Абзалов Булат Вилнурович (группа БПМ-22-1)

Отчет по задаче «Guns and Cores» (Пушки и ядра)

Ссылка на все части кода (библиотека Geom и сам исполняемый файл) в папке в GitHub: <https://github.com/bruhbzaaal/abzalov_b_v/tree/main/abzalov_b_v/guns_cores_modulation>

**Условие задачи.**

Векторно даны координаты двух пушек. Векторно задана скорость Пушки 1 . В момент времени Ʈ = 0 первая пушка выстреливает Ядром 1 со скоростью . Через промежуток времени ∆t, за который звук выстрела первой пушки доходит до второй пушки, Пушка 2 должна выстрелить с такой скоростью , чтобы Ядро 2 столкнулось с Ядром 1 через время t, которое задает пользователь. При этом задается ограничение модуля скорости второй пушки – она не может стрелять быстрее заданной скорости.

**Решение задачи.**

1. Используя функцию norm() библиотеки Geom находим расстояние между пушками, зная их координаты. Полученное время делим на 331 м/с (скорость звука в воздухе), находим ∆t.
2. Уравнения движения первого ядра

* По оси X:
* По оси Y:

1. Уравнения движения второго ядра

* По оси X:
* По оси Y:

1. Пользователь задает время, в которое он ожидает увидеть столкновение ядер.
2. Из уравнений (2) находим координаты первого ядра в заданный момент времени. В этих же координатах в тот же момент времени должно оказаться второе ядро. Из уравнений (3) выражаем и .
3. Если пользователь задаст момент времени, когда первое ядро уже приземлилось, код уведомит об этом пользователя (Collision is impossible at this time - core 1 landed). Если при найденных и модуль скорости V2 будет превосходить заданное ограничение по скорости, код уведомит об этом пользователя и укажет, какая скорость по модулю нужна для столкновения ядер в этот момент времени (Collision is impossible at this time - speed limit. Needed speed: 110.03 (пример)). Если все в порядке, код выводит искомый вектор скорости, его модуль (для сравнения с ограничением), а так же координаты точки столкновения ядер.
4. Затем код моделирует траектории движения каждого ядра, находя их координаты в каждый момент времени с частотой дискретизации 0.1 с, до тех пор, пока оба ядра не приземлятся. Когда приземляется одно из ядер, код «замораживает» его координаты. Результаты моделирования выводятся в файле table.tsv для дальнейшей копии в Exсel.

**Пример модели 1.**

Rdec2D gun\_1{ 10.0, 5.0 }; //координаты первой пушки

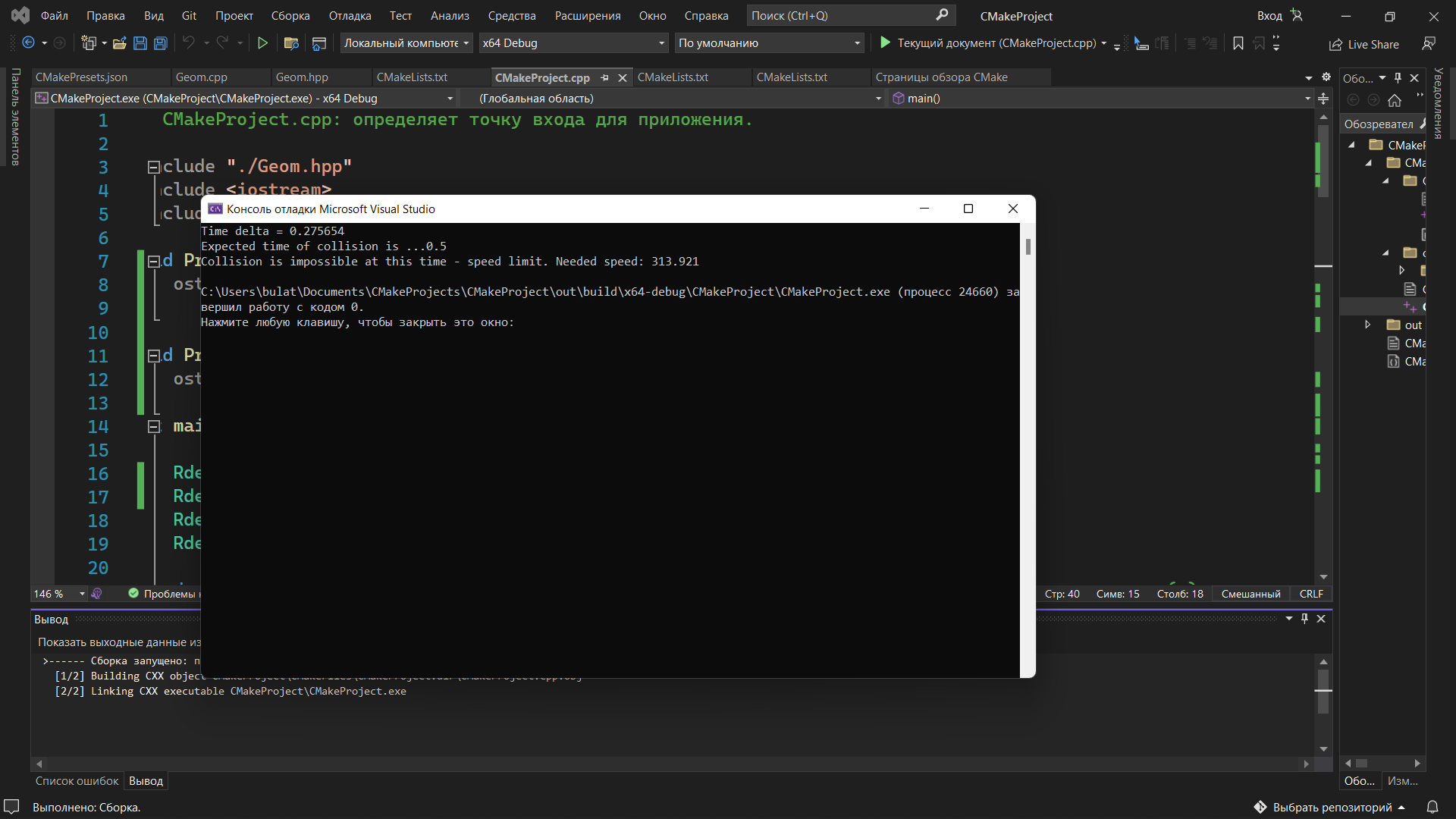
Rdec2D gun\_2{ 100.0, 20.0 }; //координаты второй пушки

Rdec2D speed\_1{ 50.0, 50.0 }; //скорость первой пушки (м/с)

…

double speed\_limit = 100; // скаляр предельной скорости второй пушки (м/с)

1. Ожидаемое время столкновения: 0.5 с

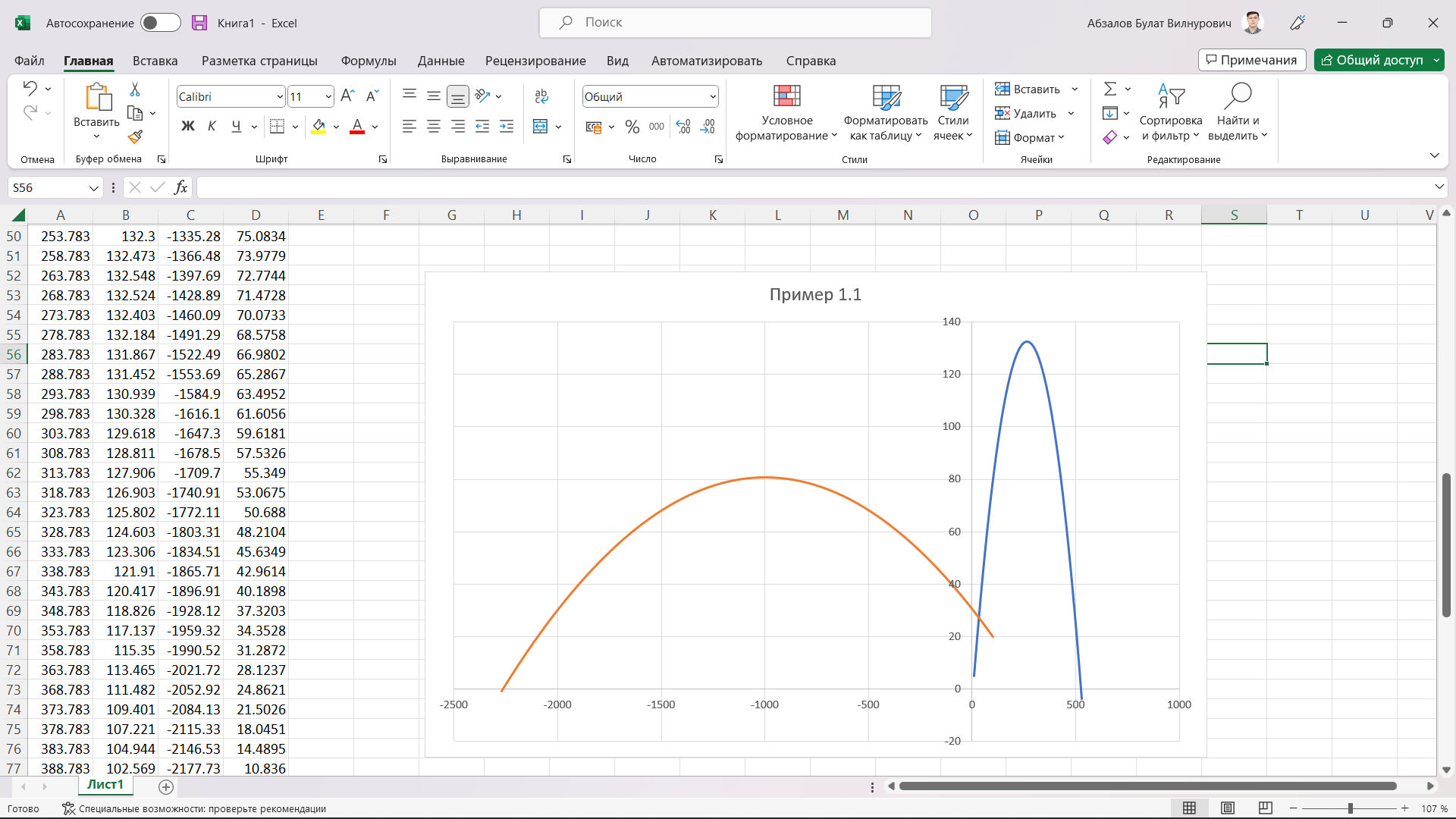


Код моделирует координаты, беря невозможную скорость, чтобы показать траектории, если бы не было ограничения:

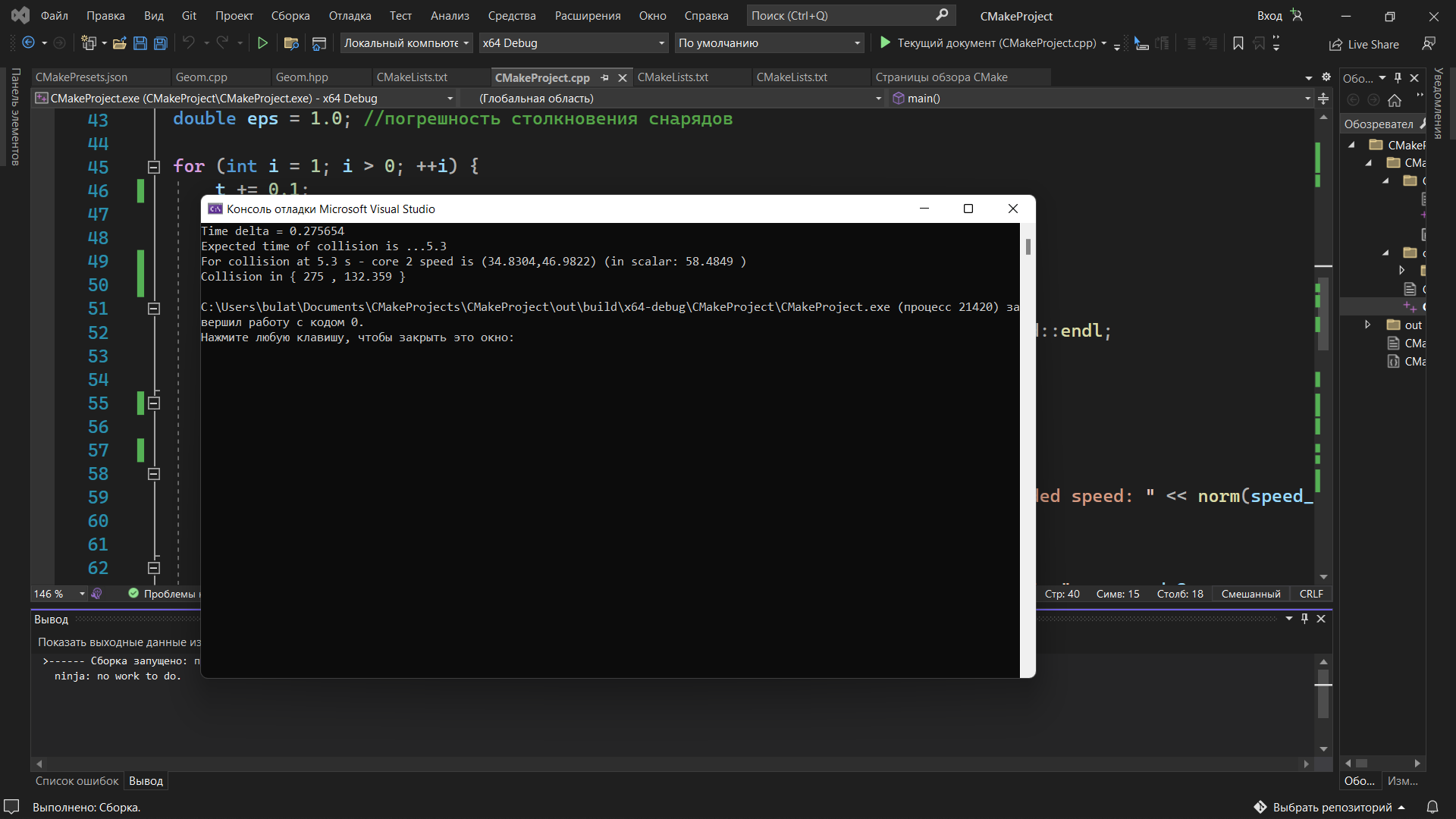
Изображение выглядит как текст, электроника, компьютер, снимок экрана

Автоматически созданное описание

График в Excel:



1. Ожидаемое время столкновения: 5.3 с

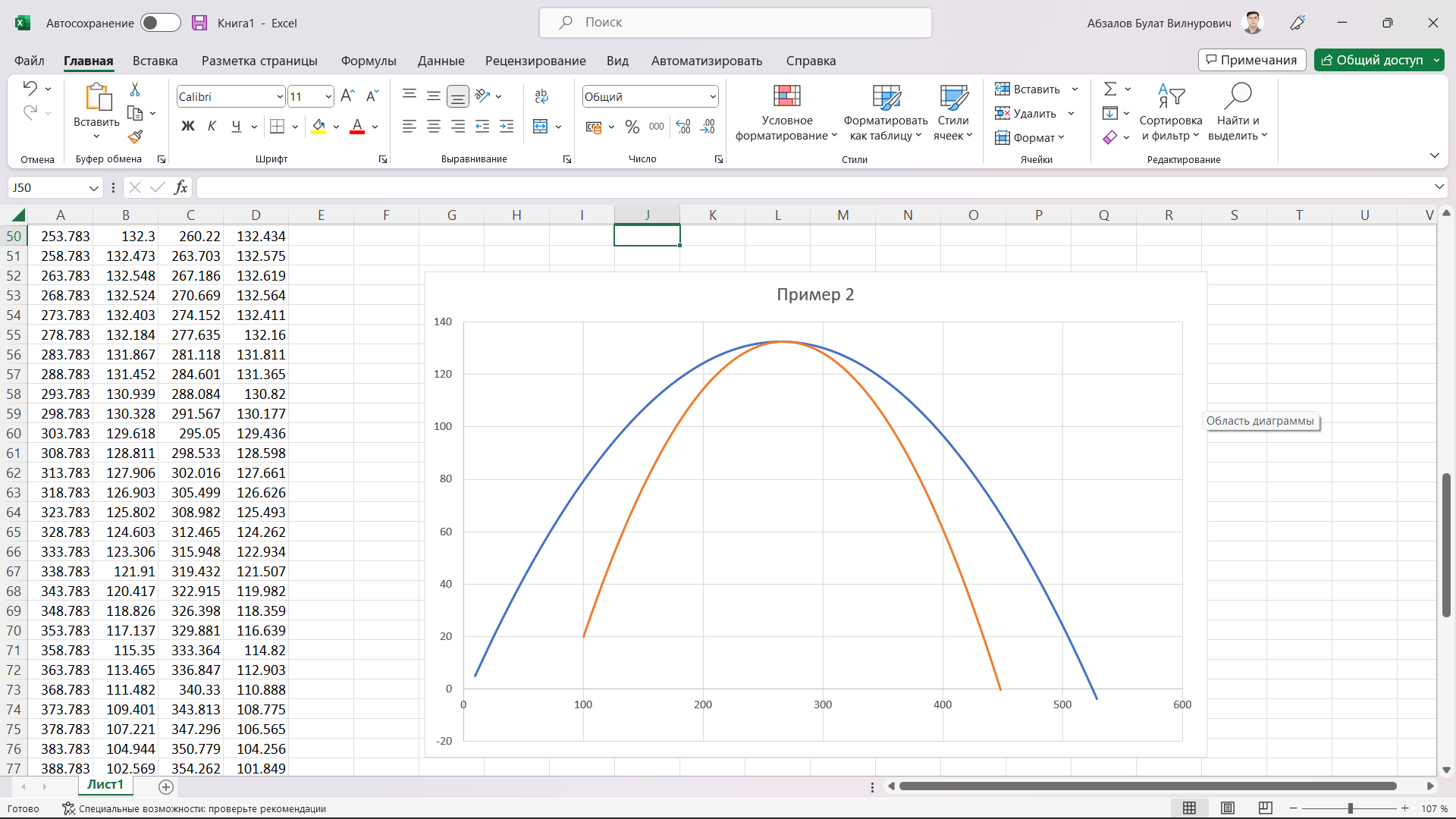


Моделирование:

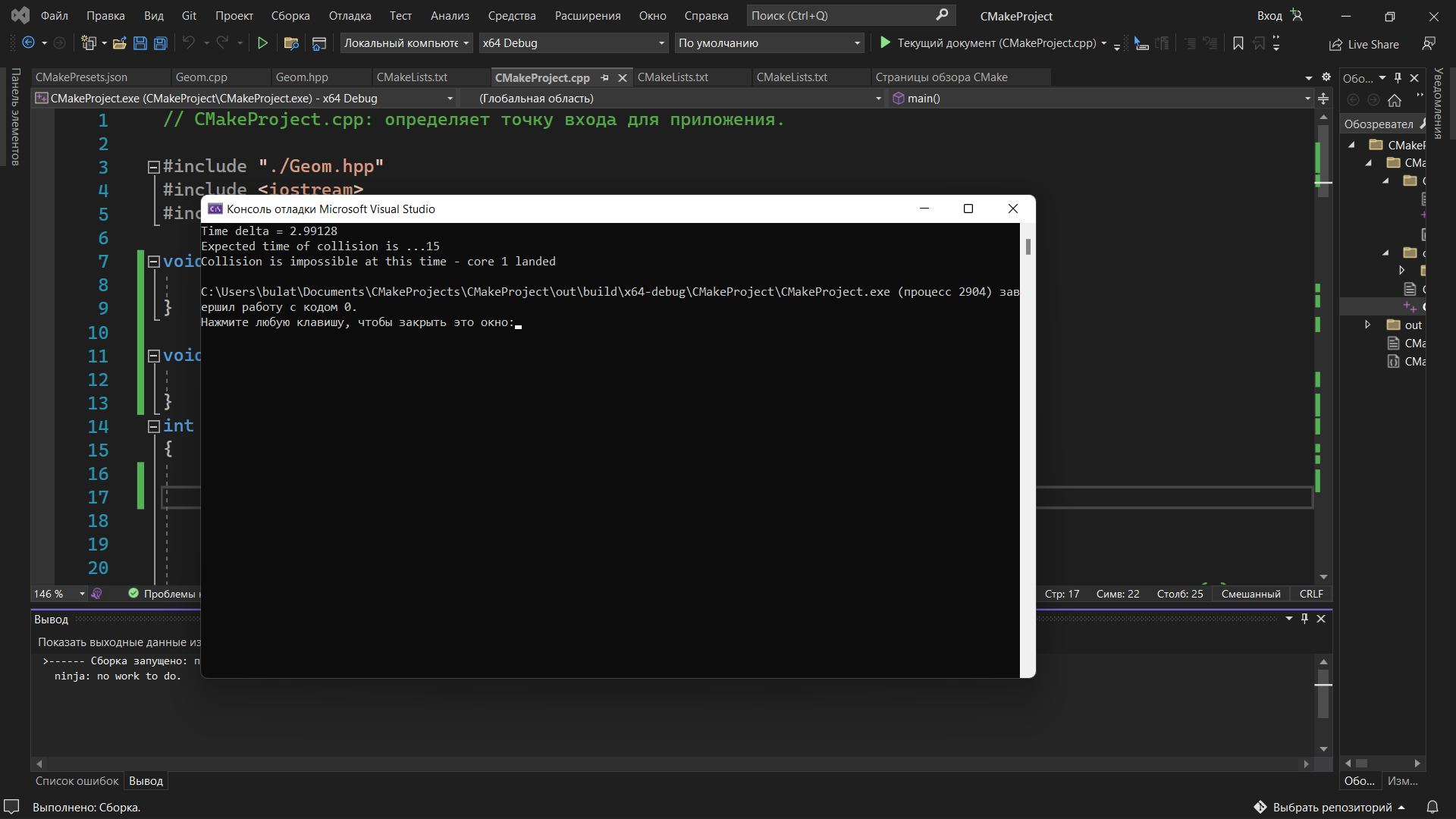
Изображение выглядит как текст, снимок экрана, компьютер

Автоматически созданное описание

График в Excel:



1. Ожидаемое время столкновения: 15 с



**Пример модели 2.**

Rdec2D gun\_1{ 10.0, 5.0 }; //координаты первой пушки

Rdec2D gun\_2{ 1000.0, 20.0 }; //координаты второй пушки

Rdec2D speed\_1{ 50.0, 50.0 }; //скорость первой пушки (м/с)

double speed\_limit = 100; // скаляр предельной скорости второй пушки (м/с)

Ожидаемое время столкновения: 10 с

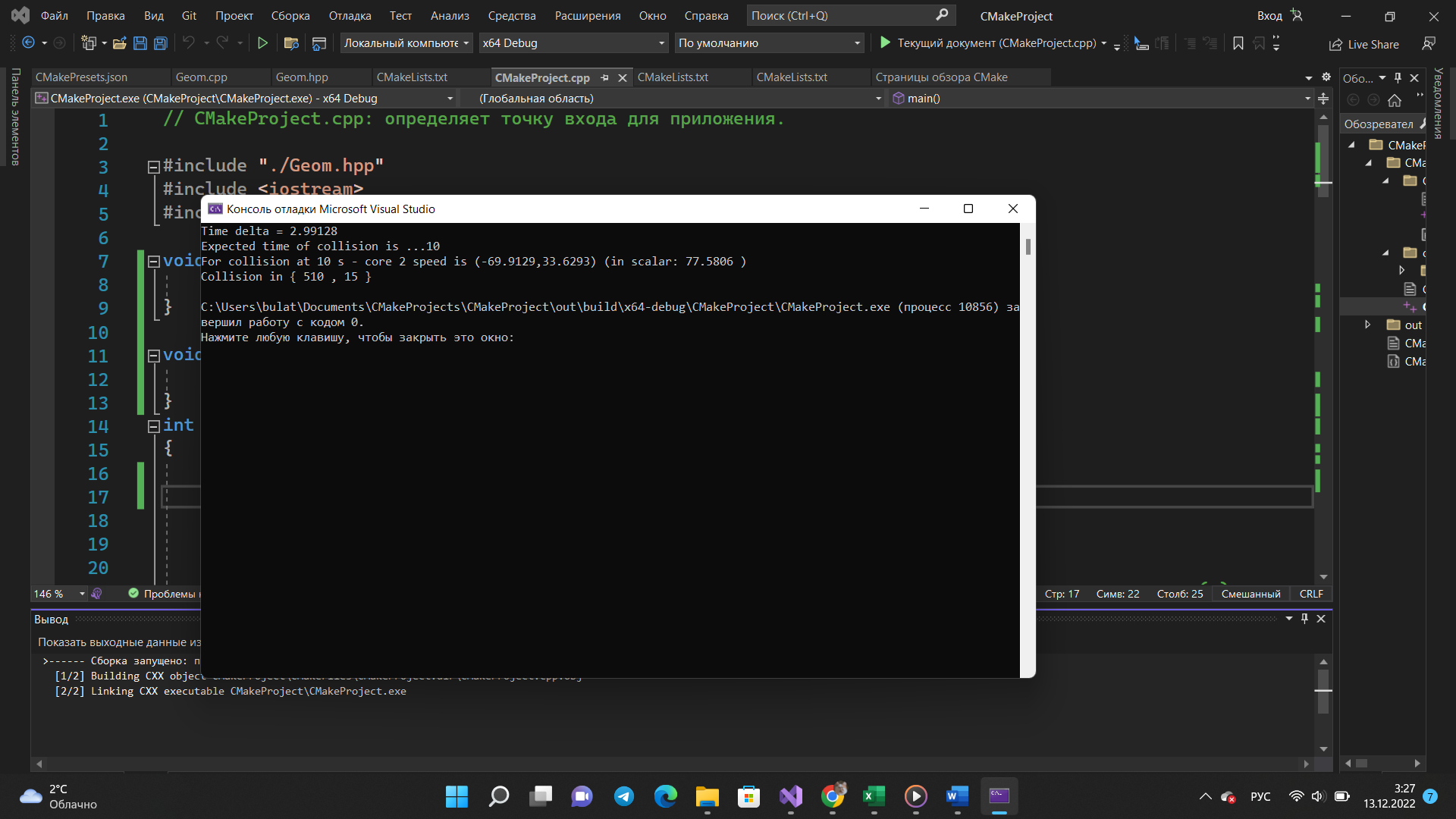


График Excel:

