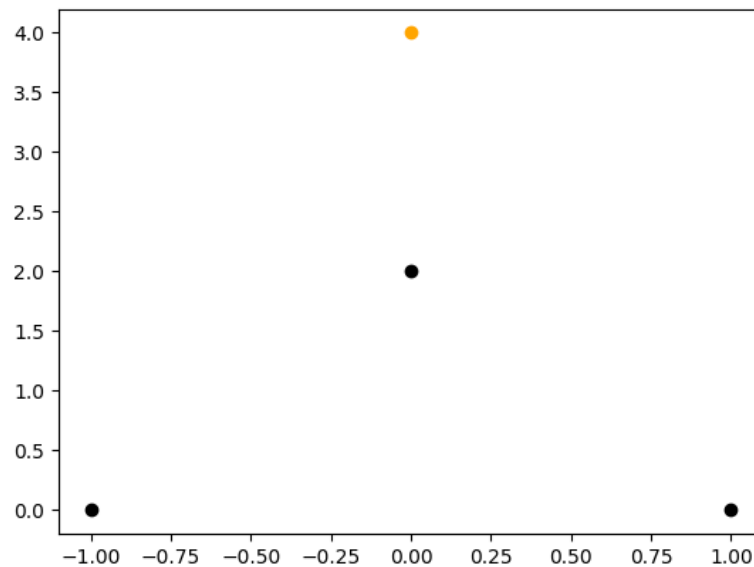
Реальный результат: $\{0, 1.11022e-16, 3\}$.

Ожидаемый результат: $\{0, 0, 3\}$.

Вывод: нам удалось определить местоположение молнии с достаточной точностью.

- 6) Проверка детекции молнии, расположенной за вершиной треугольника детекторов.

Визуализация:



Входные данные: $\{-1, 0, 5.061552812808831\}$,
 $\{1, 0, 5.061552812808831\}$,
 $\{0, 2, 4\}$, 2.

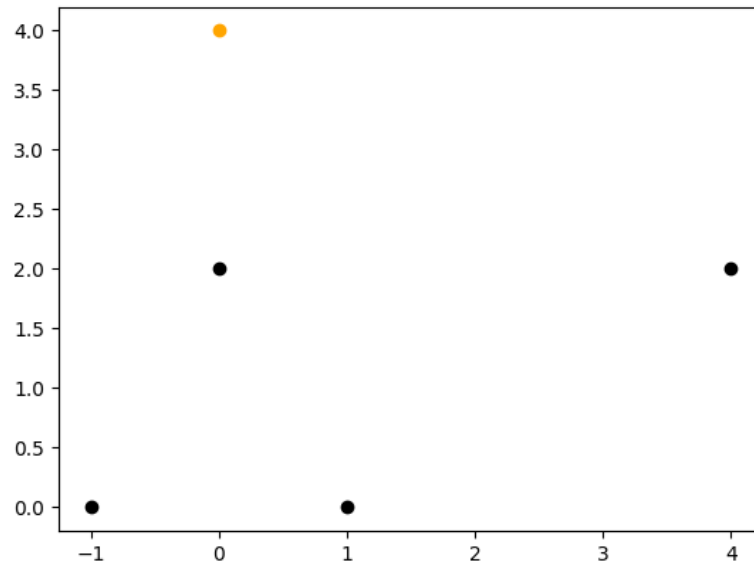
Реальный результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}$.

Ожидаемый результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}$.

Вывод: тест пройден успешно, хотя и не удалось определить место положение молнии. Это было ожидаемо, так как молния находится в зоне неопределённости.

- 7) Проверка детекции молнии при числе детекторов более трёх.

Визуализация:



Входные данные: $\{ \{-1, 0, 5.061552812808831\},$
 $\{1, 0, 5.061552812808831\},$
 $\{0, 2, 4\},$
 $\{4, 2, 5.23606797749979\} \}, 2$

Реальный результат: $\{-1.9984e-15, 4, 3\}.$

Ожидаемый результат: $\{0, 4, 3\}.$

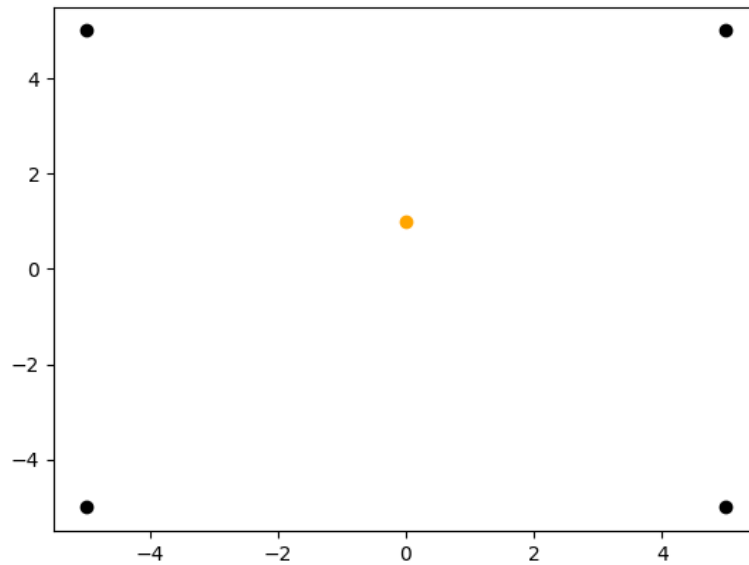
Вывод: нам удалось определить местоположение молнии с достаточной точностью.

Таким образом видно, что векторный метод реализован корректно.

Метод четырёхугольника.

- 1) Проверка детекции молнии, расположенной в центре квадрата детекторов.

Визуализация:



Входные данные: $\{\{-5, -5, 11.781024967590666\},$
 $\{-5, 5, 11.640312423743286\},$
 $\{5, 5, 11.640312423743286\},$
 $\{5, -5, 11.781024967590666\}\}, 10$

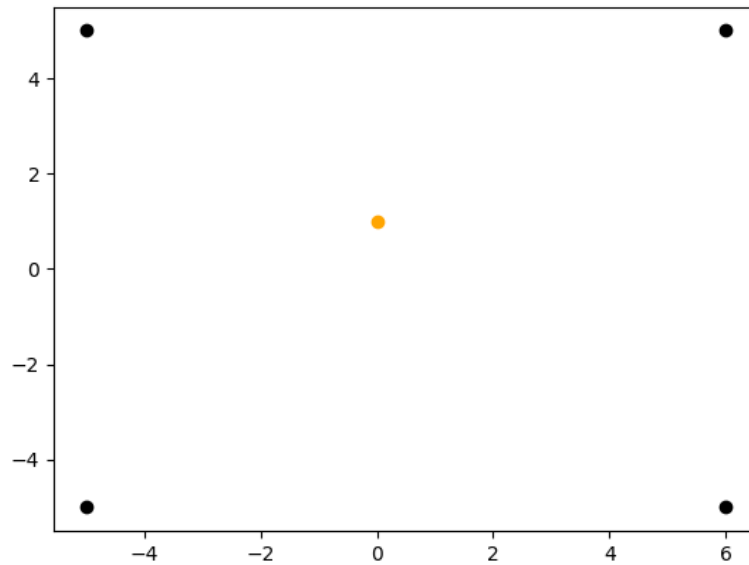
Реальный результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}.$

Ожидаемый результат: $\{\text{NaN}, \text{NaN}, \text{NaN}\}.$

Вывод: тест пройден успешно, хотя нам не удалось определить место положение молнии. Это связано с тем, что получаемая система является вырожденной.

- 2) Проверка детекции молнии, расположенной в квадрате детекторов, но не в его центре.

Визуализация:



Входные данные: $\{\{-5, -5, 11.781024967590666\},$
 $\{-5, 5, 11.640312423743286\},$
 $\{6, 5, 11.721110255092798\},$
 $\{6, -5, 11.848528137423857\}\}, 10$

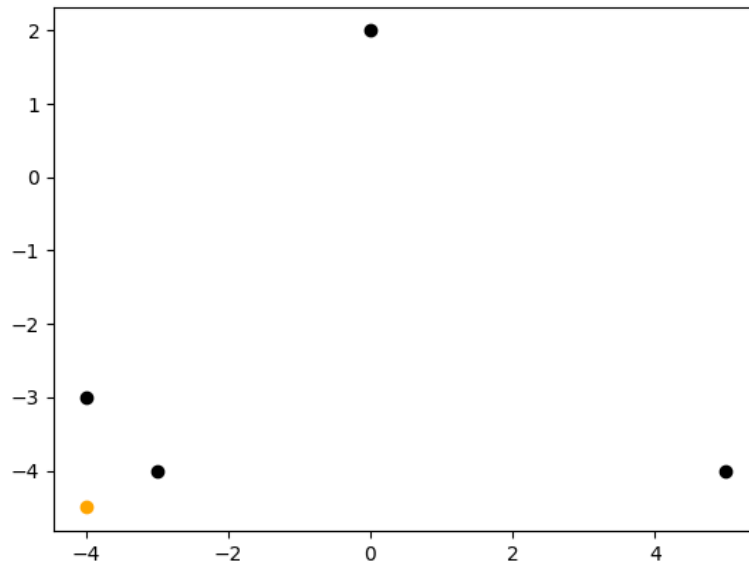
Реальный результат: $\{-6.21725e-15, 1, 11\}$.

Ожидаемый результат: $\{0, 1, 11\}$.

Вывод: нам удалось определить местоположение молнии с достаточной точностью.

- 3) Проверка детекции молнии, расположенной вне четырёхугольника детекторов со стороны наименьшего ребра.

Визуализация:



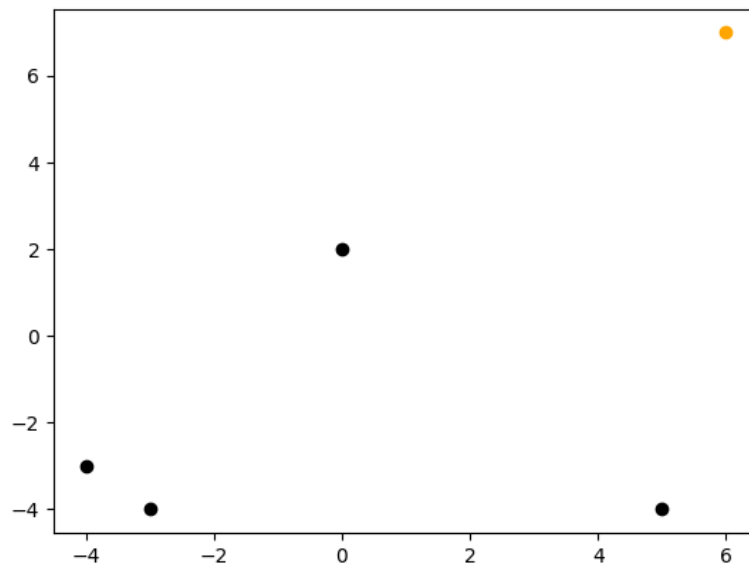
Входные данные: $\{\{-3, -4, 5.279508497187473\},$
 $\{-4, -3, 5.375\},$
 $\{0, 2, 6.908042190309218\},$
 $\{5, -4, 7.253469547164993\}\}, 4$

Реальный результат: $\{-4, -4.5, 5\}.$

Ожидаемый результат: $\{-4, -4.5, 5\}.$

- 4) Проверка детекции молнии, расположенной вне четырёхугольника детекторов со стороны наибольшего ребра.

Визуализация:



Входные данные: $\{\{-3, -4, 8.553167600887974\},$
 $\{-4, -3, 8.535533905932738\},$

$\{0, 2, 6.9525624189766635\},$
 $\{5, -4, 7.761340254296815\}\}, 4$

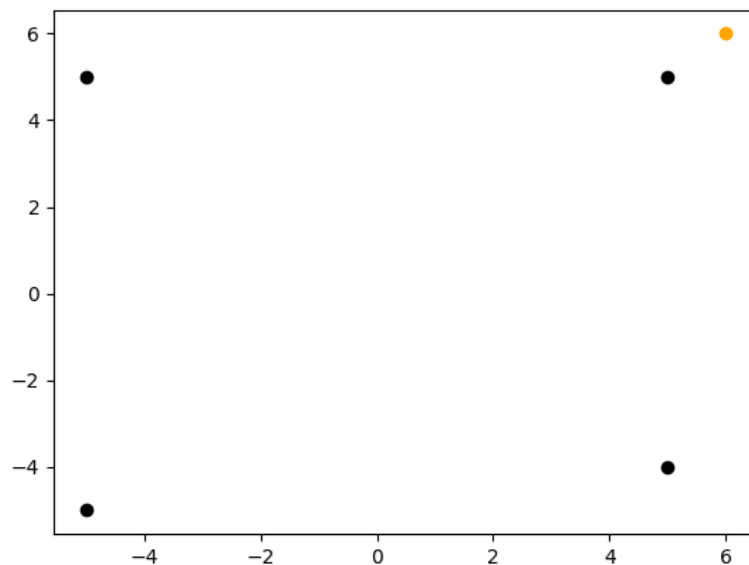
Реальный результат: $\{6, 7, 5\}$.

Ожидаемый результат: $\{6, 7, 5\}$.

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 5) Проверка детекции молнии, расположенной за вершиной четырёхугольника детекторов.

Визуализация:



Входные данные: $\{-5, -5, 12.555634918610405\},$
 $\{-5, 5, 12.104536101718725\},$
 $\{5, 5, 11.14142135623731\},$
 $\{5, -4, 12.004987562112088\}\}, 10$

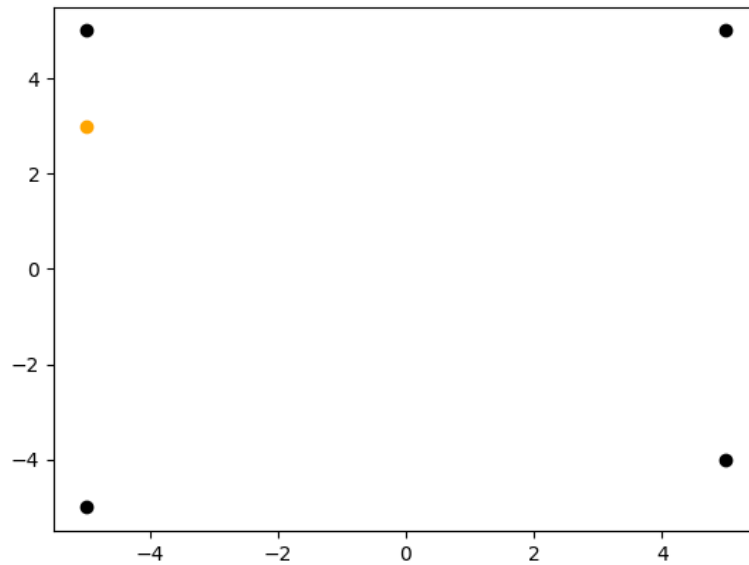
Реальный результат: $\{6, 6, 11\}$.

Ожидаемый результат: $\{6, 6, 11\}$.

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

- 6) Проверка детекции молнии, расположенной на ребре четырёхугольника детекторов.

Визуализация:



Входные данные: $\{\{-5, -5, 9.666666666666666\},$
 $\{-5, 5, 7.666666666666667\},$
 $\{5, 5, 10.399346342395189\},$
 $\{5, -4, 11.068851871911235\}\}, 3$

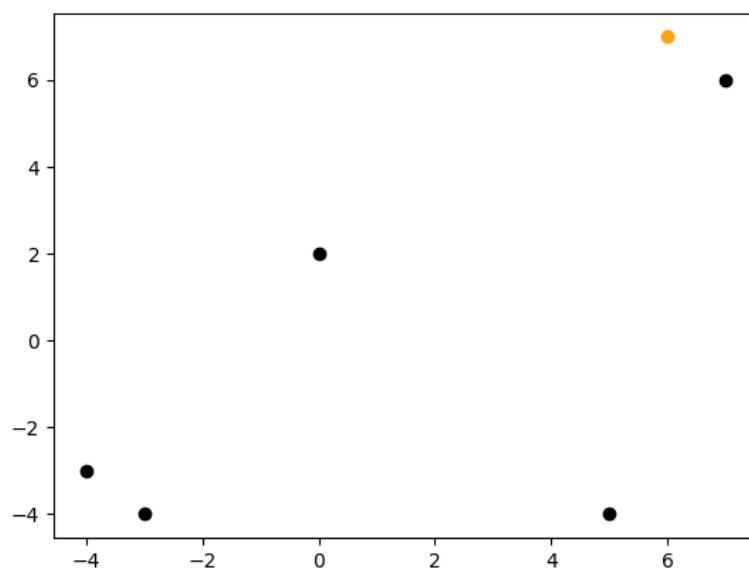
Реальный результат: $\{-5, 3, 7\}$.

Ожидаемый результат: $\{-5, 3, 7\}$.

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

7) Проверка детекции молнии при числе детекторов большем четырёх.

Визуализация:



Входные данные: $\{\{-3, -4, 8.553167600887974\},$

{-4, -3, 8.535533905932738},
 {0, 2, 6.9525624189766635},
 {5, -4, 7.761340254296815},
 {7, 6, 5.353553390593274}}, 4

Реальный результат: {6, 7, 5}.

Ожидаемый результат: {6, 7, 5}.

Вывод: как мы видим тест пройден успешно, так как результаты, возвращённые программой, полностью совпали с ожидаемыми.

Таким образом видно, что метод четырёхугольника реализован корректно.

Моделирование.

Исследуем поведение полученных методов. Для этого построим матрицы погрешностей для различных случаев расположения детекторов и методов детекции. Это можно сделать, воспользовавшись программой, расположенной в директории *Modeling/DetectorsLocation*. В ней будем строить сетку от {-5, -5} до {5, 5} в 400 узлов по x и в 400 по y . В результате работы данной программы, в зависимости от её параметров, мы получим два изображения содержащих сетку погрешности в расстоянии и во времени, на которых:

- чёрные точки – местоположение детекторов,
- красный пиксель – зона неопределённости,
- пиксели от зелёного до синего цвета показывают величину погрешности вычислений, которые можно интерпретировать согласно градиенту под изображением,

или те же данные, но визуализированные с помощью изолиний, где значения изолиний определяется следующими таблицами:

- для расстояния:

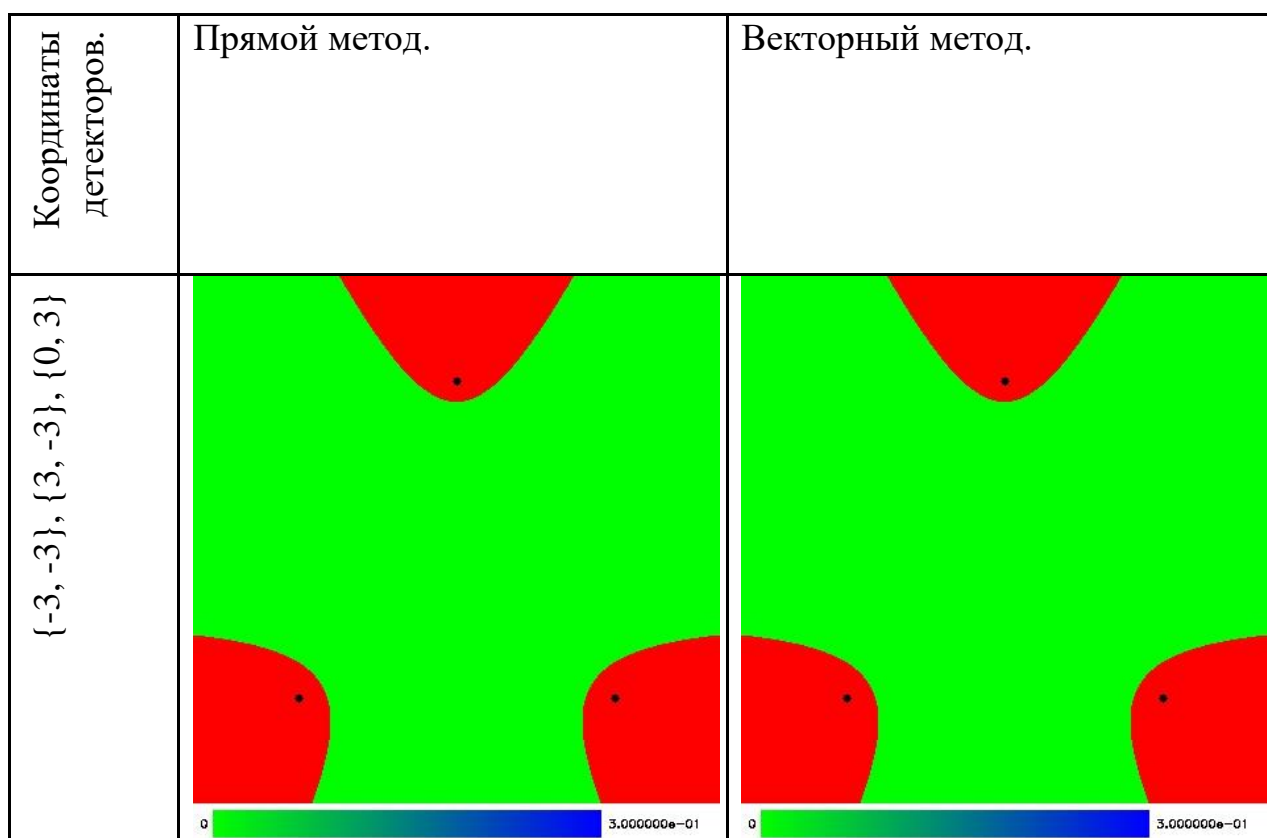
— 0.0025	— 0.0050	— 0.0075	— 0.0100
— 0.0250	— 0.0500	— 0.0750	— 0.1000
— 0.2500	— 0.5000	— 0.7500	— 1.0000

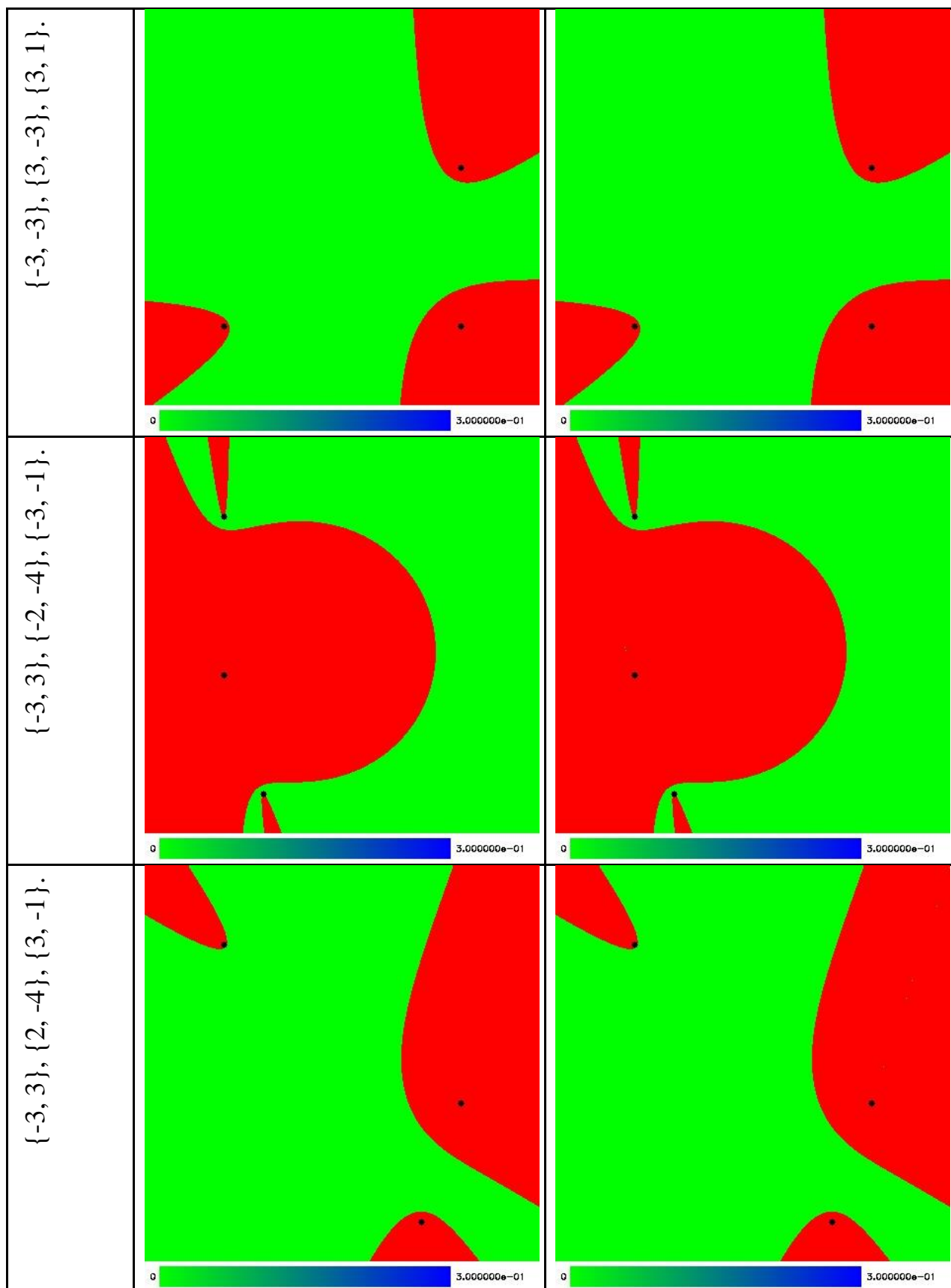
- для времени:



При этом изолинии красного цвета обозначает границы зон неопределённости. В дальнейшем, при изучении только зон неопределённости будем визуализировать данные с помощью сетки, во всех остальных случаях – используя изолинии.

Рассмотрим неопределённости, возникающие в прямом и векторном методах. Установим временную и скоростную погрешность в 0. Так как неопределённость в координатах в данных методах может возникнуть исключительно при неопределённости во времени, рассмотрим только матрицу погрешности для времени.

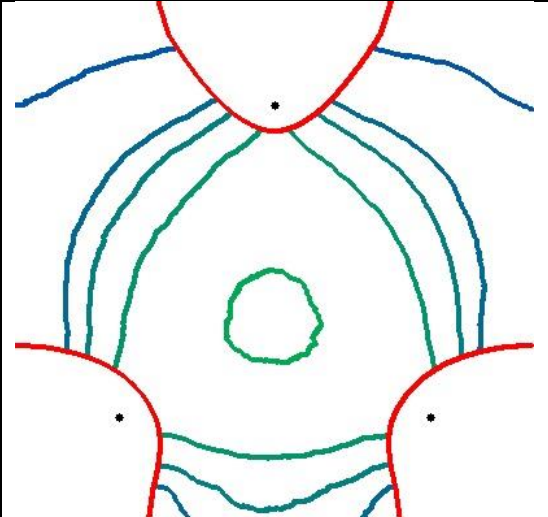
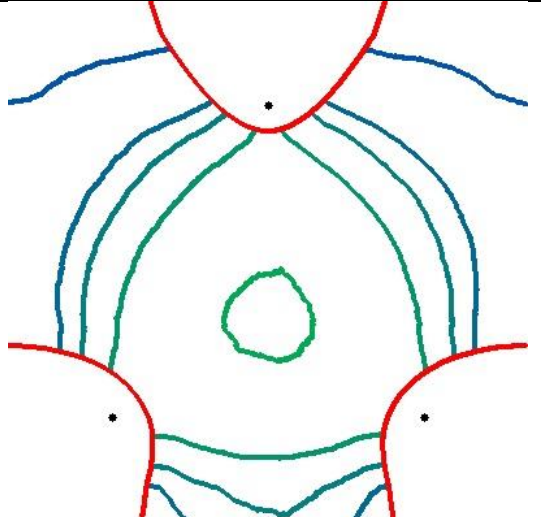
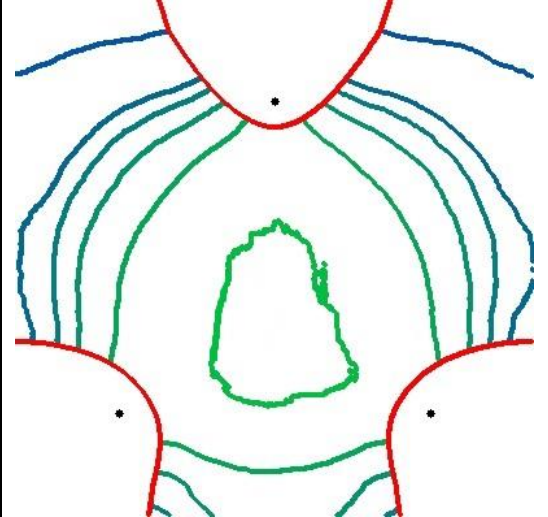
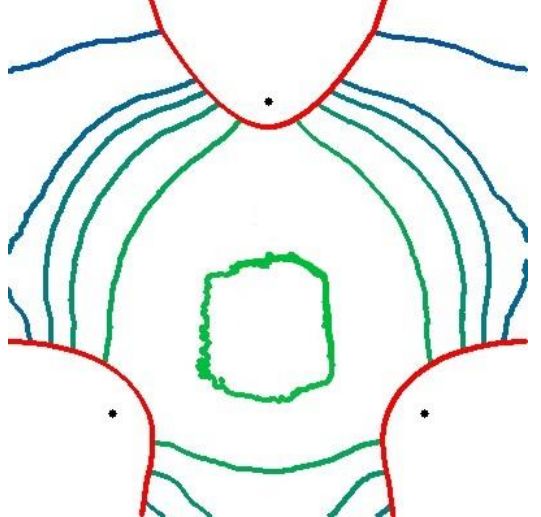




Таким образом мы видим, что данные методы имеют области неопределённости, положение которых зависит от координат детекторов. Также заметно, что области неопределённости у данных методов совпадают. Помимо этого, мы не наблюдаем серьёзных отклонений реальных

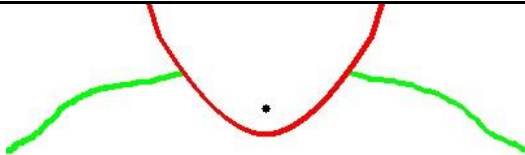
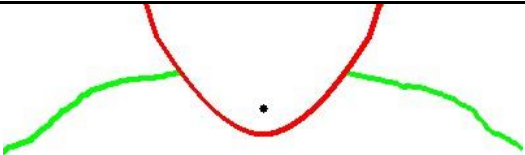


результатов от ожидаемых (так как на изображениях отсутствуют пиксели синего оттенка), из чего следует, что методы были реализованы корректно.

Рассмотрим, как ведут себя данные методы при добавлении погрешности. Добавим случайную погрешность при вычислениях к скорости, не превышающую 1%. Тогда, получим:

Координаты.	Данные в сетке.	Прямой метод.	Векторный метод.
{-3, -3}, {3, -3}, {0, 3}.	Расстояние.		
	Время.		

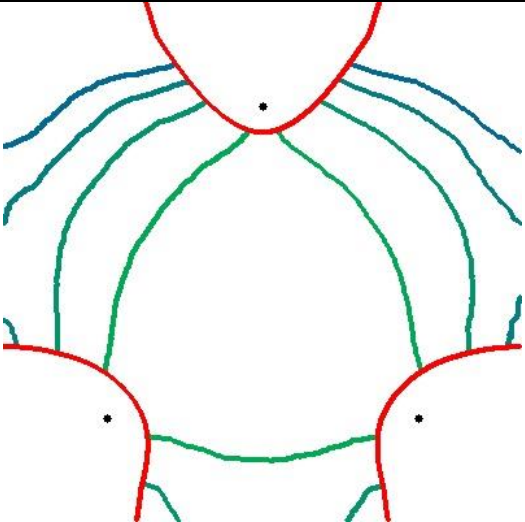
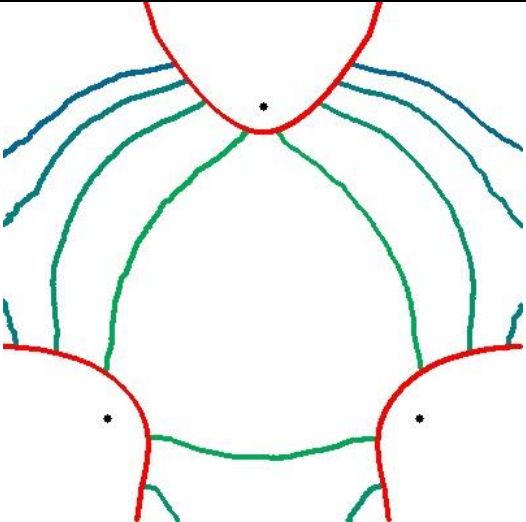
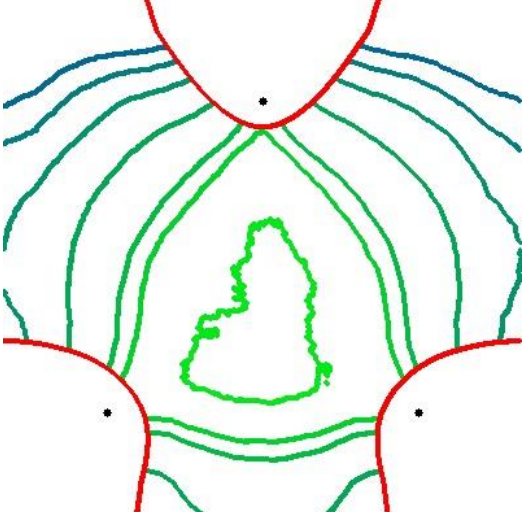
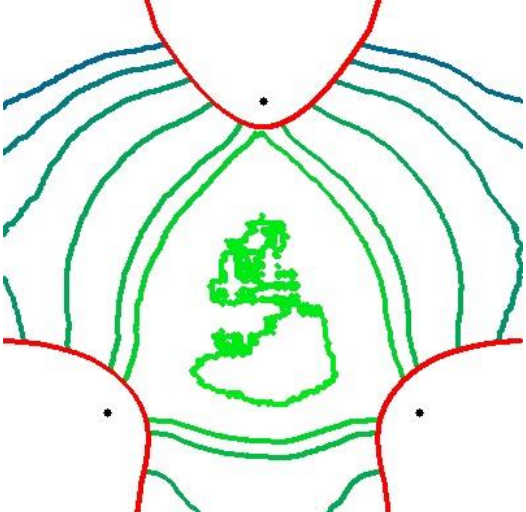
Несложно заметить, что ошибки в методах практически полностью совпадают. На изображениях мы наблюдаем появление погрешностей во времени и в координатах определения положения молнии в зонах, наиболее отдалённых от центра треугольника, образованного детекторами.

Установим максимальное отклонение погрешности на 0.01 единиц по времени (для скорости она равна 0). Получаем:

Координаты. Данные в сетке.	{-3, -3}, {3, -3}, {0, 3}.		Прямой метод.	Векторный метод.
	Расстояние.			
	Время.			

Как мы видим места появления ошибок полностью совпадают с ведением погрешности только по скорости. Также видно, что прямой и векторный метод ведут себя схоже, в плане появления ошибок в результате добавления погрешности ко времени.

Теперь же установим максимальное отклонение погрешности на 0.005 единиц по времени и на 0.5% по скорости. Получаем:

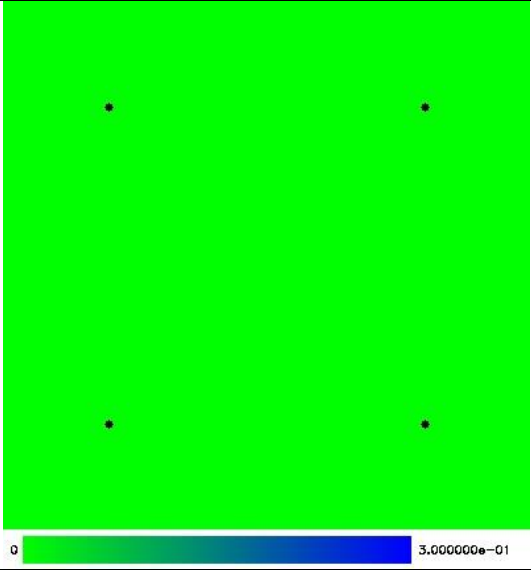
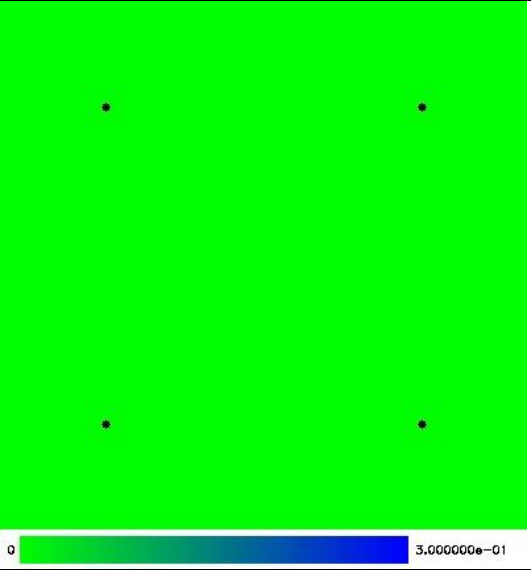
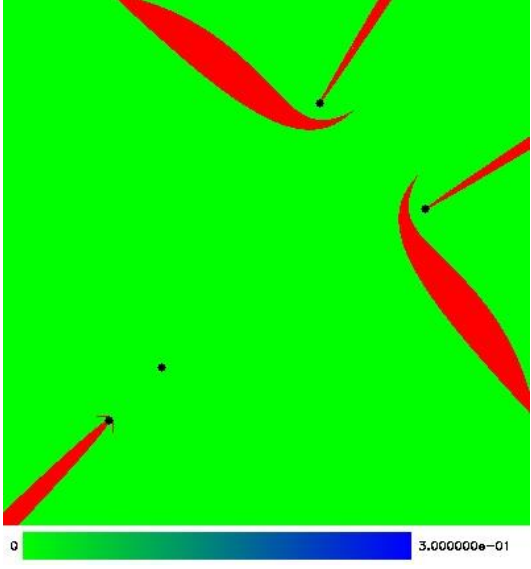
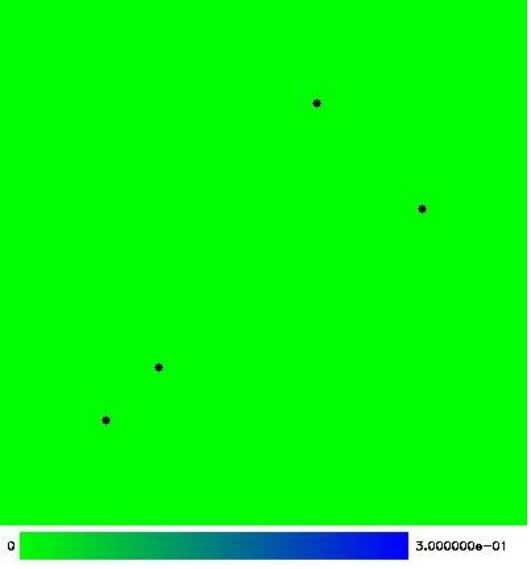
Координаты.	Данные в сетке.	Прямой метод.		Векторный метод.	
{-3, -3}, {3, -3}, {0, 3}.	Расстояние.				
	Время.				

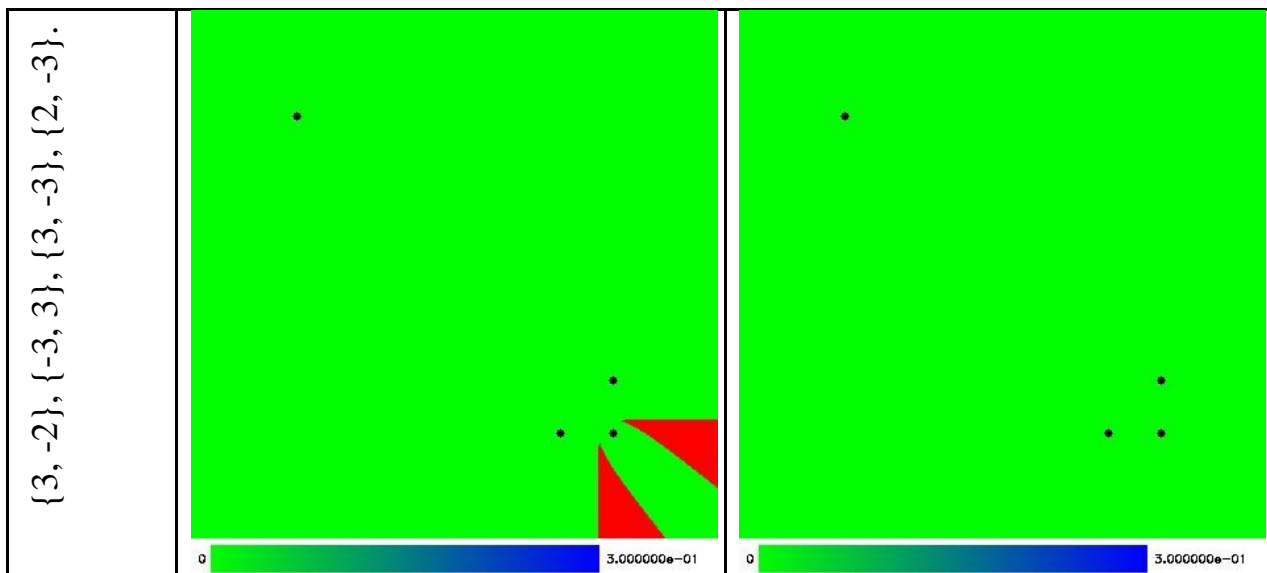
Не сложно заметить, что места появления ошибок не изменились. Помимо этого, методы продолжают вести себя аналогично в смысле появления ошибок и зон неопределённостей.

Таким образом видно, что прямой и векторный метод ведут себя аналогично. Поэтому, для дальнейших сравнений будем использовать только один из них – векторный метод, так как он требует меньше вычислительных операций.

Сравним векторный метод и метод четырёхугольника на неопределённости. Для этого установим максимальную погрешность по времени и скорости в 0. Так как неопределённость в координатах в данных

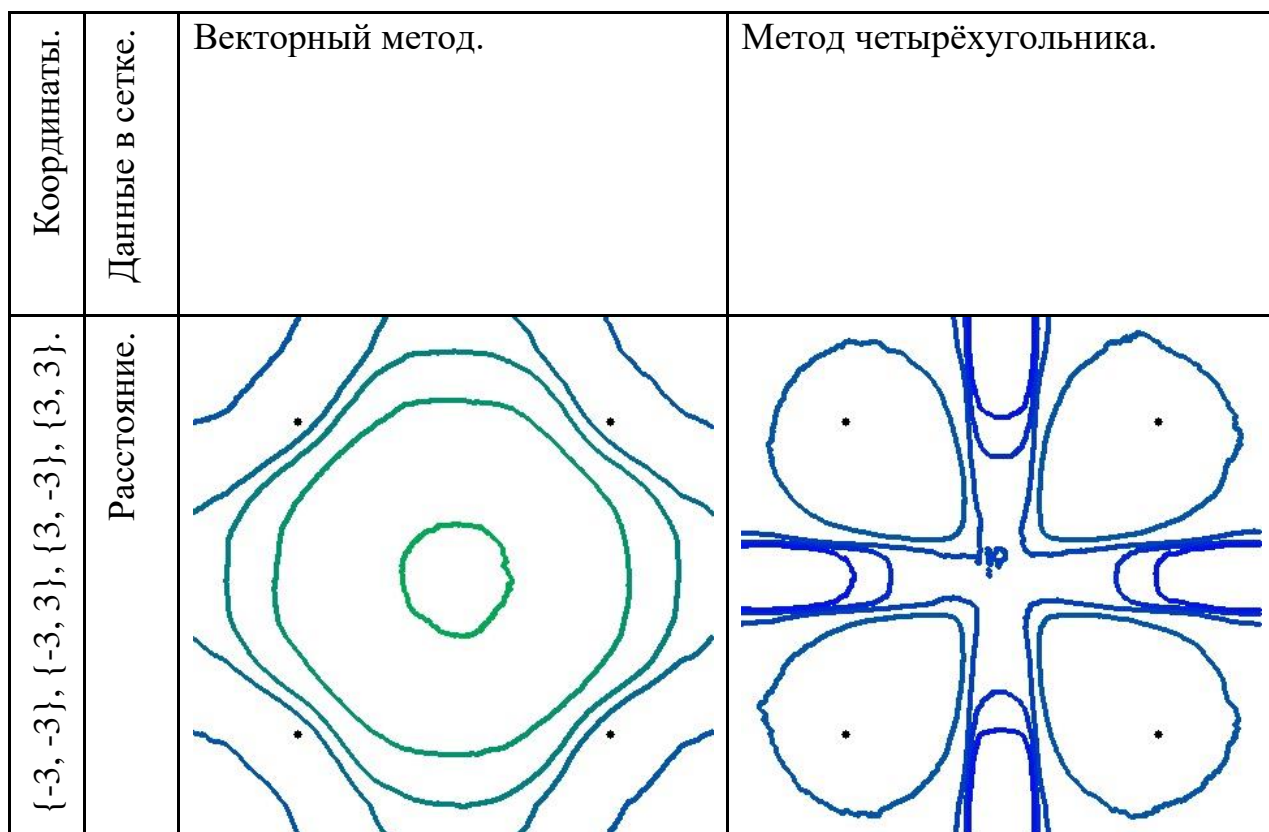
методах может возникнуть исключительно при неопределённости во времени, рассмотрим только матрицу погрешности для времени:

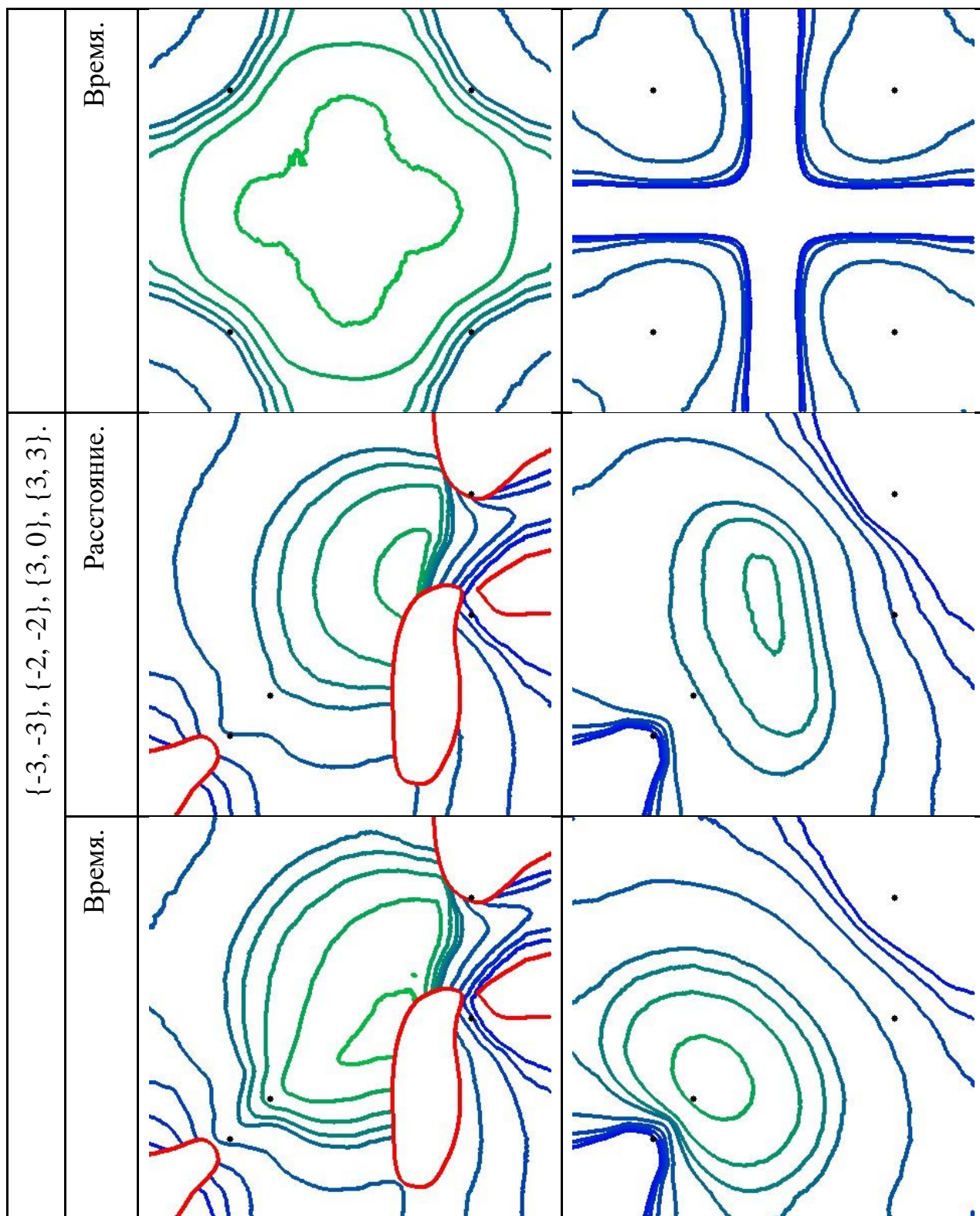
Координаты детекторов.	Векторный метод.	Метод четырёхугольника.
$\{-3, 3\}, \{-3, -3\}, \{3, -3\}, \{3, 3\}$.		
$\{-3, -3\}, \{-2, -2\}, \{1, 3\}, \{3, 1\}$.		



Мы видим, что методы с точки зрения областей слепых зон – различны. В векторном методе довольно часто встречаются расстановки из четырёх детекторов, которые имеют слепые зоны, что нельзя сказать о методе четырёх угольника. Также отметим, что мы не наблюдаем серьёзных отклонений реальных результатов от ожидаемых (так как на изображениях отсутствуют пиксели с синим оттенком), из чего следует, что методы были реализованы корректно.

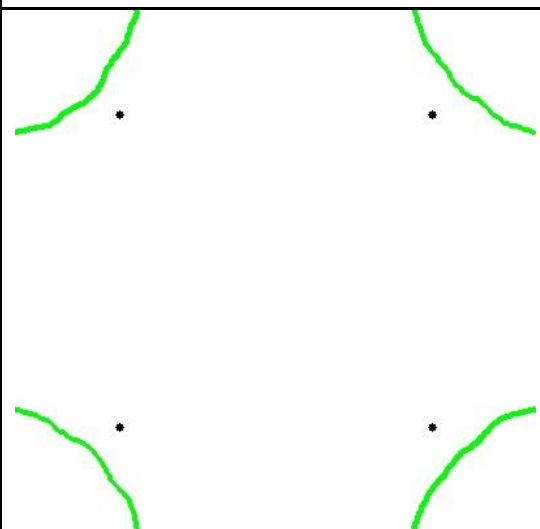
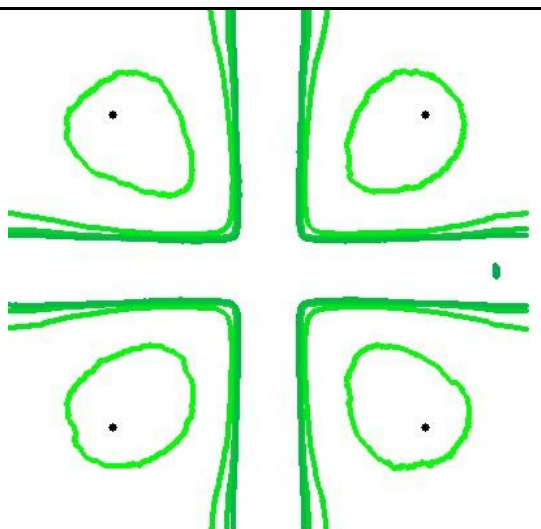
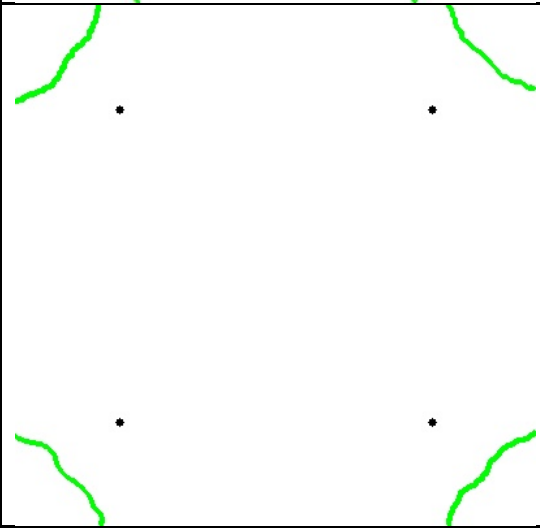
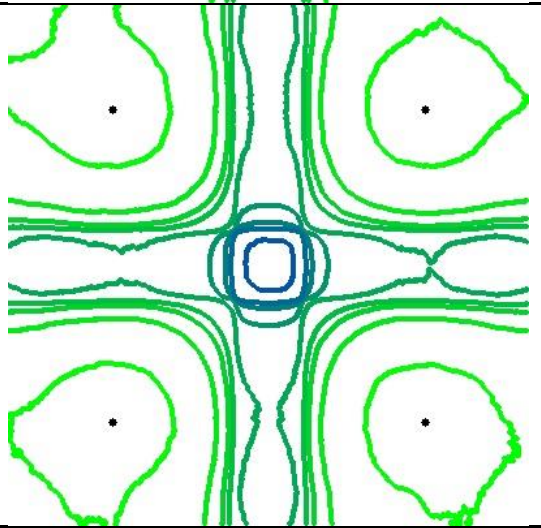
Теперь установим максимальную погрешность в скорости в 1%. Тогда:

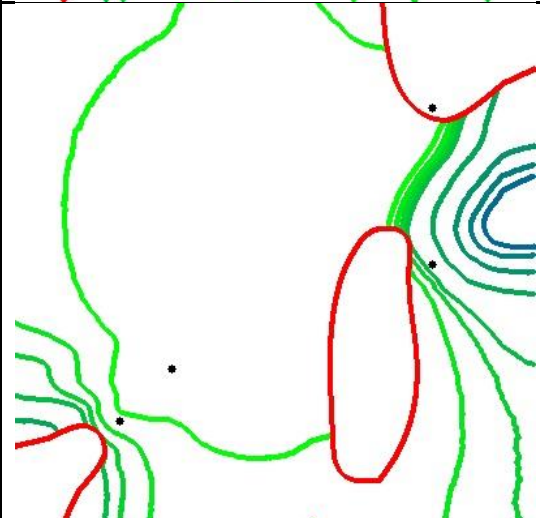
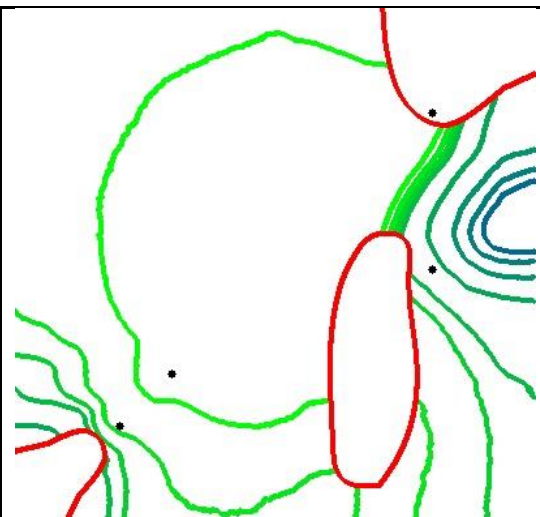
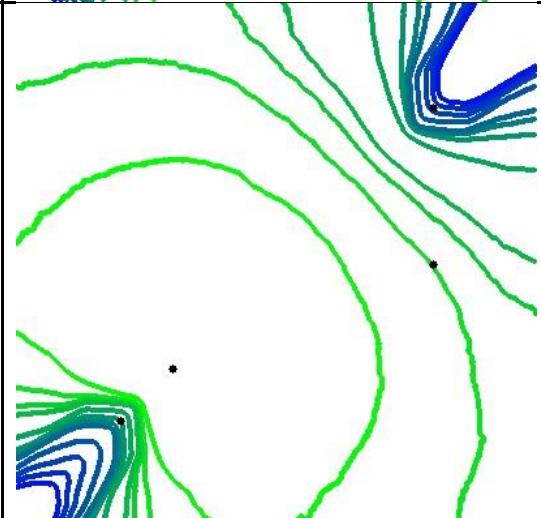
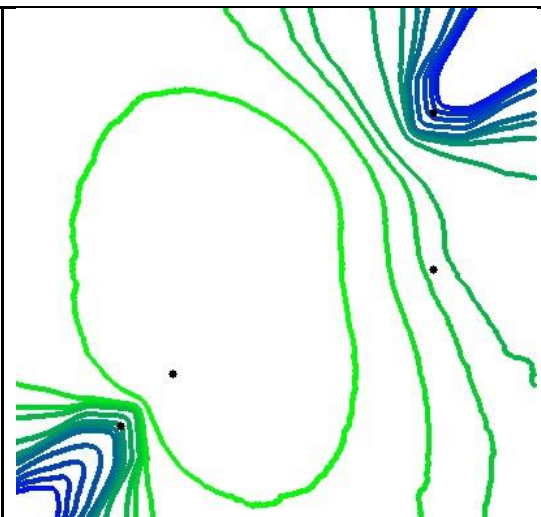
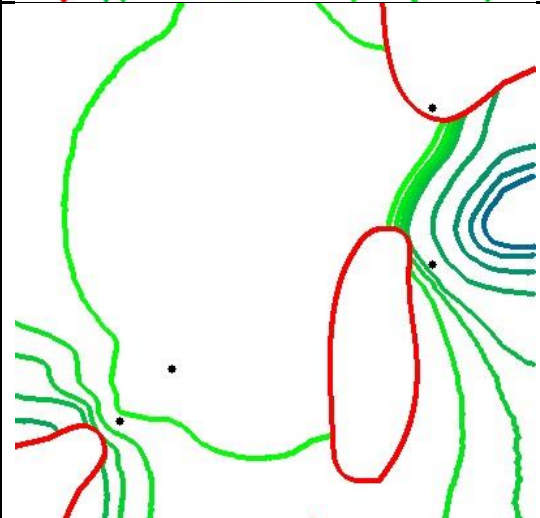
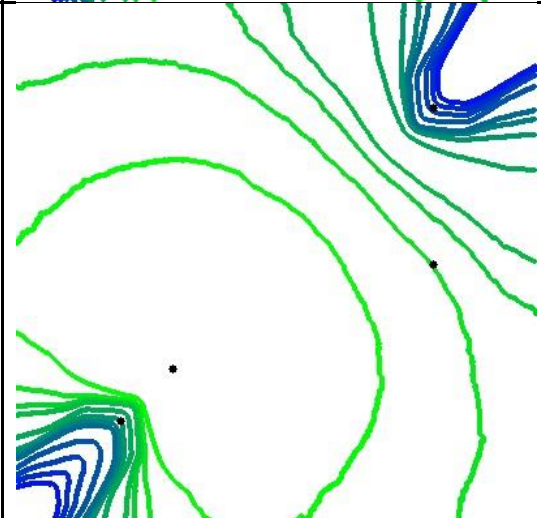




Мы видим, что методы, в смысле погрешностей и наличия зон неопределённостей, ведут себя различно: векторный метод при определённых расстановках детекторов имеет зоны неопределённости, что не наблюдается у метода четырехугольника, но при этом, последний способен показать худшие результаты в вопросе точности на более простых расположениях детекторов.

Установим максимальное отклонение погрешности на 0.01 единиц по времени (для скорости она равна 0). Получаем:

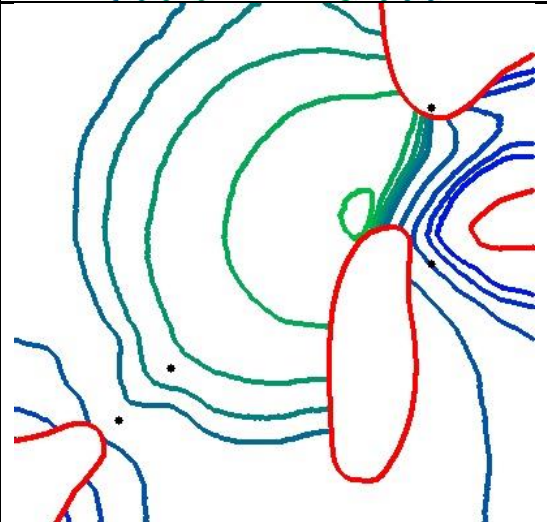
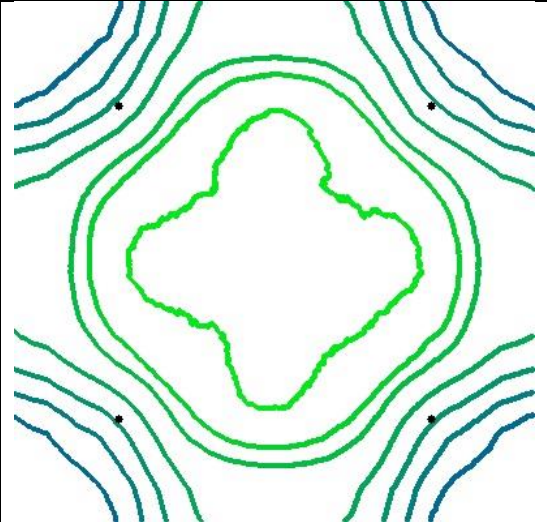
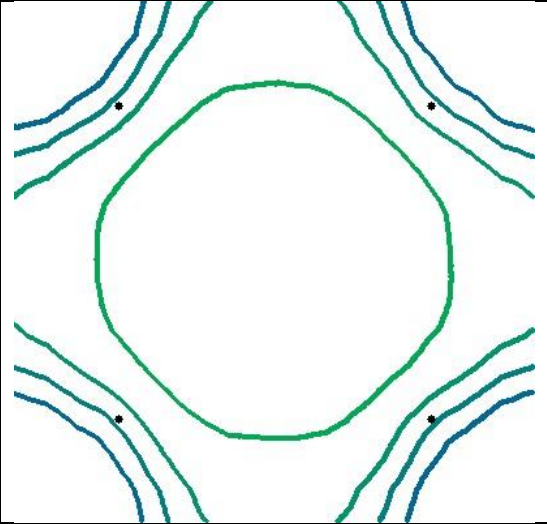
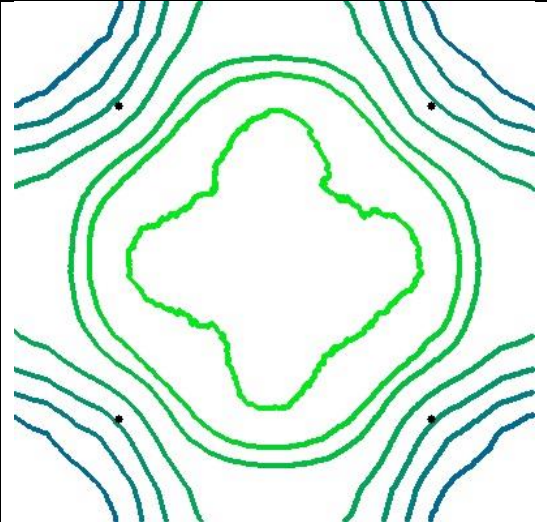
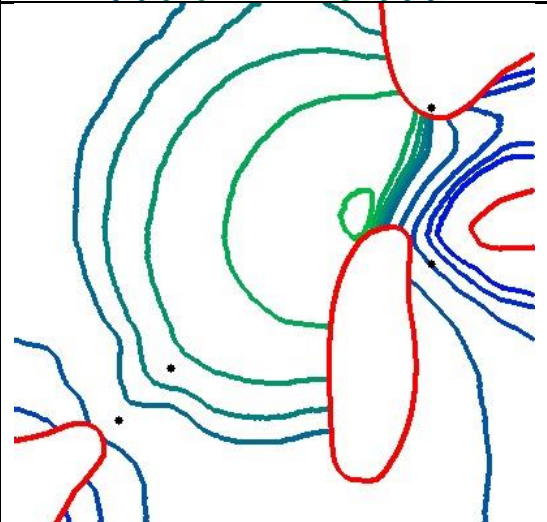
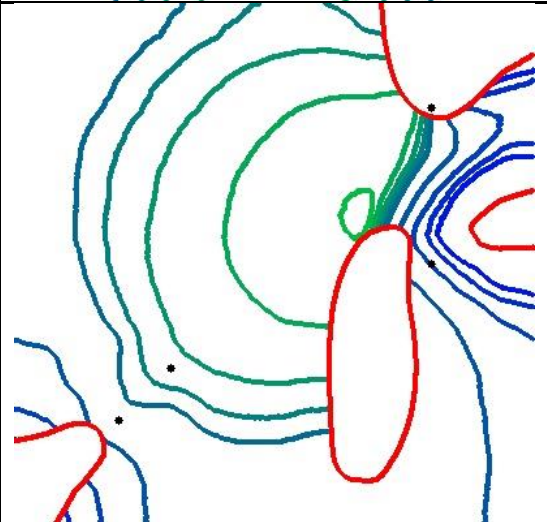
Координаты.	Данные в сетке.	Векторный метод.	Метод четырёхугольника.
{-3, -3}, {-3, 3}, {3, -3}, {3, 3}.	Расстояние.		
	Время.		

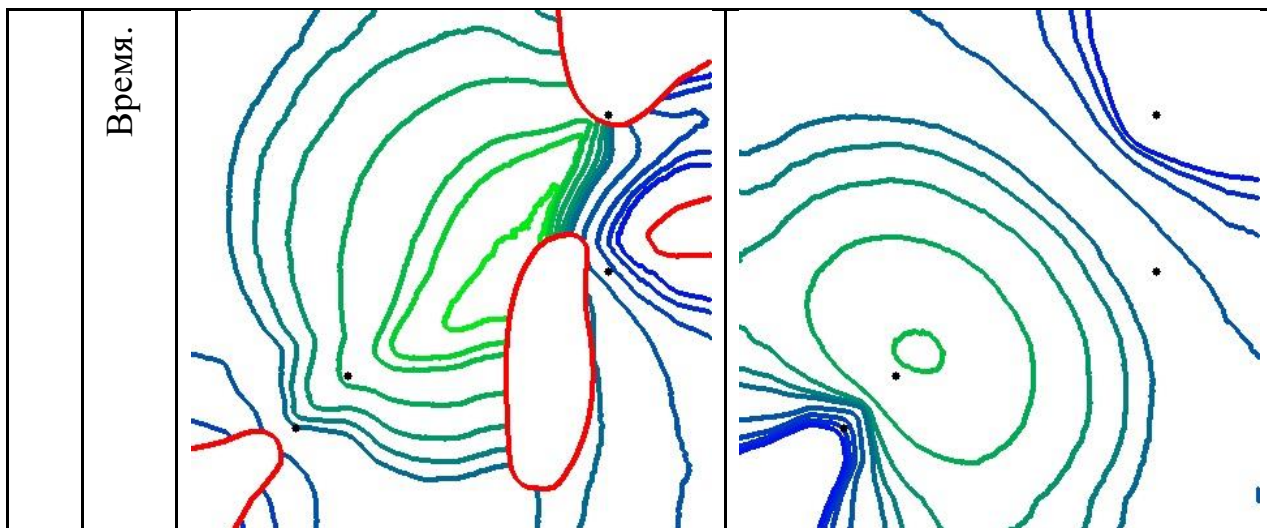
	Время.	{-3, -3}, {-2, -2}, {3, 0}, {3, 3}.	
		Расстояние.	
			
			

Заметно, что зоны появления погрешностей, в соответствии с методами, практически совпадают, с зонами, получаемыми при добавлении погрешности только к скорости. При этом методы относятся друг другу в плане погрешностей и наличия зон неопределённости аналогично наличию погрешности только по скорости.

Теперь же установим максимальное отклонение погрешности на 0.005 единиц по времени и на 0.5% по скорости. Получаем:

Координаты.	Данные в сетке.	Векторный метод.	Метод четырёхугольника.

$\{-3, -3\}, \{-2, -2\}, \{3, 0\}, \{3, 3\}.$	Расстояние.	
$\{-3, -3\}, \{-3, 3\}, \{3, -3\}, \{3, 3\}.$	Время.	
	Расстояние.	
	Время.	
	Расстояние.	
	Время.	



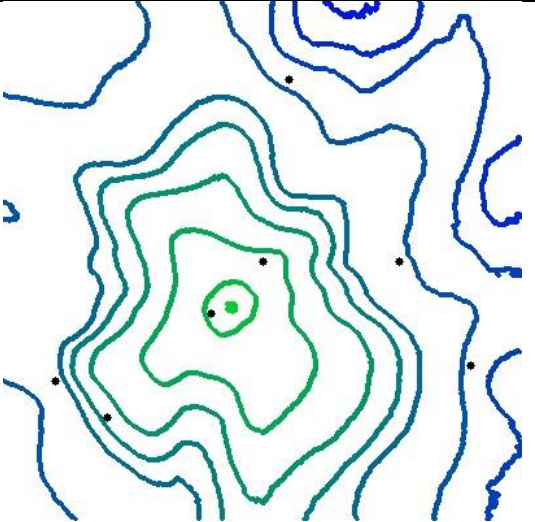
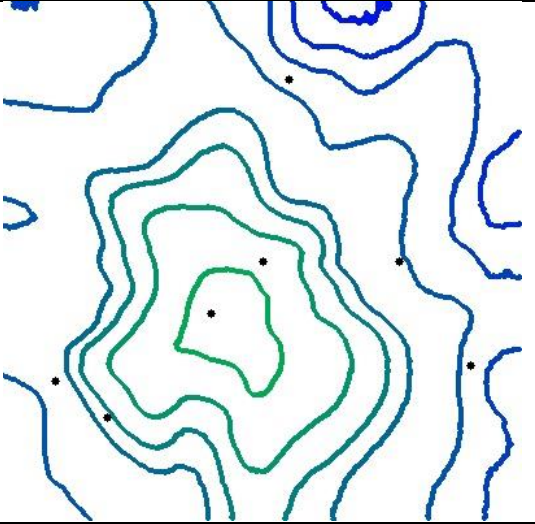
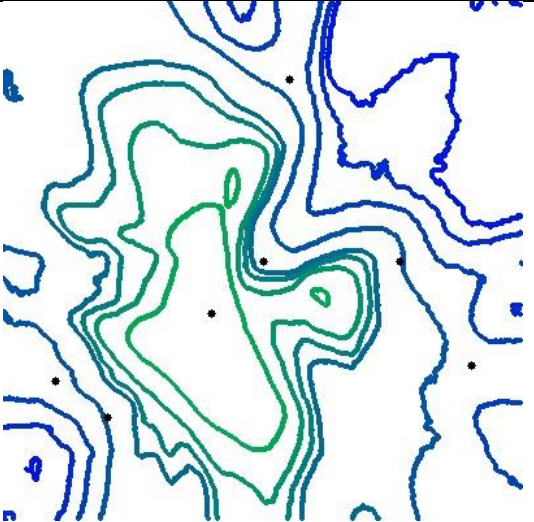
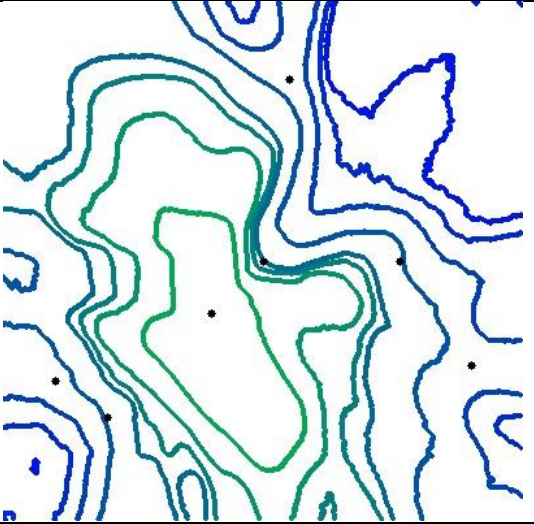
Отметим, что места появления ошибок для каждого из методов не изменились.

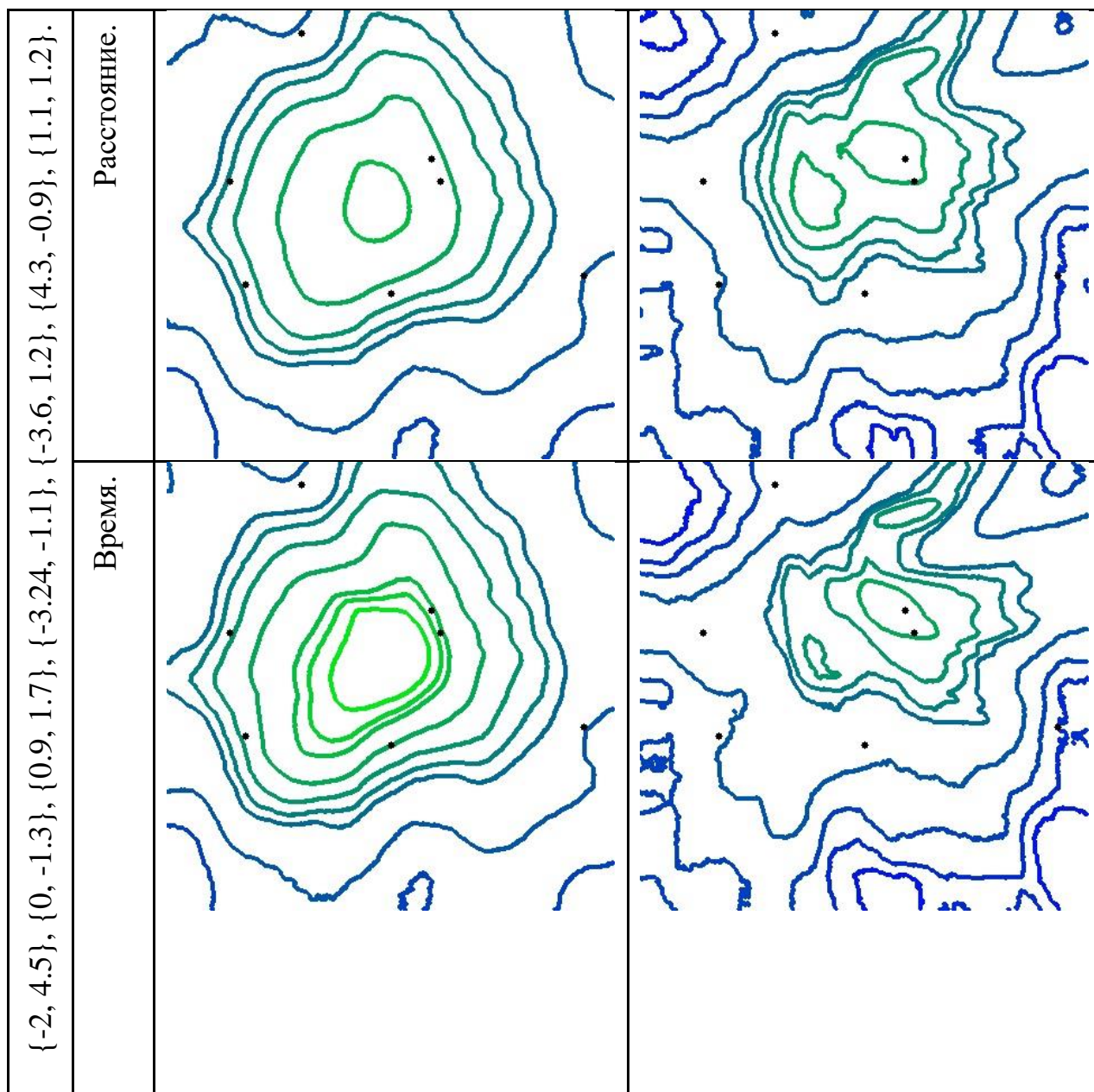
Таким образом мы видим, что данные методы ведут себя различно в плане наличия зон неопределённости и погрешностей:

- в методе четырёхугольника зоны неопределённости встречаются реже, чем в векторном методе,
- в зависимости от расположения детекторов и рассматриваемой точки один из методов может показать лучшие результаты чем другой.

Теперь рассмотрим поведение метода прямоугольников и векторного метода при семи детекторах с установленным максимальным отклонением погрешности на 0.005 единиц по времени и на 0.5% по скорости:

Координаты.	Данные в сетке.	Векторный метод.	Метод четырёхугольника.

$\{-3, -3\}, \{0, 0\}, \{4, -2\}, \{0.5, 3.5\}, \{2.63, 0\}, \{-4, -2.3\}, \{-1, -1\}.$	
Время.	Расстояние.
	
	



Мы видим, что методы различны в плане зон ошибочного определения времени и координат молний. Также несложно заметить, что объединение результатов методом усреднения ответа является недостаточным. Тогда, возможны следующие варианты объединения:

- брать медианное значение,
- отбрасывать 20% значений с левого края, 20% значений с правого края, из оставшихся 60% взять среднее,
- ввести вероятность (массу) уверенности в ответе, которую затем учитывать в момент объединения.

Разрешение неопределённостей.

Порой, по тем или иным причинам, необходимо избавиться от неопределённостей в конкретных областях. Рассмотрим, как это можно сделать.

- 1) Изменить положение детекторов, если это возможно.
- 2) Увеличить число детекторов.
- 3) Изменить метод детекции. Так, заменив прямой / векторный метод на метод четырёхугольника мы уменьшим площадь слепых зон, но при этом увеличим погрешность.
- 4) Ввести силовую характеристику полученных данных с детекторов, учитываемую всеми перечисленными методами.

Промоделировать происходящие изменения слепых зон при использовании пунктов 1 - 3 можно с помощью программы, расположенной в директории *Modeling/DetectorsLocation*.

Вывод.

Таким образом нам удалось разработать программу способную определять время и местоположение на плоскости возникшей молнии, использующую данные с различного числа детекторов. В ходе испытаний которой мы выяснили:

- Наилучшим методом является векторный, так как он более устойчив к погрешности по сравнению с методом прямоугольника, а также очень схож с прямым методом, но при этом требует меньшего количества операций. Стоит отметить, что векторный метод не лишён недостатков, главным из которых является наличие слепых зон.
- Реализованный метод слияния полученных результатов необходимо заменить, например ввести массу уверенности в полученном ответе и производить объединение согласно ей.

Источники.

- Пермяков М. С., Поталова Е. Ю., Клещева Т. И. Грозовая активность в Приморском крае // Метеорология и гидрология. – 2019 – 12 – С. 43 – 52.
- Koshak W. J., Blakeslee R. J., Data Retrieval Algorithms for Validating the Optical Transient Detector and the Lightning Imaging Sensor // Journal of Atmospheric and Oceanic Technology. – March 2000 – С. 279 – 297.